3. Index

3.1 HOWTOs

The following Linux HOWTOs are currently available:

- 3Dfx HOWTO, by Bernd Kreimeier <bk@gamers.org>. How to use 3Dfx graphics accelerator chip support. Updated 6 February 1998.
- AX25 HOWTO, by Terry Dawson <terry@perf.no.itg.telecom.com.au>. How to configure AX25 networking for Linux. Updated 17 October 1997.
- Access HOWTO, by Michael De La Rue <access-howto@ed.ac.uk>. How to use adaptive technology with Linux. Updated 28 March 1997.
- Alpha HOWTO, by David Mosberger <davidm@azstarnet.com>. Overview of Alpha systems and processors. Updated 6 June 1997.
- Assembly HOWTO, by François-René Rideau <fare@tunes.org>. Information on programming in x86 assembly. Updated 22 June 1999.
- Bash Prompt HOWTO, by Giles Orr <giles@interlog.com>. Creating and controlling terminal and xterm prompts. Updated 07 January 1999.
- Belgian HOWTO, by Dag Wieers <dag@mind.be>. Information on using Linux with Belgian-specific features. Updated 07 January 1999.
- Benchmarking HOWTO, by André D. Balsa <andrewbalsa@usa.net>. How to do basic benchmarking. Updated 15 August 1997.
- Beowulf HOWTO, by Jacek Radajewski <jacek@usq.edu.au> Douglas Eadline plogic.com>. Introduces the Beowulf Supercomputer architecture and provides background information on parallel programming.. Updated 22 November 1998.
- BootPrompt HOWTO, by Paul Gortmaker <gpg109@rsphy1.anu.edu.au>. List of boot time arguments and overview of booting software. Updated May 15, 1999.
- Bootdisk HOWTO, by Tom Fawcett <fawcett@croftj.net>. How to create a boot/root maintenance disk for Linux. Updated May 15, 1999.
- Busmouse HOWTO, by Chris Bagwell <cbagwell@sprynet.com>. Information on bus mouse compatibility with Linux. Updated 15 June 1998.
- CD Writing HOWTO, by Winfried Trümper <winni@xpilot.org>. How to write CDs. Updated 7 August 1999.
- CDROM HOWTO, by Jeff Tranter <jeff_tranter@pobox.com>. Information on CDROM drive compatibility for Linux. Updated 24 March 1999.
- Chinese HOWTO, by Chih-Wei Huang <cwhuang@phys.ntu.edu.tw>. How to configure Linux for use with the Chinese characterset. Updated 2 June 1998.
- Commercial HOWTO, by Mr. Poet <poet@linuxports.com>. Listing of commercial software products for Linux. Updated 7 March 1999.
- Config HOWTO, by Guido Gonzato <guido@ibogfs.cineca.it>. How to fine-tune and customize your

Linux system. Updated 19 January 1999.

- Consultants HOWTO, by Mr. Poet <poet@linuxports.com>. Listing of Linux consultants. Updated 3 July 1999.
- Cyrillic HOWTO, by Alexander L. Belikoff <abel@bfr.co.il>. How to configure Linux for use with the Cyrillic characterset. Updated 23 January 1998.
- DNS HOWTO, by Nicolai Langfeldt <janl@math.uio.no>. How to set up DNS. Updated 11 February 1999.
- DOS/Win to Linux HOWTO, by Guido Gonzato <guido@ibogfs.cineca.it>. How to move from DOS/Windows to Linux. Updated 22 February 1999.
- DOSEMU HOWTO, by Uwe Bonnes <bon@elektron.ikp.physik.th-darmstadt.de> David Hodges erewhon.ak.planet.co.nz>. HOWTO about the Linux MS-DOS Emulator, DOSEMU. Updated dosemu-0.98.8, 24 April 1999.
- Danish HOWTO, by Niels Kristian Bech Jensen <nkbj@image.dk>. How to configure Linux for use with the Danish characterset. Updated 20 February 1999.
- Diskless HOWTO, by Robert Nemkin <buci@math.klte.hu>. How to set up a diskless Linux box. Updated 13 May 1999.
- Distribution HOWTO, by Eric S. Raymond <esr@snark.thyrsus.com>. A list of Linux distributions. Updated 8 May 1999.
- Emacspeak HOWTO, by Jim Van Zandt <jrv@vanzandt.mv.com>. How to use 'emacspeak' with Linux. Updated 10 April 1999.
- Esperanto HOWTO, by Wolfram Diestel <diestel@rzaix340.rz.uni-leipzig.de>. How to use Esperanto in general and ISO-8859-3 in special with Linux. Updated August 1999.
- Ethernet HOWTO, by Paul Gortmaker <gpg109@rsphy1.anu.edu.au>. Information on Ethernet hardware compatibility for Linux. Updated 5 May 1999.
- Finnish HOWTO, by Pekka Taipale <pjt@iki.fi>. How to configure Linux for use with the Finnish characterset. Updated 14 February 1996.
- Firewall HOWTO, by Mark Grennan <markg@netplus.net>. How to set up a firewall using Linux. Updated 8 November 1996.
- French HOWTO, by Guylhem Aznar <guylhem@danmark.linux.eu.org>. How to configure Linux for use with the French characterset.
- Ftape HOWTO, by Kevin Johnson <kjj@pobox.com>. Information on ftape drive compatibility with Linux. Updated August 1998.
- GCC HOWTO, by Daniel Barlow <daniel.barlow@linux.org>. How to set up the GNU C compiler and development libraries. Updated 28 February 1996.
- German HOWTO, by Winfried Trümper <winni@xpilot.org>. Information on using Linux with German-specific features. Updated 19 March 1997.
- Glibc2 HOWTO, by Eric Green <ejg3@cornell.edu>. How to install and migrate to the glibc2 library. Updated 8 February 1998.
- HAM HOWTO, by Terry Dawson <terry@perf.no.itg.telecom.com.au>. HOWTO configure amateur radio software for Linux. Updated 1 April 1997.
- HOWTO HOWTO, by Mark F. Komarinski <markk@cgipc.com>. Getting a new LDP author up and running with tools, ideas, and. Updated 1 April 1997.

TOC

- HOWTO Index, by Tim Bynum <linux-howto@metalab.unc.edu>. Index of HOWTO documents about Linux. Updated 12 September 1999.
- Hardware Compatibility HOWTO, by Patrick Reijnen <antispam.patrickr@antispam.bart.nl>. A list of hardware known to work with Linux. Updated 20 March 1999.
- Hebrew HOWTO, by Yair G. Rajwan <yair@hobbes.jct.ac.il>. How to configure Linux for use with the Hebrew characterset. Updated 12 September 1995.
- INFO-SHEET, by Michael K. Johnson <johnsonm@redhat.com>. Generic introduction to the Linux operating system. Updated 1 September 1998.
- IPCHAINS HOWTO, by Paul Russell <Paul.Russell@rustcorp.com.au>. Install and configure the enhanced IP firewalling chains software. Updated 12 March 1999.
- IPX HOWTO, by Terry Dawson <terry@perf.no.itg.telecom.com.au>. How to install and configure IPX networking. Updated 06 May 1998.
- IR HOWTO, by Werner Heuser <r2d2c3po@zedat.fu-berlin.de>. An introduction to the software provided by the Linux/IR project. Updated 9 February 1999.
- ISP Hookup HOWTO, by Egil Kvaleberg <egil@kvaleberg.no>. Basic introduction to hooking up to an ISP. Updated 5 March 1998.
- Installation HOWTO, by Eric S. Raymond <esr@snark.thyrsus.com>. How to obtain and install Linux. Updated 20 November 1998.
- Intranet Server HOWTO, by Pramod Karnad <karnad@indiamail.com>. How to setup a Linux Intranet server. Updated 7 August 1997.
- Italian HOWTO, by Marco ``Gaio'' Gaiarin <gaio@dei.unipd.it>. How to configure Linux for use with the Italian characterset. Updated 3 November 1998.
- Java-CGI HOWTO, by David H. Silber <dhs@orbits.com>. How to set up Java-capable CGI bin. Updated 1 December 1998.
- Kernel HOWTO, by Brian Ward <ward@blah.math.tu-graz.ac.at>. Upgrading and compiling the Linux kernel. Updated 5 June 1999.
- Keyboard and Console HOWTO, by Andries Brouwer <aeb@cwi.nl>. Information about the Linux keyboard, console and non-ASCII characters. Updated 25 February 1998.
- KickStart HOWTO, by Martin Hamilton <martinh@gnu.org>. Briefly describes how to use the RedHat Linux KickStart system to rapidly install large numbers of identical Linux boxes. Updated 11 January 1999.
- LinuxDoc+Emacs+Ispell HOWTO, by Philippe Martin <feloy@wanadoo.fr>. Assist writers and translators of Linux HOWTOs or any other paper for the Linux Documentation Project. Updated 27 February 1998.
- META-FAQ, by Michael K. Johnson <johnsonm@redhat.com>. A listing of Linux sources of information. Updated 25 October 1997.
- MGR HOWTO, by Vincent Broman <broman@nosc.mil>. Information on the MGR graphics interface for Linux. Updated 30 May 1996.
- MILO HOWTO, by David A. Rusling <david.rusling@reo.mts.dec.com>. How to use the Alpha Linux Miniloader (MILO). Updated 6 December 1996.
- MIPS HOWTO, by Ralf Baechle <ralf@gnu.org>. Describes the MIPS port of the Linux operating system, common problems and their solutions, availability and more. Updated 31 March 1999.
- Mail HOWTO, by Guylhem Aznar <guylhem@danmark.linux.eu.org>. Information on electronic mail servers and clients. Updated July 1999.

TOC

- Mail User HOWTO, by Eric S. Raymond <esr@thyrsus.com>. This document is an introduction to the world of electronic mail. Updated 07 May 1999.
- Modem HOWTO, by David S. Lawyer <bf347@lafn.org>. Help with selecting, connecting, configuring, trouble-shooting, and understanding modems for a PC. Updated August 1999.
- Multi-Disk HOWTO, by Stein Gjoen <sgjoen@nyx.net>. How to setup multiple hard disk drives. Updated 27 May 1999.
- Multicast HOWTO, by Juan-Mariano de Goyeneche <jmseyas@dit.upm.es>. This HOWTO tries to cover most aspects related to multicast over TCP/IP networks. Updated 20 March 1998.
- NET3-4 HOWTO, by Mr. Poet <poet@linuxports.com>. Howto configure TCP/IP networking under Linux. Updated August 1999.
- NFS HOWTO, by Nicolai Langfeldt <janl@math.uio.no>. How to set up NFS clients and servers. Updated 3 November 1997.
- NIS HOWTO, by Thorsten Kukuk <kukuk@suse.de>. Information on using NIS/YP on Linux systems. Updated 9 March 1999.
- Networking Overview HOWTO, by Daniel López Ridruejo <ridruejo@esi.us.es>. The purpose of this document is to give an overview of the networking capabilities of the Linux Operating System providing pointers for further information and implementation details. Updated 10 July 1998.
- Optical Disk HOWTO, by Skip Rye <Skip_Rye@faneuil.com>. How to use optical disk drives with Linux. Updated 11 December 1998.
- Oracle HOWTO, by Paul Haigh <paul@nailed.demon.co.uk>. How to setup Oracle as a database server. Updated 4 August 1998.
- PCI HOWTO, by Michael Will <Michael.Will@student.uni-tuebingen.de>. Information on PCI-architecture compatibility with Linux. Updated 30 March 1997.
- PCMCIA HOWTO, by Dave Hinds <dhinds@allegro.stanford.edu>. How to install and use PCMCIA Card Services. Updated 15 May 1999.
- PPP HOWTO, by Robert Hart <hartr@interweft.com.au>. Information on using PPP networking with Linux. Updated 31 March 1997.
- PalmOS HOWTO, by David H. Silber <pilot@orbits.com>. How to use your Palm OS device with a Linux system. Updated 20 September 1998.
- Parallel Processing HOWTO, by Hank Dietz <pplinux@ecn.purdue.edu>. Discussion of parallel processing approaches for Linux. Updated 5 January 1998.
- Plug and Play HOWTO, by David Lawyer <bf347@lafn.org>. How to get your Linux system to support Plug-and-Play. Updated August 1999.
- Polish HOWTO, by Sergiusz Pawlowicz <ser@arch.pwr.wroc.pl>. Information on using Linux with Polish-specific features. Updated 8 February 1999.
- Portuguese-HOWTO, by Carlos Augusto Moreira dos Santos <casantos@cpmet.ufpel.tche.br>. Este documento pretende ser um guia de referência de configuraçao do Linux e seus programas.... Updated 24 May 1999.
- PostgreSQL HOWTO, by Al Dev (Alavoor Vasudevan) <aldev@hotmail.com>. How to setup PostgreSQL as a database server. Updated 8 January 1999.
- Printing HOWTO, by Grant Taylor <gtaylor+pht@picante.com>. HOWTO on printing software for Linux. Updated 9 March 1999.

TOC

- Printing Usage HOWTO, by Mark Komarinski <markk@auratek.com>. How to use the printing system a variety of file types and options. Updated 6 February 1998.
- Quake HOWTO, by Bob Zimbinski <bobz@mr.net> Thomas Mike Hallock medina.net>. This document explains how to install, run and troubleshoot Quake, QuakeWorld and Quake II on an Intel Linux system. Updated 30 August 1998.
- RPM HOWTO, by Donnie Barnes <djb@redhat.com>. How to use the Red Hat Package Manager (.rpm). Updated 8 April 1997.
- Reading List HOWTO, by Eric S. Raymond <esr@snark.thyrsus.com>. Interesting books pertaining to Linux subjects. Updated 20 April 1999.
- Root RAID HOWTO, by Michael A. Robinton <michael@bzs.org>. How to create a root-mounted RAID filesystem. Updated 25 March 1998.
- SCSI Programming HOWTO, by Heiko Eissfeldt <heiko@colossus.escape.de>. Information on programming the generic Linux SCSI interface. Updated 7 May 1996.
- SMB HOWTO, by David Wood <dwood@plugged.net.au>. How to use the Session Message Block (SMB) protocol with Linux. Updated 25 June 1999.
- SRM HOWTO, by David Mosberger <davidm@azstarnet.com>. How to boot Linux/Alpha using the SRM firmware. Updated 17 August 1996.
- Security HOWTO, by Kevin Fenzi <kevin@scrye.com>. General overview of security issues. Updated 25 April 1999.
- Serbian HOWTO, by Zoltan Csala <zolika@sezampro.PRVO-SKLONI-OVO.yu>. Information on using Linux with Serbian-specific features. Updated November 1998.
- Serial HOWTO, by David Lawyer <bf347@lafn.org>. This HOWTO covers basic info on the Serial Port and multiport serial cards. Updated August 1999.
- Serial Programming HOWTO, by Peter H. Baumann <Peter.Baumann@dlr.de>. How to use serial ports in programs. Updated 22 January 1998.
- Shadow Password HOWTO, by Michael H. Jackson <mhjack@tscnet.com>. How to obtain, install, and configure shadow passwords. Updated 3 April 1996.
- Slovenian HOWTO, by Primoz Peterlin <primoz.peterlin@biofiz.mf.uni-lj.si>. Information on using Linux with Slovenian-specific features. Updated 15 February 1999.
- Software Building-HOWTO, by Mendel Leo Cooper <thegrendel@theriver.com>. How to build software packages. Updated 27 July 1999.
- Software Release Practice HOWTO, by Eric S. Raymond <esr@snark.thyrsus.com>. Describes good release practices for Linux open-source projects. Updated 18 June 1999.
- Sound HOWTO, by Jeff Tranter <jeff_tranter@pobox.com>. Sound hardware and software for the Linux operating system. Updated 24 March 1999.
- Sound Playing HOWTO, by Yoo C. Chung <wacko@laplace.snu.ac.kr>. How to play various sound formats under Linux. Updated 24 March 1999.
- Spanish HOWTO, by Gonzalo Garcia Agullo <Gonzalo.Garcia-Agullo@jrc.es>. Information on using Linux with Spanish-specific features. Updated 20 August 1996.
- TclTk HOWTO, by Luca Rossetti <lukaros@tin.it>. This document describes the Linux approach to Tcl, a scripting language. Updated 07 November 1998.
- teTeX HOWTO, by Robert Kiesling <kiesling@terracom.net>. How to install the teTeX package (TeX and

LaTeX) under Linux. Updated 9 November 1998.

- Text-Terminal HOWTO, by David S. Lawyer <bf347@lafn.org>. This document explains what text terminals are, how they work, how to install and configure them. Updated August 1999.
- Thai HOWTO, by Poonlap Veeratanabutr <poon-v@fedu.uec.ac.jp>. How to configure Linux for use with the Thai characterset. Updated 4 August 1998.
- Tips HOWTO, by Paul Anderson <paul@geeky1.ebtech.net>. HOWTO on miscellaneous tips and tricks for Linux. Updated June 1998.
- UMSDOS HOWTO, by Jacques Gelinas <jacques@solucorp.qc.ca>. How to install and use the UMSDOS filesystem. Updated 13 November 1995.
- UPS HOWTO, by Harvey J. Stein <abel@netvision.net.il>. Information on using a UPS power supply with Linux. Updated 18 November 1997.
- UUCP HOWTO, by Guylhem Aznar <guylhem@danmark.linux.eu.org>. Information on UUCP software for Linux. Updated 6 February 1998.
- Unix and Internet Fundamentals HOWTO, by Eric S. Raymond <esr@snark.thyrsus.com>. Describes the working basics of PC-class computers, Unix-like operating systems, and the Internet in non-technical language. Updated 3 December 1998.
- User Group HOWTO, by Kendall Grant Clark <kclark@ntlug.org>. Tips on founding, maintaining, and growing a Linux User Group. Updated 24 April 1998.
- VAR HOWTO, by Mr. Poet <poet@linuxports.com>. Listing of Linux value added resellers. Updated 22 August 1999.
- VME HOWTO, by John Huggins and Michael Wyrick <vmelinux@va.net>. How to run Linux on your VMEbus Pentium and other PCI local bus based VMEbus processor designs. Updated 30 July 1998.
- VMS to Linux HOWTO, by Guido Gonzato <guido@ibogfs.cineca.it>. How to move from VMS to Linux. Updated 20 April 1998.
- VPN-Masquerade HOWTO, by John D. Hardin <jhardin@wolfenet.com>. This document describes how to configure a Linux firewall to. Updated 9 September 1999.
- Virtual Services HOWTO, by Brian Ackerman <brian@nycrc.net>. How to set up virtual hosting services. Updated 15 August 1998.
- WWW HOWTO, by Wayne Leister <n3mtr@qis.net>. How to set up WWW clients and servers. Updated 21 August 1999.
- WWW mSQL HOWTO, by Oliver Corff <corff@zedat.fu-berlin.de>. How to set up a web server database with mSQL. Updated 17 September 1997.
- XFree86 HOWTO, by Eric S. Raymond <esr@snark.thyrsus.com>. How to obtain, install, and configure XFree86 3.2 (X11R6). Updated 12 May 1999.
- XFree86 Video Timings HOWTO, by Eric S. Raymond <esr@snark.thyrsus.com>. How to compose a mode line for XFree86. Updated 13 June 1999.
- X Window User HOWTO, by Ray Brigleb <ray@croftj.net>. Information on configuring the X Window environment for the Linux user. Updated 1 September 1999.

3.2 mini-HOWTOs

TOC

The following mini-HOWTOs are available:

- 3 Button Mouse mini-HOWTO, by Geoff Short <geoff@kipper.york.ac.uk>. How to configure your mouse to use 3 buttons. Updated 31 May 1998.
- ADSM Backup mini-HOWTO, by Thomas Koenig <Thomas.Koenig@ciw.uni-karlsruhe.de>. How to install and use the ADSM backup program. Updated 15 January 1997.
- Asymmetric Digital Subscriber Loop (ADSL) mini-HOWTO, by David Fannin <dfannin@dnai.com>. Addresses the ordering, installation and configuration. Updated 10 April 1999.
- AI-Alife mini-HOWTO, by John A. Eikenberry <jae@ai.uga.edu>. Information about AI software for Linux. Updated 13 January 1998.
- Advocacy mini-HOWTO, by Paul L. Rogers <Paul.L.Rogers@li.org>. Suggestions on how to advocate the use of Linux. Updated 7 May 1998.
- Alsa Sound mini-HOWTO, by Valentijn Sessink. Describes the installation of the Alsa sound drivers for Linux. Updated 18 May 1999.
- Apache SSL PHP/FI frontpage mini-HOWTO, by Marcus Faure <marcus@faure.de>. Build a multipurpose webserver. Updated July 1998.
- Automount mini-HOWTO, by Don <don@sabotage.org>. This file describes the autofs automounter, how to configure it, and points out some problems to avoid. Updated 17 April 1999.
- Backup with MSDOS mini-HOWTO, by Christopher Neufeld <neufeld@physics.utoronto.ca>. How to backup Linux machines with MSDOS. Updated 5 August 1997.
- Battery Powered mini-HOWTO, by Hanno Mueller <hanno@lava.de>. How to reduce a Linux system's power consumption. Updated 21 December 1997.
- Boca mini-HOWTO, by David H Dennis <david@freelink.net>. How to install a Boca 16-port serial card (Boca 2016). Updated 1 August 1997.
- BogoMips mini-HOWTO, by Wim C.A. van Dorst <baron@clifton.hobby.nl>. Information about BogoMips. Updated 8 February 1999.
- Bridge mini-HOWTO, by Chris Cole <cole@lynkmedia.com>. How to setup an ethernet bridge. Updated 7 September 1998.
- Bridge+Firewall mini-HOWTO, by Peter Breuer <ptb@it.uc3m.es>. How to setup and ethernet bridge and firewall. Updated 19 December 1997.
- Bzip2 mini-HOWTO, by David Fetter <dfetter@best.com>. How to use the new bzip2 compression program. Updated 22 August 1998.
- Cable Modem mini-HOWTO, by Vladimir Vuksan <vuksan@veus.hr>. How to use a cable modem with a cable ISP. Updated 22 June 1999.
- Cipe+Masquerading mini-HOWTO, by Anthony Ciaravalo <acj@home.com>. How to setup a Virtual Private Network between your LAN and other LAN's using cipe through linux masquerading firewall machines. Updated 28 October 1998.
- Clock mini-HOWTO, by Ron Bean <rbean@execpc.com>. How to set and keep your clock on time. Updated July 1999.
- Coffee mini-HOWTO, by Georgatos Photis <gef@ceid.upatras.gr>. Thoughts about making coffee with Linux (humorous). Updated 15 January 1998.

TOC

- Cyrus IMAP mini-HOWTO, by Kevin Mitchell <kevin@iserv.net>. How to install the Cyrus IMAP se Updated 21 January 1998.
- DHCP mini-HOWTO, by Vladimir Vuksan <vuksan@veus.hr>. How to setup a DHCP Server and Client. Updated 22 June 1999.
- DPT Hardware RAID mini-HOWTO, by Ram Samudrala <me@ram.org>. How to configure hardware RAID. Updated 9 April 1999.
- Diald mini-HOWTO, by Harish Pillay <h.pillay@ieee.org>. How to use 'diald' to dial an ISP. Updated 3 June 1996.
- Ext2fs Undeletion mini-HOWTO, by Aaron Crane <aaronc@pobox.com>. How to retrieve deleted files from an ext2 filesystem. Updated 2 February 1999.
- Fax Server mini-HOWTO, by Erez Strauss <erez@newplaces.com>. How to setup a fax server. Update 8 November 1997.
- Firewall Piercing mini-HOWTO, by François-René Rideau <rideau@ens.fr>. Using ppp over telnet transparently through an Internet firewall. Updated 27 November 1998.
- GIS-GRASS mini-HOWTO, by David A. Hastings <dah@ngdc.noaa.gov>. How to install Geographic Information System (GIS) software. Updated 13 November 1997.
- GTEK BBS-550 mini-HOWTO, by Wajihuddin Ahmed <wahmed@sdnpk.undp.org>. How to setup the GTEK BBS-550 multiport board with Linux. Updated 20 August 1997.
- Hard Disk Upgrade mini-HOWTO, by Yves Bellefeuille <yan@ottawa.com>. How to copy a Linux system from one hard disk to another. Updated 31 January 1998.
- IO Port Programming mini-HOWTO, by Riku Saikkonen <Riku.Saikkonen@hut.fi>. How to use I/O ports in C programs. Updated 28 December 1997.
- IP Alias mini-HOWTO, by Harish Pillay <h.pillay@ieee.org>. How to use IP aliasing. Updated 13 January 1997.
- IP Masquerade mini-HOWTO, by Ambrose Au <ambrose@writeme.com>. How to use IP masquerading. Updated 07 February 1999.
- IP Subnetworking mini-HOWTO, by Robert Hart <hartr@interweft.com.au>. Why and how to subnetwork an IP network. Updated 31 March 1997.
- ISP Connectivity mini-HOWTO, by Michael Strates <mstrates@croftj.net>. How to get mail and news over a dialup connection. Updated 6 November 1997.
- Install From ZIP mini-HOWTO, by Kevin Snively <k.snively@seaslug.org>. How to install Linux from a parallel port ZIP drive. Updated 29 April 1998.
- Kerneld mini-HOWTO, by Henrik Storner <storner@osiris.ping.dk>. How to use 'kerneld' (dynamic module loading). Updated 19 July 1997.
- LBX mini-HOWTO, by Paul D. Smith <psmith@baynetworks.com>. How to use Low-Bandwidth X (LBX). Updated 11 December 1997.
- LILO mini-HOWTO, by Alessandro Rubini <rubini@linux.it>. Examples of typical LILO installations. Updated 16 August 1998.
- Large Disk mini-HOWTO, by Andries Brouwer <aeb@cwi.nl>. How to use disks with > 1024 cylinders. Updated 12 September 1999.
- Leased Line mini-HOWTO, by Rob van der Putten <rob@sput.webster.nl>. How to setup leased line

TOC

modems. Updated July 1998.

- Linux+DOS+Win95+OS2 mini-HOWTO, by Mike Harlan <r3mdh@raex.com>. How to use Linux and DOS and OS/2 and Win95 together. Updated 11 November 1997.
- Linux+FreeBSD mini-HOWTO, by Niels Kristian Bech Jensen <nkbj@image.dk> How to use Linux and FreeBSD together. Updated 15 January 1999.
- Linux+NT-Loader mini-HOWTO, by Bernd Reichert <reichert@dial.eunet.ch>. How to use Linux and the Windows NT boot loader together. Updated 2 September 1997.
- Linux+Win95 mini-HOWTO, by Jonathan Katz <jkatz@in.net>. How to use Linux and Windows95 together. Updated 26 October 1996.
- Loadlin+Win95 mini-HOWTO, by Chris Fischer <protek@brigadoon.com>. How to use Linux and Windows95 together, using loadlin. Updated 13 March 1999.
- Mac Terminal mini-HOWTO, by Robert Kiesling <kiesling@terracom.net>. How to use an Apple Macintosh as a serial terminal. Updated 9 November 1997.
- Mail Queue mini-HOWTO, by Leif Erlingsson <leif@lege.com>. How to queue remote mail and deliver local mail. Updated 3 September 1997.
- Mail2News mini-HOWTO, by Robert Hart <iweft@ipax.com.au>. How to set up a mail to news gateway. Updated 1 July 1999.
- Man Page mini-HOWTO, by Jens Schweikhardt <schweikh@noc.dfn.de>. How to write man pages. Updated July 1998.
- Modules mini-HOWTO, by Riley H. Williams <rhw@bigfoot.com>. How to setup and configure kernel modules. Updated 14 November 1997.
- Multiboot using LILO mini-HOWTO, by Renzo Zanelli <rzanelli@southeast.net>. Howto multiboot between Windows 95, Windows NT, and Linux. Updated 26 March 1998.
- NCD X Terminal mini-HOWTO, by Ian Hodge <ihodge@nortel.ca>. Describes how to connect an NCD X terminal to a UNIX host. Updated 3 April 1998.
- NFS-Root mini-HOWTO, by Andreas Kostyrka <andreas@ag.or.at>. How to set up diskless Linux machines. Updated 8 August 1997.
- NFS-Root-Client mini-HOWTO, by Ofer Maor <ofer@hadar.co.il>. How to set up diskless Linux machines using NFS. Updated 02 February 1999.
- Netrom-Node mini-HOWTO, by Karl Larsen <k5di@yahoo.com>. How to setup the ax25-utilities package for Amateur Radio such making Netrom Nodes. Updated 19 October 1998.
- Netscape+Proxy mini-HOWTO, by Sarma Seetamraju <sarma@usa.net>. How to set up a proxy server for Netscape. Updated 15 August 1997.
- Netstation mini-HOWTO, by Kris Buytaert <Kris.Buytaert@advalvas.be>. How to hook up a IBM Netstation to your local network using a Linux box as server. Updated 22 February 1998.
- News Leafsite mini-HOWTO, by Florian Kuehnert <sutok@gmx.de>. How to set up a leaf news site. Updated 4 January 1998.
- Offline Mailing mini-HOWTO, by Gunther Voet <freaker@tuc.ml.org>. How to set up email addresses without a dedicated Internet connection. Updated 4 June 1998.
- PLIP mini-HOWTO, by Andrea Controzzi <controzz@cli.di.unipi.it>. How to set up PLIP (Parallel Line Interface Protocol). Updated 12 March 1998.

TOC

- Partition mini-HOWTO, by Kristian Koehntopp <kris@koehntopp.de>. How to choose disk partitions. Updated 3 November 1997.
- Partition Rescue mini-HOWTO, by Rolf Klausen <rolfk@romsdal.vgs.no>. How to rescue deleted Linux partitions. Updated 22 October 1997.
- Path mini-HOWTO, by Esa Turtiainen <etu@dna.fi>. How to use the PATH environment variable. Updated 15 November 1997.
- Pre-installation Checklist mini-HOWTO, by S. Parthasarathy <algolog@hd1.vsnl.net.in>. Pre-installation checklist and questionnaire. Updated 29 August 1998.
- Process Accounting mini-HOWTO, by Albert M.C. Tam <bertie@scn.org>. How to set up process accounting. Updated 8 August 1997.
- Proxy ARP Subnet mini-HOWTO, by Bob Edwards <bob@faceng.anu.edu.au>. How to use proxy ARP with subnetting. Updated August 1997.
- Public Web Browser mini-HOWTO, by Donald B. Marti Jr. <dmarti@best.com>. How to set up a guest account to use a WWW browser. Updated 5 January 1998.
- Qmail+MH mini-HOWTO, by Christopher Richardson <rdn@tara.n.eunet.de>. How to install qmail and MH. Updated 5 March 1998.
- Quota mini-HOWTO, by Albert M.C. Tam <bertie@scn.org>. How to set up disk quotas. Updated 8 August 1997.
- RCS mini-HOWTO, by Robert Kiesling <kiesling@terracom.net>. How to use RCS (Revision Control System). Updated 14 August 1997.
- RPM+Slackware mini-HOWTO, by Dave Whitinger <dave@whitinger.net>. How to install the Red Hat Package Manager (RPM) under Slackware. Updated 13 April 1998.
- RedHat CD mini-HOWTO, by Morten Kjeldgaard <mok@imsb.au.dk> Peter von der Ahé <pahe+rhcd@daimi.au.dk>. How to make your own CDs from the Red Hat Linux distribution equivalent to the ones commercially available from Red Hat. Updated 9 September 1998.
- Remote Boot mini-HOWTO, by Marc Vuilleumier Stückelberg <Marc.VuilleumierStuckelberg@cui.unige.ch>. How to set up a server-based boot selector. Updated February 1999.
- Remote X Apps mini-HOWTO, by Vincent Zweije <zweije@xs4all.nl>. How to run remote X applications. Updated 14 July 1998.
- SLIP-PPP Emulator mini-HOWTO, by Irish <irish@eskimo.com>. How to use SLIP-PPP emulators with Linux. Updated 7 August 1997.
- Sendmail Address Rewrite mini-HOWTO, by Thomas Roessler <roessler@guug.de>. Set up sendmail's configuration file for the home user's dial-up access. Updated 6 May 1998.
- Sendmail+UUCP mini-HOWTO, by Jamal Hadi Salim <jamal@glcom.com>. How to use sendmail and UUCP together. Updated August 1998.
- Secure POP via SSH mini-HOWTO, by Manish Singh <yosh@gimp.org>. How to set up secure POP connections using ssh. Updated 30 September 1998.
- Small Memory mini-HOWTO, by Todd Burgess <tburgess@uoguelph.ca>. How to run Linux on a system with a small amount of memory. Update 29 October 1997.
- Software RAID mini-HOWTO, by Linas Vepstas <linas@fc.net>. How to configure software RAID. Updated 21 November 1998.

TOC

- Soundblaster AWE mini-HOWTO, by Marcus Brinkmann <Marcus.Brinkmann@ruhr-uni-bochum.de>. How to install the Soundblaster AWE 32/64. Updated 11 January 1998.
- StarOffice mini-HOWTO, by Matthew Borowski <mkb@poboxes.com>. Information on installing the StarOffice suite. Updated 2 June 1998.
- Term Firewall mini-HOWTO, by Barak Pearlmutter <bap@cs.unm.edu>. How to use 'term' over a firewall. Updated 15 July 1997.
- TkRat mini-HOWTO, by Dave Whitinger <dave@whitinger.net>. How to install and use the TkRat mail program. Updated 2 February 1998.
- Token Ring mini-HOWTO, by Mike Eckhoff <mike.e@emissary.aus-etc.com>. How to use token ring cards. Updated 7 January 1998.
- Ultra-DMA mini-HOWTO, by Brion Vibber <brion@pobox.com>. How to use Ultra-DMA drives and controllers. Updated 27 May 1999.
- Update mini-HOWTO, by Stein Gjoen <sgjoen@nyx.net>. How to stay updated about Linux development. Updated 3 February 1998.
- Upgrade mini-HOWTO, by Greg Louis <glouis@dynamicro.on.ca>. How to upgrade your Linux distribution. Updated 6 June 1996.
- VAIO mini-HOWTO, by Hideki Saito <hideki@chatlink.com>. This document explains installation of Linux on Sony VAIO computers. Updated 16 September 1998.
- Vesafb mini-HOWTO, by Alex Buell <alex.buell@tahallah.demon.co.uk>. How to use the vesafb device. Updated 2 August 1998.
- VPN mini-HOWTO, by Árpád Magosányi <mag@bunuel.tii.matav.hu>. How to set up a VPN (Virtual Private Network). Updated 7 August 1997.
- Visual Bell mini-HOWTO, by Alessandro Rubini <rubini@linux.it>. How to disable audible bells, and enable visual bells. Updated 11 November 1997.
- Windows Modem Sharing mini-HOWTO, by Friedemann Baitinger <baiti@toplink.net>. How to setup Windows to use a shared modem on a Linux machine. Updated 2 November 1997.
- WordPerfect mini-HOWTO, by Wade Hampton <whampton@staffnet.com>. How to set up WordPerfect for Linux. Updated 13 August 1997.
- X Big Cursor mini-HOWTO, by Joerg Schneider <schneid@ira.uka.de>. How to use enlarged cursors with XWindows. Updated 11 August 1997.
- XFree86-XInside mini-HOWTO, by Marco Melgazzi <marco@techie.com>. How to convert XFree86 to XInside modelines. Updated September 1997.
- xterm Title mini-HOWTO, by Ric Lister <ric@giccs.georgetown.edu>. How to put strings into the titlebar of an xterm. Updated 7 January 1998.
- ZIP Install mini-HOWTO, by John Wiggins <jwiggins@comp.uark.edu>. How to install Linux onto a ZIP drive. Updated 10 January 1999.
- ZIP Drive mini-HOWTO, by Kyle Dansie <dansie@ibm.net>. Provides a quick reference quide on setting up and using the Iomega ZIP drive with Linux. Updated 10 January 1999.

3.3 Special HOWTOs

The *High Availability HOWTO*, by Harald Milz <hm@seneca.muc.de> is available at http://metalab.unc.edu/pub/Linux/ALPHA/linux-ha/High-Availability-HOWTO.html. It is not included with the HOWTO collection because it relies on figures and cannot be distributed in all supported formats.

The *Graphics mini-HOWTO*, by Michael J. Hammel <mjhammel@graphics-muse.org> is available at http://www.graphics-muse.org/linux/lgh.html. It is not included with the HOWTO collection because it needs to use a lot of images, which don't translate to other output formats.

3.4 Unmaintained HOWTOs and mini-HOWTOs

There are a number of unmaintained documents at ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/unmaintained. These are kept around since old documentation is sometimes better than none. However, you should *be aware that you are reading old documentation*.

---

5. Copyright

Copyright (c) 1995 - 1998 by Tim Bynum.

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at `linux-howto@metalab.unc.edu` via email.

The Linux 3Dfx HOWTO

Bernd Kreimeier (bk@gamers.org)

v1.16, 6 February 1998

This document describes 3Dfx graphics accelerator chip support for Linux. It lists some supported hardware, describes how to configure the drivers, and answers frequently asked questions.

1. Introduction

- 1.1 Contributors and Contacts
- 1.2 Acknowledgments
- 1.3 Revision History
- 1.4 New versions of this document
- 1.5 Feedback
- 1.6 Distribution Policy

2. Graphics Accelerator Technology

- 2.1 Basics
- 2.2 Hardware configuration
- 2.3 A bit of Voodoo Graphics (tm) architecture

3. Installation

- 3.1 Installing the board
- 3.2 Setting up the Displays
- 3.3 Installing the Glide distribution

4. Answers To Frequently Asked Questions

TOC

5. FAQ: Requirements?

- 5.1 What are the system requirements?
- 5.2 Does it work with Linux-Alpha?
- 5.3 Which 3Dfx chipsets are supported?
- 5.4 Is the Voodoo Rush (tm) supported?
- 5.5 Which boards are supported?
- 5.6 How do boards differ?
- 5.7 What about AGP?

6. FAQ: Voodoo Graphics (tm)? 3Dfx?

- 6.1 Who is 3Dfx?
- 6.2 Who is Quantum3D?
- 6.3 What is the Voodoo Graphics (tm)?
- 6.4 What is the Voodoo Rush (tm)?
- 6.5 What is the Voodoo 2 (tm)?
- 6.6 What is VGA pass-though?
- 6.7 What is Texelfx or TMU?
- 6.8 What is a Pixelfx unit?
- 6.9 What is SLI mode?
- 6.10 Is there a single board SLI setup?
- 6.11 How much memory? How many buffers?
- 6.12 Does the Voodoo Graphics (tm) do 24 or 32 bit color?
- 6.13 Does the Voodoo Graphics (tm) store 24 or 32 bit z-buffer per pixel?
- 6.14 What resolutions does the Voodoo Graphics (tm) support?
- 6.15 What texture sizes are supported?
- 6.16 Does the Voodoo Graphics (tm) support paletted textures?
- 6.17 What about overclocking?
- 6.18 Where could I get additional info on Voodoo Graphics (tm)?

TOC

7. FAQ: Glide? TexUS?

- 7.1 What is Glide anyway?
- 7.2 What is TexUS?
- 7.3 Is Glide freeware?
- 7.4 Where do I get Glide?
- 7.5 Is the Glide source available?
- 7.6 Is Linux Glide supported?
- 7.7 Where could I post Glide questions?
- 7.8 Where to send bug reports?
- 7.9 Who is maintaining it?
- 7.10 How can I contribute to Linux Glide?
- 7.11 Do I have to use Glide?
- 7.12 Should I program using the Glide API?
- 7.13 What is the Glide current version?
- 7.14 Does it support multiple Texelfx already?
- 7.15 Is Linux Glide identical to DOS/Windows Glide?
- 7.16 Where to I get information on Glide?
- 7.17 Where to get some Glide demos?
- 7.18 What is ATB?

8. FAQ: Glide and XFree86?

- 8.1 Does it run with XFree86?
- 8.2 Does it only run full screen?
- 8.3 What is the problem with AT3D/Voodoo Rush (tm) boards?
- 8.4 What about GLX for XFree86?
- 8.5 Glide and commerical X Servers?
- 8.6 Glide and SVGA?
- 8.7 Glide and GGI?

TOC

9. FAQ: OpenGL/Mesa?

- 9.1 What is OpenGL?
- 9.2 Where to get additional information on OpenGL?
- 9.3 Is Glide an OpenGL implementation?
- 9.4 Is there an OpenGL driver from 3Dfx?
- 9.5 Is there a commercial OpenGL for Linux and 3Dfx?
- 9.6 What is Mesa?
- 9.7 Does Mesa work with 3Dfx?
- 9.8 How portable is Mesa with Glide?
- 9.9 Where to get info on Mesa?
- 9.10 Where to get information on Mesa Voodoo?
- 9.11 Does Mesa support multitexturing?
- 9.12 Does Mesa support single pass trilinear mipmapping?
- 9.13 What is the Mesa "Window Hack"?
- 9.14 How about GLUT?

10. FAQ: But Quake?

- 10.1 What about that 3Dfx GL driver for Quake?
- 10.2 Is there a 3Dfx based glQuake for Linux?
- 10.3 Does glQuake run in an XFree86 window?
- 10.4 Known Linux Quake problems?
- 10.5 Know Linux Quake security problems?
- 10.6 Does LinuxQuake use multitexturing?
- 10.7 Where can I get current information on Linux glQuake?

11. FAQ: Troubleshooting?

- 11.1 Has this hardware been tested?
- 11.2 Failed to change I/O privilege?
- 11.3 Does it work without root privilege?
- 11.4 Displayed images looks awful (single screen)?
- 11.5 The last frame is still there (single or dual screen)?

- 11.6 Powersave kicks in (dual screen)?
- 11.7 My machine seem to lock (X11, single screen)?
- 11.8 My machine locks (single or dual screen)?
- 11.9 My machine locks (used with S3 VGA board)?
- 11.10 No address conflict, but locks anyway?
- 11.11 Mesa runs, but does not access the board?
- 11.12 Resetting dual board SLI?
- 11.13 Resetting single board SLI?

1. Introduction

This is the Linux 3Dfx HOWTO document. It is intended as a quick reference covering everything you need to know to install and configure 3Dfx support under Linux. Frequently asked questions regarding the 3Dfx support are answered, and references are given to some other sources of information on a variety of topics related to computer generated, hardware accelerated 3D graphics.

This information is only valid for Linux on the Intel platform. Some information may be applicable to other processor architectures, but I have no first hand experience or information on this. It is only applicable to boards based on 3Dfx technology, any other graphics accelerator hardware is beyond the scope of this document.

1.1 Contributors and Contacts

This document would not have been possible without all the information contributed by other people - those involved in the Linux Glide port and the beta testing process, in the development of Mesa and the Mesa Voodoo drivers, or rewieving the document on behalf of 3Dfx and Quantum3D. Some of them contributed entire sections to this document.

Daryll Strauss daryll@harlot.rb.ca.us did the port, Paul J. Metzger pjm@rbd.com modified the Mesa Voodoo driver (written by David Bucciarelli tech.hmw@plus.it) for Linux, Brian Paul brianp@RA.AVID.COM integrated it with his famous Mesa library. With respect to Voodoo Graphics (tm) accelerated Mesa, additional thanks has to go to Henri Fousse, Gary McTaggart, and the maintainer of the 3Dfx Mesa for DOS, Charlie Wallace Charlie.Wallace@unistudios.com. The folks at 3Dfx, notably Gary Sanders, Rod Hughes, and Marty Franz, provided valuable input, as did Ross Q. Smith of Quantum3D. The pages on the Voodoo Extreme and Operation 3Dfx websites provided useful info as well, and in some case I relied on the 3Dfx local Newsgroups. The Linux glQuake2 port that uses Linux Glide and Mesa is maintained by Dave Kirsch zoid@idsoftware.com. Thanks to all those who sent e-mail regarding corrections and updates, and special thanks to Mark Atkinson for reminding me of the dual cable setup.

Thanks to the SGML-Tools package (formerly known as Linuxdoc-SGML), this HOWTO is available in several formats, all generated from a common source file. For information on SGML-Tools see its homepage at pobox.com/~cg/sgmltools.

1.2 Acknowledgments

3Dfx, the 3Dfx Interactive logo, Voodoo Graphics (tm), and Voodoo Rush (tm) are registered trademarks of 3Dfx Interactive, Inc. Glide, TexUS, Pixelfx and Texelfx are trademarks of 3Dfx Interactive, Inc. OpenGL is a registered trademark of Silicon Graphics. Obsidian is a trademark of Quantum3D. Other product names are trademarks of the respective holders, and are hereby considered properly

acknowledged.

1.3 Revision History

Version 1.03

First version for public release.

Version 1.16

Current version v1.16 6 February 1998.

1.4 New versions of this document

You will find the most recent version of this document at www.gamers.org/dEngine/xf3D/.

New versions of this document will be periodically posted to the comp.os.linux.answers newsgroup. They will also be uploaded to various anonymous ftp sites that archive such information including ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/.

Hypertext versions of this and other Linux HOWTOs are available on many World-Wide-Web sites, including sunsite.unc.edu/LDP/. Most Linux CD-ROM distributions include the HOWTOs, often under the `/usr/doc/`directory, and you can also buy printed copies from several vendors.

If you make a translation of this document into another language, let me know and I'll include a reference to it here.

1.5 Feedback

I rely on you, the reader, to make this HOWTO useful. If you have any suggestions, corrections, or comments, please send them to me (bk@gamers.org), and I will try to incorporate them in the next revision. Please add `HOWTO 3Dfx` to the Subject-line of the mail, so procmail will dump it in the appropriate folder.

Before sending bug reports or questions, *please read all of the information in this HOWTO*, and *send detailed information about the problem*.

If you publish this document on a CD-ROM or in hardcopy form, a complimentary copy would be appreciated. Mail me for my postal address. Also consider making a donation to the Linux Documentation Project to help support free documentation for Linux. Contact the Linux HOWTO coordinator, Tim Bynum (linux-howto@sunsite.unc.edu), for more information.

1.6 Distribution Policy

Copyright (c) 1997, 1998 by Bernd Kreimeier. This document may be distributed under the terms set forth in the LDP license at sunsite.unc.edu/LDP/COPYRIGHT.html.

This HOWTO is free documentation; you can redistribute it and/or modify it under the terms of the LDP license. This document is distributed in the hope that it will be useful, but **without any warranty**; without even the implied warranty of **merchantability** or **fitness for a particular purpose**. See the LDP license for more details.

---

2. Graphics Accelerator Technology

2.1 Basics

This section gives a *very* cursory overview of computer graphics accelerator technology, in order to help you understand the concepts used later in the document. You should consult e.g. a book on OpenGL in order to learn more.

2.2 Hardware configuration

Graphics accelerators come in different flavors: either as a separate PCI board that is able to pass through the video signal of a (possibly 2D or video accelerated) VGA board, or as a PCI board that does both VGA and 3D graphics (effectively replacing older VGA controllers). The 3Dfx boards based on the Voodoo Graphics (tm) belong to the former category. We will get into this again later.

If there is no address conflict, any 3D accelerator board could be present under Linux without interfering, but in order to access the accelerator, you will need a driver. A combined 2D/3D accelerator might behave differently.

2.3 A bit of Voodoo Graphics (tm) architecture

Usually, accessing texture memory and frame/depth buffer is a major bottleneck. For each pixel on the screen, there are at least one (nearest), four (bi-linear), or eight (tri-linear mipmapped) read accesses to texture memory, plus a read/write to the depth buffer, and a read/write to frame buffer memory.

The Voodoo Graphics (tm) architecture separates texture memory from frame/depth buffer memory by introducing two separate rendering stages, with two corresponding units (Pixelfx and Texelfx), each having a separate memory interface to dedicated memory. This gives an above-average fill rate, paid for restrictions in memory management (e.g. unused framebuffer memory can not be used for texture caching).

Moreover, a Voodoo Graphics (tm) could use two TMU's (texture management or texelfx units), and finally, two Voodoo Graphics (tm) could be combined with a mechanism called Scan-Line Interleaving (SLI). SLI essentially means that each Pixelfx unit effectively provides only every other scanline, which decreases bandwidth impact on each Pixelfx' framebuffer memory.

3. Installation

Configuring Linux to support 3Dfx accelerators involves the following steps:

1. Installing the board.
2. Installing the Glide distribution.
3. Compiling, linking and/or running the application.

The next sections will cover each of these steps in detail.

3.1 Installing the board

Follow the manufacturer's instructions for installing the hardware or have your dealer perform the installation. It should not be necessary to select settings for IRQ, DMA channel, either Plug&Pray (tm) or factory defaults should work. The add-on boards described here are memory mapped devices and do not use IRQ's. The only kind of conflict to avoid is memory overlap with other devices.

As 3Dfx does not develop or sell any boards, do not contact them on any problems.

Troubleshooting the hardware installation

To check the installation and the memory mapping, do `cat /proc/pci`. The output should contain something like

```
  Bus  0, device  12, function  0:
    VGA compatible controller: S3 Inc. Vision 968 (rev 0).
      Medium devsel.  IRQ 11.
      Non-prefetchable 32 bit memory at 0xf4000000.

  Bus  0, device   9, function  0:
    Multimedia video controller: Unknown vendor Unknown device (rev 2).
      Vendor id=121a. Device id=1.
      Fast devsel.  Fast back-to-back capable.
      Prefetchable 32 bit memory at 0xfb000000.
```

for a Diamond Monster 3D used with a Diamond Stealth-64. Additionally a `cat /proc/cpuinfo /proc/meminfo` might be helpfull for tracking down conflicts and/or submitting a bug report.

With current kernels, you will probably get a boot warning like

```
Jun 12 12:31:52 hal kernel: Warning : Unknown PCI device (121a:1).
Please read include/linux/pci.h
```

which could be safely ignored. If you happen to have a board not very common, or have encountered a ne revision, you should take the time to follow the advice in `/usr/include/linux/pci.h` and send al necessary information to linux-pcisupport@cao-vlsi.ibp.fr.

TOC

If you experience any problems with the board, you should try to verify that DOS and/or Win95 or NT support works. You will probably not receive any useful response from a board manufacturer on a bug report or request regarding Linux. Having dealt with the Diamond support e-mail system, I would not expect useful responses for other operating systems either.

Configuring the kernel

There is no kernel configuration necessary, as long as PCI support is enabled. The Linux Kernel HOWTO should be consulted for the details of building a kernel.

Configuring devices

The current drivers do not (yet) require any special devices. This is different from other driver developments (e.g. the sound drivers, where you will find a `/dev/dsp` and `/dev/audio`). The driver uses the `/dev/mem` device which should always be available. In consequence, you need to use `setuid` or root privileges to access the accelerator board.

3.2 Setting up the Displays

There are two possible setups with add-on boards. You could either pass-through the video signal from your regular VGA board via the accelerator board to the display, or you could use two displays at the same time. Rely to the manual provided by the board manufacturer for details. Both configurations have been tried with the Monster 3D board.

Single screen display solution

This configuration allows you to check basic operations of the accelerator board - if the video signal is not transmitted to the display, hardware failure is possible.

Beware that the video output signal might deteoriate significantly if passed through the video board. To a degree, this is inevitable. However, reviews have complained about below-average of the cables provided e.g. with the Monster 3D, and judging from the one I tested, this has not changed.

There are other pitfalls in single screen configurations. Switching from the VGA display mode to the accelerated display mode will change resolution and refresh rate as well, even if you are using 640x480 e.g. with X11, too. Moreover, if you are running X11, your application is responsible for demanding all keyboard and mouse events, or you might get stuck because of changed scope and exposure on the X11 display (that is effectively invisible when the accelerated mode is used) You could use SVGA console mode instead of X11.

If you are going to use a single screen configuration and switch modes often, remember that your monitor hardware might not enjoy this kind of use.

TOC

Single screen dual cable setup

Some high end monitors (e.g. the EIZO F-784-T) come with two connectors, one with 5 BNC connectors for RGB, HSync, VSync, the other e.g. a regular VGA or a 13W3 Sub-D VGA. These displays usually also feature a front panel input selector to safely switch from one to the other. It is thus possible to use e.g. a VGA-to-BNC cable with your high end 2D card, and a VGA-to-13W3 Sub-D cable with your 3Dfx, and effectively run dual screen on one display.

Dual screen display solution

The accelerator board does not need the VGA input signal. Instead of routing the common video output through the accelerator board, you could attach a second monitor to its output, and use both at the same time. This solution is more expensive, but gives best results, as your main display will still be hires and without the signal quality losses involved in a pass-through solution. In addition, you could use X11 and the accelerated full screen display in parallel, for development and debugging.

A common problem is that the accelerator board will not provide any video signal when not used. In consequence, each time the graphics application terminates, the hardware screensave/powersave might kick in depending on your monitors configuration. Again, your hardware might not enjoy being treated like this. You should use

```
setenv SST_DUALSCREEN 1
```

to force continued video output in this setup.

3.3 Installing the Glide distribution

The Glide driver and library are provided as a single compressed archive. Use `tar` and `gzip` to unpack, and follow the instructions in the README and INSTALL accompanying the distribution. Read the install script and run it. Installation puts everything in /usr/local/glide/include,lib,bin and sets the ld.conf to look there. Where it installs and setting ld.conf are independent actions. If you skip the ld.conf step then you need the LD_LIBRARY_PATH.

You will need to install the header files in a location available at compile time, if you want to compile your own graphics applications. If you do not want to use the installation as above (i.e. you insist on a different location), make sure that any application could access the shared libary at runtime, or you will get a response like `can't load library 'libglide.so'`.

Using the detect program

There is a `bin/detect` program in the distribution (the source is not available). You have to run it as root, and you will get something like

```
slot  vendorId   devId   baseAddr0  command  description
----  --------  ------  ----------  -------  -----------
  00    0x8086  0x122d  0x00000000   0x0006  Intel:430FX (Triton)
```

```
 07     0x8086  0x122e  0x00000000   0x0007  Intel:ISA bridge
 09     0x121a  0x0001  0xfb000008   0x0002  3Dfx:video multimedia adap
 10     0x1000  0x0001  0x0000e401   0x0007  ???:SCSI bus controller
 11     0x9004  0x8178  0x0000e001   0x0017  Adaptec:SCSI bus controller
 12     0x5333  0x88f0  0xf4000000   0x0083  S3:VGA-compatible display co
```

as a result. If you do not have root privileges, the program will bail out with

```
Permission denied: Failed to change I/O privilege. Are you root?
```

output might come handy for a bug report as well.

Using the test programs

Within the Glide distribution, you will find a folder with test programs. Note that these test programs are under 3Dfx copyright, and are legally available for use only if you have purchased a board with a 3Dfx chipset. See the LICENSE file in the distribution, or their web site www.3dfx.com for details.

It is recommend to compile and link the test programs even if there happen to be binaries in the distribution. Note that some of the programs will requires some files like `alpha.3df` from the distribution to be available in the same folder. All test programs use the 640x480 screen resolution. Some will request a veriety of single character inputs, others will just state `Press A Key To Begin Test`. Beware of loss of input scope if running X11 on the same screen at the same time.

See the README.test for a list of programs, and other details.

4. Answers To Frequently Asked Questions

The following section answers some of the questions that (will) have been asked on the Usenet news groups and mailing lists. The FAQ has been subdivided into several parts for convenience, namely

- FAQ: Requirements?
- FAQ: Voodoo Graphics (tm)? 3Dfx?
- FAQ: Glide?
- FAQ: Glide and SVGA?
- FAQ: Glide and XFree86?
- FAQ: Glide versus OpenGL/Mesa?
- FAQ: But Quake?
- FAQ: Troubleshooting?

Each section lists several questions and answers, which will hopefully address most problems.

5. FAQ: Requirements?

5.1 What are the system requirements?

A Linux PC, PCI 2.1 compliant, a monitor capable of 640x480, and a 3D accelerator board based on the 3Dfx Voodoo Graphics (tm). It will work on a P5 or P6, with or without MMX. The current version does not use MMX, but it has some optimized code paths for P6.

At one point, some 3Dfx statements seemed to imply that using Linux Glide required using a RedHat distribution. Note that while Linux Glide has originally been ported in a RedHat 4.1 environment, it has been used and tested with many other Linux distributions, including homebrew, Slackware, and Debian 1.3.1.

5.2 Does it work with Linux-Alpha?

There is currently no Linux Glide distribution available for any platform besides i586. As the Glide sources are not available for distribution, you will have to wait for the binary. Quantum3D has DEC Alpha support announced for 2H97. Please contact Daryll Strauss if you are interested in supporting this.

There is also the issue of porting the the assembly modules. While there are alternative C paths in the code, the assembly module in Glide (essentially triangle setup) offered significant performance gains depending on the P5 CPU used.

5.3 Which 3Dfx chipsets are supported?

Currently, the 3Dfx Voodoo Graphics (tm) chipset is supported under Linux. The Voodoo Rush (tm) chipset is not yet supported.

5.4 Is the Voodoo Rush (tm) supported?

The current port of Glide to Linux does not support the Voodoo Rush (tm). An update is in the works.

The problem is that at one point the Voodoo Rush (tm) driver code in Glide depended on Direct Draw. There was an SST96 based DOS portion in the library that could theoretically be used for Linux, as soon as all portions residing in the 2D/Direct Draw/D3D combo driver are replaced.

Thus Voodoo Rush (tm) based boards like the *Hercules Stingray 128/3D* or *Intergraph Intense Rush* are not supported yet.

TOC

5.5 Which boards are supported?

There are no officially supported boards, as 3Dfx does not sell any boards. This section does not attempt to list all boards, it will just give an overview, and will list only boards that have been found to cause trouble.

It is important to recognize that Linux support for a given board does not only require a driver for the 3D accelerator component. If a board features its own VGA core as well, support by either Linux SVGA or XFree86 is required as well (see section about Voodoo Rush (tm) chipset). Currently, an add-on solution is recommended, as it allows you to choose a regular graphics board well supported for Linux. There are other aspects discussed below.

All Quantum3D Obsidian boards, independend of texture memory, frame buffer memory, number of Pixelfx and Texelfx units, and SLI should work. Same for all other Voodoo Graphics (tm) based boards, like Orchid Righteous 3D, Canopus Pure 3D, Flash 3D, and Diamond Monster 3D. Voodoo Rush (tm) based boards are not yet supported.

Boards that are not based on 3Dfx chipsets (e.g. manufactured by S3, Matrox, 3Dlabs, Videologic) do *not* work with the 3Dfx drivers and are beyond the scope of this document.

5.6 How do boards differ?

As the board manufacturers are using the same chipset, any differences are due to board design. Examples are quality of the pass-through cable and connectors (reportedly, Orchid provided better quality than Diamond), availability of a TV-compliant video signal output (Canopus Pure 3D), and, most notably, memory size on board.

Most common were boards for games with 2MB texture cache and 2 MB framebuffer memory, however, the Canopus Pure3D comes with a maximal 4 MB texture cache, which is an advantage e.g. with games using dynamically changed textures, and/or illumation textures (Quake, most notably). The memory architecture of a typical Voodoo Graphics (tm) board is described below, in a separate section.

Quantum 3D offers the widest selection of 3Dfx-based boards, and is probably the place to go if you are looking for a high end Voodoo Graphics (tm) based board configuration. Quantum 3D is addressing the visual simulation market, while most of the other vendors are only targetting the consumer-level PC-game market.

5.7 What about AGP?

There is no Voodoo Graphics (tm) or Voodoo Rush (tm) AGP board that I am aware of. I am not aware of AGP support under Linux, and I do not know whether upcmong AGP boards using 3Dfx technology might possibly be supported with Linux.

---

6. FAQ: Voodoo Graphics (tm)? 3Dfx?

6.1 Who is 3Dfx?

3Dfx is a San Jose based manufacturer of 3D graphics accelerator hardware for arcade games, game consoles, and PC boards. Their official website is www.3dfx.com. 3Dfx does not sell any boards, but other companies do, e.g. Quantum3D.

6.2 Who is Quantum3D?

Quantum3D started as a 3Dfx spin-off, manufacturing high end accelerator boards based on 3Dfx chip technology for consumer and business market, and supplying arcade game technology. See their home page at www.quantum3d.com for additional information. For general inquiries regarding Quantum3D, please send mail to info@quantum3d.

6.3 What is the Voodoo Graphics (tm)?

The Voodoo Graphics (tm) is a chipset manufactured by 3Dfx. It is used in hardware acceleration boards for the PC. See the HOWTO section on supported hardware.

6.4 What is the Voodoo Rush (tm)?

The Voodoo Rush (tm) is a derivate of the Voodoo Graphics (tm) that has an interface to cooperate with a 2D VGA video accelerator, effectively supporting accelerated graphics in windows. This combo is currently not supported with Linux.

6.5 What is the Voodoo 2 (tm)?

The Voodoo 2 (tm) is the successor of the Voodoo Graphics (tm) chipset, featuring several improvements. It is announced for late March 1998, and annoucements of Voodoo 2 (tm) based boards have been published e.g. by Quantum 3D, by Creative Labs, Orchid Technologies, and Diamond Multimedia.

The Voodoo 2 (tm) is supposed to be backwards compatible. However, a new version of Glide will have to be ported to Linux.

TOC

6.6 What is VGA pass-though?

The Voodoo Graphics (tm) (but not the Voodoo Rush (tm)) boards are add-on boards, meant to be used with a regular 2D VGA video accelerator board. In short, the video output of your regular VGA board is used as input for the Voodoo Graphics (tm) based add-on board, which by default passes it through to the display also connected to the Voodoo Graphics (tm) board. If the Voodoo Graphics (tm) is used (e.g. by a game), it will disconnect the VGA input signal, switch the display to a 640x480 fullscreen mode with the refresh rate configured by SST variables and the application/driver, and generate the video signal itself. The VGA doesn't need to be aware of this, and won't be.

This setup has several advantages: free choice of 2D VGA board, which is an issue with Linux, as XFree86 drivers aren't available for all chipsets and revisions, and a cost effective migration path to accelerated 3D graphics. It also has several disadvantages: an application using the Voodoo Graphics (tm) might not re-enable video output when crashing, and regular VGA video signal deteoriates in the the pass-through process.

6.7 What is Texelfx or TMU?

Voodoo Graphics (tm) chipsets have two units. The first one interfaces the texture memory on the board, does the texture mapping, and ultimately generates the input for the second unit that interfaces the framebuffer. This one is called Texelfx, aka Texture Management Unit, aka TMU. The neat thing about this is that a board can use two Texelfx instead of only one, like some of the Quantum3D Obsidian boards did, effectively doubling the processing power in some cases, depending on the application.

As each Texelfx can address 4MB texture memory, a dual Texelfx setup has an effective texture cache of up to 8MB. This can be true even if only one Texelfx is actually needed by a particular application, as textures can be distributed to both Texelfx, which are used depending on the requested texture. Both Texelfx are used together to perform certain operations as trilinear filtering and illumination texture/lightmap passes (e.g. in glQuake) in a single pass instead of the two passes that are required with only one Texelfx. To actually exploit the theoretically available speedup and cache size increase, a Glide application has to use both Texelfx properly.

The two Texelfx can not be used separately to each draw a textured triangle at the same time. A triangle is always drawn using whatever the current setup is, which can be to use both Texelfx for a single pass operation combining two textures, or one Texelfx for only a single texture. Each Texelfx can only access its own memory.

6.8 What is a Pixelfx unit?

Voodoo Graphics (tm) chipsets have two units. The second one interfaces the framebuffer and ultimately generates the depth buffer and pixel color updates. This one is called Pixelfx. The neat thing here is that two Pixelfx units can cooperate in SLI mode, like with some of the Quantum3D Obsidian boards, effectively doubling the frame rate.

TOC

6.9 What is SLI mode?

SLI means "Scanline Interleave". In this mode, two Pixelfx are connected and render in alternate turns, one handling odd, the other handling even scanlines of the actual output. Inthis mode, each Pixelfx stores only half of the image and half of the depth buffer data in its own local framebuffer, effectively doubling the number of pixels.

The Pixelfx in question can be on the same board, or on two boards properly connected. Some Quantum3D Obsidian boards support SLI with Voodoo Graphics (tm).

As two cards can decode the same PCI addresses and receive the same data, there is not necessarily additional bus bandwidth required by SLI. On the other hand, texture data will have to be replicated on both boards, thus the amount of texture memory effectively stays the same.

6.10 Is there a single board SLI setup?

There are now two types of Quantum3D SLI boards. The intial setup used two boards, two PCI slots, and an interconnect (e.g. the Obsidian 100-4440). The later revision which performs identically is contained on one full-length PCI board (e.g. Obsidian 100-4440SB). Thus a single board SLI solution is possible, and has been done.

6.11 How much memory? How many buffers?

The most essential difference between different boards using the Voodoo Graphics (tm) chipset is the amount and organization of memory. Quantum3D used a three digit scheme to descibe boards. Here is a slightly modifed one (anticipating Voodoo 2 (tm)). Note that if you use more than one Texelfx, they need the same amount of texture cache memory each, and if you combine two Pixelfx, each needs the same amount of frame buffer memory.

```
"SLI / Pixelfx / Texelfx1 / Texelfx2 "
```

It means that a common 2MB+2MB board would be a `1/2/2/0` solution, with the minimally required total 4Mb of memory. A Canopus Pure 3D would be `1/2/4/0`, or 6MB. An Obsidian-2220 board with two Texelfx would be `1/2/2/2`, and an Obsidian SLI-2440 board would be `2/2/4/4`. A fully featured dual board solution (2 Pixelfx, each with 2 Texelfx and 4MB frame buffer, each Texelfx 4 MB texture cache) would be `2/4/4/4`, and the total amount of memory would be `SLI*(Pixelfx+Texelfx1+Texelfx2)`, or 24 MB.

So there.

6.12 Does the Voodoo Graphics (tm) do 24 or 32 bit color?

TOC

No. The Voodoo Graphics (tm) architecture uses 16bpp internally. This is true for Voodoo Graphics (tm), Voodoo Rush (tm) and Voodoo 2 (tm) alike. Quantum3D claims to implement 22-bpp effective color depth with an enhanced 16-bpp frame buffer, though.

6.13 Does the Voodoo Graphics (tm) store 24 or 32 bit z-buffer per pixel?

No. The Voodoo Graphics (tm) architecture uses 16bpp internally for the depth buffer, too. This again is true for Voodoo Graphics (tm), Voodoo Rush (tm) and Voodoo 2 (tm) alike. Again, Quantum3D claims that using the floating point 16-bits per pixel (bpp) depth buffering provides 22-bpp effective Z-buffer precision.

6.14 What resolutions does the Voodoo Graphics (tm) support?

The Voodoo Graphics (tm) chipset supports up to 4 MB frame buffer memory. Presuming double buffering and a depth buffer, a 2MB framebuffer will support a resolution of 640x480. With 4 MB frame buffer, 800x600 is possible.

Unfortunately 960x720 is not supported. The Voodoo Graphics (tm) chipset requires that the amount of memory for a particular resolution must be such that the vertical and horizontal resolutions must be evenly divisible by 32. The video refresh controller, though can output any particular resolution, but the "virtual" size required for the memory footprint must be in dimensions evenly divisible by 32. So, 960x720 actually requires 960x736 amount of memory, and 960x736x2x3 = 4.04MBytes.

However, using two boards with SLI, or a dual Pixelfx SLI board means that each framebuffer will only have to store half of the image. Thus 2 times 4 MB in SLI mode are good up to 1024x768, which is the maximum because of the overall hardware design. You will be able to do 1024x768 tripled buffered with Z, but you will not be able to do e.g. 1280x960 with double buffering.

Note that triple buffering (no VSync synchonization required by the application), stereo buffering (for interfacing LCD shutters) and other more demanding setups will severely decrease the available resolution.

6.15 What texture sizes are supported?

The maximum texture size for the Voodoo Graphics (tm) chipset is 256x256, and you have to use powers of two. Note that for really small textures (e.g. 16x16) you are better off merging them into a large texture, and adjusting your effective texture coordinates appropriately.

TOC

6.16 Does the Voodoo Graphics (tm) support paletted textures?

The Voodoo Graphics (tm) hardware and Glide support the palette extension to OpenGL. The most recent version of Mesa does support the `GL_EXT_paletted_texture` and `GL_EXT_shared_texture_palette` extensions.

6.17 What about overclocking?

If you want to put aside considerations about warranty and overheating, and want to do overclocking to boost up performance even further, there is related info out on the web. The basic mechanism is to use Glide environment variables to adjust the clock.

Note that the actual recommended clock is board dependend. While the default clock speed is 50 Mhz, the Diamond Monster 3D property sheet lets you set up a clock of 57 MHz. It all comes down to the design of a specific board, and which components are used with the Voodoo Graphics (tm) chipset - most notably access speed of the RAM in question. If you exceed the limits of your hardware, rendering artifacts will occur to say the least. Reportedly, 57 MHz usually works, while 60 MHz or more is already pushing it.

Increasing the clock frequency also means increasing the waste heat disposed in the chips, in a nonlinear dependency (10% increase in frequency means a lot larger increase in heating). In consequence, for permanent overclocking you might want to educate yourself about ways to add cooling fans to the board in a way that does not affect warranty. A very recommendable source is the "3Dfx Voodoo Heat Report" by Eric van Ballegoie, available on the web.

6.18 Where could I get additional info on Voodoo Graphics (tm)?

There is a FAQ by 3Dfx, which should be available at their web site. You will find retail information at the following locations: www.3dfx.com and www.quantum3d.com.

Inofficial sites that have good info are "Voodoo Extreme" at www.ve3d.com, and "Operation 3Dfx" at www.ve3d.com.

---

7. FAQ: Glide? TexUS?

7.1 What is Glide anyway?

Glide is a proprietary API plus drivers to access 3D graphics accelerator hardware based on chipsets manufactured by 3Dfx. Glide has been developed and implemented for DOS, Windows, and Macintosh, and has been ported to Linux by Daryll Strauss.

7.2 What is TexUS?

In the distribution is a `libtexus.so`, which is the 3Dfx Interactive Texture Utility Software. It is an image processing libary and utility program for preparing images for use with the 3Dfx Interactive Glide library. Features of TexUS include file format conversion, MIPmap creation, and support for 3Dfx Interactive Narrow Channel Compression textures.

The TexUS utility program `texus` reads images in several popular formats (TGA, PPM, RGT), generates MIPmaps, and writes the images as 3Dfx Interactive textures files (see e.g. alpha.3df, as found in the distribution) or as an image file for inspection. For details on the parameters for `texus`, and the API, see the TexUS documentation.

7.3 Is Glide freeware?

Nope. Glide is neither GPL'ed nor subject to any other public license. See LICENSE in the distribution for any details. Effectively, by downloading and using it, you agree to the End User License Agreement (EULA) on the 3Dfx web site. Glide is provided as binary only, and you should neither use nor distribute any files but the ones released to the public, if you have not signed an NDA. The Glide distribution including the test program sources are copyrighted by 3Dfx.

The same is true for all the sources in the Glide distribution. In the words of 3Dfx: These are not public domain, but they can be freely distributed to owners of 3Dfx products only. No card, No code!

7.4 Where do I get Glide?

The entire 3Dfx SDK is available for download off their public web-site located at www.3dfx.com/software/download_glide.html. Anything else 3Dfx publicly released by 3Dfx is nearby on their website, too.

There is also an FTP site, ftp.3dfx.com. The FTP has a longer timeout, and some of the larger files have been broken into 3 files (approx. 3MB each).

TOC

7.5 Is the Glide source available?

Nope. The Glide source is made available only based on a special agreement and NDA with 3Dfx.

7.6 Is Linux Glide supported?

Currently, Linux Glide is unsupported. Basically, it is provided under the same disclaimers as the 3Dfx GL DLL (see below).

However, 3Dfx definitely wants to provide as much support as possible, and is in the process of setting up some prerequisites. For the time being, you will have to rely on the 3Dfx newsgroup (see below).

In addition, the Quantum3D web page claims that Linux support (for Obsidian) is planned for both Intel and AXP architecture systems in 2H97.

7.7 Where could I post Glide questions?

There are newsgroups currently available only on the NNTP server news.3dfx.com run by 3Dfx. This USENET groups are dedicated to 3Dfx and Glide in general, and will mainly provide assistance for DOS, Win95, and NT. The current list includes:

```
3dfx.events
3dfx.games.glquake
3dfx.glide
3dfx.glide.linux
3dfx.products
3dfx.test
```

---

and the `3dfx.oem.products.*` group for specific boards, eg. `3dfx.oem.products.quantum3d.obsidian`. Please use news.3dfx.com/3dfx.glide.linux for all Lnux Glide related questions.

A mailing list dedicated to Linux Glide is in preparation for 1Q98. Send mail to majordomo@gamers.org, no subject, body of the message `info linux-3dfx` to get information about the posting guidelines, the hypermail archive and how to subscribe to the list or the digest.

7.8 Where to send bug reports?

Currently, you should rely on the newsgroup (see above), that is news.3dfx.com/3dfx.glide.linux. There is no official support e-mail set up yet. For questions not specific to Linux Glide, make sure to use the other newsgroups.

TOC

7.9 Who is maintaining it?

3Dfx will appoint an official maintainer soon. Currently, inofficial maintainer of the Linux Glide port is Daryll Strauss. Please post bug reports in the newsgroup (above). If you are confident that you found a bug not previously reported, please mail to Daryll at daryll@harlot.rb.ca.us

7.10 How can I contribute to Linux Glide?

You could submit precise bug reports. Providing sample programs to be included in the distribution is another possibility. A major contribution would be adding code to the Glide based Mesa Voodoo driver source. See section on Mesa Voodoo below.

7.11 Do I have to use Glide?

Yes. As of now, there is no other Voodoo Graphics (tm) driver available for Linux. At the lowest level, Glide is the only interface that talks directly to the hardware. However, you can write OpenGL code without knowing anything about Glide, and use Mesa with the Glide based Mesa Voodoo driver. It helps to be aware of the involvement of Glide for recognizing driver limitations and bugs, though.

7.12 Should I program using the Glide API?

That depends on the application you are heading for. Glide is a proprietary API that is partly similar to OpenGL or Mesa, partly contains features only available as EXTensions to some OpenGL implementations, and partly contains features not available anywhere but within Glide.

If you want to use the OpenGL API, you will need Mesa (see below). Mesa, namely the Mesa Voodoo driver, offers an API resembling the well documented and widely used OpenGL API. However, the Mesa Voodoo driver is in early alpha, and you will have to accept performance losses and lack of support for some features.

In summary, the decision is up to you - if you are heading for maximum performance while accepting potential problems with porting to non-3Dfx hardware, Glide is not a bad choice. If you care about maintenance, OpenGL might be the best bet in the long run.

7.13 What is the Glide current version?

The current version of Linux Glide is 2.4. The next version will probably be identical to the current version for DOS/Windows, which is 2.4.3, which comes in two distributions. Right now, various parts of Glide are different for Voodoo Rush (tm) (VR) and Voodoo Graphics (tm) (VG) boards. Thus you have to pick up separate distributions (under Windows) for VR and VG. The same will be true for Linux. There will possibly be another chunk of code and another distribution for Voodoo 2 (tm) (V2) boards.

There is also a Glide 3.0 in preparation that will extend the API for use of triangle fans and triangle

strips, and provide better state change optimization. Support for fans and strips will in some situation significantly reduce the amount of data sent ber triangle, and the Mesa driver will benefit from this, as the OpenGL API has separate modes for this. For a detailed explanation on this see e.g. the OpenGL documentation.

TOC

7.14 Does it support multiple Texelfx already?

Multiple Texelfx/TMU's can be used for single pass trilinear mipmapping for improvement image quality without performance penalty in current Linux Glide already. You will need a board with two Texelfx (that is, one of the appropriate Quantum3D Obsidian boards). The application needs to specify the use of both Texelfx accordingly, it does not happen automatically.

Note that because most applications are implemented for consumer boards with a single Texelfx, they might not query the presence of a second Texelfx, and thus not use it. This is not a flaw of Glide but of the application.

7.15 Is Linux Glide identical to DOS/Windows Glide?

The publicly available version of Linux Glide should be identical to the respective DOS/Windows versions. Delays in releasing the Linux port of newer DOS/Windows releases are possible.

7.16 Where to I get information on Glide?

There is exhaustive information available from 3Dfx. You could download it from their home page at www.3dfx.com/software/download_glide.html. These are for free, presuming you bought a 3Dfx hardware based board. Please read the licensing regulations.

Basically, you should look for some of the following:

- *Glide Release Notes*
- *Glide Programming Guide*
- *Glide Reference Manual*
- *Glide Porting Guide*
- *TexUs Texture Utility Software*
- *ATB Release Notes*
- *Installing and Using the Obsidian*

These are available as Microsoft Word documents, and part of the Windows Glide distribution, i.e. the self-extracting archive file. Postscript copies for separate download should be available at www.3dfx.com as well. Note that the release numbers are not always in sync with those of Glide.

TOC

7.17 Where to get some Glide demos?

You will find demo sources for Glide within the distribution (test programs), and on the 3Dfx home page. The problem with the latter is that some require ATB. To port these demos to Linux, the event handling has to be completely rewritten.

In addition, you might find useful some of the OpenGL demo sources accompanying Mesa and GLUT. While the Glide API is different from the OpenGL API, they target the same hardware rendering pipeline.

7.18 What is ATB?

Some of the 3Dfx demo programs for Glide depend not only on Glide but also on 3Dfx's proprietary Arcade Toolbox (ATB), which is available for DOS and Win32, but has not been ported for Linux. If you are a devleoper, the sources are available within the Total Immersion program, so porting ATB to Linux would be possible.

---

8. FAQ: Glide and XFree86?

8.1 Does it run with XFree86?

Basically, the Voodoo Graphics (tm) hardware does not care about X. The X server will not even notice that the video signal generated by the VGA hardware does not reach the display in single screen configurations. If your application is not written X aware, Glide switching to full screen mode might cause problems (see troubleshooting section). If you do not want the overhead of writing an X11-aware application, you might want to use SVGA console mode instead.

So yes, it does run with XFree86, but no, it is not cooperating if you don't write your application accordingly. You can use the Mesa "window hack", which will be significantly slower than fullscreen, but still a lot faster than software rendering (see section below).

8.2 Does it only run full screen?

See above. The Voodoo Graphics (tm) hardware is not window environment aware, neither is Linux Glide. Again, the experimental Mesa "window hack" covered below will allow for pasting the Voodoo Graphics (tm) board framebuffer's content into an X11 window.

8.3 What is the problem with AT3D/Voodoo Rush (tm) boards?

There is an inherent problem when using Voodoo Rush (tm) boards with Linux: Basically, these boards are meant to be VGA 2D/3D accelerator boards, either as a single board solution, or with a Voodoo Rush (tm) based daughterboard used transparently. The VGA component tied to the Voodoo Rush (tm) is a Alliance Semiconductor's ProMotion-AT3D multimedia accelerator. To use this e.g. with XFree86 at all, you need a driver for the AT3D chipset.

There is a mailing list on this, and a web site with FAQ at www.frozenwave.com/linux-stingray128. Look there for most current info. There is a SuSE maintained driver at ftp.suse.com/suse_update/special/xat3d.tgz. Reportedly, the XFree86 SVGA server also works, supporting 8, 16 and 32 bpp. Official support will probably be in XFree86 4.0. XFree86 decided to prepare an intermediate XFree86 3.3.2 release as well, which might already address the issues.

The following `XF86Config` settings reportedly work.

```
# device section settings
Chipset "AT24"
```

```
Videoram 4032

# videomodes tested by Oliver Schaertel
#  25.18  28.32  for 640 x 480   (70hz)
#  61.60         for 1024 x 786  (60hz)
#  120           for 1280 x 1024 (66hz)
```

In summary, there is nothing prohibiting this except for the fact that the drivers in XFree86 are not yet finished.

If you want a more technical explanation: Voodoo Rush (tm) support requires X server changes to support grabbing a buffer area in the video memory on the AT3D board, as the Voodoo Rush (tm) based boards need to store their back buffer and z buffer there. This memory allocation and locking requirement is not a 3Dfx specific problem, it is also needed e.g. for support of TV capture cards, and is thus under active development for XFree86. This means changes at the device dependend X level (thus XAA), which are currently implemented as an extension to XFree86 DGA (Direct Graphics Access, an X11 extension proposal implemented in different ways by Sun and XFree86, that is not part of the final X11R6.1 standard and thus not portable). It might be part of an XFree86 GLX implementation later on. The currently distributed X servers assume they have full control of the framebuffer, and use anything that is not used by the visual region of the framebuffer as pixmap cache, e.g. for caching fonts.

8.4 What about GLX for XFree86?

There are a couple of problems.

The currently supported Voodoo Graphics (tm) hardware and the available revision of Linux Glide are full screen only, and not set up to share a framebuffer with a window environment. Thus GLX or other integration with X11 is not yet possible.

The Voodoo Rush (tm) might be capable of cooperating with XFree86 (that is, an SVGA compliant board will work with the XFree86 SVGA server), but it is not yet supported by Linux Glide, nor do S3 or other XFree86 servers support these boards yet.

In addition, GLX is tied to OpenGL or, in the Linux case, to Mesa. The XFree86 team is currently working on integrating Mesa with their X Server. GLX is in beta, XFree86 3.3 has the hooks for GLX. See Steve Parker's GLX pages at www.cs.utah.edu/~sparker/xfree86-3d/ for the most recent information. Moreover, there is a joint effort by XFree86 and SuSe, which includes a GLX, see www.suse.de/~sim/. Currently, Mesa still uses its GLX emulation with Linux.

8.5 Glide and commerical X Servers?

I have not received any mail regarding use of Glide and/or Mesa with commercial X Servers. I would be interested to get confirmation on this, especially on Mesa and Glide with a commercial X Server that has GLX support.

TOC

8.6 Glide and SVGA?

You should have no problems running Glide based applications either single or dual screen using VGA modes. It might be a good idea to set up the 640x480 resolution in the SVGA modes, too, if you are using a single screen setup.

8.7 Glide and GGI?

A GGI driver for Glide is under development by Jon M. Taylor, but has not officially been released and was put on hold till completion of GGI 0.0.9. For information about GGI see synergy.caltech.edu/~ggi/. If you are adventurous, you might find the combination of XGGI (a GGI based X Server for XFree86) and GGI for Glide an interesting prospect. There is also a GGI driver interfacing the OpenGL API; tested with unaccelerated Mesa. Essentially, this means X11R6 running on a Voodoo Graphics (tm), using either Mesa or Glide directly.

---

9. FAQ: OpenGL/Mesa?

9.1 What is OpenGL?

OpenGL is an immediate mode graphics programming API originally developed by SGI based on their previous proprietary Iris GL, and became in industry standard several years ago. It is defined and maintained by the Architectural Revision Board (ARB), an organization that includes members as SGI, IBM, and DEC, and Microsoft.

OpenGL provides a complete feature set for 2D and 3D graphics operations in a pipelined hardware accelerated architecture for triangle and polygon rendering. In a broader sense, OpenGL is a powerful and generic toolset for hardware assisted computer graphics.

9.2 Where to get additional information on OpenGL?

The official site for OpenGL maintained by the members of the ARB, is www.opengl.org,

A most recommended site is Mark Kilgard's Gateway to OpenGL Info at reality.sgi.com/mjk_asd/opengl-links.html: it provides pointers to book, online manual pages, GLUT, GLE, Mesa, ports to several OS, tons of demos and tools.

If you are interested in game programming using OpenGL, there is the `OpenGL-GameDev-L@fatcity.com` at `Listserv@fatcity.com`. Be warned, this is a high traffic list with very technical content, and you will probably prefer to use `procmail` to handle the 100 messages per day coming in. You cut down bandwidth using the `SET OpenGL-GameDev-L DIGEST` command. It is also not appropriate if you are looking for introductions. The archive is handled by the ListServ software, use the `INDEX OpenGL-GameDev-L` and `GET OpenGL-GameDev-L "filename"` commands to get a preview before subscribing.

9.3 Is Glide an OpenGL implementation?

No, Glide is a proprietary 3Dfx API which several features specific to the Voodoo Graphics (tm) and Voodoo Rush (tm). A 3Dfx OpenGL is in preparation (see below). Several Glide features would require EXTensions to OpenGL, some of which already found in other implementations (e.g. paletted textures).

The closest thing to a hardware accelerated Linux OpenGL you could currently get is Brian Paul's Mesa along with David Bucciarelli's Mesa Voodoo driver (see below).

TOC

9.4 Is there an OpenGL driver from 3Dfx?

Both the 3Dfx website and the Quantum3D website announced OpenGL for Voodoo Graphics (tm) to be available 4Q97. The driver is currently in Beta, and accessible only to registered deverloper's under written Beta test agreement.

A linux port has not been announced yet.

9.5 Is there a commercial OpenGL for Linux and 3Dfx?

I am not aware of any third party commercial OpenGL that supports the Voodoo Graphics (tm). Last time I paid attention, neither MetroX nor XInside OpenGL did.

9.6 What is Mesa?

Mesa is a free implementation of the OpenGL API, designed and written by Brian Paul, with contributions from many others. Its performance is competitive, and while it is not officially certified, it is an almost fully compliant OpenGL implementation conforming to the ARB specifications - more complete than some commercial products out, actually.

9.7 Does Mesa work with 3Dfx?

The latest Mesa MesaVer; release works with Linux Glide 2.4. In fact, support was included in earlier versions, however, this driver is still under development, so be prepared for bugs and less than optimal performance. It is steadily improving, though, and bugs are usually fixed very fast.

You will need to get the Mesa library archive from the iris.ssec.wisc.edu FTP site. It is recommended to subscribe to the mailing list as well, especially when trying to track down bugs, hardware, or driver limitations. Make sure to get the most recent distribution. A Mesa-3.0 is in preparation.

9.8 How portable is Mesa with Glide?

It is available for Linux and Win32, and any application based on Mesa will only have the usual system specific code, which should usually mean XWindows vs. Windows, or GLX vs. WGL. If you use e.g. GLUT or Qt, you should get away with any system specifics at all for virtually most applications. There are only a few issues (like sampling relative mouse movement) that are not adressed by the available portable GUI toolkits.

Mesa/Glide is also available for DOS. The port which is 32bit DOS is maintained by Charlie Wallace and kept up to date with the main Mesa base. See www.geocities.com/~charlie_x/.for the most current releases.

TOC

9.9 Where to get info on Mesa?

The Mesa home page is at www.ssec.wisc.edu/~brianp/Mesa.html. There is an archive of the Mesa mailing list. at www.iqm.unicamp.br/mesa/. This list is not specific to 3Dfx and Glide, but if you are interested in using 3Dfx hardware to accelerate Mesa, it is a good place to start.

9.10 Where to get information on Mesa Voodoo?

For latest information on the Mesa Voodoo driver maintained by David Bucciarelli tech.hmw@plus.it see the home page at www-hmw.caribel.pisa.it/fxmesa/.

9.11 Does Mesa support multitexturing?

Not yet (as of Mesa 2.6), but it is on the list. In Mesa you will probably have to use the OpenGL `EXT_multitexture` extension once it is available. There is no final specification for multitextures in OpenGL, which is supposed to be part of the upcoming OpenGL 1.2 revision. There might be a Glide driver specific implementation of the extension in upcoming Mesa releases, but as long as only certain Quantum3D Obsidian boards come with multiple TMU's, it is not a top priority. This will surely change once Voodoo 2 (tm) based boards are in widespread use.

9.12 Does Mesa support single pass trilinear mipmapping?

Multiple TMU's should be used for single pass trilinear mipmapping for improvement image quality without performance penalty in current Linux Glide already. Mesa support is not yet done (as of Mesa 2.6), but is in preparation.

9.13 What is the Mesa "Window Hack"?

The most recent revisions of Mesa contain an experimental feature for Linux XFree86. Basically, the GLX emulation used by Mesa copies the contents of the Voodoo Graphics (tm) board's most recently finished framebuffer content into video memory on each `glXSwapBuffers` call. This feature is also available with Mesa for Windows.

This obviously puts some drain on the PCI, doubled by the fact that this uses X11 MIT SHM, not XFree86 DGA to access the video memory. The same approach could theoretically be used with e.g. SVGA. The major benefit is that you could use a Voodoo Graphics (tm) board for accelerated rendering into a window, and that you don't have to use the VGA passthrough mode (video output of the VGA board deteoriates in passing through, which is very visible with high end monitors like e.g. EIZO F784-T).

Note that this experimental feature is *NOT* Voodoo Rush (tm) support by any means. It applies only to

the Voodoo Graphics (tm) based boards. Moreover, you need to use a modified GLUT, as interfacing window management system and handling the events appropriately has to be done by the application, is not handled in the driver.

Make really sure that you have enabled the following environment variables:

```
export SST_VGA_PASS=1           # to stop video signal switching
export SST_NOSHUTDOWN=1         # to stop video signal switching
export MESA_GLX_FX="window"     # to initiate Mesa window mode
```

If you manage to forget one of the SST variables, your VGA board will be shut off, and you will loose the display (but not the actual X). It is pretty hard to get that back being effectively blind.

Finally, note that the libMesaGL.a (or .so) library can contain multiple client interfaces. I.e. the GLX, OSMesa, and fxMesa (and even SVGAMesa) interfaces call all be compiled into the same libMesaGL.a. The client program can use any of them freely, even simultaneously if it's careful.

9.14 How about GLUT?

Mark Kilgard's GLUT distribution is a very good place to get sample applications plus a lot of useful utilities. You will find it at reality.sgi.com/mjk_asd/glut3/, and you should get it anyway. The current release is GLUT 3.6, and discussion on a GLUT 3.7 (aka GameGLUT) has begun. Note that Mark Kilgard has left SGI recently, so the archive might move some time this year - for the time being it will be kept at SGI.

There is also a GLUT mailing list, `glut@perp.com`. Send mail to majordomo@perp.com, with the (on of the) following in the body of your email message:

```
help
info glut
subscribe glut
end
```

As GLUT handles double buffers, windows, events, and other operations closely tied to hardware and operating system, using GLUT with Voodoo Graphics (tm) requires support, which is currently in development within GLX for Mesa. It already works for most cases.

10. FAQ: But Quake?

10.1 What about that 3Dfx GL driver for Quake?

The 3Dfx Quake GL, aka mini-driver, aka miniport, aka Game GL, aka 3Dfx GL alpha, implemented only a Quake-specific subset of OpenGL (see http://www.cs.unc.edu/~martin/3dfx.html for an inofficial list of supported code paths). It is not supported, and not updated anymore. It was a Win32 DLL (`opengl32.dll`) released by 3Dfx and was available for Windows only. This DLL is not, and will not be ported to Linux.

10.2 Is there a 3Dfx based glQuake for Linux?

Yes. A Quake linuxquake v0.97 binary has been released based on Mesa with Glide. The Quake2 q2test binary for Linux and Voodoo Graphics (tm) has been made available as well. A full Quake2 for Linux was released in January 1998, with linuxquake2-3.10. Dave "Zoid" Kirsch is the official maintainer of all Linux ports of Quake, Quakeworld, and Quake2, including all the recent Mesa based ports. Note that all Linux ports, including the Mesa based ones, are not officially supported by id Software.

See ftp.idsoftware.com/idstuff/quake/unix/ for the latest releases.

10.3 Does glQuake run in an XFree86 window?

A revision of Mesa and the Mesa-based Linux glQuake is in preparation. Mesa already does support this by GLX, but Linux glQuake does not use GLX.

10.4 Known Linux Quake problems?

Here is an excerpt, as of January 7th, 1998. I omitted most stuff not specific to &3Dfx; hardware.

- You really should run Quake2 as root when using the SVGALib and/or GL renders. You don't have to run as root for the X11 refresh, but the modes on the mouse and sound devices must be read/writable by whatever user you run it as. Dedicated server requires no special permissions.
- X11 has some garbage on the screen when 'loading'. This is normal in 16bit color mode. X11 doesn't work in 24bit (TrueColor). It would be very slow in any case.
- Some people are experiencing crashes with the GL renderer. Make sure you install the libMesa that comes with Quake2! Older versions of libMesa don't work properly.
- If you are experience video 'lag' in the GL renderer (the frame rate feels like it's lagging behind your mouse movement) type "gl_finish 1" in the console. This forces update on a per frame basis.
- When running the GL renderer, make sure you have killed selection and/or gpm or the mouse

TOC

won't work as they won't "release" it while Quake2 is running in GL mode.

10.5 Know Linux Quake security problems?

As Dave Kirsch posted on January 28th, 1998: an exploit for Quake2 under Linux has been published. Quake2 is using shared libraries. While the READMRE so far does not specifically mention it, note that Quake2 should not be `setuid`.

If you want to use the `ref_soft` and `ref_gl` renderers, you should run Quake2 as root. Do not make the binary setuid. You can only run both those renderers at the console only, so being root is not that much of an issue.

The X11 render does not need any root permissions (if `/dev/dsp` is writable by others for sound). The dedicated server mode does not need to be root either, obviously.

Problems such as root requirements for games has been sort of a sore spot in Linux for a number of years now. This is one of the goals that e.g. GGI is targetting to fix. A `ref_ggi` might be supported in the near future.

10.6 Does LinuxQuake use multitexturing?

To my understadnding, glQuake will use a multitexture EXTension if the OpenGL driver in question offers it. The current Mesa implementation and the Glide driver for Linux do not yet support this extension, so for the time being the answer is no. See section on Mesa and multitexturing for details.

10.7 Where can I get current information on Linux glQuake?

Try some of these sites: the "The Linux Quake Resource" at linuxquake.telefragged.com, or the "Linux Quake Page" at www.planetquake.com/threewave/linux/. Alternatively, you could look for Linux Quake sites in the "SlipgateCentral" database at www.slipgatecentral.com.

---

11. FAQ: Troubleshooting?

11.1 Has this hardware been tested?

See hardware requirements list above. I currently do not maintain a conclusive list of vendors and boards, as no particular board specific problems have been verified. Currently, only 3Dfx and Quantum3D provide boards for testing to the developers, so Quantum3D consumer boards are a safe bet. Every other Voodoo Graphics (tm) based board should work, too. I have reports regarding the Orchid Righteous 3D, Guillemot Maxi 3D Gamer, and Diamond Monster 3D.

If you are a board manufacturer who wants to make sure his Voodoo Graphics (tm), Voodoo Rush (tm) or Voodoo 2 (tm) boards work with upcoming releases of Linux, Xfree86, Linux Glide and/or Mesa, please contact me, and I will happily forward your request to the persons maintaining the drivers in question. If you are interested in support for Linux Glide on other then the PC platfrom, e.g. DEC Alpha, please contact the maintainer of Linux Glide Daryll Strauss, at daryll@harlot.rb.ca.us

11.2 Failed to change I/O privilege?

You need to be root, or `setuid` your application to run a Glide based application. For DMA, the driver accesses `/dev/mem`, which is not writeable for anybody but root, with good reasons. See the README in the Glide distribution for Linux.

11.3 Does it work without root privilege?

There are compelling case where the setuid requirement is a problem, obviously. There are currently solutions in preparation, which require changes to the library internals itself.

11.4 Displayed images looks awful (single screen)?

If you are using the analog pass through configuration, the common SVGA or X11 display might look pretty bad. You could try to get a better connector cable than the one provided with the accelerator board (the ones delivered with the Diamond Monster 3D are reportedly worse then the one accompanying the Orchid Righteous 3D), but up to a degree there will inevitably be signal loss with an additional transmission added.

If the 640x480 full screen image created by the accelerator board does look awful, this might indicate a real hardware problem. You will have to contact the board manufacturer, not 3Dfx for details, as the quality of the video signal has nothing to do with the accelerator - the board manufacturer chooses the RAMDAC, output drivers, and other components responsible.

TOC

11.5 The last frame is still there (single or dual screen)?

You terminated your application with Ctrl-C, or it did not exit normally. The accelerator board will dutifully provide the current content of the framebuffer as a video signal unless told otherwise.

11.6 Powersave kicks in (dual screen)?

When you application terminates in dual screen setups, the accelerator board does not provide video output any longer. Thus powersave kicks each time. To avoid this, use

```
setenv SST_DUALSCREEN 1
```

11.7 My machine seem to lock (X11, single screen)?

If you are running X when calling a Glide application, you probably moved the mouse out of the window, and the keyboard inputs do not reach the application anymore.

If you application is supposed to run concurrently with X11, it is recommend to expose a full screen window, or use the `XGrabPointer` and `XGrabServer` functions to redirect all inputs to the application while the X server cannot access the display. Note that grabbing all input with `XGrabPointer` and `XGrabServer` does not qualify as well-behaved application, and that your program might block the entire system.

If you experience this problem without running X, be sure that there is no hardware conflict (see below).

11.8 My machine locks (single or dual screen)?

If the system definitely does not respond to any inputs (you are running two displays and know about the loss of focus), you might experience a more or less subtle hardware conflict. See installation troubleshooting section for details.

If there is no obvious address conflict, there might still be other problems (below). If you are writing your own code the most common reason for locking is that you didn't snap your vertices. See the section on snapping in the Glide documentation.

11.9 My machine locks (used with S3 VGA board)?

It is possible you have a problem with memory region overlap specific to S3. There is some info and a patch to the so-called S3 problem in the 3Dfx web site, but these apply to Windows only. To my understanding, the cause of the problem is that some S3 boards (older revisions of Diamond Stealth S3

968) reserve more memory space than actually used, thus the Voodoo Graphics (tm) has to be mapp a different location. However, this has not been reported as a problem with Linux, and might be Windows-specific.

TOC

11.10 No address conflict, but locks anyway?

If you happen to use a motherboard with non-standard or incomplete PCI support, you could try to shuffle the boards a bit. I am running an ASUS TP4XE that has that non-standard modified "Media Slot", i.e. PCI slot4 with additional connector for ASUS-manufactured SCSI/Sound combo boards, and I experienced severe problems while running a Diamond Monster 3D in that slot. The system operates flawlessly since I put the board in one of the regular slots.

11.11 Mesa runs, but does not access the board?

Be sure that you recompiled all the libraries (including the toolkits the demo programs use - remember that GLUT does not yet support Voodoo Graphics (tm)), and that you removed the older libraries, run `ldconfig`, and/or set your `LD_LIBRARY_PATH` properly. Mesa supports several drivers in parallel (you could use X11 SHM, off screen rendering, and Mesa Voodoo at the same time), and you might have to create and switch contexts explicitely (see `MakeCurrent` function) if the Voodoo Graphics (tm) isn't chosen by default.

11.12 Resetting dual board SLI?

If a Quantum 3D Obsidian board using in an SLI setup exits abruptly (i.e., the application crashes, or is aborted by user), the boards are left in an undefined state. With the dual-board set, you can run a program called `resetsli` to reset them. Until you run the `resetsli` program, you will not be able to re-initialize the Obsidian board.

11.13 Resetting single board SLI?

The `resetsli` program mentioned above does not yet work with a single board Obsidian SLI (e.g. the Obsidian 100-4440SB). You will have to reboot your system by reset in order to reset the board.

Linux AX25-HOWTO, Amateur Radio.

Terry Dawson, VK2KTJ, terry@perf.no.itg.telstra.com.au

v1.5, 17 October 1997

The Linux Operating System is perhaps the only operating system in the world that can boast native and standard support for the AX.25 packet radio protocol utilised by Amateur Radio Operators worldwide. This document aims to describe how to install and configure this support.

1. Introduction.

- 1.1 Changes from the previous version
- 1.2 Where to obtain new versions of this document.
- 1.3 Other related documentation.

2. The Packet Radio Protocols and Linux.

- 2.1 How it all fits together.

3. The AX.25/NetRom/Rose software components.

- 3.1 Finding the kernel, tools and utility packages.

4. Installing the AX.25/NetRom/Rose software.

- 4.1 Compiling the kernel.
- 4.2 The network configuration tools.
- 4.3 The AX.25 user and utility programs.

TOC

5. A note on callsigns, addresses and things before we start.

- 5.1 What are all those T1, T2, N2 and things ?
- 5.2 Run time configurable parameters

6. Configuring an AX.25 port.

- 6.1 Creating the AX.25 network device.
- 6.2 Creating the `/etc/ax25/axports` file.
- 6.3 Configuring AX.25 routing.

7. Configuring an AX.25 interface for TCP/IP.

8. Configuring a NetRom port.

- 8.1 Configuring `/etc/ax25/nrports`
- 8.2 Configuring `/etc/ax25/nrbroadcast`
- 8.3 Creating the NetRom Network device
- 8.4 Starting the NetRom daemon
- 8.5 Configuring NetRom routing.

9. Configuring a NetRom interface for TCP/IP.

10. Configuring a Rose port.

- 10.1 Configuring `/etc/ax25/rsports`
- 10.2 Creating the Rose Network device.
- 10.3 Configuring Rose Routing

11. Making AX.25/NetRom/Rose calls.

TOC

12. Configuring Linux to accept Packet connections.

- 12.1 Creating the `/etc/ax25/ax25d.conf` file.
- 12.2 A simple example `ax25d.conf` file.
- 12.3 Starting *ax25d*

13. Configuring the *node* software.

- 13.1 Creating the `/etc/ax25/node.conf` file.
- 13.2 Creating the `/etc/ax25/node.perms` file.
- 13.3 Configuring *node* to run from *ax25d*
- 13.4 Configuring *node* to run from *inetd*

14. Configuring *axspawn*.

- 14.1 Creating the `/etc/ax25/axspawn.conf` file.

15. Configuring the *pms*

- 15.1 Create the `/etc/ax25/pms.motd` file.
- 15.2 Create the `/etc/ax25/pms.info` file.
- 15.3 Associate AX.25 callsigns with system users.
- 15.4 Add the PMS to the `/etc/ax25/ax25d.conf` file.
- 15.5 Test the PMS.

16. Configuring the *user_call* programs.

17. Configuring the Rose Uplink and Downlink commands

- 17.1 Configuring a Rose downlink
- 17.2 Configuring a Rose uplink

18. Associating AX.25 callsigns with Linux users.

19. The /proc/ file system entries.

20. AX.25, NetRom, Rose network programming.

- 20.1 The address families.
- 20.2 The header files.
- 20.3 Callsign mangling and examples.

21. Some sample configurations.

- 21.1 Small Ethernet LAN with Linux as a router to Radio LAN
- 21.2 IPIP encapsulated gateway configuration.
- 21.3 AXIP encapsulated gateway configuration
- 21.4 Linking NOS and Linux using a pipe device

22. Where do I find more information about .... ?

- 22.1 Packet Radio
- 22.2 Protocol Documentation
- 22.3 Hardware Documentation

23. Discussion relating to Amateur Radio and Linux.

24. Acknowledgements.

25. Copyright.

1. Introduction.

This document was originally an appendix to the HAM-HOWTO, but grew too large to be reasonably managed in that fashion. This document describes how to install and configure the native AX.25, NetRom and Rose support for Linux. A few typical configurations are described that could be used as models to work from.

The Linux implementation of the amateur radio protocols is very flexible. To people relatively unfamiliar with the Linux operating system the configuration process may look daunting and complicated. It will take you a little time to come to understand how the whole thing fits together. You will find configuration very difficult if you have not properly prepared yourself by learning about Linux in general. You cannot expect to switch from some other environment to Linux without learning about Linux itself.

1.1 Changes from the previous version

```
Additions:
        Joerg Reuters Web Page
        "More Information" section
        ax25ipd configuration.

Corrections/Updates:
        Changed pty's to a safer range to prevent possible conflicts
        Updated module and ax25-utils versions.

ToDo:
        Fix up the SCC section, this is probably wrong.
        Expand on the programming section.
```

1.2 Where to obtain new versions of this document.

The best place to obtain the latest version of this document is from a Linux Documentation Project archive. The Linux Documentation Project runs a Web Server and this document appears there as the AX25-HOWTO. This document is also available in various formats from the sunsite.unc.edu ftp archive.

You can always contact me, but I pass new versions of the document directly to the LDP HOWTO coordinator, so if it isn't there then chances are I haven't finished it.

1.3 Other related documentation.

TOC

There is a lot of related documentation. There are many documents that relate to Linux networking in more general ways and I strongly recommend you also read these as they will assist you in your efforts and provide you with stronger insight into other possible configurations.

They are:

The HAM-HOWTO,

the NET-3-HOWTO,

the Ethernet-HOWTO,

and:

the Firewall-HOWTO

More general Linux information may be found by reference to other Linux HOWTO documents.

---

2. The Packet Radio Protocols and Linux.

The *AX.25* protocol offers both connected and connectionless modes of operation, and is used either by itself for point-point links, or to carry other protocols such as TCP/IP and NetRom.

It is similar to X.25 level 2 in structure, with some extensions to make it more useful in the amateur radio environment.

The NetRom protocol is an attempt at a full network protocol and uses AX.25 at its lowest layer as a datalink protocol. It provides a network layer that is an adapted form of AX.25. The NetRom protocol features dynamic routing and node aliases.

The Rose protocol was conceived and first implemented by Tom Moulton W2VY and is an implementation of the X.25 packet layer protocol and is designed to operate with AX.25 as its datalink layer protocol. It too provides a network layer. Rose addresses take the form of 10 digit numbers. The first four digits are called the Data Network Identification Code (DNIC) and are taken from Appendix B of the CCITT X.121 recommendation. More information on the Rose protocol may be ontained from the RATS Web server.

Alan Cox developed some early kernel based AX.25 software support for Linux. Jonathon Naylor `<g4klx@g4klx.demon.co.uk>` has taken up ongoing development of the code, has added NetRom and Rose support and is now the developer of the AX.25 related kernel code. DAMA support was developed by Joerg, DL1BKE, `jreuter@poboxes.com`. Baycom and SoundModem support were added by Thomas Sailer, `<sailer@ife.ee.ethz.ch>`. The AX.25 utility software is now maintained by me.

The Linux code supports KISS based TNC's (Terminal Node Controllers), the Ottawa PI card, the Gracilis PacketTwin card and other Z8530 SCC based cards with the generic SCC driver and both the Parallel and Serial port Baycom modems. Thomas's new soundmodem driver supports Soundblaster and soundcards based on the Crystal chipset.

The User programs contain a simple PMS (Personal Message System), a beacon facility, a line mode connect program, `listen' an example of how to capture all AX.25 frames at raw interface level and programs to configure the NetRom protocol. Included also are an AX.25 server style program to handle and despatch incoming AX.25 connections and a NetRom daemon which does most of the hard work for NetRom support.

2.1 How it all fits together.

The Linux AX.25 implementation is a brand new implementation. While in many ways it may looks similar to NOS, or BPQ or other AX.25 implementations, it is none of these and is not identical to any of them. The Linux AX.25 implementation is capable of being configured to behave almost identically to other implementations, but the configuration process is very different.

To assist you in understanding how you need to think when configuring this section describes some of the structural features of the AX.25 implementation and how it fits into the context of the overall Linux structure.

Simplified Protocol Layering Diagram

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

TOC

```
| AF_AX25 | AF_NETROM |  AF_INET     | AF_ROSE |
|=========|===========|==============|=========|
|         |           |              |         |
|         |           |    TCP/IP    |         |
|         |           ----------     |         |
|         |   NetRom            |    | Rose    |
|         ------------------------------------
|            AX.25                             |
-----------------------------------------------
```

This diagram simply illustrates that NetRom, Rose and TCP/IP all run directly on top of AX.25, but that each of these protocols is treated as a seperate protocol at the programming interface. The `AF_' names are simply the names given to the `*Address Family*' of each of these protocols when writing programs to use them. The important thing to note here is the implicit dependence on the configuration of your AX.25 devices before you can configure your NetRom, Rose or TCP/IP devices.

Software module diagram of Linux Network Implementation

```
-----------------------------------------------------------------------------
 User     | Programs  |   call        node    ||  Daemons | ax25d  mheardd
          |           |   pms         mheard  ||          | inetd  netromd
-----------------------------------------------------------------------------
          | Sockets   | open(), close(), listen(), read(), write(), connect()
          |           |-------------------------------------------------------
          |           |    AF_AX25   |  AF_NETROM  |   AF_ROSE   |  AF_INET
          |-----------------------------------------------------------------
Kernel    | Protocols |    AX.25     |   NetRom    |     Rose    | IP/TCP/UDP
          |-----------------------------------------------------------------
          | Devices   |    ax0,ax1   |  nr0,nr1    | rose0,rose1 | eth0,ppp0
          |-----------------------------------------------------------------
          | Drivers   |  Kiss   PI2   PacketTwin   SCC   BPQ     | slip ppp
          |           |      Soundmodem      Baycom              | ethernet
-----------------------------------------------------------------------------
Hardware | PI2 Card, PacketTwin Card, SCC card, Serial port, Ethernet Card
-----------------------------------------------------------------------------
```

This diagram is a little more general than the first. This diagram attempts to show the relationship between user applications, the kernel and the hardware. It also shows the relationship between the Socket application programming interface, the actual protocol modules, the kernel networking devices and the device drivers. Anything in this diagram is dependent on anything underneath it, and in general you must configure from the bottom of the diagram upwards. So for example, if you want to run the *call* program you must also configure the Hardware, then ensure that the kernel has the appropriate device driver, that you create the appropriate network device, that the kernel includes the desired protocol that presents a programming interface that the *call* program can use. I have attempted to lay out this document in roughly that order.

---

3. The AX.25/NetRom/Rose software components.

The AX.25 software is comprised of three components, the kernel source, the network configuration tools and the utility programs.

The version 2.0.xx Linux kernels include the AX.25, NetRom, Z8530 SCC, PI card and PacketTwin drivers by default. These have been significantly enhanced in the 2.1.* kernels. Unfortunately, the rest of the 2.1.* kernels makes them fairly unstable at the moment and not a good choice for a production system. To solve this problem Jonathon Naylor has prepared a patch kit which will bring the amateur radio protocol support in a 2.0.28 kernel up to the standard of the 2.1.* kernels. This is very simple to apply, and provides a range of facilities not present in the standard kernel such as Rose support.

3.1 Finding the kernel, tools and utility packages.

The kernel source:

The kernel source can be found in its usual place at: **ftp.kernel.org**

```
/pub/linux/kernel/v2.0/linux-2.0.31.tar.gz
```

The current version of the AX25 upgrade patch is available at: **ftp.pspt.fi**

```
/pub/linux/ham/ax25/ax25-module-14e.tar.gz
```

The network tools:

The latest alpha release of the standard Linux network tools support AX.25 and NetRom and can be found at: **ftp.inka.de**

```
/pub/comp/Linux/networking/net-tools/net-tools-1.33.tar.gz
```

The latest ipfwadm package can be found at: **ftp.xos.nl**

```
/pub/linux/ipfwadm/
```

The AX25 utilities:

There are two different families of AX25-utilities. One is for the `2.0.*` kernels and the other will work with either the `2.1.*` kernels or the `2.0.*+moduleXX` kernels. The ax25-utils version number indicates the oldest version of kernel that they will work with. Please choose a version of the ax25-utils appropriate to your kernel. The following are working combinations. You **must** use one of the following

combinations, any other combination will not work, or will not work well.

```
Linux Kernel              AX25 Utility set
----------------------    --------------------------
linux-2.0.29              ax25-utils-2.0.12c.tar.gz **
linux-2.0.28+module12     ax25-utils-2.1.22b.tar.gz **
linux-2.0.30+module14c    ax25-utils-2.1.42a.tar.gz
linux-2.0.31+module14d    ax25-utils-2.1.42a.tar.gz
linux-2.1.22 ++           ax25-utils-2.1.22b.tar.gz
linux-2.1.42 ++           ax25-utils-2.1.42a.tar.gz
```

Note: the `ax25-utils-2.0.*` series (marked above with the '**' symbol) is now obsolete and is no longer supported. This document covers configuration using the versions of software recommended above the table. While there are differences between the releases, most of the information will be relevant to earlier releases of code.

The AX.25 utility programs can be found at: ftp.pspt.fi

or at: sunsite.unc.edu

---

4. Installing the AX.25/NetRom/Rose software.

To successfully install AX.25 support on your linux system you must configure and install an appropriate kernel and then install the AX.25 utilities.

4.1 Compiling the kernel.

If you are already familiar with the process of compiling the Linux Kernel then you can skip this section, just be sure to select the appropriate options when compiling the kernel. If you are not, then read on.

The normal place for the kernel source to be unpacked to is the `/usr/src` directory into a subdirectory called `linux`. To do this you should be logged in as `root` and execute a series of commands similar to the following:

```
# mv linux linux.old
# cd /usr/src
# tar xvfz linux-2.0.31.tar.gz
# tar xvfz /pub/net/ax25/ax25-module-14e.tar.gz
# patch -p0 </usr/src/ax25-module-14/ax25-2.0.31-2.1.47-2.diff
# cd linux
```

After you have unpacked the kernel source and applied the upgrade, you need to run the configuration script and choose the options that suit your hardware configuration and the options that you wish built into your kernel. You do this by using the command:

```
# make menuconfig
```

You might also try:

```
# make config
```

I'm going to describe the full screen method (menuconfig) because it is easier to move around, but you use whichever you are most comfortable with.

In either case you will be offered a range of options at which you must answer `Y' or `N'. (Note you may also answer `M' if you are using modules. For the sake of simplicity I will assume you are not, please make appropriate modifications if you are).

The options most relevant to an AX.25 configuration are:

```
Code maturity level options  --->
    ...
```

```
    [*] Prompt for development and/or incomplete code/drivers
    ...
General setup  --->
    ...
    [*] Networking support
    ...
Networking options  --->
    ...
    [*] TCP/IP networking
    [?] IP: forwarding/gatewaying
    ...
    [?] IP: tunneling
    ...
    [?] IP: Allow large windows (not recommended if <16Mb of memory)
    ...
    [*] Amateur Radio AX.25 Level 2
    [?] Amateur Radio NET/ROM
    [?] Amateur Radio X.25 PLP (Rose)
    ...
Network device support  --->
    [*] Network device support
    ...
    [*] Radio network interfaces
    [?] BAYCOM ser12 and par96 driver for AX.25
    [?] Soundcard modem driver for AX.25
    [?] Soundmodem support for Soundblaster and compatible cards
    [?] Soundmodem support for WSS and Crystal cards
    [?] Soundmodem support for 1200 baud AFSK modulation
    [?] Soundmodem support for 4800 baud HAPN-1 modulation
    [?] Soundmodem support for 9600 baud FSK G3RUH modulation
    [?] Serial port KISS driver for AX.25
    [?] BPQ Ethernet driver for AX.25
    [?] Gracilis PackeTwin support for AX.25
    [?] Ottawa PI and PI/2 support for AX.25
    [?] Z8530 SCC KISS emulation driver for AX.25
    ...
```

The options I have flagged with a `*' are those that you **must** must answer `Y' to. The rest are dependent on what hardware you have and what other options you want to include. Some of these options are described in more detail later on, so if you don't know what you want yet, then read ahead and come back to this step later.

After you have completed the kernel configuration you should be able to cleanly compile your new kernel:

```
    # make dep
```

TOC

```
# make clean
# make zImage
```

maake sure you move your `arch/i386/boot/zImage` file wherever you want it and then edit your `/etc/lilo.conf` file and rerun *lilo* to ensure that you actually boot from it.

A word about Kernel modules

I recommend that you **don't** compile any of the drivers as modules. In nearly all installations you gain nothing but additional complexity. Many people have problems trying to get the modularised components working, not because the software is faulty but because modules are more complicated to install and configure.

If you've chosen to compile any of the components as modules, then you'll also need to use:

```
# make modules
# make modules_install
```

to install your modules in the appropriate location.

You will also need to add some entries into your `/etc/conf.modules` file that will ensure that the *kerneld* program knows how to handle the kernel modules. You should add/modify the following:

```
alias net-pf-3     ax25
alias net-pf-6     netrom
alias net-pf-11    rose
alias tty-ldisc-1  slip
alias tty-ldisc-3  ppp
alias tty-ldisc-5  mkiss
alias bc0          baycom
alias nr0          netrom
alias pi0a         pi2
alias pt0a         pt
alias scc0         optoscc    (or one of the other scc drivers)
alias sm0          soundmodem
alias tunl0        newtunnel
alias char-major-4 serial
alias char-major-5 serial
alias char-major-6 lp
```

What's new in 2.0.*+ModuleXX or 2.1.* Kernels ?

The `2.1.*` kernels have enhanced versions of nearly all of the protocols and drivers. The most significant of the enhancements are:

modularised

the protocols and drivers have all been modularised so that you can *insmod* and *rmmod* them

whenever you wish. This reduces the kernel memory requirements for infrequently used modu and makes development and bug hunting much simpler. That being said, it also makes configuration slightly more difficult.

All drivers are now network drivers

all of the network devices such as Baycom, SCC, PI, Packettwin etc now present a normal network interface, that is they now look like the ethernet driver does, they no longer look like KISS TNC's. A new utility called *net2kiss* allows you to build a kiss interface to these devices if you wish.

bug fixed

there have been many bug fixes and new features added to the drivers and protocols. The Rose protocol is one important addition.

4.2 The network configuration tools.

Now that you have compiled the kernel you should compile the new network configuration tools. These tools allow you to modify the configuration of network devices and to add routes to the routing table.

The new alpha release of the standard `net-tools` package includes support for AX.25 and NetRom support. I've tested this and it seems to work well for me.

A patch kit that adds Rose support and fixes some bugs.

The standard net-tools-1.33.tar.gz package has some small bugs that affect the AX.25 and NetRom support. I've made a small patch kit that corrects these and adds Rose support to the tools as well.

You can get the patch from: zone.pspt.fi.

Building the standard net-tools release.

Don't forget to read the `Release` file and follow any instructions there. The steps I used to compile the tools were:

```
# cd /usr/src
# tar xvfz net-tools-1.33.tar.gz
# zcat net-tools-1.33.rose.tjd.diff.gz | patch -p0
# cd net-tools-1.33
# make config
```

At this stage you will be presented with a series of configuration questions, similar to the kernel configuration questions. Be sure to include support for all of the protocols and network devices types that you intend to use. If you do not know how to answer a particular question then answer `Y'.

When the compilation is complete, you should use the:

```
# make install
```

TOC

command to install the programs in their proper place.

If you wish to use the IP firewall facilities then you will need the latest firewall administration tool `ipfwadm`. This tool replaces the older `ipfw` tool which will not work with new kernels.

I compiled the `ipfwadm` utility with the following commands:

```
# cd /usr/src
# tar xvfz ipfwadm-2.0beta2.tar.gz
# cd ipfwadm-2.0beta2
# make install
# cp ipfwadm.8 /usr/man/man8
# cp ipfw.4 /usr/man/man4
```

4.3 The AX.25 user and utility programs.

After you have successfully compiled and booted your new kernel, you need to compile the user programs. To compile and install the user programs you should use a series of commands similar to the following:

```
# cd /usr/src
# tax xvfz ax25-utils-2.1.42a.tar.gz
# cd ax25-utils-2.1.42a
# make config
# make
# make install
```

The files will be installed under the `/usr` directory by default in subdirectories: `bin`, `sbin`, `etc` and `man`.

If this is a first time installation, that is you've never installed any ax25 utilities on your machine before you should also use the:

```
# make installconf
```

command to install some sample configuration files into the `/etc/ax25/` directory from which to work.

If you get messages something like:

```
gcc -Wall -Wstrict-prototypes -O2 -I../lib -c call.c
call.c: In function `statline':
call.c:268: warning: implicit declaration of function `attron'
call.c:268: `A_REVERSE' undeclared (first use this function)
call.c:268: (Each undeclared identifier is reported only once
call.c:268: for each function it appears in.)
```

then you should double check that you have the *ncurses* package properly installed on your system. configuration script attempts to locate your ncurses packages in the common locations, but some installations have ncurses badly installed and it is unable to locate them.

---

5. A note on callsigns, addresses and things before we start.

Each AX.25 and NetRom port on your system must have a callsign/ssid allocated to it. These are configured in the configuration files that will be described in detail later on.

Some AX.25 implementations such as NOS and BPQ will allow you to configure the same callsign/ssid on each AX.25 and NetRom port. For somewhat complicated technical reasons Linux does not allow this. This isn't as big a problem in practise as it might seem.

This means that there are things you should be aware of and take into consideration when doing your configurations.

1. Each AX.25 and NetRom port must be configured with a unique callsign/ssid.
2. TCP/IP will use the callsign/ssid of the AX.25 port it is being transmitted or received by, ie the one you configured for the AX.25 interface in point 1.
3. NetRom will use the callsign/ssid specified for it in its configuration file, but this callsign is only used when your NetRom is speaking to another NetRom, this is **not** the callsign/ssid that AX.25 users who wish to use your NetRom `node' will use. More on this later.
4. Rose will, by default, use the callsign/ssid of the AX.25 port, unless the Rose callsign has been specifically set using the `*rsparms*' command. If you set a callsign/ssid using the `*rsparms*' command then Rose will use this callsign/ssid on all ports.
5. Other programs, such as the `*ax25d*' program can listen using any callsign/ssid that they wish and these may be duplicated across different ports.
6. If you are careful with routing you can configure the same IP address on all ports if you wish.

5.1 What are all those T1, T2, N2 and things ?

Not every AX.25 implementation is a TNC2. Linux uses nomenclature that differs in some respects from that you will be used to if your sole experience with packet is a TNC. The following table should help you interpret what each of the configurable items are, so that when you come across them later in this text you'll understand what they mean.

Linux	TAPR TNC	Description
T1	FRACK	How long to wait before retransmitting an unacknowledged frame.
T2	RESPTIME	The minimum amount of time to wait for another frame to be received before transmitting an acknowledgement.

```
T3     | CHECK    | The period of time we wait between sending
       |          | a check that the link is still active.
-------------------------------------------------------------------
N2     | RETRY    | How many times to retransmit a frame before
       |          | assuming the connection has failed.
-------------------------------------------------------------------
Idle   |          | The period of time a connection can be idle
       |          | before we close it down.
-------------------------------------------------------------------
Window | MAXFRAME | The maximum number of unacknowledged
       |          | transmitted frames.
-------------------------------------------------------------------
```

5.2 Run time configurable parameters

The `2.1.*` and `2.0.* +moduleXX` kernels have a new feature that allows you to change many previously unchangable parameters at run time. If you take a careful look at the `/proc/sys/net/` directory structure you will see many files with useful names that describe various parameters for the network configuration. The files in the `/proc/sys/net/ax25/` directory each represents one configured AX.25 port. The name of the file relates to the name of the port.

The structure of the files in `/proc/sys/net/ax25/<portname>/` is as follows:

FileName	Meaning	Values	Default
ip_default_mode	IP Default Mode	0=DG 1=VC	0
ax25_default_mode	AX.25 Default Mode	0=Normal 1=Extended	0
backoff_type	Backoff	0=Linear 1=Exponential	1
connect_mode	Connected Mode	0=No 1=Yes	1
standard_window_size	Standard Window	1 <= N <= 7	2
extended_window_size	Extended Window	1 <= N <= 63	32
t1_timeout	T1 Timeout	1s <= N <= 30s	10s
t2_timeout	T2 Timeout	1s <= N <= 20s	3s
t3_timeout	T3 Timeout	0s <= N <= 3600s	300s
idle_timeout	Idle Timeout	0m <= N	20m
maximum_retry_count	N2	1 <= N <= 31	10
maximum_packet_length	AX.25 Frame Length	1 <= N <= 512	256

In the table T1, T2 and T3 are given in seconds, and the Idle Timeout is given in minutes. But please note that the values used in the sysctl interface are given in internal units where the time in seconds is multiplied by 10, this allows resolution down to 1/10 of a second. With timers that are allowed to be zero, eg T3 and Idle, a zero value indicates that the timer is disabled.

The structure of the files in `/proc/sys/net/netrom/` is as follows:

FileName	Values	Default
default_path_quality		10
link_fails_count		2
network_ttl_initialiser		16

```
obsolescence_count_initialiser                          6
routing_control                                         1
transport_acknowledge_delay                             50
transport_busy_delay                                    1800
transport_maximum_tries                                 3
transport_requested_window_size                         4
transport_timeout                                       1200
```

The structure of the files in `/proc/sys/net/rose/` is as follows:

```
FileName                        Values                  Default
acknowledge_hold_back_timeout                           50
call_request_timeout                                    2000
clear_request_timeout                                   1800
link_fail_timeout                                       1200
maximum_virtual_circuits                                50
reset_request_timeout                                   1800
restart_request_timeout                                 1800
routing_control                                         1
window_size                                             3
```

To set a parameter all you need to do is write the desired value to the file itself, for example to check and set the Rose window size you'd use something like:

```
# cat /proc/sys/net/rose/window_size
3
# echo 4 >/proc/sys/net/rose/window_size
# cat /proc/sys/net/rose/window_size
4
```

---

6. Configuring an AX.25 port.

Each of the AX.25 applications read a particular configuration file to obtain the parameters for the various AX.25 ports configured on your Linux machine. For AX.25 ports the file that is read is the `/etc/ax25/axport` file. You must have an entry in this file for each AX.25 port you want on your system.

6.1 Creating the AX.25 network device.

The network device is what is listed when you use the `*ifconfig*' command. This is the object that the Linux kernel sends and receives network data from. Nearly always the network device has a physical port associated with it, but there are occasions where this isn't necessary. The network device does relate directly to a device driver.

In the Linux AX.25 code there are a number of device drivers. The most common is probably the KISS driver, but others are the SCC driver(s), the Baycom driver and the SoundModem driver.

Each of these device drivers will create a network device when it is started.

Creating a KISS device.

Kernel Compile Options:

```
General setup  --->
    [*] Networking support
Network device support  --->
    [*] Network device support
    ...
    [*] Radio network interfaces
    [*] Serial port KISS driver for AX.25
```

Probably the most common configuration will be for a KISS TNC on a serial port. You will need to have the TNC preconfigured and connected to your serial port. You can use a communications program like *minicom* or *seyon* to configure the TNC into kiss mode.

To create a KISS device you use the *kissattach* program. In it simplest form you can use the *kissattach* program as follows:

```
# /usr/sbin/kissattach /dev/ttyS0 radio
# kissparms -p radio -t 100 -s 100 -r 25
```

The *kissattach* command will create a KISS network device. These devices are called `ax[0-9]'. The first time you use the *kissattach* command it creates `ax0', the second time it creates `ax1' etc. Each KISS device has an associated serial port.

The *kissparms* command allows you to set various KISS parameters on a KISS device.

Specifically the example presented would create a KISS network device using the serial device `/dev/ttyS0' and the entry from the `/etc/ax25/axports` with a port name of `radio'. It then configures it with a *txdelay* and *slottime* of 100 milliseconds and a *ppersist* value of 25.

Please refer to the *man* pages for more information.

TOC

Configuring for Dual Port TNC's

The *mkiss* utility included in the ax25-utils distribution allows you to make use of both modems on a dual port TNC. Configuration is fairly simple. It works by taking a single serial device connected to a single multiport TNC and making it look like a number of devices each connected to a single port TNC. You do this **before** you do any of the AX.25 configuration. The devices that you then do the AX.25 configuration on are pseudo-TTY interfaces, (`/dev/ttyq*`), and not the actual serial device. Pseudo-TTY devices create a kind of pipe through which programs designed to talk to tty devices can talk to other programs designed to talk to tty devices. Each pipe has a master and a slave end. The master end is generally called `` `/dev/ptyq*' `` and the slave ends are called `` `/dev/ttyq*' ``. There is a one to one relationship between masters and slaves, so `/dev/ptyq0` is the master end of a pipe with `/dev/ttyq0` as its slave. You must open the master end of a pipe before opening the slave end. *mkiss* exploits this mechanism to split a single serial device into seperate devices.

Example: if you have a dual port tnc and it is connected to your `/dev/ttyS0` serial device at 9600 bps, the command:

```
# /usr/sbin/mkiss -s 9600 /dev/ttyS0 /dev/ptyq0 /dev/ptyq1
# /usr/sbin/kissattach /dev/ttyq0 port1
# /usr/sbin/kissattach /dev/ttyq1 port2
```

would create two pseudo-tty devices that each look like a normal single port TNC. You would then treat `/dev/ttyq0` and `/dev/ttyq1` just as you would a conventional serial device with TNC connected. This means you'd then use the *kissattach* command as described above, on each of those, in the example for AX.25 ports called `port1` and `port2`. You shouldn't use *kissattach* on the actual serial device as the *mkiss* program uses it.

The *mkiss* command has a number of optional arguments that you may wish to use. They are summarised as follows:

-c

enables the addition of a one byte checksum to each KISS frame. This is not supported by most KISS implementation, it is supported by the G8BPG KISS rom.

-s <speed>

sets the speed of the serial port.

-h

enables hardware handshaking on the serial port, it is off by default. Most KISS implementation do not support this, but some do.

-l

enables logging of information to the *syslog* logfile.

Creating a Baycom device.

Kernel Compile Options:

```
Code maturity level options  --->
    [*] Prompt for development and/or incomplete code/drivers
General setup  --->
    [*] Networking support
Network device support  --->
    [*] Network device support
    ...
    [*] Radio network interfaces
    [*] BAYCOM ser12 and par96 driver for AX.25
```

Thomas Sailer, `<sailer@ife.ee.ethz.ch>`, despite the popularly held belief that it would not work very well, has developed Linux support for Baycom modems. His driver supports the `Ser12` serial port, `Par96` and the enhanced `PicPar` parallel port modems. Further information about the modems themselves may be obtained from the Baycom Web site.

TOC

Your first step should be to determine the i/o and addresses of the serial or parallel port(s) you have Baycom modem(s) connected to. When you have these you must configure the Baycom driver with them.

The BayCom driver creates network devices called: `bc0`, `bc1`, `bc2` etc. when it is configured.

The *sethdlc* utility allows you to configure the driver with these parameters, or, if you have only one Baycom modem installed you may specify the parameters on the *insmod* commmand line when you load the Baycom module.

For example, a simple configuration. Disable the serial driver for COM1: then configure the Baycom driver for a Ser12 serial port modem on COM1: with the software DCD option enabled:

```
# setserial /dev/ttyS0 uart none
# insmod hdlcdrv
# insmod baycom mode="ser12*" iobase=0x3f8 irq=4
```

Par96 parallel port type modem on LPT1: using hardware DCD detection:

```
# insmod hdlcdrv
# insmod baycom mode="par96" iobase=0x378 irq=7 options=0
```

This is not really the preferred way to do it. The *sethdlc* utility works just as easily with one device as with many.

The *sethdlc man* page has the full details, but a couple of examples will illustrate the most important aspects of this configuration. The following examples assume you have already loaded the Baycom module using:

```
# insmod hdlcdrv
# insmod baycom
```

or that you compiled the kernel with the driver inbuilt.

Configure the `bc0` device driver as a Parallel port Baycom modem on LPT1: with software DCD:

```
# sethdlc -p -i bc0 mode par96 io 0x378 irq 7
```

Configure the `bc1` device driver as a Serial port Baycom modem on COM1:

```
# sethdlc -p -i bc1 mode "ser12*" io 0x3f8 irq 4
```

Configuring the AX.25 channel access parameters.

The AX.25 channel access parameters are the equivalent of the KISS ppersist, txdelay and slottime type parameters. Again you use the *sethdlc* utility for this.

Again the *sethdlc* man page is the source of the most complete information but another example of two won't hurt:

Configure the `bc0` device with TxDelay of 200 mS, SlotTime of 100 mS, PPersist of 40 and half duplex:

```
# sethdlc -i bc0 -a txd 200 slot 100 ppersist 40 half
```

Note that the timing values are in milliseconds.

Configuring the Kernel AX.25 to use the BayCom device

The BayCom driver creates standard network devices that the AX.25 Kernel code can use. Configuration is much the same as that for a PI or PacketTwin card.

The first step is to configure the device with an AX.25 callsign. The *ifconfig* utility may be used to perform this.

```
# /sbin/ifconfig bc0 hw ax25 VK2KTJ-15 up
```

will assign the BayCom device `bc0` the AX.25 callsign `VK2KTJ-15`. Alternatively you can use the *axparms* command you'll still need to use the *ifconfig* command to bring the device up though:

```
# ifconfig bc0 up
# axparms -setcall bc0 vk2ktj-15
```

The next step is to create an entry in the `/etc/ax25/axports` file as you would for any other device. The entry in the `axports` file is associated with the network device you've configured by the callsign you configure. The entry in the `axports` file that has the callsign that you configured the BayCom device with is the one that will be used to refer to it.

You may then treat the new AX.25 device as you would any other. You can configure it for TCP/IP, add it to ax25d and run NetRom or Rose over it as you please.

Creating a SoundModem device.

Kernel Compile Options:

```
Code maturity level options  --->
    [*] Prompt for development and/or incomplete code/drivers
General setup  --->
    [*] Networking support
Network device support  --->
    [*] Network device support
    ...
    [*] Radio network interfaces
    [*] Soundcard modem driver for AX.25
    [?] Soundmodem support for Soundblaster and compatible cards
    [?] Soundmodem support for WSS and Crystal cards
    [?] Soundmodem support for 1200 baud AFSK modulation
    [?] Soundmodem support for 4800 baud HAPN-1 modulation
    [?] Soundmodem support for 9600 baud FSK G3RUH modulation
```

Thomas Sailer has built a new driver for the kernel that allows you to use your soundcard as a modem. Connect your radio directly to your soundcard to play packet! Thomas recommends at least a 486DX2/66 if you want to use this software as all of the digital signal processing is done by the main CPU.

The driver currently emulates 1200 bps AFSK, 4800 HAPN and 9600 FSK (G3RUH compatible) modem types. The only sound cards currently supported are SoundBlaster and WindowsSoundSystem Compatible models. The sound cards require some circuitry to help them drive the Push-To-Talk circuitry, and information on this is available from Thomas's SoundModem PTT circuit web page. There are quite a few possible options, they are: detect the sound output from the soundcard, or use output from a parallel port, serial port or midi port. Circuit examples for each of these are on Thomas's site.

The SoundModem driver creates network devices called: `sm0`, `sm1`, `sm2` etc when it is configured.

Note: the SoundModem driver competes for the same resources as the Linux sound driver. So if you wish to use the SoundModem driver you must ensure that the Linux sound driver is not installed. You can of course compile them both as modules and insert and remove them as you wish.

Configuring the sound card.

The SoundModem driver does not initialise the sound card. The ax25-utils package includes a utility to do this called `*setcrystal*' that may be used for SoundCards based on the Crystal chipset. If you have some other card then you will have to use some other software to initialise it. Its syntax is fairly straightforward:

```
setcrystal [-w wssio] [-s sbio] [-f synthio] [-i irq] [-d dma] [-c dma2]
```

So, for example, if you wished to configure a soundblaster card at i/o base address 0x388, irq 10 and DMA 1 you would

```
# setcrystal -s 0x388 -i 10 -d 1
```

To configure a WindowSoundSystem card at i/o base address 0x534, irq 5, DMA 3 you would use:

```
# setcrystal -w 0x534 -i 5 -d 3
```

The `[-f synthio]` parameter is the set the synthesiser address, and the `[-c dma2]` parameter is to set the second DMA channel to allow full duplex operation.

Configuring the SoundModem driver.

When you have configured the soundcard you need to configure the driver telling it where the sound card is located and what sort of modem you wish it to emulate.

The *sethdlc* utility allows you to configure the driver with these parameters, or, if you have only one soundcard installed you may specify the parameters on the *insmod* commmand line when you load the SoundModem module.

For example, a simple configuration, with one SoundBlaster soundcard configured as described above emulating a 1200 bps modem:

```
# insmod hdlcdrv
# insmod soundmodem mode="sbc:afsk1200" iobase=0x220 irq=5 dma=1
```

This is not really the preferred way to do it. The *sethdlc* utility works just as easily with one device as with many.

The *sethdlc* man page has the full details, but a couple of examples will illustrate the most important aspects of this configuration. The following examples assume you have already loaded the SoundModem modules using:

```
# insmod hdlcdrv
# insmod soundmodem
```

or that you compiled the kernel with the driver inbuilt.

Configure the driver to support the WindowsSoundSystem card we configured above to emulate a G3RUH 9600 compatible modem as device `sm0` using a parallel port at 0x378 to key the Push-To-Talk:

```
# sethdlc -p -i sm0 mode wss:fsk9600 io 0x534 irq 5 dma 3 pario 0x378
```

Configure the driver to support the SoundBlaster card we configured above to emulate a 4800 bps HAPN modem as device `sm1` using the serial port located at 0x2f8 to key the Push-To-Talk:

```
# sethdlc -p -i sm1 mode sbc:hapn4800 io 0x388 irq 10 dma 1 serio 0x2f8
```

Configure the driver to support the SoundBlaster card we configured above to emulate a 1200 bps AFSK modem as device `sm1` using the serial port located at 0x2f8 to key the Push-To-Talk:

```
# sethdlc -p -i sm1 mode sbc:afsk1200 io 0x388 irq 10 dma 1 serio 0x2f8
```

Configuring the AX.25 channel access parameters.

The AX.25 channel access parameters are the equivalent of the KISS ppersist, txdelay and slottime type parameters. You use the *sethdlc* utility for this as well.

Again the *sethdlc* man page is the source of the most complete information but another example of two won't hurt:

Configure the `sm0` device with TxDelay of 100 mS, SlotTime of 50mS, PPersist of 128 and full duplex:

TOC

```
# sethdlc -i sm0 -a txd 100 slot 50 ppersist 128 full
```

Note that the timing values are in milliseconds.

Setting the audio levels and tuning the driver.

It is very important that the audio levels be set correctly for any radio based modem to work. This is equally true of the SoundModem. Thomas has developed some utility programs that make this task easier. They are called *smdiag* and *smmixer*.

smdiag

provides two types of display, either an oscilloscope type display or an eye pattern type display.

smmixer

allows you to actually adjust the transmit and receive audio levels.

To start the *smdiag* utility in 'eye' mode for the SoundModem device `sm0` you would use:

```
# smdiag -i sm0 -e
```

To start the *smmixer* utility for the SoundModem device `sm0` you would use:

```
# smmixer -i sm0
```

Configuring the Kernel AX.25 to use the SoundModem

The SoundModem driver creates standard network devices that the AX.25 Kernel code can use. Configuration is much the same as that for a PI or PacketTwin card.

The first step is to configure the device with an AX.25 callsign. The *ifconfig* utility may be used to perform this.

```
# /sbin/ifconfig sm0 hw ax25 VK2KTJ-15 up
```

will assign the SoundModem device `sm0` the AX.25 callsign `VK2KTJ-15`. Alternatively you can use the *axparms* command, but you still need the *ifconfig* utility to bring the device up:

```
# ifconfig sm0 up
# axparms -setcall sm0 vk2ktj-15
```

The next step is to create an entry in the `/etc/ax25/axports` file as you would for any other device. The entry in the `axports` file is associated with the network device you've configured by the callsign you configure. The entry in the `axports` file that has the callsign that you configured the SoundModem device with is the one that will be used to refer to it.

You may then treat the new AX.25 device as you would any other. You can configure it for TCP/IP, add it to ax25d and run NetRom or Rose over it as you please.

Creating a PI card device.

Kernel Compile Options:

```
General setup  --->
    [*] Networking support
Network device support  --->
    [*] Network device support
    ...
    [*] Radio network interfaces
    [*] Ottawa PI and PI/2 support for AX.25
```

TOC

The PI card device driver creates devices named `pi[0-9][ab]'. The first PI card detected will be allocated `pi0', the second `pi1' etc. The `a' and `b' refer to the first and second physical interface on the PI card. If you have built your kernel to include the PI card driver, and the card has been properly detected then you can use the following command to configure the network device:

```
# /sbin/ifconfig pi0a hw ax25 VK2KTJ-15 up
```

This command would configure the first port on the first PI card detected with the callsign `VK2KTJ-15` and make it active. To use the device all you now need to do is to configure an entry into your `/etc/ax25/axports` file with a matching callsign/ssid and you will be ready to continue on.

The PI card driver was written by `David Perry, <dp@hydra.carleton.edu>`

Creating a PacketTwin device.

Kernel Compile Options:

```
General setup  --->
    [*] Networking support
Network device support  --->
    [*] Network device support
    ...
    [*] Radio network interfaces
    [*] Gracilis PackeTwin support for AX.25
```

The PacketTwin card device driver creates devices named `pt[0-9][ab]'. The first PacketTwin card detected will be allocated `pt0', the second `pt1' etc. The `a' and `b' refer to the first and second physical interface on the PacketTwin card. If you have built your kernel to include the PacketTwin card driver, and the card has been properly detected then you can use the following command to configure the network device:

```
# /sbin/ifconfig pt0a hw ax25 VK2KTJ-15 up
```

This command would configure the first port on the first PacketTwin card detected with the callsign `VK2KTJ-15` and make it active. To use the device all you now need to do is to configure an entry into your `/etc/ax25/axports` file with a matching callsign/ssid and you will be ready to continue on.

The PacketTwin card driver was written by `Craig Small VK2XLZ, <csmall@triode.apana.org.au>`.

Creating a generic SCC device.

Kernel Compile Options:

```
General setup  --->
    [*] Networking support
Network device support  --->
    [*] Network device support
    ...
    [*] Radio network interfaces
    [*] Z8530 SCC KISS emulation driver for AX.25
```

Joerg Reuter, DL1BKE, `jreuter@poboxes.com` has developed generic support for Z8530 SCC based cards. His driver is configurable to support a range of different types of cards and present an interface that looks like a KISS TNC so you can treat it as though it were a KISS TNC.

Obtaining and building the configuration tool package.

While the kernel driver is included in the standard kernel distribution, Joerg distributes more recent versions of his driver with the suite of configuration tools that you will need to obtain as well.

You can obtain the configuration tools package from:

Joerg's web page

or:

db0bm.automation.fh-aachen.de

```
/incoming/dl1bke/
```

or:

insl1.etec.uni-karlsruhe.de

```
/pub/hamradio/linux/z8530/
```

or:

ftp.ucsd.edu

```
/hamradio/packet/tcpip/linux
/hamradio/packet/tcpip/incoming/
```

You will find multiple versions, choose the one that best suits the kernel you intend to use:

```
z8530drv-2.4a.dl1bke.tar.gz   2.0.*
z8530drv-utils-3.0.tar.gz    2.1.6 or greater
```

The following commands were what I used to compile and install the package for kernel version 2.0.30:

```
# cd /usr/src
# gzip -dc z8530drv-2.4a.dl1bke.tar.gz | tar xvpofz -
# cd z8530drv
# make clean
# make dep
# make module         # If you want to build the driver as a module
# make for_kernel     # If you want the driver to built into your kernel
# make install
```

After the above is complete you should have three new programs installed in your `/sbin` directory: *gencfg*, *sccinit* and *sccstat*. It is these programs that you will use to configure the driver for your card.

You will also have a group of new special device files created in your `/dev` called `scc0-scc7`. These will be used later and will be the `KISS' devices you will end up using.

If you chose to 'make for_kernel' then you will need to recompile your kernel. To ensure that you include support for the z8530 driver you must be sure to answer `Y' to: `Z8530 SCC kiss emulation driver for AX.25' when asked during a kernel `make config'.

If you chose to 'make module' then the new `scc.o` will have been installed in the appropriate `/lib/modules` directory and you do not need to recompile your kernel. Remember to use the *insmod* command to load the module before your try and configure it.

TOC

Configuring the driver for your card.

The z8530 SCC driver has been designed to be as flexible as possible so as to support as many different types of cards as possible. With this flexibility has come some cost in configuration.

There is more comprehensive documentation in the package and you should read this if you have any problems. You should particularly look at `doc/scc_eng.doc` or `doc/scc_ger.doc` for more detailed information. I've paraphrased the important details, but as a result there is a lot of lower level detail that I have not included.

The main configuration file is read by the *sccinit* program and is called `/etc/z8530drv.conf`. This file is broken into two main stages: Configuration of the hardware parameters and channel configuration. After you have configured this file you need only add:

```
# sccinit
```

into the `rc` file that configures your network and the driver will be initialised according to the contents of the configuration file. You must do this before you attempt to use the driver.

Configuration of the hardware parameters.

The first section is broken into stanzas, each stanza representing an 8530 chip. Each stanza is a list of keywords with arguments. You may specify up to four SCC chips in this file by default. The `#define MAXSCC 4` in `scc.c` can be increased if you require support for more.

The allowable keywords and arguments are:

chip

the `chip` keyword is used to separate stanzas. It will take anything as an argument. The arguments are not used.

data_a

this keyword is used to specify the address of the data port for the z8530 channel `A'. The argument is a hexadecimal number e.g. 0x300

ctrl_a

this keyword is used to specify the address of the control port for the z8530 channel `A'. The arguments is a hexadecimal number e.g. 0x304

data_b

this keyword is used to specify the address of the data port for the z8530 channel `B'. The argument is a hexadecimal number e.g. 0x301

ctrl_b

this keyword is used to specify the address of the control port for the z8530 channel `B'. The arguments is a hexadecimal number e.g. 0x305

irq

this keyword is used to specify the IRQ used by the 8530 SCC described in this stanza. The argument is an integer e.g. 5

pclock

this keyword is used to specify the frequency of the clock at the PCLK pin of the 8530. The argument is an integer frequency in Hz which defaults to 4915200 if the keyword is not supplied.

board

the type of board supporting this 8530 SCC. The argument is a character string. The allowed values are:

PA0HZP

the PA0HZP SCC Card

EAGLE

the Eagle card

PC100

the DRSI PC100 SCC card

PRIMUS

the PRIMUS-PC (DG9BL) card

BAYCOM

BayCom (U)SCC card

escc

this keyword is optional and is used to enable support for the Extended SCC chips (ESCC) such as the 8580, 85180, or the 85280. The argument is a character string with allowed values of `yes' or `no'. The default is `no'.

vector

this keyword is optional and specifies the address of the vector latch (also known as "intack port") for PA0HZP cards. There can be only one vector latch for all chips. The default is 0.

special

this keyword is optional and specifies the address of the special function register on several cards. The default is 0.

option

this keyword is optional and defaults to 0.

Some example configurations for the more popular cards are as follows:

BayCom USCC

```
chip    1
data_a  0x300
ctrl_a  0x304
data_b  0x301
ctrl_b  0x305
irq     5
board   BAYCOM
#
# SCC chip 2
#
chip    2
data_a  0x302
ctrl_a  0x306
data_b  0x303
ctrl_b  0x307
board   BAYCOM
```

PA0HZP SCC card

```
chip 1
data_a 0x153
data_b 0x151
ctrl_a 0x152
ctrl_b 0x150
irq 9
```

```
        pclock 4915200
        board PA0HZP
        vector 0x168
        escc no
        #
        #
        #
        chip 2
        data_a 0x157
        data_b 0x155
        ctrl_a 0x156
        ctrl_b 0x154
        irq 9
        pclock 4915200
        board PA0HZP
        vector 0x168
        escc no
```

DRSI SCC card

```
        chip 1
        data_a 0x303
        data_b 0x301
        ctrl_a 0x302
        ctrl_b 0x300
        irq 7
        pclock 4915200
        board DRSI
        escc no
```

If you already have a working configuration for your card under NOS, then you can use the *gencfg* command to convert the PE1CHL NOS driver commands into a form suitable for use in the z8530 driver configuration file.

To use *gencfg* you simply invoke it with the same parameters as you used for the PE1CHL driver in NET/NOS. For example:

```
    # gencfg 2 0x150 4 2 0 1 0x168 9 4915200
```

will generate a skeleton configuration for the OptoSCC card.

Channel Configuration

The Channel Configuration section is where you specify all of the other parameters associated with the port you are configuring. Again this section is broken into stanzas. One stanza represents one logical port, and therefore there would be two of these for each one of the hardware parameters stanzas as each 8530 SCC supports two ports.

These keywords and arguments are also written to the `/etc/z8530drv.conf` file and must appear **after** the hardware parameters section.

Sequence is very important in this section, but if you stick with the suggested sequence it should work ok. The keywords and arguments are:

device

this keyword must be the first line of a port definition and specifies the name of the special device file that the rest of the configuration applies to. e.g. `/dev/scc0`

speed

this keyword specifies the speed in bits per second of the interface. The argument is an integer: e.g. `1200`

TOC

clock

this keyword specifies where the clock for the data will be sourced. Allowable values are:

dpll

normal halfduplex operation

external

MODEM supplies its own Rx/Tx clock

divider

use fullduplex divider if installed.

mode

this keyword specifies the data coding to be used. Allowable arguments are: `nrzi` or `nrz`

rxbuffers

this keyword specifies the number of receive buffers to allocate memory for. The argument is an integer, e.g. 8.

txbuffers

this keyword specifies the number of transmit buffers to allocate memory for. The argument is an integer, e.g. 8.

bufsize

this keyword specifies the size of the receive and transmit buffers. The arguments is in bytes and represents the total length of the frame, so it must also take into account the AX.25 headers and not just the length of the data field. This keyword is optional and default to `384`

txdelay

the KISS transmit delay value, the argument is an integer in mS.

persist

the KISS persist value, the argument is an integer.

slot

the KISS slot time value, the argument is an integer in mS.

tail

the KISS transmit tail value, the argument is an integer in mS.

fulldup

the KISS full duplex flag, the argument is an integer. `1`==Full Duplex, `0`==Half Duplex.

wait

the KISS wait value, the argument is an integer in mS.

min

the KISS min value, the argument is an integer in S.

maxkey

the KISS maximum keyup time, the argument is an integer in S.

idle

the KISS idle timer value, the argument is an integer in S.

maxdef

the KISS maxdef value, the argument is an integer.

group

TOC

the KISS group value, the argument is an integer.

txoff

the KISS txoff value, the argument is an integer in mS.

softdcd

the KISS softdcd value, the argument is an integer.

slip

the KISS slip flag, the argument is an integer.

Using the driver.

To use the driver you simply treat the `/dev/scc*` devices just as you would a serial tty device with a KISS TNC connected to it. For example, to configure Linux Kernel networking to use your SCC card you could use something like:

```
# kissattach -s 4800 /dev/scc0 VK2KTJ
```

You can also use NOS to attach to it in precisely the same way. From JNOS for example you would use something like:

```
attach asy scc0 0 ax25 scc0 256 256 4800
```

The *sccstat* and *sccparam* tools.

To assist in the diagnosis of problems you can use the *sccstat* program to display the current configuration of an SCC device. To use it try:

```
# sccstat /dev/scc0
```

you will displayed a very large amount of information relating to the configuration and health of the `/dev/scc0` SCC port.

The *sccparam* command allows you to change or modify a configuration after you have booted. Its syntax is very similar to the NOS `param` command, so to set the `txtail` setting of a device to 100mS you would use:

```
# sccparam /dev/scc0 txtail 0x8
```

Creating a BPQ ethernet device.

Kernel Compile Options:

```
General setup  --->
    [*] Networking support
Network device support  --->
    [*] Network device support
    ...
    [*] Radio network interfaces
    [*] BPQ Ethernet driver for AX.25
```

Linux supports BPQ Ethernet compatibility. This enables you to run the AX.25 protocol over your Ethernet LAN and to interwork your linux machine with other BPQ machines on the LAN.

The BPQ network devices are named `bpq[0-9]'. The `bpq0' device is associated with the `eth0' device, the `bpq1' device with the `eth1' device etc.

TOC

Configuration is quite straightforward. You firstly must have configured a standard Ethernet device. This means you wi have compiled your kernel to support your Ethernet card and tested that this works. Refer to the Ethernet-HOWTO for r information on how to do this.

To configure the BPQ support you need to configure the Ethernet device with an AX.25 callsign. The following command will do this for you:

```
# /sbin/ifconfig bpq0 hw ax25 vk2ktj-14 up
```

Again, remember that the callsign you specify should match the entry in the `/etc/ax25/axports` file that you wish to use for this port.

Configuring the BPQ Node to talk to the Linux AX.25 support.

BPQ Ethernet normally uses a multicast address. The Linux implementation does not, and instead it uses the normal Ethernet broadcast address. The NET.CFG file for the BPQ ODI driver should therefore be modifified to look similar to this:

```
LINK SUPPORT

        MAX STACKS 1
        MAX BOARDS 1

LINK DRIVER E2000                    ; or other MLID to suit your card

        INT 10                       ;
        PORT 300                     ; to suit your card

        FRAME ETHERNET_II

        PROTOCOL BPQ 8FF ETHERNET_II ; required for BPQ - can change PID

BPQPARAMS                            ; optional - only needed if you want
                                     ; to override the default target addr

        ETH_ADDR  FF:FF:FF:FF:FF:FF  ; Target address
```

6.2 Creating the `/etc/ax25/axports` file.

The `/etc/ax25/axports` is a simple text file that you create with a text editor. The format of the `/etc/ax25/axports` file is as follows:

```
portname  callsign  baudrate  paclen  window  description
```

where:

portname

is a text name that you will refer to the port by.

callsign

is the AX.25 callsign you want to assign to the port.

baudrate

is the speed at which you wish the port to communicate with your TNC.

paclen

is the maximum packet length you want to configure the port to use for AX.25 connected mode connections.

window

is the AX.25 window (K) parameter. This is the same as the `MAXFRAME` setting of many tnc's.

description

is a textual description of the port.

In my case, mine looks like:

```
radio    VK2KTJ-15       4800        256     2       4800bps 144.800 MHz
ether    VK2KTJ-14       10000000    256     2       BPQ/ethernet device
```

Remember, you must assign unique callsign/ssid to each AX.25 port you create. Create one entry for each AX.25 device you want to use, this includes KISS, Baycom, SCC, PI, PT and SoundModem ports. Each entry here will describe exactly one AX.25 network device. The entries in this file are associated with the network devices by the callsign/ssid. This is at least one good reason for requiring unique callsign/ssid.

6.3 Configuring AX.25 routing.

You may wish to configure default digipeaters paths for specific hosts. This is useful for both normal AX.25 connections and also IP based connections. The *axparms* command enables you to do this. Again, the *man* page offers a complete description, but a simple example might be:

```
# /usr/sbin/axparms -route add radio VK2XLZ VK2SUT
```

This command would set a digipeater entry for `VK2XLZ` via `VK2SUT` on the AX.25 port named `radio`.

---

7. Configuring an AX.25 interface for TCP/IP.

It is very simple to configure an AX.25 port to carry TCP/IP. If you have KISS interfaces then there are two methods for configuring an IP address. The *kissattach* command has an option that allows you to do specify an IP address. The more conventional method using the *ifconfig* command will work on all interface types.

So, modifying the previous KISS example:

```
# /usr/sbin/kissattach -i 44.136.8.5 -m 512 /dev/ttyS0 radio
# /sbin/route add -net 44.136.8.0 netmask 255.255.255.0 ax0
# /sbin/route add default ax0
```

to create the AX.25 interface with an IP address of `44.136.8.5` and an `MTU` of `512` bytes. You should still use the *ifconfig* to configure the other parameters if necessary.

If you have any other interface type then you use the *ifconfig* program to configure the ip address and netmask details for the port and add a route via the port, just as you would for any other TCP/IP interface. The following example is for a PI card device, but would work equally well for any other AX.25 network device:

```
# /sbin/ifconfig pi0a 44.136.8.5 netmask 255.255.255.0 up
# /sbin/ifconfig pi0a broadcast 44.136.8.255 mtu 512
# /sbin/route add -net 44.136.8.0 netmask 255.255.255.0 pi0a
# /sbin/route add default pi0a
```

The commands listed above are typical of the sort of configuration many of you would be familiar with if you have used NOS or any of its derivatives or any other TCP/IP software. Note that the default route might not be required in your configuration if you have some other network device configured.

To test it out, try a ping or a telnet to a local host.

```
# ping -i 5 44.136.8.58
```

Note the use of the `` `-i 5' `` arguments to *ping* to tell it to send pings every 5 seconds instead of its default of 1 second.

8. Configuring a NetRom port.

The NetRom protocol relies on, and uses the AX.25 ports you have created. The NetRom protocol rides on top of the AX.25 protocol. To configure NetRom on an AX.25 interface you must configure two files. One file describes the Netrom interfaces, and the other file describes which of the AX.25 ports will carry NetRom. You can configure multiple NetRom ports, each with its own callsign and alias, the same procedure applies for each.

8.1 Configuring `/etc/ax25/nrports`

The first is the `/etc/ax25/nrports` file. This file describes the NetRom ports in much the same way as the `/etc/ax25/axports` file describes the AX.25 ports. Each NetRom device you wish to create must have an entry in the `/etc/ax25/nrports` file. Normally a Linux machine would have only one NetRom device configured that would use a number of the AX.25 ports defined. In some situations you might wish a special service such as a BBS to have a seperate NetRom alias and so you would create more than one.

This file is formatted as follows:

```
name callsign  alias  paclen   description
```

Where:

name

is the text name that you wish to refer to the port by.

callsign

is the callsign that the NetRom traffic from this port will use. Note, this is **not** that address that users should connect to to get access to a *node* style interface. (The node program is covered later). This callsign/ssid should be unique and should not appear elsewhere in either of the `/etc/ax25/axports` or the `/etc/ax25/nrports` files.

alias

is the NetRom alias this port will have assigned to it.

paclen

is the maximum size of NetRom frames transmitted by this port.

description

is a free text description of the port.

An example would look something like the following:

```
netrom  VK2KTJ-9        LINUX   236     Linux Switch Port
```

This example creates a NetRom port known to the rest of the NetRom network as `LINUX:VK2KTJ-9'.

This file is used by programs such as the *call* program.

8.2 Configuring /etc/ax25/nrbroadcast

The second file is the `/etc/ax25/nrbroadcast` file. This file may contain a number of entries. There would normally be one entry for each AX.25 port that you wish to allow NetRom traffic on.

This file is formatted as follows:

```
axport min_obs def_qual worst_qual verbose
```

Where:

axport

is the port name obtained from the `/etc/ax25/axports` file. If you do not have an entry in `/etc/ax25/nrbroadcasts` for a port then this means that no NetRom routing will occur and any received NetRom broadcasts will be ignored for that port.

min_obs

is the minimum obselesence value for the port.

def_qual

is the default quality for the port.

worst_qual

is the worst quality value for the port, any routes under this quality will be ignored.

verbose

is a flag determining whether full NetRom routing broadcasts will occur from this port or only a routing broadcast advertising the node itself.

An example would look something like the following:

```
radio    1       200      100         1
```

8.3 Creating the NetRom Network device

When you have the two configuration files completed you must create the NetRom device in much the same way as you did for the AX.25 devices. This time you use the *nrattach* command. The *nrattach* works in just the same way as the *axattach* command except that it creates NetRom network devices called `nr[0-9]'. Again, the first time you use the *nrattach* command it creates the `nr0' device, the

second time it creates the `nr1' network devices etc. To create the network device for the NetRom p we've defined we would use:

```
# nrattach netrom
```

This command would start the NetRom device (`nr0`) named `netrom` configured with the details specified in the `/etc/ax25/nrports` file.

8.4 Starting the NetRom daemon

The Linux kernel does all of the NetRom protocol and switching, but does not manage some functions. The NetRom daemon manages the NetRom routing tables and generates the NetRom routing broadcasts. You start NetRom daemon with the command:

```
# /usr/sbin/netromd -i
```

You should soon see the `/proc/net/nr_neigh` file filling up with information about your NetRom neighbours.

Remember to put the `/usr/sbin/netromd` command in your *rc* files so that it is started automatically each time you reboot.

8.5 Configuring NetRom routing.

You may wish to configure static NetRom routes for specific hosts. The *nrparms* command enables you to do this. Again, the *man* page offers a complete description, but a simple example might be:

```
# /usr/sbin/nrparms -nodes VK2XLZ-10 + #MINTO 120 5 radio VK2SUT-9
```

This command would set a NetRom route to `#MINTO:VK2XLZ-10` via a neighbour `VK2SUT-9` on my AX.25 port called `radio'.

You can manually create entries for new neighbours using the *nrparms* command as well. For example:

```
# /usr/sbin/nrparms -routes radio VK2SUT-9 + 120
```

This command would create `VK2SUT-9` as a NetRom neighbour with a quality of `120` and this will be locked and will not be deleted automatically.

---

9. Configuring a NetRom interface for TCP/IP.

Configuring a NetRom interface for TCP/IP is almost identical to configuring an AX.25 interface for TCP/IP.

Again you can either specify the ip address and mtu on the *nrattach* command line, or use the *ifconfig* and *route* commands, but you need to manually add *arp* entries for hosts you wish to route to because there is no mechanism available for your machine to learn what NetRom address it should use to reach a particular IP host.

So, to create an `nr0` device with an IP address of `44.136.8.5`, an mtu of `512` and configured with the details from the `/etc/ax25/nrports` file for a NetRom port named `netrom` you would use:

```
# /usr/sbin/nrattach -i 44.136.8.5 -m 512 netrom
# route add 44.136.8.5 nr0
```

or you could use something like the following commands manually:

```
# /usr/sbin/nrattach netrom
# ifconfig nr0 44.136.8.5 netmask 255.255.255.0 hw netrom VK2KTJ-9
# route add 44.136.8.5 nr0
```

Then for each IP host you wish to reach via NetRom you need to set route and arp entries. To reach a destination host with an IP address of `44.136.80.4` at NetRom address `BBS:VK3BBS` via a NetRom neighbour with callsign `VK2SUT-0` you would use commands as follows:

```
# route add 44.136.80.4 nr0
# arp -t netrom -s 44.136.80.4 vk2sut-0
# nrparms -nodes vk3bbs + BBS 120 6 sl0 vk2sut-0
```

The `` `120' `` and `` `6' `` arguments to the *nrparms* command are the NetRom *quality* and *obsolescence count* values for the route.

10. Configuring a Rose port.

The Rose packet layer protocol is similar to layer three of the X.25 specification. The kernel based Rose support is a **modified** version of the FPAC Rose implementation.

The Rose packet layer protocol protocol relies on, and uses the AX.25 ports you have created. The Rose protocol rides on top of the AX.25 protocol. To configure Rose you must create a configuration file that describes the Rose ports you want. You can create multiple Rose ports if you wish, the same procedure applies for each.

10.1 Configuring `/etc/ax25/rsports`

The file where you configure your Rose interfaces is the `/etc/ax25/rsports` file. This file describes the Rose port in much the same way as the `/etc/ax25/axports` file describes the AX.25 ports.

This file is formatted as follows:

```
name  addresss  description
```

Where:

name

is the text name that you wish to refer to the port by.

address

is the 10 digit Rose address you wish to assign to this port.

description

is a free text description of the port.

An example would look something like the following:

```
rose  5050294760  Rose Port
```

Note that Rose will use the default callsign/ssid configured on each AX.25 port unless you specify otherwise.

To configure a seperate callsign/ssid for Rose to use on each port you use the *rsparms* command as follows:

```
# /usr/sbin/rsprams -call VK2KTJ-10
```

This example would make Linux listen for and use the callsign/ssid `VK2KTJ-10` on all of the confi AX.25 ports for Rose calls.

TOC

10.2 Creating the Rose Network device.

When you have created the `/etc/ax25/rsports` file you may create the Rose device in much the same way as you did for the AX.25 devices. This time you use the *rsattach* command. The *rsattach* command creates network devices named `` `rose[0-5]' ``. The first time you use the *rsattach* command it create the `` `rose0' ``device, the second time it creates the `` `rose1' ``device etc. For example:

```
# rsattach rose
```

This command would start the Rose device (`rose0`) configured with the details specified in the `/etc/ax25/rsports` file for the entry named `` `rose' ``.

10.3 Configuring Rose Routing

The Rose protocol currently supports only static routing. The *rsparms* utility allows you to configure your Rose routing table under Linux.

For example:

```
# rsparms -nodes add 5050295502 radio vk2xlz
```

would add a route to Rose node `5050295502` via an AX.25 port named `` `radio' ``in your `/etc/ax25/axports` file to a neighbour with the callsign `VK2XLZ`.

You may specify a route with a mask to capture a number of Rose destinations into a single routing entry. The syntax looks like:

```
# rsparms -nodes add 5050295502/4 radio vk2xlz
```

which would be identical to the previous example except that it would match any destination address that matched the first four digits supplied, in this case any address commencing with the digits `5050`. An alternate form for this command is:

```
# rsparms -nodes add 5050/4 radio vk2xlz
```

which is probably the less ambiguous form.

---

11. Making AX.25/NetRom/Rose calls.

Now that you have all of your AX.25, NetRom and Rose interfaces configured and active, you should be able to make test calls.

The AX25 Utilities package includes a program called `*call*' which is a splitscreen terminal program for AX.25, NetRom and Rose.

A simple AX.25 call would look like:

```
/usr/bin/call radio VK2DAY via VK2SUT
```

A simple NetRom call to a node with an alias of `SUNBBS` would look like:

```
/usr/bin/call netrom SUNBBS
```

A simple Rose call to `HEARD` at node `5050882960` would look like:

```
/usr/bin/call rose HEARD 5050882960
```

Note: you must tell call which port you wish to make the call on, as the same destination node might be reachable on any of the ports you have configured.

The *call* program is a linemode terminal program for making AX.25 calls. It recognises lines that start with `~' as command lines. The `~.' command will close the connection.

Please refer to the man page in `/usr/man` for more information.

12. Configuring Linux to accept Packet connections.

Linux is a powerful operating system and offers a great deal of flexibility in how it is configured. With this flexibility comes a cost in configuring it to do what you want. When configuring your Linux machine to accept incoming AX.25, NetRom or Rose connections there are a number of questions you need to ask yourself. The most important of which is: "What do I want users to see when they connect?". People are developing neat little applications that may be used to provide services to callers, a simple example is the *pms* program included in the AX25 utilities, a more complex example is the *node* program also included in the AX25 utilities. Alternatively you might want to give users a login prompt so that they can make use of a shell account, or you might even have written your own program, such as a customised database or a game, that you want people to connect to. Whatever you choose, you must tell the AX.25 software about this so that it knows what software to run when it accepts an incoming AX.25 connection.

The *ax25d* program is similar to the *inetd* program commonly used to accept incoming TCP/IP connections on unix machines. It sits and listens for incoming connections, when it detects one it goes away and checks a configuration file to determine what program to run and connect to that connection. Since this the standard tool for accepting incoming AX.25, NetRom and Rose connections I'll describe how to configure it.

12.1 Creating the `/etc/ax25/ax25d.conf` file.

This file is the configuration file for the *ax25d* AX.25 daemon which handles incoming AX.25, NetRom and Rose connections.

The file is a little cryptic looking at first, but you'll soon discover it is very simple in practice, with a small trap for you to be wary of.

The general format of the `ax25d.conf` file is as follows:

```
# This is a comment and is ignored by the ax25d program.
[port_name] || <port_name> || {port_name}
<peer1>    window T1 T2 T3 idle N2 <mode> <uid> <cmd> <cmd-name> <arguments>
<peer2>    window T1 T2 T3 idle N2 <mode> <uid> <cmd> <cmd-name> <arguments>
parameters window T1 T2 T3 idle N2 <mode>
<peer3>    window T1 T2 T3 idle N2 <mode> <uid> <cmd> <cmd-name> <arguments>
   ...
default    window T1 T2 T3 idle N2 <mode> <uid> <cmd> <cmd-name> <arguments>
```

Where:

#

at the start of a line marks a comment and is completely ignored by the *ax25d* program.

<port_name>

is the name of the AX.25, NetRom or Rose port as specified in the `/etc/ax25/axports`, `/etc/ax25/nrports` and `/etc/ax25/rsports` files. The name of the port is surrounded by the `[]' brackets if it is an AX.25 port, the `<>' brackets if it is a NetRom port, or the `{}' brackets if it is a Rose port. There is an alternate form for this field, and that is use prefix the port name with `callsign/ssid via' to indicate that you wish accept calls to the callsign/ssid via this interface. The example should more clearly illustrate this.

<peer>

is the callsign of the peer node that this particular configuration applies to. If you don't specify an SSID here any SSID will match.

window

is the AX.25 Window parameter (K) or MAXFRAME parameter for this configuration.

T1

is the Frame retransmission (T1) timer in half second units.

T2

is the amount of time the AX.25 software will wait for another incoming frame before preparing a response in 1 second units.

T3

is the amount of time of inactivity before the AX.25 software will disconnect the session in 1 second units.

idle

is the idle timer value in seconds.

N2

is the number of consecutive retransmissions that will occur before the connection is closed.

<mode>

provides a mechanism for determining certain types of general permissions. The modes are enabled or disabled by supplying a combination of characters, each representing a permission. The characters may be in either upper or lower case and must be in a single block with no spaces.

u/U

UTMP - currently unsupported.

v/V

Validate call - currently unsupported.

q/Q

Quiet - Don't log connection

n/N

check NetRom Neighbour - currently unsupported.

d/D

Disallow Digipeaters - Connections must be direct, not digipeated.

l/L

Lockout - Don't allow connection.

***/0**

marker - place marker, no mode set.

<uid>

is the userid that the program to be run to support the connection should be run as.

<cmd>

is the full pathname of the command to be run, with no arguments specified.

<cmd-name>

is the text that should appear in a *ps* as the command name running (normally the same as <cmd> except without the directory path information.

<arguments>

are the command line argument to be passed to the <:cmd> when it is run. You pass useful information into these arguments by use of the following tokens:

%d

Name of the port the connection was received on.

%U

AX.25 callsign of the connected party without the SSID, in uppercase.

%u

AX.25 callsign of the connected party without the SSID, in lowercase.

%S

AX.25 callsign of the connected party with the SSID, in uppercase.

%s

AX.25 callsign of the connected party with the SSID, in lowercase.

%P

AX.25 callsign of the remote node that the connection came in from without the SSID, in uppercase.

%p

AX.25 callsign of the remote node that the connection came in from without the SSID, in lowercase.

%R

AX.25 callsign of the remote node that the connection came in from with the SSID, in uppercase.

%r

AX.25 callsign of the remote node that the connection came in from with the SSID, in lowercase.

You need one section in the above format for each AX.25, NetRom or Rose interface you want to accept incoming AX.25, NetRom or Rose connections on.

There are two special lines in the paragraph, one starts with the string `parameters' and the other starts with the string `default' (yes there is a difference). These lines serve special functions.

The `default' lines purpose should be obvious, this line acts as a catch-all, so that any incoming connection on the <interface_call> interface that doesn't have a specific rule will match the `default' rule. If you don't have a `default' rule, then any connections not matching any specific rule will be disconnected immediately without notice.

The `parameters' line is a little more subtle, and here is the trap I mentioned earlier. In any of the fields for any definition for a peer you can use the `*' character to say `use the default value'. The `parameters' line is what sets those default values. The kernel software itself has some defaults which will be used if you don't specify any using the `parameters' entry. The trap is that the these defaults apply **only** to those rules **below** the `parameters' line, not to those above. You may have more than one `parameters' rule per interface definition, and in this way you may create groups of default configurations. It is important to note that the `parameters' rule does not allow you to set the `uid' or `command' fields.

12.2 A simple example ax25d.conf file.

TOC

Ok, an illustrative example:

```
# ax25d.conf for VK2KTJ - 02/03/97
# This configuration uses the AX.25 port defined earlier.

# <peer> Win T1  T2  T3  idl N2 <mode> <uid> <exec> <argv[0]>[<args....>]

[VK2KTJ-0 via radio]
parameters 1    10  *  *  *   *   *
VK2XLZ     *     *  *  *  *   *   *    root  /usr/sbin/axspawn axspawn %u +
VK2DAY     *     *  *  *  *   *   *    root  /usr/sbin/axspawn axspawn %u +
NOCALL     *     *  *  *  *   *   L
default    1    10  5 100 180 5   *    root  /usr/sbin/pms pms -a -o vk2ktj

[VK2KTJ-1 via radio]
default    *     *    *   *   *   0    root /usr/sbin/node node

<netrom>
parameters 1    10  *  *  *   *   *
NOCALL     *     *  *  *  *   *   L
default    *     *  *  *  *   *   0        root /usr/sbin/node node

{VK2KTJ-0 via rose}
parameters 1    10  *  *  *   *   *
VK2XLZ     *     *  *  *  *   *   *    root  /usr/sbin/axspawn axspawn %u +
VK2DAY     *     *  *  *  *   *   *    root  /usr/sbin/axspawn axspawn %u +
NOCALL     *     *  *  *  *   *   L
default    1    10  5 100 180 5   *    root  /usr/sbin/pms pms -a -o vk2ktj

{VK2KTJ-1 via rose}
default    *     *    *   *   *   0    root /usr/sbin/node node radio
```

This example says that anybody attempting to connect to the callsign `VK2KTJ-0' heard on the AX.25 port called `radio' will have the following rules applied:

Anyone whose callsign is set to `NOCALL' should be locked out, note the use of mode `L'.

The `parameters` line changes two parameters from the kernel defaults (Window and T1) and will run the */usr/sbin/axspawn* program for them. Any copies of */usr/sbin/axspawn* run this way will appear as *axspawn* in a *ps* listing for convenience. The next two lines provide definitions for two stations who will receive those permissions.

The last line in the paragraph is the `catch all' definition that everybody else will get (including VK2XLZ and VK2DAY using any other SSID other than -1). This definition sets all of the parameters implicitly and will cause the *pms* program to be run with a command line argument indicating that it is being run for an AX.25 connection, and that the owner callsign is `VK2KTJ`. (See the `Configuring the PMS' section below for more details).

The next configuration accepts calls to `VK2KTJ-1` via the `radio` port. It runs the *node* program for everybody that connects to it.

The next configuration is a NetRom configuration, note the use of the greater-then and less-than braces instead of the square brackets. These denote a NetRom configuration. This configuration is simpler, it simply says that anyone

TOC

connecting to our NetRom port called `netrom' will have the *node* program run for them, unless they have a calls of `NOCALL' in which case they will be locked out.

The last two configurations are for incoming Rose connections. The first for people who have placed calls to `vk2ktj-0' and the second for `VK2KTJ-1 at the our Rose node address. These work precisely the same way. Not the use of the curly braces to distinguish the port as a Rose port.

This example is a contrived one but I think it illustrates clearly the important features of the syntax of the configuration file. The configuration file is explained fully in the `ax25d.conf` *man* page. A more detailed example is included in the `ax25-utils` package that might be useful to you too.

12.3 Starting *ax25d*

When you have the two configuration files completed you start *ax25d* with the command:

```
# /usr/sbin/ax25d
```

When this is run people should be able to make AX.25 connections to your Linux machine. Remember to put the `ax25d` command in your *rc* files so that it is started automatically when you reboot each time.

---

13. Configuring the *node* software.

The *node* software was developed by Tomi Manninen `<tomi.manninen@hut.fi>` and was based on the original PMS program. It provides a fairly complete and flexible node capability that is easily configured. It allows users once they are connected to make Telnet, NetRom, Rose, and AX.25 connections out and to obtain various sorts of information such as Finger, Nodes and Heard lists etc. You can configure the node to execute any Linux command you wish fairly simply.

The node would normally be invoked from the *ax25d* program although it is also capable of being invoked from the TCP/IP *inetd* program to allow users to telnet to your machine and obtain access to it, or by running it from the command line.

13.1 Creating the `/etc/ax25/node.conf` file.

The `node.conf` file is where the main configuration of the node takes place. It is a simple text file and its format is as follows:

```
# /etc/ax25/node.conf
# configuration file for the node(8) program.
#
# Lines beginning with '#' are comments and are ignored.

# Hostname
# Specifies the hostname of the node machine
hostname        radio.gw.vk2ktj.ampr.org

# Local Network
# allows you to specify what is consider 'local' for the
# purposes of permission checking using nodes.perms.
localnet        44.136.8.96/29

# Hide Ports
# If specified allows you to make ports invisible to users. The
# listed ports will not be listed by the (P)orts command.
hiddenports     rose netrom

# Node Identification.
# this will appear in the node prompt
NodeId          LINUX:VK2KTJ-9

# NetRom port
# This is the name of the netrom port that will be used for
# outgoing NetRom connections from the node.
NrPort          netrom

# Node Idle Timeout
```

TOC

```
# Specifies the idle time for connections to this node in seconds.
idletimout      1800

# Connection Idle Timeout
# Specifies the idle timer for connections made via this node in
# seconds.
conntimeout     1800

# Reconnect
# Specifies whether users should be reconnected to the node
# when their remote connections disconnect, or whether they
# should be disconnected complete.
reconnect       on

# Command Aliases
# Provide a way of making complex node commands simple.
alias           CONV    "telnet vk1xwt.ampr.org 3600"
alias           BBS     "connect radio vk2xsb"

# Externam Command Aliases
# Provide a means of executing external commands under the node.
# extcmd <cmdname> <flag> <userid> <command>
# Flag == 1 is the only implemented function.
# <command> is formatted as per ax25d.conf
extcmd          PMS     1       root    /usr/sbin/pms pms -u %U -o VK2KTJ

# Logging
# Set logging to the system log. 3 is the noisiest, 0 is disabled.
loglevel        3

# The escape character
# 20 = (Control-T)
EscapeChar      20
```

13.2 Creating the `/etc/ax25/node.perms` file.

The *node* allows you to assign permissions to users. These permissions allow you to determine which users should be allowed to make use of options such as the (T)elnet, and (C)onnect commands, for example, and which shouldn't. The `node.perms` file is where this information is stored and contains five key fields. For all fields an asterisk `*' character matches anything. This is useful for building default rules.

user

The first field is the callsign or user to which the permissions should apply. Any SSID value is ignored, so you should just place the base callsign here.

method

Each protocol or access method is also given permissions. For example you might allow users who have connected via AX.25 or NetRom to use the (C)onnect option, but prevent others, such as those who are telnet connected from a non-local node from having access to it. The second field therefore allows you to select

which access method this permissions rule should apply to. The access methods allowed are:

```
method  description
------  ------------------------------------------------------------
ampr    User is telnet connected from an amprnet address (44.0.0.0)
ax25    User connected by AX.25
host    User started node from command line
inet    user is telnet connected from a non-loca, non-ampr address.
local   User is telnet connected from a 'local' host
netrom  User connected by NetRom
rose    User connected by Rose
*       User connected by any means.
```

port

For AX.25 users you can control permissions on a port by port basis too if you choose. This allows you to determine what AX.25 are allowed to do based on which of your ports they have connected to. The third field contains the port name if you are using this facility. This is useful only for AX.25 connections.

password

You may optionally configure the node so that it prompts users to enter a password when they connect. This might be useful to help protect specially configured users who have high authority levels. If the fourth field is set then its value will be the password that will be accepted.

permissions

The permissions field is the final field in each entry in the file. The permissions field is coded as a bit field, with each facility having a bit value which if set allows the option to be used and if not set prevents the facility being used. The list of controllable facilities and their corresponding bit values are:

```
value   description
-----   ------------------------------------------------
 1      Login allowed.
 2      AX25 (C)onnects allowed.
 4      NetRom (C)onnects allowed.
 8      (T)elnet to local hosts allowed.
 16     (T)elnet to amprnet (44.0.0.0) hosts allowed.
 32     (T)elnet to non-local, non-amprnet hosts allowed.
 64     Hidden ports allowed for AX.25 (C)onnects.
 128    Rose (C)onnects allowed.
```

To code the permissions value for a rule, simply take each of the permissions you want that user to have and add their values together. The resulting number is what you place in field five.

A sample `nodes.perms` might look like:

```
# /etc/ax25/node.perms
#
# The node operator is VK2KTJ, has a password of 'secret' and
# is allowed all permissions by all connection methods
vk2ktj  *       *       secret  255

# The following users are banned from connecting
```

```
NOCALL   *        *        *        0
PK232    *        *        *        0
PMS      *        *        *        0

# INET users are banned from connecting.
*        inet     *        *        0

# AX.25, NetRom, Local, Host and AMPR users may (C)onnect and (T)elnet
# to local and ampr hosts but not to other IP addresses.
*        ax25     *        *        159
*        netrom   *        *        159
*        local    *        *        159
*        host     *        *        159
*        ampr     *        *        159
```

13.3 Configuring *node* to run from *ax25d*

The *node* program would normally be run by the *ax25d* program. To do this you need to add appropriate rules to the `/etc/ax25/ax25d.conf` file. In my configuration I wanted users to have a choice of either connecting to the *node* or connecting to other services. *ax25d* allows you to do this by cleverly creating creating port aliases. For example, given the *ax25d* configuration presented above, I want to configure *node* so that all users who connect to `VK2KTJ-1` are given the node. To do this I add the following to my `/etc/ax25/ax25d.conf` file:

```
[vk2ktj-1 via radio]
default    *     *    *    *    *    0    root /usr/sbin/node node
```

This says that the Linux kernel code will answer any connection requests for the callsign `VK2KTJ-1' heard on the AX.25 port named `radio', and will cause the *node* program to be run.

13.4 Configuring *node* to run from *inetd*

If you want users to be able to telnet a port on your machine and obtain access to the *node* you can go this fairly easily. The first thing to decide is what port users should connect to. In this example I've arbitrarily chosen port 4000, though Tomi gives details on how you could replace the normal telnet daemon with the *node* in his documentation.

You need to modify two files.

To `/etc/services` you should add:

```
node     3694/tcp         #OH2BNS's node software
```

and to `/etc/inetd.conf` you should add:

```
node     stream  tcp      nowait  root     /usr/sbin/node node
```

When this is done, and you have restarted the *inetd* program any user who telnet connects to port 3694 of your machine will be prompted to login and if configured, their password and then they will be connected to the *node*.

14. Configuring *axspawn*.

The *axspawn* program is a simple program that allows AX.25 stations who connect to be logged in to your machine. It may be invoked from the *ax25d* program as described above in a manner similar to the *node* program. To allow a user to log in to your machine you should add a line similar to the following into your `/etc/ax25/ax25d.conf` file:

```
default * * * * * 1 root /usr/sbin/axspawn axspawn %u
```

If the line ends in the `+` character then the connecting user must hit return before they will be allowed to login. The default is to not wait. Any individual host configurations that follow this line will have the *axspawn* program run when they connect. When *axspawn* is run it first checks that the command line argument it is supplied is a legal callsign, strips the SSID, then it checks that `/etc/passwd` file to see if that user has an account configured. If there is an account, and the password is either `""` (null) or `+` then the user is logged in, if there is anything in the password field the user is prompted to enter a password. If there is not an existing account in the `/etc/passwd` file then *axspawn* may be configured to automatically create one.

14.1 Creating the `/etc/ax25/axspawn.conf` file.

You can alter the behaviour of *axspawn* in various ways by use of the `/etc/ax25/axspawn.conf` file. This file is formatted as follows:

```
# /etc/ax25/axspawn.conf
#
# allow automatic creation of user accounts
create    yes
#
# guest user if above is 'no' or everything else fails. Disable with "no"
guest     no
#
# group id or name for autoaccount
group     ax25
#
# first user id to use
first_uid 2001
#
# maximum user id
max_uid   3000
#
# where to add the home directory for the new users
home      /home/ax25
#
# user shell
shell     /bin/bash
#
# bind user id to callsign for outgoing connects.
```

```
associate yes
```

The eight configurable characteristics of *axspawn* are as follows:

#

indicates a comment.

create

if this field is set to `yes` then *axspawn* will attempt to automatically create a user account for any user who connects and does not already have an entry in the `/etc/passwd` file.

guest

this field names the login name of the account that will be used for people who connect who do not already have accounts if *create* is set to `no`. This is usually `ax25` or `guest`.

group

this field names the group name that will be used for any users who connect and do not already have an entry in the `/etc/passwd` file.

first_uid

this is the number of the first userid that will be automatically created for new users.

max_uid

this is the maximum number that will be used for the userid of new users.

home

this is the home (login) directory of new users.

shell

this is the login shell of any new users.

associate

this flag indicates whether outgoing AX.25 connections made by this user after they login will use their own callsign, or your stations callsign.

---

15. Configuring the *pms*

The *pms* program is an implementation of a simple personal message system. It was originally written by Alan Cox. Dave Brown, N2RJT, `<dcb@vectorbd.com>` has taken on further development of it. At present it is still very simple, supporting only the ability to send mail to the owner of the system and to obtain some limited system information but Dave is working to expand its capability to make it more useful.

After that is done there are a couple of simple files that you should create that give users some information about the system and then you need to add appropriate entries into the `ax25d.conf` file so that connected users are presented with the PMS.

15.1 Create the `/etc/ax25/pms.motd` file.

The `/etc/ax25/pms.motd` file contains the `message of the day' that users will be presented with after they connect and receive the usual BBS id header. The file is a simple text file, any text you include in this file will be sent to users.

15.2 Create the `/etc/ax25/pms.info` file.

The `/etc/ax25/pms.info` file is also a simple text file in which you would put more detailed information about your station or configuration. This file is presented to users in response to their issuing of the `Info` command from the `PMS>` prompt.

15.3 Associate AX.25 callsigns with system users.

When a connected user sends mail to an AX.25 callsign, the *pms* expects that callsign to be mapped, or associated with a real system user on your machine. This is described in a section of its own.

15.4 Add the PMS to the `/etc/ax25/ax25d.conf` file.

Adding the *pms* to your `ax25d.conf` file is very simple. There is one small thing you need to think about though. Dave has added command line arguments to the PMS to allow it to handle a number of different text end-of-line conventions. AX.25 and NetRom by convention expect the end-of-line to be *carriage return, linefeed* while the standard unix end-of-line is just *newline*. So, for example, if you wanted to add an entry that meant that the default action for a connection received on an AX.25 port is to start the PMS then you would add a line that looked something like:

```
default  1  10 5 100 5   0    root  /usr/sbin/pms pms -a -o vk2
```

This simply runs the *pms* program, telling it that it is an AX.25 connection it is connected to and that the PMS owner is `vk2ktj`. Check the *man* page for what you should specify for other connection methods.

15.5 Test the PMS.

To test the PMS, you can try the following command from the command line:

```
# /usr/sbin/pms -u vk2ktj -o vk2ktj
```

Substitute your own callsign for mine and this will run the pms, telling it that it is to use the unix end-of-line convention, and that user logging in is `vk2ktj`. You can do all the things connected users can.

Additionally you might try getting some other node to connect to you to confirm that your `ax25d.conf` configuration works.

---

16. Configuring the *user_call* programs.

The `*user_call*' programs are really called: *ax25_call* and *netrom_call*. They are very simple programs designed to be called from *ax25d* to automate network connections to remote hosts. They may of course be called from a number of other places such as shell scripts or other daemons such as the *node* program.

They are like a very simple *call* program. They don't do any meddling with the data at all, so the end of line handling you'll have to worry about yourself.

Let's start with an example of how you might use them. Imagine you have a small network at home and that you have one linux machine acting as your Linux radio gateway and another machine, lets say a BPQ node connected to it via an ethernet connection.

Normally if you wanted radio users to be able to connect to the BPQ node they would either have to digipeat through your linux node, or connect to the node program on your linux node and then connect from it. The *ax25_call* program can simplify this if it is called from the *ax25d* program.

Imagine the BPQ node has the callsign `VK2KTJ-9` and that the linux machine has the AX.25/ethernet port named `bpq'. Let us also imagine the Linux gateway machine has a radio port called `radio'.

An entry in the `/etc/ax25/ax25d.conf` that looked like:

```
[VK2KTJ-1 via radio]
default     * * * *   *   *   *
                root /usr/sbin/ax25_call ax25_call bpq %u vk2ktj-9
```

would enable users to connect direct to `VK2KTJ-1' which would actually be the Linux *ax25d* daemon and then be automatically switched to an AX.25 connection to `VK2KTJ-9' via the `bpq' interface.

There are all sorts of other possible configurations that you might try. The `*netrom_call*' and `*rose_call*' utilities work in similar ways. One amateur has used this utility to make connections to a remote BBS easier. Normally the users would have to manually enter a long connection string to make the call so he created an entry that made the BBS appear as though it were on the local network by having his *ax25d* proxy the connection to the remote machine.

17. Configuring the Rose Uplink and Downlink commands

If you are familiar with the ROM based Rose implementation you will be familiar with the method by which AX.25 users make calls across a Rose network. If a users local Rose node has the callsign `VK2KTJ-5` and the AX.25 user wants to connect to `VK5XXX` at remote Rose node `5050882960` then they would issue the command:

```
c vk5xxx v vk2ktj-5 5050 882960
```

At the remote node, `VK5XXX` would see an incoming connection with the local AX.25 users callsign and being digipeated via the remote Rose nodes callsign.

The Linux Rose implementation does not support this capability in the kernel, but there are two application programs called *rsuplnk* and *rsdwnlnk* which perform this function.

17.1 Configuring a Rose downlink

To configure your Linux machine to accept a Rose connection and establish an AX.25 connection to any destination callsign that is not being listened for on your machine you need to add an entry to your `/etc/ax25/ax25d.conf` file. Normally you would configure this entry to be the default behaviour for incoming Rose connections. For example you might have Rose listeners operating for destinations like `NODE-0` or `HEARD-0` that you wish to handle locally, but for all other destination calls you may want to pass them to the *rsdwnlink* command and assume they are AX.25 users.

A typical configuration would look like:

```
#
{* via rose}
NOCALL   * * * * * *  L
default  * * * * * *  - root  /usr/sbin/rsdwnlnk rsdwnlnk 4800 vk2ktj-5
#
```

With this configuration any user who established a Rose connection to your Linux nodes address with a destination call of something that you were not specifically listening for would be converted into an AX.25 connection on the AX.25 port named `4800` with a digipeater path of `VK2KTJ-5`.

17.2 Configuring a Rose uplink

To configure your Linux machine to accept AX.25 connections in the same way that a ROM Rose node would you must add an entry into your `/etc/ax25/ax25d.conf` file that looks similar to the following:

```
#
[VK2KTJ-5* via 4800]
```

```
NOCALL   * * * * * *  L
default  * * * * * *  - root  /usr/sbin/rsuplnk rsuplnk rose
#
```

Note the special syntax for the local callsign. The `*' character indicates that the application should be invoked if the callsign is heard in the digipeater path of a connection.

This configuration would allow an AX.25 user to establish Rose calls using the example connect sequence presented in the introduction. Anybody attempting to digipeat via `VK2KTJ-5` on the AX.25 port named `4800` would be handled by the *rsuplnk* command.

---

18. Associating AX.25 callsigns with Linux users.

There are a number of situations where it is highly desirable to associate a callsign with a linux user account. One example might be where a number of amateur radio operators share the same linux machine and wish to use their own callsign when making calls. Another is the case of PMS users wanting to talk to a particular user on your machine.

The AX.25 software provides a means of managing this association of linux user account names with callsigns. We've mentioned it once already in the PMS section, but I'm spelling it out here to be sure you don't miss it.

You make the association with the *axparms* command. An example looks like:

```
# axparms -assoc vk2ktj terry
```

This command associates that AX.25 callsign `vk2ktj` with the user `terry` on the machine. So, for example, any mail for `vk2ktj` on the *pms* will be sent to Linux account `terry`.

Remember to put these associations into your *rc* file so that they are available each time your reboot.

Note you should never associate a callsign with the `root` account as this can cause configuration problems in other programs.

19. The `/proc/` file system entries.

The `/proc` filesystem contains a number of files specifically related to the AX25 and NetRom kernel software. These files are normally used by the AX52 utilities, but they are plainly formatted so you may be interested in reading them. The format is fairly easily understood so I don't think much explanation will be necessary.

/proc/net/arp

contains the list of Address Resolution Protocol mappings of IP addresses to MAC layer protocol addresses. These can can AX.25, ethernet or some other MAC layer protocol.

/proc/net/ax25

contains a list of AX.25 sockets opened. These might be listening for a connection, or active sessions.

/proc/net/ax25_bpqether

contains the AX25 over ethernet BPQ style callsign mappings.

/proc/net/ax25_calls

contains the linux userid to callsign mappings set my the *axparms -assoc* command.

/proc/net/ax25_route

contains AX.25 digipeater path information.

/proc/net/nr

contains a list of NetRom sockets opened. These might be listening for a connection, or active sessions.

/proc/net/nr_neigh

contains information about the NetRom neighbours known to the NetRom software.

/proc/net/nr_nodes

contains information about the NetRom nodes known to the NetRom software.

/proc/net/rose

contains a list of Rose sockets opened. These might be listening for a connection, or active sessions.

/proc/net/rose_nodes

contains a mapping of Rose destinations to Rose neighbours.

/proc/net/rose_neigh

contains a list of known Rose neighbours.

/proc/net/rose_routes

contains a list of all established Rose connections.

20. AX.25, NetRom, Rose network programming.

Probably the biggest advantage of using the kernel based implementations of the amateur packet radio protocols is the ease with which you can develop applications and programs to use them.

While the subject of Unix Network Programming is outside the scope of this document I will describe the elementary details of how you can make use of the AX.25, NetRom and Rose protocols within your software.

20.1 The address families.

Network programming for AX.25, NetRom and Rose is quite similar to programming for TCP/IP under Linux. The major differences being the address families used, and the address structures that need to be mangled into place.

The address family names for AX.25, NetRom and Rose are `AF_AX25`, `AF_NETROM` and `AF_ROSE` respectively.

20.2 The header files.

You must always include the `ax25.h' header file, and also the `netrom.h' or `rose.h' header files if you are dealing with those protocols. Simple top level skeletons would look something like the following:

For AX.25:

```
#include <ax25.h>
int s, addrlen = sizeof(struct full_sockaddr_ax25);
struct full_sockaddr_ax25 sockaddr;
sockaddr.fsa_ax25.sax25_family = AF_AX25
```

For NetRom:

```
#include <ax25.h>
#include <netrom.h>
int s, addrlen = sizeof(struct full_sockaddr_ax25);
struct full_sockaddr_ax25 sockaddr;
sockaddr.fsa_ax25.sax25_family = AF_NETROM;
```

For Rose:

```
#include <ax25.h>
```

```
#include <rose.h>
int s, addrlen = sizeof(struct sockaddr_rose);
struct sockaddr_rose sockaddr;
sockaddr.srose_family = AF_ROSE;
```

20.3 Callsign mangling and examples.

There are routines within the `lib/ax25.a` library built in the AX25 utilities package that manage the callsign conversions for you. You can write your own of course if you wish.

The *user_call* utilities are excellent examples from which to work. The source code for them is included in the AX25 utilities package. If you spend a little time working with those you will soon see that ninety percent of the work is involved in just getting ready to open the socket. Actually making the connection is easy, the preparation takes time.

The example are simple enough to not be very confusing. If you have any questions, you should feel to direct them to the `linux-hams` mailing list and someone there will be sure to help you.

---

21. Some sample configurations.

Following are examples of the most common types of configurations. These are guides only as there are as many ways of configuring your network as there are networks to configure, but they may give you a start.

21.1 Small Ethernet LAN with Linux as a router to Radio LAN

Many of you may have small local area networks at home and want to connect the machines on that network to your local radio LAN. This is the type of configuration I use at home. I arranged to have a suitable block of addresses allocated to me that I could capture in a single route for convenience and I use these on my Ethernet LAN. Your local IP coordinator will assist you in doing this if you want to try it as well. The addresses for the Ethernet LAN form a subset of the radio LAN addresses. The following configuration is the actual one for my linux router on my network at home:

```
                                          .      .   .    .    . .
  ---                                 .
   | Network       /---------\     .    Network
   | 44.136.8.96/29|         |    .     44.136.8/24        \ | /
   |               | Linux   |   .                          \|/
   |               |         |  .                            |
   |          eth0 | Router  |  .  /-----\    /----------\   |
   |---------------|         |-----| TNC |----| Radio    |---/
   |   44.136.8.97 |  and    |  .  \-----/    \----------/
   |               |         | sl0
   |               | Server  | 44.136.8.5
   |               |         |    .
   |               |         |     .
   |               \_________/       .
  ---                                     .      .   .    .    . .
```

```
#!/bin/sh
# /etc/rc.net
# This configuration provides one KISS based AX.25 port and one
# Ethernet device.

echo "/etc/rc.net"
echo "  Configuring:"

echo -n "    loopback:"
/sbin/ifconfig lo 127.0.0.1
/sbin/route add 127.0.0.1
echo " done."

echo -n "    ethernet:"
/sbin/ifconfig eth0 44.136.8.97 netmask 255.255.255.248 \
                broadcast 44.136.8.103 up
/sbin/route add 44.136.8.97 eth0
/sbin/route add -net 44.136.8.96 netmask 255.255.255.248 eth0
echo " done."
```

TOC

```
    echo -n "    AX.25: "
    kissattach -i 44.136.8.5 -m 512 /dev/ttyS1 4800
    ifconfig sl0 netmask 255.255.255.0 broadcast 44.136.8.255
    route add -host 44.136.8.5 sl0
    route add -net 44.136.8.0 window 1024 sl0

    echo -n "    Netrom: "
    nrattach -i 44.136.8.5 netrom

    echo "  Routing:"
    /sbin/route add default gw 44.136.8.68 window 1024 sl0
    echo "    default route."
    echo done.

    # end
```

/etc/ax25/axports

```
    # name  callsign        speed   paclen  window  description
    4800    VK2KTJ-0        4800    256     2       144.800 MHz
```

/etc/ax25/nrports

```
    # name  callsign        alias   paclen  description
    netrom  VK2KTJ-9        LINUX   235     Linux Switch Port
```

/etc/ax25/nrbroadcast

```
    # ax25_name     min_obs def_qual        worst_qual      verbose
    4800            1       120             10              1
```

- You must have IP_FORWARDING enabled in your kernel.
- The AX.25 configuration files are pretty much those used as examples in the earlier sections, refer to those where necessary.
- I've chosen to use an IP address for my radio port that is not within my home network block. I needn't have done so, I could have easily used `44.136.8.97` for that port too.
- `44.136.8.68` is my local IPIP encapsulated gateway and hence is where I point my default route.
- Each of the machines on my Ethernet network have a route:

  ```
  route add -net 44.0.0.0 netmask 255.0.0.0 \
          gw 44.136.8.97 window 512 mss 512 eth0
  ```

  The use of the *mss* and *window* parameters means that I can get optimum performance from both local Ethernet and radio based connections.
- I also run my smail, http, ftp and other daemons on the router machine so that it needs to be the only machine to provide others with facilities.
- The router machine is a lowly 386DX20 with a 20Mb harddrive and a very minimal linux configuration.

21.2 IPIP encapsulated gateway configuration.

TOC

Linux is now very commonly used for TCP/IP encapsulated gateways around the world. The new tunnel driver supports multiple encapsulated routes and makes the older *ipip* daemon obsolete.

A typical configuration would look similar to the following.

```
                                                .       .   .    .    . .
  ---                                   .
   | Network       /---------\     .    Network
   | 154.27.3/24   |         |    .     44.136.16/24       \ | /
   |               | Linux   |   .                          \|/
   |               |         |  .                            |
   |          eth0 | IPIP    |  .  /-----\    /----------\   |
---|---------------|         |-----| TNC |----| Radio    |---/
   |   154.27.3.20 | Gateway |  .  \-----/    \----------/
   |               |         | sl0
   |               |         | 44.136.16.1
   |               |         |    .
   |               |         |     .
   |               \_________/       .
  ---                                           .       .   .    .    . .
```

The configuration files of interest are:

```
# /etc/rc.net
# This file is a simple configuration that provides one KISS AX.25
# radio port, one Ethernet device, and utilises the kernel tunnel driver
# to perform the IPIP encapsulation/decapsulation
#
echo "/etc/rc.net"
echo "  Configuring:"
#
echo -n "    loopback:"
/sbin/ifconfig lo 127.0.0.1
/sbin/route add 127.0.0.1
echo " done."
#
echo -n "    ethernet:"
/sbin/ifconfig eth0 154.27.3.20 netmask 255.255.255.0 \
                broadcast 154.27.3.255 up
/sbin/route add 154.27.3.20 eth0
/sbin/route add -net 154.27.3.0 netmask 255.255.255.0 eth0
echo " done."
#
echo -n "    AX.25: "
kissattach -i 44.136.16.1 -m 512 /dev/ttyS1 4800
/sbin/ifconfig sl0 netmask 255.255.255.0 broadcast 44.136.16.255
/sbin/route add -host 44.136.16.1 sl0
/sbin/route add -net 44.136.16.0 netmask 255.255.255.0 window 1024 sl0
#
echo -n "    tunnel:"
/sbin/ifconfig tunl0 44.136.16.1 mtu 512 up
```

```
#
echo done.
#
echo -n "Routing ... "
source /etc/ipip.routes
echo done.
#
# end.
```

and:

```
# /etc/ipip.routes
# This file is generated using the munge script
#
/sbin/route add -net 44.134.8.0 netmask 255.255.255.0 tunl0 gw 134.43.26.1
/sbin/route add -net 44.34.9.0 netmask 255.255.255.0 tunl0 gw 174.84.6.17
/sbin/route add -net 44.13.28.0 netmask 255.255.255.0 tunl0 gw 212.37.126.3
   ...
   ...
   ...
```

`/etc/ax25/axports`

```
# name  callsign        speed   paclen  window  description
4800    VK2KTJ-0        4800    256     2       144.800 MHz
```

Some points to note here are:

- The new tunnel driver uses the *gw* field in the routing table in place of the *pointopoint* parameter to specify the address of the remote IPIP gateway. This is why it now supports multiple routes per interface.
- You **can** configure two network devices with the same address. In this example both the `sl0` and the `tunl0` devices have been configured with the IP address of the radio port. This is done so that the remote gateway sees the correct address from your gateway in encapsulated datagrams sent to it.
- The route commands used to specify the encapsulated routes can be automatically generated by a modified version of the *munge* script. This is included below. The route commands would then be written to a separate file and read in using the *bash* `source /etc/ipip.routes` command (assuming you called the file with the routing commands `/etc/ipip.routes`) as illustrated. The source file must be in the NOS route command format.
- Note the use of the *window* argument on the *route* command. Setting this parameter to an appropriate value improves the performance of your radio link.

The new tunnel-munge script:

```
#!/bin/sh
#
# From: Ron Atkinson <n8fow@hamgate.cc.wayne.edu>
#
#  This script is basically the 'munge' script written by Bdale N3EUA
#  for the IPIP daemon and is modified by Ron Atkinson N8FOW. It's
#  purpose is to convert a KA9Q NOS format gateways route file
#  (usually called 'encap.txt') into a Linux routing table format
#  for the IP tunnel driver.
#
#       Usage: Gateway file on stdin, Linux route format file on stdout.
#              eg.  tunnel-munge < encap.txt > ampr-routes
```

TOC

```
#
# NOTE: Before you use this script be sure to check or change the
#       following items:
#
#     1) Change the 'Local routes' and 'Misc user routes' sections
#        to routes that apply to your own area (remove mine please!)
#     2) On the fgrep line be sure to change the IP address to YOUR
#        gateway Internet address. Failure to do so will cause serious
#        routing loops.
#     3) The default interface name is 'tunl0'. Make sure this is
#        correct for your system.

echo "#"
echo "# IP tunnel route table built by $LOGNAME on `date`"
echo "# by tunnel-munge script v960307."
echo "#"
echo "# Local routes"
echo "route add -net 44.xxx.xxx.xxx netmask 255.mmm.mmm.mmm dev sl0"
echo "#"
echo "# Misc user routes"
echo "#"
echo "# remote routes"

fgrep encap | grep "^route" | grep -v " XXX.XXX.XXX.XXX" | \
awk '{
        split($3, s, "/")
        split(s[1], n,".")
        if      (n[1] == "")    n[1]="0"
        if      (n[2] == "")    n[2]="0"
        if      (n[3] == "")    n[3]="0"
        if      (n[4] == "")    n[4]="0"
        if      (s[2] == "1")   mask="128.0.0.0"
        else if (s[2] == "2")   mask="192.0.0.0"
        else if (s[2] == "3")   mask="224.0.0.0"
        else if (s[2] == "4")   mask="240.0.0.0"
        else if (s[2] == "5")   mask="248.0.0.0"
        else if (s[2] == "6")   mask="252.0.0.0"
        else if (s[2] == "7")   mask="254.0.0.0"
        else if (s[2] == "8")   mask="255.0.0.0"
        else if (s[2] == "9")   mask="255.128.0.0"
        else if (s[2] == "10")  mask="255.192.0.0"
        else if (s[2] == "11")  mask="255.224.0.0"
        else if (s[2] == "12")  mask="255.240.0.0"
        else if (s[2] == "13")  mask="255.248.0.0"
        else if (s[2] == "14")  mask="255.252.0.0"
        else if (s[2] == "15")  mask="255.254.0.0"
        else if (s[2] == "16")  mask="255.255.0.0"
        else if (s[2] == "17")  mask="255.255.128.0"
        else if (s[2] == "18")  mask="255.255.192.0"
        else if (s[2] == "19")  mask="255.255.224.0"
        else if (s[2] == "20")  mask="255.255.240.0"
        else if (s[2] == "21")  mask="255.255.248.0"
        else if (s[2] == "22")  mask="255.255.252.0"
```

```
        else if (s[2] == "23")  mask="255.255.254.0"
        else if (s[2] == "24")  mask="255.255.255.0"
        else if (s[2] == "25")  mask="255.255.255.128"
        else if (s[2] == "26")  mask="255.255.255.192"
        else if (s[2] == "27")  mask="255.255.255.224"
        else if (s[2] == "28")  mask="255.255.255.240"
        else if (s[2] == "29")  mask="255.255.255.248"
        else if (s[2] == "30")  mask="255.255.255.252"
        else if (s[2] == "31")  mask="255.255.255.254"
        else                    mask="255.255.255.255"

if (mask == "255.255.255.255")
        printf "route add -host %s.%s.%s.%s gw %s dev tunl0\n"\
                ,n[1],n[2],n[3],n[4],$5
else
        printf "route add -net %s.%s.%s.%s gw %s netmask %s dev tunl0\n"\
                ,n[1],n[2],n[3],n[4],$5,mask
 }'

echo "#"
echo "# default the rest of amprnet via mirrorshades.ucsd.edu"
echo "route add -net 44.0.0.0 gw 128.54.16.18 netmask 255.0.0.0 dev tunl0"
echo "#"
echo "# the end"
```

21.3 AXIP encapsulated gateway configuration

Many Amateur Radio Internet gateways encapsulate AX.25, NetRom and Rose in addition to tcp/ip. Encapsulation of AX.25 frames within IP datagrams is described in RFC-1226 by Brian Kantor. Mike Westerhof wrote an implementation of an AX.25 encapsulation daemon for unix in 1991. The ax25-utils package includes a marginally enhanced version of it for Linux.

An AXIP encapsulation program accepts AX.25 frames at one end, looks at the destination AX.25 address to determine what IP address to send them to, encapsulates them in a tcp/ip datagram and then transmits them to the appropriate remote destination. It also accepts tcp/ip datagrams that contain AX.25 frames, unwraps them and processes them as if it had received them directly from an AX.25 port. To distinguish IP datagrams containing AX.25 frames from other IP datagrams which don't, AXIP datagrams are coded with a protocol id of 4 (or 94 which is now deprecated). This process is described in RFC-1226.

The *ax25ipd* program included in the ax25-utils package presents itself as a program supporting a KISS interface across which you pass AX.25 frames, and an interface into the tcp/ip protocols. It is configured with a single configuration file called `/etc/ax25/ax25ipd.conf`.

AXIP configuration options.

The *ax25ipd* program has two major modes of operation. "digipeater" mode and "tnc" mode. In "tnc" mode the daemon is treated as though it were a kiss TNC, you pass KISS encapsulated frames to it and it will transmit them, this is the usual configuration. In "digipeater" mode, you treat the daemon as though it were an AX.25 digipeater. There are subtle differences between these modes.

In the configuration file you configure "routes" or mappings between destination AX.25 callsigns and the IP addresses of the hosts that you want to send the AX.25 packets too. Each route has options which will be explained later.

Other options that are configured here are

TOC

- the tty that the *ax25ipd* daemon will open and its speed (usually one end of a pipe)
- what callsign you want to use in "digipeater" mode
- beacon interval and text
- whether you want to encapsulate the AX.25 frames in IP datagrams or in UDP/IP datagrams. Nearly all AXIP gateways use IP encapsulation, but some gateways are behind firewalls that will not allow IP with the AXIP protocol id to pass and are forced to use UDP/IP. Whatever you choose must match what the tcp/ip host at the other end of the link is using.

A typical `/etc/ax25/ax25ipd.conf` file.

```
#
# ax25ipd configuration file for station floyd.vk5xxx.ampr.org
#
# Select axip transport. 'ip' is what you want for compatibility
# with most other gateways.
#
socket ip
#
# Set ax25ipd mode of operation. (digi or tnc)
#
mode tnc
#
# If you selected digi, you must define a callsign.  If you selected
# tnc mode, the callsign is currently optional, but this may change
# in the future! (2 calls if using dual port kiss)
#
#mycall vk5xxx-4
#mycall2 vk5xxx-5
#
# In digi mode, you may use an alias. (2 for dual port)
#
#myalias svwdns
#myalias2 svwdn2
#
# Send an ident every 540 seconds ...
#
#beacon after 540
#btext ax25ip -- tncmode rob/vk5xxx -- Experimental AXIP gateway
#
# Serial port, or pipe connected to a kissattach in my case
#
device /dev/ttyq0
#
# Set the device speed
#
speed 9600
#
# loglevel 0 - no output
# loglevel 1 - config info only
# loglevel 2 - major events and errors
# loglevel 3 - major events, errors, and AX25 frame trace
# loglevel 4 - all events
```

TOC

```
# log 0 for the moment, syslog not working yet ...
#
loglevel 2
#
# If we are in digi mode, we might have a real tnc here, so use param to
# set the tnc parameters ...
#
#param 1 20
#
# Broadcast Address definition. Any of the addresses listed will be forwarded
# to any of the routes flagged as broadcast capable routes.
#
broadcast QST-0 NODES-0
#
# ax.25 route definition, define as many as you need.
# format is route (call/wildcard) (ip host at destination)
# ssid of 0 routes all ssid's
#
# route <destcall> <destaddr> [flags]
#
# Valid flags are:
#         b  - allow broadcasts to be transmitted via this route
#         d  - this route is the default route
#
route vk2sut-0 44.136.8.68 b
route vk5xxx 44.136.188.221 b
route vk2abc 44.1.1.1
#
#
```

Running *ax25ipd*

Create your `/etc/ax25/axports` entry:

```
# /etc/ax25/axports
#
axip    VK2KTJ-13       9600    256     AXIP port
#
```

Run the *kissattach* command to create that port:

```
/usr/sbin/kissattach /dev/ptyq0 axip
```

Run the *ax25ipd* program:

```
/usr/sbin/ax25ipd &
```

Test the AXIP link:

```
call axip vk5xxx
```

Some notes about the routes and route flags

The "`route`" command is where you specify where you want your AX.25 packets encapsulated and sent to. When the *ax25ipd* daemon receives a packet from its interface, it compares the destination callsign with each of the callsigns in its routing table. If if finds a match then the ax.25 packet is encapsulated in an IP datagram and then transmitted to the host at the specified IP address.

There are two flags you can add to any of the route commands in the `ax25ipd.conf` file. The two flags are:

b

traffic with a destination address matching any of those on the list defined by the "`broadcast`" keyword should be transmitted via this route.

d

any packets not matching any route should be transmitted via this route.

The broadcast flag is very useful, as it enables informations that is normally destined for all stations to a number of AXIP destinations. Normally axip routes are point-to-point and unable to handle 'broadcast' packets.

21.4 Linking NOS and Linux using a pipe device

Many people like to run some version of NOS under Linux because it has all of the features and facilities they are used to. Most of those people would also like to have the NOS running on their machine capable of talking to the Linux kernel so that they can offer some of the linux capabilities to radio users via NOS.

Brandon S. Allbery, KF8NH, contributed the following information to explain how to interconnect the NOS running on a Linux machine with the kernel code using the Linux pipe device.

Since both Linux and NOS support the slip protocol it is possible to link the two together by creating a slip link. You could do this by using two serial ports with a loopback cable between them, but this would be slow and costly. Linux provides a feature that many other Unix-like operating systems provide called `pipes'. These are special pseudo devices that look like a standard tty device to software but in fact loopback to another pipe device. To use these pipes the first program must open the **master** end of the pipe, and the open then the second program can open the **slave** end of the pipe. When both ends are open the programs can communicate with each other simply by writing characters to the pipes in the way they would if they were terminal devices.

To use this feature to connect the Linux Kernel and a copy of NOS, or some other program you first must choose a pipe device to use. You can find one by looking in your `/dev` directory. The master end of the pipes are named: `ptyq[1-f]` and the slave end of the pipes are known as: `ttyq[1-f]`. Remember they come in pairs, so if you select `/dev/ptyqf` as your master end then you must use `/dev/ttyqf` as the slave end.

Once you have chosen a pipe device pair to use you should allocate the master end to you linux kernel and the slave end to the NOS program, as the Linux kernel starts first and the master end of the pipe must be opened first. You must also remember that your Linux kernel must have a different IP address to your NOS, so you will need to allocate a unique address for it if you haven't already.

You configure the pipe just as if it were a serial device, so to create the slip link from your linux kernel you can use commands similar to the following:

```
# /sbin/slattach -s 38400 -p slip /dev/ptyqf &
# /sbin/ifconfig sl0 broadcast 44.255.255.255 pointopoint 44.70.248.67 /
        mtu 1536 44.70.4.88
# /sbin/route add 44.70.248.67 sl0
# /sbin/route add -net 44.0.0.0 netmask 255.0.0.0 gw 44.70.248.67
```

TOC

In this example the Linux kernel has been given IP address `44.70.4.88` and the NOS program is using IP address `44.70.248.67`. The *route* command in the last line simply tells your linux kernel to route all datagrams for the air via the slip link created by the *slattach* command. Normally you would put these commands into your `/etc/rc.d/rc.inet2` file after all your other network configuration is complete so that the slip link is created automatically when you reboot. Note: there is no advantage in using *cslip* instead of *slip* as it actually reduces performance because the link is only a virtual one and occurs fast enough that having to compress the headers first takes longer than transmitting the uncompressed datagram.

To configure the NOS end of the link you could try the following:

```
# you can call the interface anything you want; I use "linux" for convenience.
attach asy ttyqf - slip linux 1024 1024 38400
route addprivate 44.70.4.88 linux
```

These commands will create a slip port named `linux' via the slave end of the pipe device pair to your linux kernel, and a route to it to make it work. When you have started NOS you should be able to ping and telnet to your NOS from your Linux machine and vice versa. If not, double check that you have made no mistakes especially that you have the addresses configured properly and have the pipe devices around the right way.

---

22. Where do I find more information about .... ?

Since this document assumes you already have some experience with packet radio and that this might not be the case I've collected a set of references to other information that you might find useful.

22.1 Packet Radio

You can get general information about Packet Radio from these sites:

- American Radio Relay League,
- Radio Amateur Teleprinter Society
- Tucson Amateur Packet Radio Group

22.2 Protocol Documentation

- AX.25, NetRom - Jonathon Naylor has collated a variety of documents that relate to the packet radio protocols themselves. This documentation has been packaged up into ax25-doc-1.0.tar.gz

22.3 Hardware Documentation

- Information on the **PI2 Card** is provided by the Ottawa Packet Radio Group.
- Information on **Baycom hardware** is available at the Baycom Web Page.

23. Discussion relating to Amateur Radio and Linux.

There are various places that discussion relating to Amateur Radio and Linux take place. They take place in the `comp.os.linux.*` newsgroups, they also take place on the `HAMS` list on `vger.rutgers.edu`. Other places where they are held include the `tcp-group` mailing list at `ucsd.edu` (the home of amateur radio TCP/IP discussions), and you might also try the `#linpeople` channel on the `linuxnet` irc network.

To join the Linux **linux-hams** channel on the mail list server, send mail to:

```
Majordomo@vger.rutgers.edu
```

with the line:

```
subscribe linux-hams
```

in the message body. The subject line is ignored.

The **linux-hams** mailing list is archived at:

zone.pspt.fi and zone.oh7rba.ampr.org. Please use the archives when you are first starting, because many common questions are answered there.

To join the `tcp-group` send mail to:

```
listserver@ucsd.edu
```

with the line:

```
subscribe tcp-group
```

in the body of the text.

Note: Please remember that the `tcp-group` is primarily for discussion of the use of advanced protocols, of which TCP/IP is one, in Amateur Radio. *Linux specific questions should not ordinarily go there.*

24. Acknowledgements.

The following people have contributed to this document in one way or another, knowingly or unknowingly. In no particular order (as I find them): Jonathon Naylor, Thomas Sailer, Joerg Reuter, Ron Atkinson, Alan Cox, Craig Small, John Tanner, Brandon Allbery, Hans Alblas, Klaus Kudielka, Carl Makin.

25. Copyright.

The AX25-HOWTO, information on how to install and configure some of the more important packages providing AX25 support for Linux. Copyright (c) 1996 Terry Dawson.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the:

Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Linux Ethernet-Howto

by Paul Gortmaker

v2.7, 5 May 1999

This is the Ethernet-Howto, which is a compilation of information about which ethernet devices can be used for Linux, and how to set them up. Note that this Howto is focused on the hardware and low level driver aspect of the ethernet cards, and does not cover the software end of things like `ifconfig` *and* `route`*. See the Network Howto for that stuff.*

1. Introduction

- 1.1 New Versions of this Document
- 1.2 Using the Ethernet-Howto
- 1.3 HELP - It doesn't work!

2. What card should I buy for Linux?

- 2.1 So What Drivers are Stable?
- 2.2 Eight bit vs 16 bit Cards
- 2.3 32 Bit (VLB/EISA/PCI) Ethernet Cards
- 2.4 Available 100Mbs Cards and Drivers
- 2.5 100VG versus 100BaseT
- 2.6 Type of cable that your card should support

3. Frequently Asked Questions

- 3.1 Alpha Drivers -- Getting and Using them
- 3.2 Using More than one Ethernet Card per Machine
- 3.3 The `ether=` thing didn't do anything for me. Why?
- 3.4 Problems with NE1000 / NE2000 cards (and clones)
- 3.5 Problems with SMC Ultra/EtherEZ and WD80*3 cards

- 3.6 Problems with 3Com cards
- 3.7 FAQs Not Specific to Any Card.

4. Performance Tips

- 4.1 General Concepts
- 4.2 ISA Cards and ISA Bus Speed
- 4.3 Setting the TCP Rx Window
- 4.4 Increasing NFS performance

5. Vendor/Manufacturer/Model Specific Information

- 5.1 3Com
- 5.2 Accton
- 5.3 Allied Telesyn/Telesis
- 5.4 AMD / Advanced Micro Devices
- 5.5 Ansel Communications
- 5.6 Apricot
- 5.7 Arcnet
- 5.8 AT&T
- 5.9 Boca Research
- 5.10 Cabletron
- 5.11 Cogent
- 5.12 Compaq
- 5.13 Danpex
- 5.14 D-Link
- 5.15 DFI
- 5.16 Digital / DEC
- 5.17 Farallon
- 5.18 Fujitsu
- 5.19 Hewlett Packard
- 5.20 IBM / International Business Machines
- 5.21 ICL Ethernet Cards
- 5.22 Intel Ethernet Cards

- 5.23 Kingston
- 5.24 LinkSys
- 5.25 Microdyne
- 5.26 Mylex
- 5.27 Novell Ethernet, NExxxx and associated clones.
- 5.28 Proteon
- 5.29 Pure Data
- 5.30 Racal-Interlan
- 5.31 RealTek
- 5.32 Sager
- 5.33 Schneider & Koch
- 5.34 SEEQ
- 5.35 SMC (Standard Microsystems Corp.)
- 5.36 Texas Instruments
- 5.37 Thomas Conrad
- 5.38 VIA
- 5.39 Western Digital
- 5.40 Winbond
- 5.41 Xircom
- 5.42 Zenith
- 5.43 Znyx
- 5.44 Identifying an Unknown Card
- 5.45 Drivers for Non-Ethernet Devices

6. Cables, Coax, Twisted Pair

- 6.1 Thin Ethernet (thinnet)
- 6.2 Twisted Pair
- 6.3 Thick Ethernet

7. Software Configuration and Card Diagnostics

- 7.1 Configuration Programs for Ethernet Cards
- 7.2 Diagnostic Programs for Ethernet Cards

8. Technical Information

- 8.1 Programmed I/O vs. Shared Memory vs. DMA
- 8.2 Writing a Driver
- 8.3 Driver interface to the kernel
- 8.4 Technical information from 3Com
- 8.5 Notes on AMD PCnet / LANCE Based cards
- 8.6 Multicast and Promiscuous Mode
- 8.7 The Berkeley Packet Filter (BPF)

9. Networking with a Laptop/Notebook Computer

- 9.1 Using SLIP
- 9.2 PCMCIA Support
- 9.3 ISA Ethercard in the Docking Station.
- 9.4 Pocket / parallel port adaptors.

10. Miscellaneous.

- 10.1 Passing Ethernet Arguments to the Kernel
- 10.2 Using the Ethernet Drivers as Modules
- 10.3 Related Documentation
- 10.4 Disclaimer and Copyright
- 10.5 Closing

1. Introduction

The Ethernet-Howto covers what cards you should and shouldn't buy; how to set them up, how to run more than one, and other common problems and questions. It contains detailed information on the current level of support for all of the most common ethernet cards available.

It does *not* cover the software end of things, as that is covered in the NET-3 Howto. Also note that general non-Linux specific questions about Ethernet are not (or at least they should not be) answered here. For those types of questions, see the excellent amount of information in the *comp.dcom.lans.ethernet* FAQ. You can FTP it from `rtfm.mit.edu` just like all the other newsgroup FAQs.

This present revision covers distribution kernels up to and including 2.2.7.

The Ethernet-Howto is by:

Paul Gortmaker, `p_gortmaker@yahoo.com`

The primary source of information for the initial ASCII-only version of the Ethernet-Howto was:

Donald J. Becker, `becker@cesdis.gsfc.nasa.gov`

who we should thank for writing the vast majority of ethernet card drivers that are presently available for Linux. He also is the author of the original NFS server too. Thanks Donald!

This document is Copyright (c) 1993-1999 by Paul Gortmaker. Please see the Disclaimer and Copying information at the end of this document (copyright) for information about redistribution of this document and the usual `we are not responsible for what you manage to break...' type legal stuff.

1.1 New Versions of this Document

New versions of this document can be retrieved from:

Ethernet-HOWTO

or for those wishing to use FTP and/or get non-HTML formats:

Sunsite HOWTO Archive

This is the `official' location - it can also be found on various Linux WWW/ftp mirror sites. Updates will be made as new information and/or drivers becomes available. If this copy that you are reading is more than 6 months old, then you should check to see if an updated copy is available.

This document is available in various formats (postscript, dvi, ASCII, HTML, etc.). I would recommend viewing it in HTML (via a WWW browser) or the Postscript/dvi format. Both of these contain cross-references that are not included in the plain text ASCII format.

TOC

1.2 Using the Ethernet-Howto

As this guide is getting bigger and bigger, you probably don't want to spend the rest of your afternoon reading the whole thing. And the good news is that you don't *have* to read it all. The HTML and Postscript/dvi versions have a table of contents which will really help you find what you need a lot faster.

Chances are you are reading this document beacuse you can't get things to work and you don't know what to do or check. The next section (HELP - It doesn't work!) is aimed at newcomers to linux and will point you in the right direction.

Typically the same problems and questions are asked *over and over* again by different people. Chances are your specific problem or question is one of these Frequently Asked Questions, and is answered in the FAQ portion of this document . (The FAQ section). Everybody should have a look through this section before posting for help.

If you haven't got an ethernet card, then you will want to start with deciding on a card. (What card should I buy...)

If you have already got an ethernet card, but are not sure if you can use it with Linux, then you will want to read the section which contains specific information on each manufacturer, and their cards. (Vendor Specific...)

If you are interested in some of the technical aspects of the Linux device drivers, then you can have a browse of the section with this type of information. (Technical Information)

1.3 HELP - It doesn't work!

Okay, don't panic. This will lead you through the process of getting things working, even if you have no prior background in linux or ethernet hardware.

First thing you need to do is figure out what model your card is so you can determine if Linux has a driver for that particular card. Different cards typically have different ways of being controlled by the host computer, and the linux driver (if there is one) contains this control information in a format that allows linux to use the card. If you don't have any manuals or anything of the sort that tell you anything about the card model, then you can try the section on helping with mystery cards (reference section: Identifying an Unknown Card).

Now that you know what type of card you have, read through the details of your particular card in the card specific section (reference section: Vendor Specific...) which lists in alphabetical order, card manufacturers, individual model numbers and whether it has a linux driver or not. If it lists it as `Not Supported' you can pretty much give up here. If you can't find your card in that list, then check to see if your card manual lists it as being `compatible' with another known card type. For example there are hundreds, if not thousands of different cards made to be compatible with the original Novell NE2000 design.

Assuming you have found out that a linux driver exists for your card, you now have to find it and make

use of it. Just because linux has a driver for your card does *not* mean that it is built into every kernel. (The kernel is the core operating system that is first loaded at boot, and contains drivers for various pieces of hardware, among other things.) Depending on who made the particular linux distribution you are using, there may be only a few pre-built kernels, and a whole bunch of drivers as smaller separate modules, or there may be a whole lot of kernels, covering a vast combination of built-in driver combinations.

Most linux distributions now ship with a bunch of small modules that are the various drivers. The required modules are typically loaded late in the boot process, or on-demand as a driver is needed to access a particualr device. You will need to attach this module to the kernel after it has booted up. See the information that came with your distribution on installing and using modules, along with the module section in this document. (Using the Ethernet Drivers as Modules)

If you didn't find either a pre-built kernel with your driver, or a module form of the driver, chances are you have a typically uncommon card, and you will have to build your own kernel with that driver included. Once you have linux installed, building a custom kernel is not difficult at all. You essentially answer yes or no to what you want the kernel to contain, and then tell it to build it. There is a Kernel-HowTo that will help you along.

At this point you should have somehow managed to be booting a kernel with your driver built in, or be loading it as a module. About half of the problems people have are related to not having driver loaded one way or another, so you may find things work now.

If it still doesn't work, then you need to verify that the kernel is indeed detecting the card. To do this, you need to type `dmesg | more` when logged in after the system has booted and all modules have been loaded. This will allow you to review the boot messages that the kernel scrolled up the screen during the boot process. If the card has been detected, you should see somewhere in that list a message from your card's driver that starts with `eth0`, mentions the driver name and the hardware parameters (interrupt setting, input/output port address, etc) that the card is set for. (Note: At boot, linux lists all the PCI cards installed in the system, regardless of what drivers are available - do not mistake this for the driver detection which comes later!)

If you don't see a driver indentification message like this, then the driver didn't detect your card, and that is why things aren't working. See the FAQ (The FAQ Section) for what to do if your card is not detected. If you have a NE2000 compatible, there is also some NE2000 specific tips on getting a card detected in the FAQ section as well.

If the card is detected, but the detection message reports some sort of error, like a resource conflict, then the driver probably won't have initialized properly and the card still wont be useable. Most common error messages of this sort are also listed in the FAQ section, along with a solution.

If the detection message seems okay, then double check the card resources reported by the driver against those that the card is physically set for (either by little black jumpers on the card, or by a software utility supplied by the card manufacturer.) These must match exactly. For example, if you have the card jumpered or configured to IRQ 15 and the driver reports IRQ 10 in the boot messages, things will not work. The FAQ section discusses the most common cases of drivers incorrectly detecting the configuration information of various cards.

TOC

At this point, you have managed to get you card detected with all the correct parameters, and hopefully everything is working. If not, then you either have a software configuration error, or a hardware configuration error. A software configuration error is not setting up the right network addresses for the `ifconfig` and `route` commands, and details of how to do that are fully described in the Network HowTo and the `Network Administrator's Guide' which both probably came on the CD-ROM you installed from.

A hardware configuration error is when some sort of resource conflict or mis-configuration (that the driver didn't detect at boot) stops the card from working properly. This typically can be observed in several different ways. (1) You get an error message when `ifconfig` tries to open the device for use, such as ``SIOCSFFLAGS: Try again''. (2) The driver reports `eth0` error messages (viewed by `dmesg | more`) or strange inconsistencies for each time it tries to send or receive data. (3) Typing `cat /proc/net/dev` shows non-zero numbers in one of the errs, drop, fifo, frame or carrier columns for `eth0`. (4) Typing `cat /proc/interrupts` shows a zero interrupt count for the card. Most of the typical hardware configuration errors are also discussed in the FAQ section.

Well, if you have got to this point and things still aren't working, read the FAQ section of this document, read the vendor specific section detailing your particular card, *and if it still doesn't work* then you may have to resort to posting to an appropriate newsgroup for help. If you do post, please detail all relevant information in that post, such as the card brand, the kernel version, the driver boot messages, the output from `cat /proc/net/dev`, a clear description of the problem, and of course what you have already tried to do in an effort to get things to work.

You would be surprised at how many people post useless things like ``Can someone help me? My ethernet doesn't work.'' and nothing else. Readers of the newsgroups tend to ignore such silly posts, whereas a detailed and informational problem description may allow a `linux-guru' to spot your problem right away.

---

2. What card should I buy for Linux?

The answer to this question depends heavily on exactly what you intend on doing with your net connection, and how much traffic it will see.

If you only expect a single user to be doing the occasional ftp session or WWW connection, then even an old 8 bit ISA card will probably keep you happy.

If you intend to set up a server, and you require the CPU overhead of Rx'ing and Tx'ing network data to be kept to a minimum, you probably want to look at one of the PCI cards that uses a chip with bus-mastering capapbility, such as the DEC tulip (21xxx) chip, or the AMD PCnet-PCI chip.

If you fall somewhere in the middle of the above, then any one of the low cost PCI or 16 bit ISA cards with stable drivers will do the job for you.

2.1 So What Drivers are Stable?

Of the 16 bit ISA cards, the following drivers are very mature, and you shouldn't have any problems if you buy a card that uses these drivers.

SMC-Ultra/EtherEZ, SMC-Elite (WD80x3), 3c509, Lance, NE2000.

This is not to say that all the other drivers are unstable. It just happens that the above are the oldest and most used of all the linux drivers, making them the safest choice.

Note that some el-cheapo motherboards can have trouble with the bus-mastering that the ISA Lance cards do, and some el-cheapo NE2000 clones can have trouble getting detected at boot.

The most commonly used linux PCI drivers are probably the 3Com Vortex/Boomerang (3c59x/3c9xx), the DEC tulip (21xxx), and the Intel EtherExpressPro 100. The various PCI-NE2000 clone cards are also extremely common, but purchasing a PCI-NE2000 clone card is not recommended unless the lowest possible price is more important than having a modern high-performace design card.

2.2 Eight bit vs 16 bit Cards

You probably can't buy a new 8 bit ISA ethercard anymore, but you will find lots of them turning up at computer swap meets and the like for the next few years, at very low prices. This will make them popular for ``home-ethernet'' systems. The above holds true for 16 bit ISA cards now as well, since PCI cards are now very common.

Some 8 bit cards that will provide adequate performance for light to average use are the wd8003, the 3c503 and the ne1000. The 3c501 provides poor performance, and these poor 12 year old relics of the XT days should be avoided. (Send them to Alan, he collects them...)

The 8 bit data path doesn't hurt performance that much, as you can still expect to get about 500 to 800kB/s ftp download speed to an 8 bit wd8003 card (on a fast ISA bus) from a fast host. And if most your net-traffic is going to remote sites, then the bottleneck in the path will be elsewhere, and the only speed difference you will notice is during net activity on your local subnet.

2.3 32 Bit (VLB/EISA/PCI) Ethernet Cards

Note that a 10Mbs network typically doesn't justify requiring a 32 bit interface. See Programmed I/O vs. ... as to why having a 10Mbps ethercard on an 8MHz ISA bus is really not a bottleneck. Even though having the ethercard on a fast bus won't necessarily mean faster transfers, it will usually mean reduced CPU overhead, which is good for multi-user systems.

Of course for 100Mbps networks, which are now commonplace, the 32 bit interface is a must to make use of the full bandwidth. AMD has the 32 bit PCnet-VLB and PCnet-PCI chips. See AMD PCnet-32 for info on the 32 bit versions of the LANCE / PCnet-ISA chip.

The DEC 21xxx PCI `tulip' chip is another option (see DEC 21040) for power-users. Many manufacturers produce cards that use this chip, and the prices of such no-name cards is usually quite cheap.

3Com's `Vortex' and `Boomerang' PCI cards are also another option, and the price is quite cheap if you can get one under their evaluation deal while it lasts. (see 3c590/3c595)

Intel's EtherExpress Pro 10/100 PCI cards have also been reported to work well with linux. (see EtherExpress)

Various clone manufacturers have started making PCI NE2000 clones based on a RealTek or Winbond chip. These cards are also supported by the linux ne2000 driver for v2.0.31 and newer kernels. However you only benefit from the faster bus interface, as the card is still using the age-old ne2000 driver interface. As of v2.0.34 (and above) a separate PCI-specific driver for these cards `ne2k-pci.c` is also available, which will be sightly more efficient than the ISA `ne.c` driver.

2.4 Available 100Mbs Cards and Drivers

The present list of supported 100Mbs hardware is as follows: cards with the DEC 21140 chip; the 3c595/3c90x Vortex cards; the EtherExpressPro10/100B; the PCnet-FAST; the SMC 83c170 (epic100) and the HP 100VG ANY-LAN.

Have a look at the vendor specific information for each that is in this document. You may also want to check out some of the following:

Linux and 100Mbs Ethernet

Donald's 100VG Page

Dan Kegel's Fast Ethernet Page

2.5 100VG versus 100BaseT

TOC

100BaseT is much more prominent than 100VG, and the following blurb from an older one of Donald's informative `comp.os.linux` postings summarizes the situation quite well:

``For those not in the know, there are two competing 100Mbs ethernet standards, 100VG (aka 100baseVG and 100VG-AnyLAN) and 100baseT (with 100baseTx, 100baseT4 and 100baseFx cable types).

100VG was on the market first, and I feel that it is better engineered than 100baseTx. I was rooting for it to win, but it clearly isn't going to. HP et al. made several bad choices:

1) Delaying the standard so that they could accommodate IBM and support token ring frames. It `seemed like a good idea at the time', since it would enable token ring shops to upgrade without the managers having to admit they made a very expensive mistake committing to the wrong technology. But there was nothing to be gained, as the two frame types couldn't coexist on a network, token ring is a morass of complexity, and IBM went with 100baseT anyway.

2) Producing only ISA and EISA cards. (A PCI model was only recently announced.) The ISA bus is too slow for 100mbs, and relatively few EISA machines exist. At the time VLB was common, fast, and cheap with PCI a viable choice. But "old-timer" wisdom held that servers would stay with the more expensive EISA bus.

3) Not sending me a databook. Yes, this action was the real reason for the 100VGs downfall :-). I called all over for programming info, and all I could get was a few page color glossy brochure from AT&T describing how wonderful the Regatta chipset was."

2.6 Type of cable that your card should support

If you are setting up a small ``personal" network, you will probably want to use thinnet or thin ethernet cable. This is the style with the standard BNC connectors. The thinnet, or thin ethernet cabling, (RG-58 coaxial cable) with the BNC (metal push and turn-to-lock) connectors is technically called 10Base2.

Most ethercards also come in a `Combo' version for only $10-$20 more. These have both twisted pair and thinnet transceiver built-in, allowing you to change your mind later.

The twisted pair cables, with the RJ-45 (giant phone jack) connectors is technically called 10BaseT. You may also hear it called UTP (Unsheilded Twisted Pair).

The older thick ethernet (10mm coaxial cable) which is only found in older installations is called 10Base5. The 15 pin D-shaped plug found on some ethernet cards (the AUI connector) is used to connect to thick ethernet and external transcievers.

Large corporate installations will most likely use 10BaseT instead of 10Base2. 10Base2 does not offer any upgrade path to 100Base-whatever.

See Cables, Coax... for other concerns with different types of ethernet cable.

---

3. Frequently Asked Questions

Here are some of the more frequently asked questions about using Linux with an Ethernet connection. Some of the more specific questions are sorted on a `per manufacturer basis'. Chances are the question you want an answer for has already been asked (and answered!) by someone else, so even if you don't find your answer here, you probably can find what you want from a news archive such as Dejanews.

3.1 Alpha Drivers -- Getting and Using them

I heard that there is an updated or preliminary/alpha driver available for my card. Where can I get it?

The newest of the `new' drivers can be found on Donald's ftp site: `cesdis.gsfc.nasa.gov` in the `/pub/linux/` area. Things change here quite frequently, so just look around for it. Alternatively, it may be easier to use a WWW browser on:

Don's Linux Home Page

to locate the driver that you are looking for. (Watch out for WWW browsers that silently munge the source by replacing TABs with spaces and so on - use ftp, or at least an FTP URL for downloading if unsure.)

Now, if it really is an alpha, or pre-alpha driver, then please treat it as such. In other words, don't complain because you can't figure out what to do with it. If you can't figure out how to install it, then you probably shouldn't be testing it. Also, if it brings your machine down, don't complain. Instead, send us a well documented bug report, or even better, a patch!

Note that some of the `useable' experimental/alpha drivers have been included in the standard kernel source tree. When running `make config` one of the first things you will be asked is whether to ``Prompt for development and/or incomplete code/drivers''. You will have to answer `Y' here to get asked about including any alpha/experiemntal drivers.

3.2 Using More than one Ethernet Card per Machine

What needs to be done so that Linux can run two ethernet cards?

The answer to this question depends on whether the driver(s) is/are being used as a loadable module or are compiled directly into the kernel. Most linux distributions use modular drivers now. This saves distributing lots of kernels, each with a different driver set built in. Instead a single basic kernel is used and the individual drivers that are need for a particular user's system are loaded once the system has booted far enough to access the driver module files (usually stored in `/lib/modules/`).

With the Driver as a Module: In the case of PCI drivers, the module will typically detect all of the installed cards of that brand model automatically. However, for ISA cards, probing for a card is not a safe

TOC

operation, and hence you typically need to supply the I/O base address of the card so the module knows where to look. This information is stored in the file `/etc/conf.modules`.

As an example, consider a user that has two ISA NE2000 cards, one at `0x300` and one at `0x240` and what lines they would have in their `/etc/conf.modules` file:

```
alias eth0 ne
alias eth1 ne
options ne io=0x240,0x300
```

What this does: This says that if the administrator (or the kernel) does a `modprobe eth0` or a `modprobe eth1` then the `ne.o` driver should be loaded for either `eth0` or `eth1`. Furthermore, when the `ne.o` module is loaded, it should be loaded with the options `io=0x240,0x300` so that the driver knows where to look for the cards. Note that the `0x` is important - things like `300h` as commonly used in the DOS world won't work. Switching the order of the `0x240` and the `0x300` will switch which physical card ends up as `eth0` and `eth1`.

Most of the ISA module drivers can take multiple comma separated I/O values like this example to handle multiple cards. However, some (older?) drivers, such as the 3c501.o module are currently only able to handle one card per module load. In this case you can load the module twice to get both cards detected. The `/etc/conf.modules` file in this case would look like:

```
alias eth0 3c501
alias eth1 3c501
options eth0 -o 3c501-0 io=0x280 irq=5
options eth1 -o 3c501-1 io=0x300 irq=7
```

In this example the `-o` option has been used to give each instance of the module a unique name, since you can't have two modules loaded with the same name. The `irq=` option has also been used to to specify the hardware IRQ setting of the card. (This method can also be used with modules that accept comma separated I/O values, but it is less efficient since the module ends up being loaded twice when it doesn't really need to be.)

As a final example, consider a user with one 3c503 card at `0x350`and one SMC Elite16 (wd8013) card at `0x280`. They would have:

```
alias eth0 wd
alias eth1 3c503
options wd io=0x280
options 3c503 io=0x350
```

For PCI cards, you typically only need the `alias` lines to correlate the `ethN` interfaces with the appropriate driver name, since the I/O base of a PCI card can be safely detected.

The available modules are typically stored in `/lib/modules/`uname -r`/net` where the `uname -r` command gives the kernel version (e.g. 2.0.34). You can look in there to see which one matches your card. Once you have the correct settings in your `conf.modules` file, you can test things out with:

```
modprobe ethN
dmesg | tail
```

where `N' is the number of the ethernet interface you are testing.

With the Driver Compiled into the Kernel: If you have the driver compiled into the kernel, then the hooks for multiple ethercards are all there. However, note that at the moment only *one* ethercard is auto-probed for by default. This helps to avoid possible boot time hangs caused by probing sensitive cards.

(Note: As of late 2.1.x kernels, the boot probes have been sorted into safe and unsafe, so that all safe (e.g. PCI and EISA) probes will find all related cards automatically. Systems with more than one ethernet card with at least one of them being an ISA card will still need to do one of the following.)

There are two ways that you can enable auto-probing for the second (and third, and...) card. The easiest method is to pass boot-time arguments to the kernel, which is usually done by LILO. Probing for the second card can be achieved by using a boot-time argument as simple as `ether=0,0,eth1`. In this case `eth0` and `eth1` will be assigned in the order that the cards are found at boot. Say if you want the card at `0x300` to be `eth0` and the card at `0x280` to be `eth1` then you could use

```
LILO: linux ether=5,0x300,eth0 ether=15,0x280,eth1
```

The `ether=` command accepts more than the IRQ + I/O + name shown above. Please have a look at Passing Ethernet Arguments... for the full syntax, card specific parameters, and LILO tips.

These boot time arguments can be made permanent so that you don't have to re-enter them every time. See the LILO configuration option `append' in the LILO manual.

The second way (not recommended) is to edit the file `Space.c` and replace the `0xffe0` entry for the I/O address with a zero. The `0xffe0` entry tells it not to probe for that device -- replacing it with a zero will enable autoprobing for that device.

Note that if you are intending to use Linux as a gateway between two networks, you will have to re-compile a kernel with IP forwarding enabled. Usually using an old AT/286 with something like the `kbridge' software is a better solution.

If you are viewing this while *net-surfing*, you may wish to look at a mini-howto Donald has on his WWW site. Check out Multiple Ethercards.

3.3 The `ether=` thing didn't do anything for me. Why?

As described above, the `ether=` command *only* works for drivers that are compiled into the kernel. Now most distributions use the drivers in a modular form, and so the `ether=` command is rarely used anymore. (Some older documentation has yet to be updated to reflect this change.) If you want to apply options for a modular ethernet driver you *must* make changes to the `/etc/conf.modules` file.

If you *are* using a compiled in driver, and have added an `ether=` to your LILO configuration file, note that it won't take effect until you re-run `lilo` to process the updated configuration file.

3.4 Problems with NE1000 / NE2000 cards (and clones)

Problem: PCI NE2000 clone card is not detected at boot with v2.0.x.

Reason: The `ne.c` driver up to v2.0.30 only knows about the PCI ID number of RealTek 8029 based clone cards. Since then, several others have also released PCI NE2000 clone cards, with different PCI ID numbers, and hence the driver doesn't detect them.

Solution: The easiest solution is to upgrade to a v2.0.31 (or newer) version of the linux kernel. It knows the ID numbers of about five different NE2000-PCI chips, and will detect them automatically at boot or at module loading time. If you upgrade to 2.0.34 (or newer) there is a PCI-only specific NE2000 driver that is slightly smaller and more efficient than the original ISA/PCI driver.

Problem: PCI NE2000 clone card is reported as an ne1000 (8 bit card!) at boot or when I load the ne.o module for v2.0.x, and hence doesn't work.

Reason: Some PCI clones don't implement byte wide access (and hence are not truly 100% NE2000 compatible). This causes the probe to think they are NE1000 cards.

Solution: You need to upgrade to v2.0.31 (or newer) as described above. The driver(s) now check for this hardware bug.

Problem: PCI NE2000 card gets terrible performance, even when reducing the window size as described in the Performance Tips section.

Reason: The spec sheets for the original 8390 chip, desgined and sold over ten years ago, noted that a dummy read from the chip was required before each write operation for maximum reliablity. The driver has the facility to do this but it has been disabled by default since the v1.2 kernel days. One user has reported that re-enabling this `mis-feature' helped their performance with a cheap PCI NE2000 clone card.

Solution: Since it has only been reported as a solution by one person, don't get your hopes up. Re-enabling the read before write fix is done by simply editing the driver file in `linux/drivers/net/`, uncommenting the line containing `NE_RW_BUGFIX` and then rebuilding the kernel or module as appropriate. Please send an e-mail describing the performance difference and type of card/chip you have if this helps you. (The same can be done for the `ne2k-pci.c` driver as well).

Problem: The `ne2k-pci.c` driver reports error messages like `timeout waiting for Tx RDC` with a PCI NE2000 card and doesn't work right.

Reason: Your card and/or the card to PCI bus link can't handle the long word I/O optimization used in this driver.

Solution: Firstly, check the settings available in the BIOS/CMOS setup to see if any related to PCI bus timing are too aggressive for reliable operation. Otherwise using the ISA/PCI `ne.c` driver (or removing the `#define USE_LONGIO` from `ne2k-pci.c`) should let you use the card.

Probem: ISA Plug and Play NE2000 (such as RealTek 8019) is not detected.

TOC

Reason: The original NE2000 specification (and hence the linux NE2000 driver) does not have supp for Plug and Play.

Solution: Use the DOS configuration disk that came with the card to disable PnP, and to set the card to a specified I/O address and IRQ. Add a line to `/etc/conf.modules` like `options ne io=0xNNN` where `0xNNN` is the hex I/O address you set the card to. (This assumes you are using a modular driver; if not then use an `ether=0,0xNNN,eth0` argument at boot). You may also have to enter the BIOS/CMOS setup and mark the IRQ as Legacy-ISA instead of PnP. Alternatively, if you need to leave PnP enabled for compatibility with some other operating system, then look into the *isapnptools* package. Try `man isapnp` to see if it is already installed on your system. If not, then have a look at the following URL:

ISA PNP Tools

Problem: NE*000 driver reports `not found (no reset ack)' during boot probe.

Reason: This is related to the above change. After the initial verification that an 8390 is at the probed I/O address, the reset is performed. When the card has completed the reset, it is supposed to acknowedge that the reset has completed. Your card doesn't, and so the driver assumes that no NE card is present.

Solution: You can tell the driver that you have a bad card by using an otherwise unused `mem_end` hexidecimal value of `0xbad` at boot time. You *have* to also supply a non-zero I/O base for the card when using the `0xbad` override. For example, a card that is at `0x340` that doesn't ack the reset would use something like:

```
LILO: linux ether=0,0x340,0,0xbad,eth0
```

This will allow the card detection to continue, even if your card doesn't ACK the reset. If you are using the driver as a module, then you can supply the option `bad=0xbad` just like you supply the I/O address.

Problem: NE*000 card hangs machine at first network access.

Reason: This problem has been reported for kernels as old as 1.1.57 to the present. It appears confined to a few software configurable clone cards. It appears that they expect to be initialized in some special way.

Solution: Several people have reported that running the supplied DOS software config program and/or the supplied DOS driver prior to warm booting (i.e. loadlin or the `three-finger-salute') into linux allowed the card to work. This would indicate that these cards need to be initialized in a particular fashion, slightly different than what the present Linux driver does.

Problem: NE*000 ethercard at `0x360` doesn't get detected.

Reason: Your NE2000 card is `0x20` wide in I/O space, which makes it hit the parallel port at `0x378`. Other devices that could be there are the second floppy controller (if equipped) at `0x370` and the secondary IDE controller at `0x376--0x377`. If the port(s) are already registered by another driver, the kernel will not let the probe happen.

Solution: Either move your card to an address like `0x280, 0x340, 0x320` or compile without parallel printer support.

Problem: Network `goes away' every time I print something (NE2000)

TOC

Reason: Same problem as above, but you have an older kernel that doesn't check for overlapping I/O regions. Use the same fix as above, and get a new kernel while you are at it.

Problem: NE*000 ethercard probe at 0xNNN: 00 00 C5 ... not found. (invalid signature yy zz)

Reason: First off, do you have a NE1000 or NE2000 card at the addr. 0xNNN? And if so, does the hardware address reported look like a valid one? If so, then you have a poor NE*000 clone. All NE*000 clones are supposed to have the value `0x57` in bytes 14 and 15 of the SA PROM on the card. Yours doesn't -- it has `yy zz' instead.

Solution: There are two ways to get around this. The easiest is to use an `0xbad` mem_end value as described above for the `no reset ack' problem. This will bypass the signature check, as long as a non-zero I/O base is also given. This way no recompilation of the kernel is required.

The second method (for hackers) involves changing the driver itself, and then recompiling your kernel (or module). The driver (/usr/src/linux/drivers/net/ne.c) has a "Hall of Shame" list at about line 42. This list is used to detect poor clones. For example, the DFI cards use `DFI' in the first 3 bytes of the PROM, instead of using `0x57` in bytes 14 and 15, like they are supposed to.

Problem: The machine hangs during boot right after the `8390...' or `WD....' message. Removing the NE2000 fixes the problem.

Solution: Change your NE2000 base address to something like `0x340`. Alternatively, you can use the ``reserve=" boot argument in conjunction with the ``ether=" argument to protect the card from other device driver probes.

Reason: Your NE2000 clone isn't a good enough clone. An active NE2000 is a bottomless pit that will trap any driver autoprobing in its space. Changing the NE2000 to a less-popular address will move it out of the way of other autoprobes, allowing your machine to boot.

Problem: The machine hangs during the SCSI probe at boot.

Reason: It's the same problem as above, change the ethercard's address, or use the reserve/ether boot arguments.

Problem: The machine hangs during the soundcard probe at boot.

Reason: No, that's really during the silent SCSI probe, and it's the same problem as above.

Problem: NE2000 not detected at boot - no boot messages at all

Solution: There is no `magic solution' as there can be a number of reasons why it wasn't detected. The following list should help you walk through the possible problems.

1) Build a new kernel with only the device drivers that you need. Verify that you are indeed booting the fresh kernel. Forgetting to run lilo, etc. can result in booting the old one. (Look closely at the build time/date reported at boot.) Sounds obvious, but we have all done it before. Make sure the driver is in fact included in the new kernel, by checking the `System.map` file for names like `ne_probe`.

2) Look at the boot messages carefully. Does it ever even mention doing a ne2k probe such as `NE*000 probe at 0xNNN: not found (blah blah)' or does it just fail silently. There is a big difference. Use

TOC

`dmesg|more` to review the boot messages after logging in, or hit Shift-PgUp to scroll the screen up the boot has completed and the login prompt appears.

3) After booting, do a `cat /proc/ioports` and verify that the full iospace that the card will require is vacant. If you are at `0x300` then the ne2k driver will ask for `0x300-0x31f`. If any other device driver has registered even one port anywhere in that range, the probe will not take place at that address and will silently continue to the next of the probed addresses. A common case is having the lp driver reserve `0x378` or the second IDE channel reserve `0x376` which stops the ne driver from probing `0x360-0x380`.

4) Same as above for `cat /proc/interrupts`. Make sure no other device has registered the interrupt that you set the ethercard for. In this case, the probe will happen, and the ether driver will complain loudly at boot about not being able to get the desired IRQ line.

5) If you are still stumped by the silent failure of the driver, then edit it and add some printk() to the probe. For example, with the ne2k you could add/remove lines (marked with a `+' or `-') in `linux/drivers/net/ne.c` like:

```
    int reg0 = inb_p(ioaddr);

+    printk("NE2k probe - now checking %x\n",ioaddr);
-    if (reg0 == 0xFF)
+    if (reg0 == 0xFF) {
+       printk("NE2k probe - got 0xFF (vacant I/O port)\n");
        return ENODEV;
+    }
```

Then it will output messages for each port address that it checks, and you will see if your card's address is being probed or not.

6) You can also get the ne2k diagnostic from Don's ftp site (mentioned in the howto as well) and see if it is able to detect your card after you have booted into linux. Use the `-p 0xNNN' option to tell it where to look for the card. (The default is `0x300` and it doesn't go looking elsewhere, unlike the boot-time probe.) The output from when it finds a card will look something like:

```
Checking the ethercard at 0x300.
  Register 0x0d (0x30d) is 00
  Passed initial NE2000 probe, value 00.
8390 registers: 0a 00 00 00 63 00 00 00 01 00 30 01 00 00 00 00
SA PROM  0: 00 00 00 00 c0 c0 b0 b0 05 05 65 65 05 05 20 20
SA PROM 0x10: 00 00 07 07 0d 0d 01 01 14 14 02 02 57 57 57 57

        NE2000 found at 0x300, using start page 0x40 and end page 0x80.
```

Your register values and PROM values will probably be different. Note that all the PROM values are doubled for a 16 bit card, and that the ethernet address (00:00:c0:b0:05:65) appears in the first row, and the double `0x57` signature appears at the end of the PROM.

The output from when there is no card installed at `0x300` will look something like this:

```
Checking the ethercard at 0x300.
  Register 0x0d (0x30d) is ff
  Failed initial NE2000 probe, value ff.
8390 registers: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
SA PROM        0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
SA PROM 0x10: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

 Invalid signature found, wordlength 2.
```

The `0xff` values arise because that is the value that is returned when one reads a vacant I/O port. If you happen to have some other hardware in the region that is probed, you may see some non `0xff` values as well.

7) Try warm booting into linux from a DOS boot floppy (via loadlin) after running the supplied DOS driver or config program. It may be doing some extra (i.e. non-standard) "magic" to initialize the card.

8) Try Russ Nelson's ne2000.com packet driver to see if even it can see your card -- if not, then things do not look good. Example:

```
A:> ne2000 0x60 10 0x300
```

The arguments are software interrupt vector, hardware IRQ, and I/O base. You can get it from any msdos archive in pktdrv11.zip -- The current version may be newer than 11.

3.5 Problems with SMC Ultra/EtherEZ and WD80*3 cards

Problem: You get messages such as the following:

```
eth0: bogus packet size: 65531, status=0xff, nxpg=0xff
```

Reason: There is a shared memory problem.

Solution: The most common reason for this is PCI machines that are not configured to map in ISA memory devices. Hence you end up reading the PC's RAM (all `0xff` values) instead of the RAM on the card that contains the data from the received packet.

Other typical problems that are easy to fix are board conflicts, having cache or `shadow ROM' enabled for that region, or running your ISA bus faster than 8Mhz. There are also a surprising number of memory

TOC

failures on ethernet cards, so run a diagnostic program if you have one for your ethercard.

Problem: SMC EtherEZ doesn't work in non-shared memory (PIO) mode.

Reason: Older versions of the Ultra driver only supported the card in the shared memory mode of operation.

Solution: The driver in kernel version 2.0 and above also supports the programmed I/O mode of operation. Upgrade to v2.0 or newer.

Problem: Old wd8003 and/or jumper-settable wd8013 always get the IRQ wrong.

Reason: The old wd8003 cards and jumper-settable wd8013 clones don't have the EEPROM that the driver can read the IRQ setting from. If the driver can't read the IRQ, then it tries to auto-IRQ to find out what it is. And if auto-IRQ returns zero, then the driver just assigns IRQ 5 for an 8 bit card or IRQ 10 for a 16 bit card.

Solution: Avoid the auto-IRQ code, and tell the kernel what the IRQ that you have jumpered the card to in your module configuration file (or via a boot time argument for in-kernel drivers).

Problem: SMC Ultra card is detected as wd8013, but the IRQ and shared memory base is wrong.

Reason: The Ultra card looks a lot like a wd8013, and if the Ultra driver is not present in the kernel, the wd driver may mistake the ultra as a wd8013. The ultra probe comes before the wd probe, so this usually shouldn't happen. The ultra stores the IRQ and mem base in the EEPROM differently than a wd8013, hence the bogus values reported.

Solution: Recompile with only the drivers you need in the kernel. If you have a mix of wd and ultra cards in one machine, and are using modules, then load the ultra module first.

3.6 Problems with 3Com cards

Problem: The 3c503 picks IRQ N, but this is needed for some other device which needs IRQ N. (eg. CD ROM driver, modem, etc.) Can this be fixed without compiling this into the kernel?

Solution: The 3c503 driver probes for a free IRQ line in the order {5, 9/2, 3, 4}, and it should pick a line which isn't being used. The driver chooses when the card is `ifconfig`'ed into operation.

If you are using a modular driver, you can use module parameters to set various things, including the IRQ value.

The following selects IRQ9, base location `0x300`, <ignored value>, and if_port #1 (the external transceiver).

```
io=0x300 irq=9 xcvr=1
```

Alternately, if the driver is compiled into the kernel, you can set the same values at boot by passing parameters via LILO.

```
LILO: linux ether=9,0x300,0,1,eth0
```

The following selects IRQ3, probes for the base location, <ignored value>, and the default if_port #0 (the

internal transceiver)

```
LILO: linux ether=3,0,0,0,eth0
```

Problem: 3c503: configured interrupt X invalid, will use autoIRQ.

Reason: The 3c503 card can only use one of IRQ{5, 2/9, 3, 4} (These are the only lines that are connected to the card.) If you pass in an IRQ value that is not in the above set, you will get the above message. Usually, specifying an interrupt value for the 3c503 is not necessary. The 3c503 will autoIRQ when it gets ifconfig'ed, and pick one of IRQ{5, 2/9, 3, 4}.

Solution: Use one of the valid IRQs listed above, or enable autoIRQ by not specifying the IRQ line at all.

Problem: The supplied 3c503 drivers don't use the AUI (thicknet) port. How does one choose it over the default thinnet port?

Solution: The 3c503 AUI port can be selected at boot-time for in-kernel drivers, and at module insertion for modular drivers. The selection is overloaded onto the low bit of the currently-unused dev->rmem_start variable, so a boot-time parameter of:

```
LILO: linux ether=0,0,0,1,eth0
```

should work for in-kernel drivers.

To specify the AUI port when loading as a module, just append `xcvr=1` to the module options line along with your I/O and IRQ values.

3.7 FAQs Not Specific to Any Card.

Linux and ISA Plug and Play Ethernet Cards

For best results (and minimum aggravation) it is recommended that you use the (usually DOS) program that came with your card to disable the PnP mechanism and set it to a fixed I/O address and IRQ. Make sure the I/O address you use is probed by the driver at boot, or if using modules then supply the address as an `io=` option in `/etc/conf.modules`. You may also have to enter the BIOS/CMOS setup and mark the IRQ as Legacy-ISA instead of PnP (if your computer has this option).

Note that you typically don't need DOS installed to run a DOS based configuration program. You can usually just boot a DOS floppy disk and run them from the supplied floppy disk. You can also download OpenDOS and FreeDOS for free.

If you require PnP enabled for compatibility with some other operating system then you will have to use the isapnptools package with linux to configure the card(s) each time at boot. You will still have to make sure the I/O address chosen for the card is probed by the driver or supplied as an `io=` option.

Ethercard is Not Detected at Boot.

The usual reason for this is that people are using a kernel that does not have support for their particular card built in. For a modular kernel, it usually means that the required module has not been requested for loading, or that an I/O address needs to be specified as a module option.

If you are using a modular based kernel, such as those installed by most of the linux distributions, then and use the configuration utility for the distribution to select the module for your card. For ISA cards, it is a good idea to determine the I/O address of the card and add it as an option (e.g. `io=0x340`) if the configuration utility asks for any options. If there is no configuration utility, then you will have to add the correct module name (and options) to `/etc/conf.modules` -- see `man modprobe` for more details.

If you are using a pre-compiled kernel that is part of a distribution set, then check the documentation to see which kernel you installed, and if it was built with support for your particular card. If it wasn't, then your options are to try and get one that has support for your card, or build your own.

It is usually wise to build your own kernel with only the drivers you need, as this cuts down on the kernel size (saving your precious RAM for applications!) and reduces the number of device probes that can upset sensitive hardware. Building a kernel is not as complicated as it sounds. You just have to answer yes or no to a bunch of questions about what drivers you want, and it does the rest.

The next main cause is having another device using part of the I/O space that your card needs. Most cards are 16 or 32 bytes wide in I/O space. If your card is set at `0x300` and 32 bytes wide, then the driver will ask for `0x300-0x31f`. If any other device driver has registered even one port anywhere in that range, the probe will not take place at that address and the driver will silently continue to the next of the probed addresses. So, after booting, do a `cat /proc/ioports` and verify that the full I/O space that the card will require is vacant.

Another problem is having your card jumpered to an I/O address that isn't probed by default. The list of probed addresses for each driver is easily found just after the text comments in the driver source. Even if the I/O setting of your card is not in the list of probed addresses, you can supply it at boot (for in-kernel drivers) with the `ether=` command as described in Passing Ethernet Arguments... Modular drivers can make use of the `io=` option in `/etc/conf.modules` to specify an address that isn't probed by default.

`ifconfig` reports the wrong I/O address for the card.

No it doesn't. You are just interpreting it incorrectly. This is *not* a bug, and the numbers reported are correct. It just happens that some 8390 based cards (wd80x3, smc-ultra, etc) have the actual 8390 chip living at an offset from the first assigned I/O port. This is the value stored in `dev->base_addr`, and is what `ifconfig` reports. If you want to see the full range of ports that your card uses, then try `cat /proc/ioports` which will give the numbers you expect.

PCI machine detects card but driver fails probe.

Some PCI BIOSes may not enable all PCI cards at power-up, especially if the BIOS option `PNP OS' is enabled. This mis-feature is to support the current release of Windows which still uses some real-mode drivers. Either disable this option, or try and upgrade to a newer driver which has the code to enable a disabled card.

Shared Memory ISA cards in PCI Machine do not work (`0xffff`)

This will usually show up as reads of lots of `0xffff` values. No shared memory cards of any type will work in a PCI machine unless you have the PCI ROM BIOS/CMOS SETUP configuration set properly. You have to set it to allow shared memory access from the ISA bus for the memory region that your card is trying to use. If you can't figure out which settings are applicable then ask your supplier or local computer guru. For AMI BIOS, there is usually a "Plug and Play" section where there will be an ``ISA Shared Memory Size" and ``ISA Shared Memory Base" settings. For cards like the wd8013 and SMC Ultra, change the size from the default of `Disabled' to 16kB, and change the base to the shared memory address of your card.

Card seems to send data but never receives anything.

Do a `cat /proc/interrupts`. A running total of the number of interrupt events your card generates will be in the list given from the above. If it is zero and/or doesn't increase when you try to use the card then there is probably a physical interrupt conflict with another device installed in the computer (regardless of whether or not the other device has a driver installed/available). Change the IRQ of one of the two devices to a free IRQ.

Asynchronous Transfer Mode (ATM) Support

Werner Almesberger has been working on ATM support for linux. He has been working with the Efficient Networks ENI155p board (Efficient Networks) and the Zeitnet ZN1221 board (Zeitnet).

Werner says that the driver for the ENI155p is rather stable, while the driver for the ZN1221 is presently unfinished.

Check the latest/updated status at the following URL:

Linux ATM Support

Gigabyte Ethernet Support

Is there any gigabyte ethernet support for Linux?

Yes, there are currently at least two. A driver for the Packet Engines G-NIC PCI Gigabit Ethernet adapter is available in the v2.0 and v2.2 kernels For more details, support, and driver updates, see:

`http://cesdis.gsfc.nasa.gov/linux/drivers/yellowfin.html`

The `acenic.c` driver available in the v2.2 kernels can be used for the Alteon AceNIC Gigabit Ethernet card and other Tigon based cards such as the 3Com 3c985. The driver should also work on the NetGear GA620, however this has yet to be verified.

TOC

FDDI Support

Is there FDDI support for Linux?

Yes. Larry Stefani has written a driver for v2.0 with Digital's DEFEA (FDDI EISA) and DEFPA (FDDI PCI) cards. This was included into the v2.0.24 kernel. Currently no other cards are supported though.

Full Duplex Support

Will Full Duplex give me 20MBps? Does Linux support it?

Cameron Spitzer writes the following about full duplex 10Base-T cards: ``If you connect it to a full duplex switched hub, and your system is fast enough and not doing much else, it can keep the link busy in both directions. There is no such thing as full duplex 10BASE-2 or 10BASE-5 (thin and thick coax). Full Duplex works by disabling collision detection in the adapter. That's why you can't do it with coax; the LAN won't run that way. 10BASE-T (RJ45 interface) uses separate wires for send and receive, so it's possible to run both ways at the same time. The switching hub takes care of the collision problem. The signalling rate is 10 Mbps."

So as you can see, you still will only be able to receive or transmit at 10Mbps, and hence don't expect a 2x performance increase. As to whether it is supported or not, that depends on the card and possibly the driver. Some cards may do auto-negotiation, some may need driver support, and some may need the user to select an option in a card's EEPROM configuration. Only the serious/heavy user would notice the difference between the two modes anyway.

Ethernet Cards for Linux on SMP Machines

If you spent the extra money on a multi processor (MP) computer then buy a good ethernet card as well. For v2.0 kernels it wasn't really an issue, but it definitely is for v2.2. Most of the older non-intelligent (e.g. ISA bus PIO and shared memory design) cards were never designed with any consideration for use on a MP machine. The executive summary is to buy an intelligent modern design card and make sure the driver has been written (or updated) to handle MP operation. (The key words here are `modern design' - the PCI-NE2000's are just a 10+ year old design on a modern bus.) Looking for the text `spin_lock` in the driver source is a good indication that the driver has been written to deal with MP operation. The full details of why you should buy a good card for MP use (and what happens if you dont) follow.

In v2.0 kernels, only one processor was allowed `in kernel' (i.e. changing kernel data and/or running device drivers) at any given time. So from the point of view of the card (and the associated driver) nothing was different from uni processor (UP) operation and things just continued to work. (This was the easiest way to get a working MP version of Linux - one big lock around the whole kernel only allows one processor in at a time. This way you know that you won't have two processors trying to change the same thing at the same time!)

The downside to only allowing one processor in the kernel at a time was that you only got MP performance if the running programs were self contained and calculation intensive. If the programs did a lot of input/output (I/O) such as reading or writing data to disk or over a network, then all but one of the processors would be stalled waiting on their I/O requests to be completed while the one processor running in kernel frantically tries to run all the device drivers to fill the I/O requests. The kernel becomes the

TOC

bottleneck and since there is only one processor running in the kernel, the performance of a MP machine in the heavy I/O, single-lock case quickly degrades close to that of a single processor machine.

Since this is clearly less than ideal (esp. for file/WWW servers, routers, etc.) the v2.2 kernels have finer grained locking - meaning that more than one processor can be in the kernel at a time. Instead of one big lock around the whole kernel, there are a lot of smaller locks protecting critical data from being manipulated by more than one processor at a time - e.g. one processor can be running the driver for the network card, while another processor is running the driver for the disk drive at the same time.

Okay, with that all in mind here are the snags: The finer locking means that you can have one processor trying to send data out through an ethernet driver while another processor tries to access the same driver/card to do something else (such as get the card statistics for `cat /proc/net/dev`). Oops - your card stats just got sent out over the wire, while you got data for your stats instead. Yes, the card got confused by being asked to do two (or more!) things at once, and chances are it crashed your machine in the process.

So, the driver that worked for UP is no longer good enough - it needs to be updated with locks that control access to the underlying card so that the three tasks of receive, transmit and manipulation of configuration data are serialized to the degree required by the card for stable operation. The scary part here is that a driver not yet updated with locks for stable MP operation will probably appear to be working in a MP machine under light network load, but will crash the machine or at least exhibit strange behaviour when two (or more!) processors try to do more than one of these three tasks at the same time.

The updated MP aware ethernet driver will (at a minimum) require a lock around the driver that limits access at the entry points from the kernel into the driver to `one at a time please'. With this in place, things will be serialized so that the underlying hardware should be treated just as if it was being used in a UP machine, and so it should be stable. The downside is that the one lock around the whole ethernet driver has the same negative performance implications that having one big lock around the whole kernel had (but on a smaller scale) - i.e. you can only have one processor dealing with the card at a time. [Technical Note: The performance impact may also include increased interrupt latencies if the locks that need to be added are of the `irqsave` type and they are held for a long time.]

Possible improvements on this situation can be made in two ways. You can try to minimize the time between when the lock is taken and when it is released, and/or you can implement finer grained locking within the driver (e.g. a lock around the whole driver would be overkill if a lock or two protecting against simultaneous access to a couple of sensitive registers/settings on the card would suffice).

However, for older non-intelligent cards that were never designed with MP use in mind, neither of these improvements may be feasible. Worse yet is that the non-intelligent cards typically require the processor to move the data between the card and the computer memory, so in a worst case scenario the lock will be held the whole time that it takes to move each 1.5kB data packet over an ISA bus.

The more modern intelligent cards typically move network data directly to and from the computer memory without any help from a processor. This is a big win, since the lock is then only held for the short time it takes the processor to tell the card where in memory to get/store the next network data packet. More modern card designs are less apt to require a single big lock around the whole driver as well.

Ethernet Cards for Linux on Alpha/AXP PCI Boards

As of v2.0, only the 3c509, depca, de4x5, pcnet32, and all the 8390 drivers (wd, smc-ultra, ne, 3c503, etc.) have been made `architecture independent' so as to work on the DEC Alpha CPU based systems. Other updated PCI drivers from Donald's WWW page may also work as these have been written with architecture independence in mind.

Note that the changes that are required to make a driver architecture independent aren't that complicated. You only need to do the following:

-multiply all `jiffies` related values by HZ/100 to account for the different HZ value that the Alpha uses. (i.e `timeout=2;` becomes `timeout=2*HZ/100;`)

-replace any I/O memory (640k to 1MB) pointer dereferences with the appropriate readb() writeb() readl() writel() calls, as shown in this example.

```
-       int *mem_base = (int *)dev->mem_start;
-       mem_base[0] = 0xba5eba5e;
+       unsigned long mem_base = dev->mem_start;
+       writel(0xba5eba5e, mem_base);
```

-replace all memcpy() calls that have I/O memory as source or target destinations with the appropriate one of `memcpy_fromio()` or `memcpy_toio()`.

Details on handling memory accesses in an architecture independent fashion are documented in the file `linux/Documentation/IO-mapping.txt` that comes with recent kernels.

Ethernet for Linux on SUN/Sparc Hardware.

For the most up to date information on Sparc stuff, try the following URL:

Linux Sparc

Note that some Sparc ethernet hardware gets its MAC address from the host computer, and hence you can end up with multiple interfaces all with the same MAC address. If you need to put more than one interface on the same net then use the `hw` option to `ifconfig` to assign unique MAC address.

Issues regarding porting PCI drivers to the Sparc platform are similar to those mentioned above for the AXP platform. In addition there may be some endian issues, as the Sparc is big endian, and the AXP and ix86 are little endian.

Ethernet for Linux on Other Hardware.

There are several other hardware platforms that Linux can run on, such as Atari/Amiga (m68k). As in the Sparc case it is best to check with the home site of each Linux port to that platform to see what is currently supported. (Links to such sites are welcome here - send them in!)

TOC

Linking 10 or 100 BaseT without a Hub

Can I link 10/100BaseT (RJ45) based systems together without a hub?

You can link 2 machines easily, but no more than that, without extra devices/gizmos. See Twisted Pair -- it explains how to do it. And no, you can't hack together a hub just by crossing a few wires and stuff. It's pretty much impossible to do the collision signal right without duplicating a hub.

SIOCSIFxxx: No such device

I get a bunch of `SIOCSIFxxx: No such device' messages at boot, followed by a `SIOCADDRT: Network is unreachable' What is wrong?

Your ethernet device was not detected at boot/module insertion time, and when `ifconfig` and `route` are run, they have no device to work with. Use `dmesg | more` to review the boot messages and see if there are any messages about detecting an ethernet card.

SIOCSFFLAGS: Try again

I get `SIOCSFFLAGS: Try again' when I run `ifconfig' -- Huh?

Some other device has taken the IRQ that your ethercard is trying to use, and so the ethercard can't use the IRQ. You don't necessairly need to reboot to resolve this, as some devices only grab the IRQs when they need them and then release them when they are done. Examples are some sound cards, serial ports, floppy disk driver, etc. You can type `cat /proc/interrupts` to see which interrupts are presently *in use*. Most of the Linux ethercard drivers only grab the IRQ when they are opened for use via `ifconfig'. If you can get the other device to `let go' of the required IRQ line, then you should be able to `Try again' with ifconfig.

Using `ifconfig' and Link UNSPEC with HW-addr of 00:00:00:00:00:00

When I run ifconfig with no arguments, it reports that LINK is UNSPEC (instead of 10Mbs Ethernet) and it also says that my hardware address is all zeros.

This is because people are running a newer version of the `ifconfig' program than their kernel version. This new version of ifconfig is not able to report these properties when used in conjunction with an older kernel. You can either upgrade your kernel, `downgrade' ifconfig, or simply ignore it. The kernel knows your hardware address, so it really doesn't matter if ifconfig can't read it.

You may also get strange information if the `ifconfig` program you are using is a lot older than the kernel you are using.

Huge Number of RX and TX Errors

When I run ifconfig with no arguments, it reports that I have a huge error count in both rec'd and transmitted packets. It all seems to work ok -- What is wrong?

Look again. It says `RX packets` *big number* **PAUSE** `errors 0` **PAUSE** `dropped 0` **PAUSE**

`overrun 0`. And the same for the `TX` column. Hence the big numbers you are seeing are the total number of packets that your machine has rec'd and transmitted. If you still find it confusing, try typing `cat /proc/net/dev` instead.

Entries in `/dev/` for Ethercards

I have /dev/eth0 as a link to /dev/xxx. Is this right?

Contrary to what you have heard, the files in /dev/* are not used. You can delete any `/dev/wd0`, `/dev/ne0` and similar entries.

Linux and ``trailers''

Should I disable trailers when I `ifconfig' my ethercard?

You can't disable trailers, and you shouldn't want to. `Trailers' are a hack to avoid data copying in the networking layers. The idea was to use a trivial fixed-size header of size `H', put the variable-size header info at the end of the packet, and allocate all packets `H' bytes before the start of a page. While it was a good idea, it turned out to not work well in practice. If someone suggests the use of `-trailers', note that it is the equivalent of sacrificial goats blood. It won't do anything to solve the problem, but if problem fixes itself then someone can claim deep magical knowledge.

Access to the raw Ethernet Device

How do I get access to the raw ethernet device in linux, without going through TCP/IP and friends?

```
int s=socket(AF_INET,SOCK_PACKET,htons(ETH_P_ALL));
```

This gives you a socket receiving every protocol type. Do `recvfrom()` calls to it and it will fill the sockaddr with device type in sa_family and the device name in the sa_data array. I don't know who originally invented SOCK_PACKET for Linux (its been in for ages) but its superb stuff. You can use it to send stuff raw too via `sendto()` calls. You have to have root access to do either of course.

---

4. Performance Tips

Here are some tips that you can use if you are suffering from low ethernet throughput, or to gain a bit more speed on those ftp transfers.

The `ttcp.c` program is a good test for measuring raw throughput speed. Another common trick is to do a `ftp> get large_file /dev/null` where `large_file` is > 1MB and residing in the buffer cache on the Tx'ing machine. (Do the `get' at least twice, as the first time will be priming the buffer cache on the Tx'ing machine.) You want the file in the buffer cache because you are not interested in combining the file access speed from the disk into your measurement. Which is also why you send the incoming data to `/dev/null` instead of onto the disk.

4.1 General Concepts

Even an 8 bit card is able to receive back-to-back packets without any problems. The difficulty arises when the computer doesn't get the Rx'd packets off the card quick enough to make room for more incoming packets. If the computer does not quickly clear the card's memory of the packets already received, the card will have no place to put the new packet.

In this case the card either drops the new packet, or writes over top of a previously received packet. Either one seriously interrupts the smooth flow of traffic by causing/requesting re-transmissions and can seriously degrade performance by up to a factor of 5!

Cards with more onboard memory are able to ``buffer'' more packets, and thus can handle larger bursts of back-to-back packets without dropping packets. This in turn means that the card does not require as low a latency from the the host computer with respect to pulling the packets out of the buffer to avoid dropping packets.

Most 8 bit cards have an 8kB buffer, and most 16 bit cards have a 16kB buffer. Most Linux drivers will reserve 3kB of that buffer (for two Tx buffers), leaving only 5kB of receive space for an 8 bit card. This is room enough for only three full sized (1500 bytes) ethernet packets.

4.2 ISA Cards and ISA Bus Speed

As mentioned above, if the packets are removed from the card fast enough, then a drop/overrun condition won't occur even when the amount of Rx packet buffer memory is small. The factor that sets the rate at which packets are removed from the card to the computer's memory is the speed of the data path that joins the two -- that being the ISA bus speed. (If the CPU is a dog-slow 386sx-16, then this will also play a role.)

The recommended ISA bus clock is about 8MHz, but many motherboards and peripheral devices can be run at higher frequencies. The clock frequency for the ISA bus can usually be set in the CMOS setup, by

selecting a divisor of the mainboard/CPU clock frequency. Some ISA and PCI/ISA mainboards may have this option, and so you are stuck with the factory default.

For example, here are some receive speeds as measured by the TTCP program on a 40MHz 486, with an 8 bit WD8003EP card, for different ISA bus speeds.

```
ISA Bus Speed (MHz)     Rx TTCP (kB/s)
-------------------     --------------
6.7                     740
13.4                    970
20.0                    1030
26.7                    1075
```

You would be hard pressed to do better than 1075kB/s with *any* 10Mb/s ethernet card, using TCP/IP. However, don't expect every system to work at fast ISA bus speeds. Most systems will not function properly at speeds above 13MHz. (Also, some PCI systems have the ISA bus speed fixed at 8MHz, so that the end user does not have the option of increasing it.)

In addition to faster transfer speeds, one will usually also benefit from a reduction in CPU usage due to the shorter duration memory and I/O cycles. (Note that hard disks and video cards located on the ISA bus will also usually experience a performance increase from an increased ISA bus speed.)

Be sure to back up your data prior to experimenting with ISA bus speeds in excess of 8MHz, and thouroughly test that all ISA peripherals are operating properly after making any speed increases.

4.3 Setting the TCP Rx Window

Once again, cards with small amounts of onboard RAM and relatively slow data paths between the card and the computer's memory run into trouble. The default TCP Rx window setting is 32kB, which means that a fast computer on the same subnet as you can dump 32k of data on you without stopping to see if you received any of it okay.

Recent versions of the `route` command have the ability to set the size of this window on the fly. Usually it is only for the local net that this window must be reduced, as computers that are behind a couple of routers or gateways are `buffered' enough to not pose a problem. An example usage would be:

```
route add <whatever> ... window <win_size>
```

where `win_size` is the size of the window you wish to use (in bytes). An 8 bit 3c503 card on an ISA bus operating at a speed of 8MHz or less would work well with a window size of about 4kB. Too large a window will cause overruns and dropped packets, and a drastic reduction in ethernet throughput. You can check the operating status by doing a `cat /proc/net/dev` which will display any dropped or

overrun conditions that occurred.

4.4 Increasing NFS performance

Some people have found that using 8 bit cards in NFS clients causes poorer than expected performance, when using 8kB (native Sun) NFS packet size.

The possible reason for this could be due to the difference in on board buffer size between the 8 bit and the 16 bit cards. The maximum ethernet packet size is about 1500 bytes. Now that 8kB NFS packet will arrive as about 6 back to back maximum size ethernet packets. Both the 8 and 16 bit cards have no problem Rx'ing back to back packets. The problem arises when the machine doesn't remove the packets from the cards buffer in time, and the buffer overflows. The fact that 8 bit cards take an extra ISA bus cycle per transfer doesn't help either. What you *can* do if you have an 8 bit card is either set the NFS transfer size to 2kB (or even 1kB), or try increasing the ISA bus speed in order to get the card's buffer cleared out faster. I have found that an old WD8003E card at 8MHz (with no other system load) can keep up with a large receive at 2kB NFS size, but not at 4kB, where performance was degraded by a factor of three.

On the other hand, if the default mount option is to use 1kB size and you have at least a 16 bit ISA card, you may find a significant increase in going to 4kB (or even 8kB).

---

5. Vendor/Manufacturer/Model Specific Information

The following lists many cards in alphabetical order by vendor name and then product identifier. Beside each product ID, you will see either `Supported', `Semi-Supported' or `Not Supported'.

Supported means that a driver for that card exists, and many people are happily using it and it seems quite reliable.

Semi-Supported means that a driver exists, but at least one of the following descriptions is true: (1) The driver and/or hardware are buggy, which may cause poor performance, failing connections or even crashes. (2) The driver is new or the card is fairly uncommon, and hence the driver has seen very little use/testing and the driver author has had very little feedback. Obviously (2) is preferable to (1), and the individual description of the card/driver should make it clear which one holds true. In either case, you will probably have to answer `Y' when asked ``Prompt for development and/or incomplete code/drivers?'' when running `make config`.

Not Supported means there is not a driver currently available for that card. This could be due to a lack of interest in hardware that is rare/uncommon, or because the vendors won't release the hardware documentation required to write a driver.

Note that the difference between `Supported' and `Semi-Supported' is rather subjective, and is based on user feedback observed in newsgroup postings and mailing list messages. (After all, it is impossible for one person to test all drivers with all cards for each kernel version!!!) So be warned that you may find a card listed as semi-supported works perfectly for you (which is great), or that a card listed as supported gives you no end of troubles and problems (which is not so great).

After the status, the name of the driver given in the linux kernel is listed. This will also be the name of the driver module that would be used in the `alias eth0 driver_name` line that is found in the `/etc/conf.modules` module configuration file.

5.1 3Com

If you are not sure what your card is, but you think it is a 3Com card, you can probably figure it out from the assembly number. 3Com has a document `Identifying 3Com Adapters By Assembly Number' (ref 24500002) that would most likely clear things up. See Technical Information from 3Com for info on how to get documents from 3Com.

Also note that 3Com has a FTP site with various goodies: `ftp.3Com.com` that you may want to check out.

For those of you browsing this document by a WWW browser, you can try 3Com's WWW site as well.

3c501

Status: Semi-Supported, Driver Name: 3c501

This obsolete stone-age 8 bit card is really too brain-damaged to use. Avoid it like the plague. Do not purchase this card, even as a joke. It's performance is horrible, and it breaks in many ways.

For those not yet convinced, the 3c501 can only do one thing at a time -- while you are removing one packet from the single-packet buffer it cannot receive another packet, nor can it receive a packet while loading a transmit packet. This was fine for a network between two 8088-based computers where processing each packet and replying took 10's of msecs, but modern networks send back-to-back packets for almost every transaction.

AutoIRQ works, DMA isn't used, the autoprobe only looks at `0x280` and `0x300`, and the debug level is set with the third boot-time argument.

Once again, the use of a 3c501 is *strongly discouraged*! Even more so with a IP multicast kernel, as you will grind to a halt while listening to *all* multicast packets. See the comments at the top of the source code for more details.

EtherLink II, 3c503, 3c503/16

Status: Supported, Driver Name: 3c503 (+8390)

The 3c503 does not have ``EEPROM setup'', so a diagnostic/setup program isn't needed before running the card with Linux. The shared memory address of the 3c503 is set using jumpers that are shared with the boot PROM address. This is confusing to people familiar with other ISA cards, where you always leave the jumper set to ``disable'' unless you have a boot PROM.

These cards should be about the same speed as the same bus width WD80x3, but turn out to be actually a bit slower. These shared-memory ethercards also have a programmed I/O mode that doesn't use the 8390 facilities (their engineers found too many bugs!) The Linux 3c503 driver can also work with the 3c503 in programmed-I/O mode, but this is slower and less reliable than shared memory mode. Also, programmed-I/O mode is not as well tested when updating the drivers. You shouldn't use the programmed-I/O mode unless you need it for MS-DOS compatibility.

The 3c503's IRQ line is set in software, with no hints from an EEPROM. Unlike the MS-DOS drivers, the Linux driver has capability to autoIRQ: it uses the first available IRQ line in {5,2/9,3,4}, selected each time the card is ifconfig'ed. (Older driver versions selected the IRQ at boot time.) The ioctl() call in `ifconfig' will return EAGAIN if no IRQ line is available at that time.

Some common problems that people have with the 503 are discussed in Problems with...

If you intend on using this driver as a loadable module you should probably see Using the Ethernet Drivers as Modules for module specific information.

Note that some old diskless 386 workstations have an on board 3c503 (made by 3Com and sold under different names, like `Bull') but the vendor ID is not a 3Com ID and so it won't be detected. More details can be found in the Etherboot package, which you will need anyways to boot these diskless boxes.

TOC

Etherlink Plus 3c505

Status: Semi-Supported, Driver Name: 3c505

This is a driver that was written by Craig Southeren `geoffw@extro.ucc.su.oz.au`. These cards also use the i82586 chip. There are not that many of these cards about. It is included in the standard kernel, but it is classed as an alpha driver. See Alpha Drivers for important information on using alpha-test ethernet drivers with Linux.

There is also the file `/usr/src/linux/drivers/net/README.3c505` that you should read if you are going to use one of these cards. It contains various options that you can enable/disable.

Etherlink-16 3c507

Status: Semi-Supported, Driver Name: 3c507

This card uses one of the Intel chips, and the development of the driver is closely related to the development of the Intel Ether Express driver. The driver is included in the standard kernel release, but as an alpha driver. See Alpha Drivers for important information on using alpha-test ethernet drivers with Linux.

Etherlink III, 3c509 / 3c509B

Status: Supported, Driver Name: 3c509

This card is fairly inexpensive and has good performance for an ISA non-bus-master design. The drawbacks are that the original 3c509 requires very low interrupt latency. The 3c509B shouldn't suffer from the same problem, due to having a larger buffer. (See below.) These cards use PIO transfers, similar to a ne2000 card, and so a shared memory card such as a wd8013 will be more efficient in comparison.

The original 3c509 has a small packet buffer (4kB total, 2kB Rx, 2kB Tx), causing the driver to occasionally drop a packet if interrupts are masked for too long. To minimize this problem, you can try unmasking interrupts during IDE disk transfers (see `man hdparm`) and/or increasing your ISA bus speed so IDE transfers finish sooner.

The newer model 3c509B has 8kB on board, and the buffer can be split 4/4, 5/3 or 6/2 for Rx/Tx. This setting is changed with the DOS configuration utility, and is stored on the EEPROM. This should alleviate the above problem with the original 3c509.

3c509B users should use either the supplied DOS utility to disable the *plug and play* support, *and* to set the output media to what they require. The linux driver currently does *not* support the Autodetect media setting, so you *have* to select 10Base-T or 10Base-2 or AUI. Note that to turn off PnP entirely, you should do a `3C5X9CFG /PNP:DISABLE` and then follow that with a hard reset to ensure that it takes effect.

Some people ask about the ``Server or Workstation'' and ``Highest Modem Speed'' settings presented in the DOS configuration utility. Donald writes ``These are only hints to the drivers, and the Linux driver

TOC

does not use these parameters: it always optimizes for high throughput rather than low latency (`Serv
Low latency was critically important for old, non-windowed, IPX throughput. To reduce the latency
MS-DOS driver for the 3c509 disables interrupts for some operations, blocking serial port interrupts.
Thus the need for the `modem speed' setting. The Linux driver avoids the need to disable interrupts for long periods by operating only on whole packets e.g. by not starting to transmit a packet until it is completely transferred to the card."

Note that the ISA card detection uses a different method than most cards. Basically, you ask the cards to respond by sending data to an ID_PORT (port `0x100` to `0x1ff` on intervals of `0x10`). This detection method means that a particular card will *always* get detected first in a multiple ISA 3c509 configuration. The card with the lowest hardware ethernet address will *always* end up being `eth0`. This shouldn't matter to anyone, except for those people who want to assign a 6 byte hardware address to a particular interface. If you have multiple 3c509 cards, it is best to append `ether=0,0,ethN` commands without the I/O port specified (i.e. use I/O=zero) and allow the probe to sort out which card is first. Using a non-zero I/O value will ensure that it does not detect all your cards, so don't do it.

If this really bothers you, have a look at Donald's latest driver, as you may be able to use a `0x3c509` value in the unused mem address fields to order the detection to suit your needs.

3c515

Status: Supported, Driver Name: 3c515

This is 3Com's ISA 100Mbps offering, codenamed ``CorkScrew''. A relatively new driver from Donald for these cards is included in the v2.2 kernels. For the most up to date information, you should probably look on the Vortex page:

Vortex

3c523

Status: Semi-Supported, Driver Name: 3c523

This MCA bus card uses the i82586, and Chris Beauregard has modified the ni52 driver to work with these cards. The driver for it can be found in the v2.2 kernel source tree.

More details can be found on the MCA-Linux page at
`http://glycerine.cetmm.uni.edu/mca/`

3c527

Status: Not Supported.

Yes, another MCA card. No, not too much interest in it. Better chances with the 3c529 if you are stuck with MCA.

3c529

Status: Supported, Driver Name: 3c509

This card actually uses the same chipset as the 3c509. Donald actually put hooks into the 3c509 driver to check for MCA cards after probing for EISA cards, and before probing for ISA cards, long before MCA support was added to the kernel. The required MCA probe code is included in the driver shipped with v2.2 kernels. More details can be found on the MCA-Linux page at:

`http://glycerine.cetmm.uni.edu/mca/`

3c562

Status: Supported, Driver Name: 3c589 (distributed separately)

This PCMCIA card is the combination of a 3c589B ethernet card with a modem. The modem appears as a standard modem to the end user. The only difficulty is getting the two separate linux drivers to share one interrupt. There are a couple of new registers and some hardware interrupt sharing support. You need to use a v2.0 or newer kernel that has the support for interrupt sharing.

Thanks again to Cameron for getting a sample unit and documentation sent off to David Hinds. Look for support in David's PCMCIA package release.

See PCMCIA Support for more info on PCMCIA chipsets, socket enablers, etc.

3c575

Status: Unknown.

A driver for this PCMCIA card is under development and hopefully will be included in David's PCMCIA package in the future. Best to check the PCMCIA package to get the current status.

3c579

Status: Supported, Driver Name: 3c509

The EISA version of the 509. The current EISA version uses the same 16 bit wide chip rather than a 32 bit interface, so the performance increase isn't stunning. Make sure the card is configured for EISA addressing mode. Read the above 3c509 section for info on the driver.

3c589 / 3c589B

Status: Semi-Supported, Driver Name: 3c589

Many people have been using this PCMCIA card for quite some time now. Note that support for it is not (at present) included in the default kernel source tree. The "B" in the name means the same here as it does for the 3c509 case.

There are drivers available on Donald's ftp site and in David Hinds PCMCIA package. You will also

need a supported PCMCIA controller chipset. See PCMCIA Support for more info on PCMCIA driv chipsets, socket enablers, etc.

3c590 / 3c595

Status: Supported, Driver Name: 3c59x

These ``Vortex'' cards are for PCI bus machines, with the '590 being 10Mbps and the '595 being 3Com's 100Mbs offering. Also note that you can run the '595 as a '590 (i.e. in a 10Mbps mode). The driver is included in the v2.0 kernel source, but is also continually being updated. If you have problems with the driver in the v2.0 kernel, you can get an updated driver from the following URL:

Vortex

Note that there are two different 3c590 cards out there, early models that had 32kB of on-board memory, and later models that only have 8kB of memory. Chances are you won't be able to buy a new 3c59x for much longer, as it is being replaced with the 3c90x card. If you are buying a used one off somebody, try and get the 32kB version. The 3c595 cards have 64kB, as you can't get away with only 8kB RAM at 100Mbps!

A thanks to Cameron Spitzer and Terry Murphy of 3Com for sending cards and documentation to Donald so he could write the driver.

Donald has set up a mailing list for Vortex driver support. To join the list, just do:

```
echo subscribe | /bin/mail linux-vortex-request@cesdis.gsfc.nasa.gov
```

3c592 / 3c597

Status: Supported, Driver Name: 3c59x

These are the EISA versions of the 3c59x series of cards. The 3c592/3c597 (aka Demon) should work with the vortex driver discussed above.

3c900 / 3c905 / 3c905B

Status: Supported, Driver Name: 3c59x

These cards (aka `Boomerang', aka EtherLink III XL) have been released to take over the place of the 3c590/3c595 cards.

The support for the Cyclone `B' revision was only recently added. To use this card with older v2.0 kernels, you must obtain the updated `3c59x.c` driver from Donald's site at:

Vortex-Page

If in doubt about anything then check out the above WWW page. Donald has set up a mailing list for Vortex driver support announcements and etc. To join the list, just do:

```
echo subscribe | /bin/mail linux-vortex-request@cesdis.gsfc.nasa.gov
```

TOC

3c985

Status: Supported, Driver Name: acenic

This driver, by Jes Sorensen, is available in v2.2 kernels It supports several other Gigabit cards in addition to the 3Com model.

5.2 Accton

Accton MPX

Status: Supported, Driver Name: ne (+8390)

Don't let the name fool you. This is still supposed to be a NE2000 compatible card, and should work with the ne2000 driver.

Accton EN1203, EN1207, EtherDuo-PCI

Status: Supported, Driver Name: de4x5, tulip

This is another implementation of the DEC 21040 PCI chip. The EN1207 card has the 21140, and also has a 10Base-2 connector, which has proved troublesome for some people in terms of selecting that media. Using the card with 10Base-T and 100Base-T media have worked for others though. So as with all purchases, you should try and make sure you can return it if it doesn't work for you.

See DEC 21040 for more information on these cards, and the present driver situation.

Accton EN2209 Parallel Port Adaptor (EtherPocket)

Status: Semi-Supported, Driver Name: ?

A driver for these parallel port adapters is available but not yet part of the 2.0 or 2.1 kernel source. You have to get the driver from:

`http://www.unix-ag.uni-siegen.de/~nils/accton_linux.html`

Accton EN2212 PCMCIA Card

Status: Semi-Supported, Driver Name: ?

David Hinds has been working on a driver for this card, and you are best to check the latest release of his PCMCIA package to see what the present status is.

5.3 Allied Telesyn/Telesis

AT1500

Status: Supported, Driver Name: lance

These are a series of low-cost ethercards using the 79C960 version of the AMD LANCE. These are bus-master cards, and hence one of the faster ISA bus ethercards available.

DMA selection and chip numbering information can be found in AMD LANCE.

More technical information on AMD LANCE based Ethernet cards can be found in Notes on AMD...

AT1700

Status: Supported, Driver Name: at1700

Note that to access this driver during `make config` you still have to answer `Y' when asked ``Prompt for development and/or incomplete code/drivers?'' at the first. This is simply due to lack of feedback on the driver stability due to it being a relatively rare card. If you have problems with the driver that ships with the kernel then you may be interested in the alternative driver available at: `http://www.cc.hit-u.ac.jp/nagoya/at1700/`

The Allied Telesis AT1700 series ethercards are based on the Fujitsu MB86965. This chip uses a programmed I/O interface, and a pair of fixed-size transmit buffers. This allows small groups of packets to be sent back-to-back, with a short pause while switching buffers.

A unique feature is the ability to drive 150ohm STP (Shielded Twisted Pair) cable commonly installed for Token Ring, in addition to 10baseT 100ohm UTP (unshielded twisted pair). A fibre optic version of the card (AT1700FT) exists as well.

The Fujitsu chip used on the AT1700 has a design flaw: it can only be fully reset by doing a power cycle of the machine. Pressing the reset button doesn't reset the bus interface. This wouldn't be so bad, except that it can only be reliably detected when it has been freshly reset. The solution/work-around is to power-cycle the machine if the kernel has a problem detecting the AT1700.

AT2450

Status: Supported, Driver Name: pcnet32

This is the PCI version of the AT1500, and it doesn't suffer from the problems that the Boca 79c970 PCI card does. DMA selection and chip numbering information can be found in AMD LANCE.

More technical information on AMD LANCE based Ethernet cards can be found in Notes on AMD...

AT2500

Status: Semi-Supported, Driver Name: rtl8139

This card uses the RealTek 8139 chip - see the section RealTek 8139.

AT2540FX

Status: Semi-Supported, Driver Name: eepro100

This card uses the i82557 chip, and hence may/should work with the eepro100 driver. If you try this please send in a report so this information can be updated.

5.4 AMD / Advanced Micro Devices

Carl Ching of AMD was kind enough to provide a very detailed description of all the relevant AMD ethernet products which helped clear up this section.

AMD LANCE (7990, 79C960/961/961A, PCnet-ISA)

Status: Supported, Driver Name: lance

There really is no AMD ethernet card. You are probably reading this because the only markings you could find on your card said AMD and the above number. The 7990 is the original `LANCE' chip, but most stuff (including this document) refer to all these similar chips as `LANCE' chips. (...incorrectly, I might add.)

These above numbers refer to chips from AMD that are the heart of many ethernet cards. For example, the Allied Telesis AT1500 (see AT1500) and the NE1500/2100 (see NE1500) use these chips.

The 7990/79c90 have long been replaced by newer versions. The 79C960 (a.k.a. PCnet-ISA) essentially contains the 79c90 core, along with all the other hardware support required, which allows a single-chip ethernet solution. The 79c961 (PCnet-ISA+) is a jumperless Plug and Play version of the '960. The final chip in the ISA series is the 79c961A (PCnet-ISA II), which adds full duplex capabilities. All cards with one of these chips should work with the lance.c driver, with the exception of very old cards that used the original 7990 in a shared memory configuration. These old cards can be spotted by the lack of jumpers for a DMA channel.

One common problem people have is the `busmaster arbitration failure' message. This is printed out when the LANCE driver can't get access to the bus after a reasonable amount of time has elapsed (50us). This usually indicates that the motherboard implementation of bus-mastering DMA is broken, or some other device is hogging the bus, or there is a DMA channel conflict. If your BIOS setup has the `GAT option' (for Guaranteed Access Time) then try toggling/altering that setting to see if it helps.

Also note that the driver only looks at the addresses: `0x300, 0x320, 0x340, 0x360` for a valid card, and any address supplied by an `ether=` boot argument is silently ignored (this will be fixed) so make sure your card is configured for one of the above I/O addresses for now.

The driver will still work fine, even if more than 16MB of memory is installed, since low-memory `bounce-buffers' are used when needed (i.e. any data from above 16MB is copied into a buffer below 16MB before being given to the card to transmit.)

The DMA channel can be set with the low bits of the otherwise-unused dev->mem_start value (a.k.a. PARAM_1). (see PARAM_1) If unset it is probed for by enabling each free DMA channel in turn and checking if initialization succeeds.

The HP-J2405A board is an exception: with this board it's easy to read the EEPROM-set values for the IRQ, and DMA.

See Notes on AMD... for more info on these chips.

AMD 79C965 (PCnet-32)

Status: Supported, Driver Name: pcnet32

This is the PCnet-32 -- a 32 bit bus-master version of the original LANCE chip for VL-bus and local bus systems. chip. While these chips can be operated with the standard `lance.c` driver, a 32 bit version (`pcnet32.c`) is also available that does not have to concern itself with any 16MB limitations associated with the ISA bus.

AMD 79C970/970A (PCnet-PCI)

Status: Supported, Driver Name: pcnet32

This is the PCnet-PCI -- similar to the PCnet-32, but designed for PCI bus based systems. Please see the above PCnet-32 information. This means that you need to build a kernel with PCI BIOS support enabled. The '970A adds full duplex support along with some other features to the original '970 design.

Note that the Boca implementation of the 79C970 fails on fast Pentium machines. This is a hardware problem, as it affects DOS users as well. See the Boca section for more details.

AMD 79C971 (PCnet-FAST)

Status: Supported, Driver Name: pcnet32

This is AMD's 100Mbit chip for PCI systems, which also supports full duplex operation. It was introduced in June 1996.

AMD 79C972 (PCnet-FAST+)

Status: Unknown, Driver Name: pcnet32

This should also work just like the '971 but this has yet to be confirmed.

AMD 79C974 (PCnet-SCSI)

Status: Supported, Driver Name: pcnet32

This is the PCnet-SCSI -- which is basically treated like a '970 from an Ethernet point of view. Also see the above information. Don't ask if the SCSI half of the chip is supported -- this is the *Ethernet-HowTo*, not the SCSI-HowTo.

5.5 Ansel Communications

AC3200 EISA

Status: Semi-Supported, Driver Name: ac3200

Note that to access this driver during `make config` you still have to answer `Y' when asked ``Prompt for development and/or incomplete code/drivers?'' at the first. This is simply due to lack of feedback on the driver stability due to it being a relatively rare card.

This driver is included in the present kernel as an alpha test driver. It is based on the common NS8390 chip used in the ne2000 and wd80x3 cards. Please see Alpha Drivers in this document for important information regarding alpha drivers.

If you use it, let one of us know how things work out, as feedback has been low, even though the driver has been in the kernel since v1.1.25.

If you intend on using this driver as a loadable module you should probably see Using the Ethernet Drivers as Modules for module specific information.

5.6 Apricot

Apricot Xen-II On Board Ethernet

Status: Semi-Supported, Driver Name: apricot

This on board ethernet uses an i82596 bus-master chip. It can only be at I/O address `0x300`. By looking at the driver source, it appears that the IRQ is also hardwired to 10.

Earlier versions of the driver had a tendency to think that anything living at `0x300` was an apricot NIC. Since then the hardware address is checked to avoid these false detections.

5.7 Arcnet

Status: Supported, Driver Name: arcnet (arc-rimi, com90xx, com20020)

With the very low cost and better performance of ethernet, chances are that most places will be giving away their Arcnet hardware for free, resulting in a lot of home systems with Arcnet.

An advantage of Arcnet is that all of the cards have identical interfaces, so one driver will work for everyone. It also has built in error handling so that it supposedly never loses a packet. (Great for UDP traffic!)

Avery Pennarun's arcnet driver has been in the default kernel sources since 1.1.80. The arcnet driver uses `arc0' as its name instead of the usual `eth0' for ethernet devices. Bug reports and success stories can be mailed to:

`apenwarr@foxnet.net`

There are information files contained in the standard kernel for setting jumpers and general hints.

Supposedly the driver also works with the 100Mbs ARCnet cards as well!

5.8 AT&T

Note that AT&T's StarLAN is an orphaned technology, like SynOptics LattisNet, and can't be used in a standard 10Base-T environment, without a hub that `speaks' both.

AT&T T7231 (LanPACER+)

Status: Not Supported.

These StarLAN cards use an interface similar to the i82586 chip. At one point, Matthijs Melchior (`matthijs.n.melchior@att.com`) was playing with the 3c507 driver, and almost had something useable working. Haven't heard much since that.

5.9 Boca Research

Yes, they make more than just multi-port serial cards. :-)

Boca BEN (ISA, VLB, PCI)

Status: Supported, Driver Name: lance, pcnet32

These cards are based on AMD's PCnet chips. Perspective buyers should be warned that many users have had endless problems with these VLB/PCI cards. Owners of fast Pentium systems have been especially hit. Note that this is not a driver problem, as it hits DOS/Win/NT users as well. Boca's technical support number is (407) 241-8088, and you can also reach them at `75300.2672@compuserve.com`. The older ISA cards don't appear to suffer the same problems.

Donald did a comparitive test with a Boca PCI card and a similar Allied Telsyn PCnet/PCI implementation, which showed that the problem lies in Boca's implementation of the PCnet/PCI chip. These test results can be accessed on Don's www server.

Linux at CESDIS

Boca is offering a `warranty repair' for affected owners, which involves adding one of the missing capacitors, but it appears that this fix doesn't work 100 percent for most people, although it helps so

If you are *still* thinking of buying one of these cards, then at least try and get a 7 day unconditional return policy, so that if it doesn't work properly in your system, you can return it.

More general information on the AMD chips can be found in AMD LANCE.

More technical information on AMD LANCE based Ethernet cards can be found in Notes on AMD...

5.10 Cabletron

Donald writes: `Yes, another one of these companies that won't release its programming information. They waited for months before actually confirming that all their information was proprietary, deliberately wasting my time. Avoid their cards like the plague if you can. Also note that some people have phoned Cabletron, and have been told things like `a D. Becker is working on a driver for linux' -- making it sound like I work for them. This is NOT the case.'

Apparently Cabletron has changed their policy with respect to programming information (like Xircom) since Donald made the above comment several years ago -- send e-mail to `support@ctron.com` if you want to verify this or ask for programming information. However, at this point in time, there is little demand for modified/updated drivers for the older E20xx and E21xx cards.

E10**, E10**-x, E20**, E20**-x

Status: Semi-Supported, Driver Name: ne (+8390)

These are NEx000 almost-clones that are reported to work with the standard NEx000 drivers, thanks to a ctron-specific check during the probe. If there are any problems, they are unlikely to be fixed, as the programming information is unavailable.

E2100

Status: Semi-Supported, Driver Name: e2100 (+8390)

Again, there is not much one can do when the programming information is proprietary. The E2100 is a poor design. Whenever it maps its shared memory in during a packet transfer, it maps it into the *whole 128K region!* That means you **can't** safely use another interrupt-driven shared memory device in that region, including another E2100. It will work most of the time, but every once in a while it will bite you. (Yes, this problem can be avoided by turning off interrupts while transferring packets, but that will almost certainly lose clock ticks.) Also, if you mis-program the board, or halt the machine at just the wrong moment, even the reset button won't bring it back. You will *have* to turn it off and *leave* it off for about 30 seconds.

Media selection is automatic, but you can override this with the low bits of the dev->mem_end parameter. See PARAM_2. Module users can specify an `xcvr=N` value as an `option` in the `/etc/conf.modules` file.

Also, don't confuse the E2100 for a NE2100 clone. The E2100 is a shared memory NatSemi DP8390 design, roughly similar to a brain-damaged WD8013, whereas the NE2100 (and NE1500) use a bus-mastering AMD LANCE design.

There is an E2100 driver included in the standard kernel. However, seeing as programming info isn't available, don't expect bug-fixes. Don't use one unless you are already stuck with the card.

If you intend on using this driver as a loadable module you should probably see Using the Ethernet Drivers as Modules for module specific information.

E22**

Status: Semi-Supported, Driver Name: lance

According to information in a Cabletron Tech Bulletin, these cards use the standard AMD PC-Net chipset (see AMD PC-Net) and should work with the generic lance driver.

5.11 Cogent

Here is where and how to reach them:

```
Cogent Data Technologies, Inc.
175 West Street, P.O. Box 926
Friday Harbour, WA 98250, USA.

Cogent Sales
15375 S.E. 30th Place, Suite 310
Bellevue, WA 98007, USA.

Technical Support:
Phone (360) 378-2929 between 8am and 5pm PST
Fax (360) 378-2882
Compuserve GO COGENT
Bulletin Board Service (360) 378-5405
Internet: support@cogentdata.com
```

EM100-ISA/EISA

Status: Semi-Supported, Driver Name: smc9194

These cards use the SMC 91c100 chip and may work with the SMC 91c92 driver, but this has yet to be verified.

Cogent eMASTER+, EM100-PCI, EM400, EM960, EM964

TOC

Status: Supported, Driver Name: de4x5, tulip

These are yet another DEC 21040 implementation that should hopefully work fine with the standard 21040 driver.

The EM400 and the EM964 are four port cards using a DEC 21050 bridge and 4 21040 chips.

See DEC 21040 for more information on these cards, and the present driver situation.

5.12 Compaq

Compaq aren't really in the business of making ethernet cards, but a lot of their systems have embedded ethernet controllers on the motherboard.

Compaq Deskpro / Compaq XL (Embedded AMD Chip)

Status: Supported, Driver Name: pcnet32

Machines such as the XL series have an AMD 79c97x PCI chip on the mainboard that can be used with the standard LANCE driver. But before you can use it, you have to do some trickery to get the PCI BIOS to a place where Linux can see it. Frank Maas was kind enough to provide the details:

`` The problem with this Compaq machine however is that the PCI directory is loaded in high memory, at a spot where the Linux kernel can't (won't) reach. Result: the card is never detected nor is it usable (sideline: the mouse won't work either) The workaround (as described thoroughly in http://www-c724.uibk.ac.at/XL/) is to load MS-DOS, launch a little driver Compaq wrote and then load the Linux kernel using LOADLIN. Ok, I'll give you time to say `yuck, yuck', but for now this is the only working solution I know of. The little driver simply moves the PCI directory to a place where it is normally stored (and where Linux can find it)."

More general information on the AMD chips can be found in AMD LANCE.

Compaq Nettelligent/NetFlex (Embedded ThunderLAN Chip)

Status: Supported, Driver Name: tlan

These systems use a Texas Instruments ThunderLAN chip Information on the ThunderLAN driver can be found in ThunderLAN.

5.13 Danpex

Danpex EN9400

Status: Supported, Driver Name: de4x5, tulip

Yet another card based on the DEC 21040 chip, reported to work fine, and at a relatively cheap price.

See DEC 21040 for more information on these cards, and the present driver situation.

5.14 D-Link

DE-100, DE-200, DE-220-T, DE-250

Status: Supported, Driver Name: ne (+8390)

Some of the early D-Link cards didn't have the `0x57` PROM signature, but the ne2000 driver knows about them. For the software configurable cards, you can get the config program from `www.dlink.com`. The DE2** cards were the most widely reported as having the spurious transfer address mismatch errors with early versions of linux. Note that there are also cards from Digital (DEC) that are also named DE100 and DE200, but the similarity stops there.

DE-520

Status: Supported, Driver Name: pcnet32

This is a PCI card using the PCI version of AMD's LANCE chip. DMA selection and chip numbering information can be found in AMD LANCE.

More technical information on AMD LANCE based Ethernet cards can be found in Notes on AMD....

DE-528

Status: Supported, Driver Name: ne, ne2k-pci (+8390)

Apparently D-Link have also started making PCI NE2000 clones.

DE-530

Status: Supported, Driver Name: de4x5, tulip

This is a generic DEC 21040 PCI chip implementation, and is reported to work with the generic 21040 tulip driver.

See DEC 21040 for more information on these cards, and the present driver situation.

DE-600

Status: Supported, Driver Name: de600

Laptop users and other folk who might want a quick way to put their computer onto the ethernet may want to use this. The driver is included with the default kernel source tree. Bjorn Ekwall `bj0rn@blox.se` wrote the driver. Expect about 180kb/s transfer speed from this via the parallel port. You should read the README.DLINK file in the kernel source tree.

Note that the device name that you pass to `ifconfig` is *now* `eth0` and not the previously used `dl0`.

If your parallel port is *not* at the standard `0x378` then you will have to recompile. Bjorn writes: ``Since the DE-620 driver tries to sqeeze the last microsecond from the loops, I made the irq and port address constants instead of variables. This makes for a usable speed, but it also means that you can't change these assignements from e.g. lilo; you _have_ to recompile..." Also note that some laptops implement the on-board parallel port at `0x3bc` which is where the parallel ports on monochrome cards were/are.

DE-620

Status: Supported, Driver Name: de620

Same as the DE-600, only with two output formats. Bjorn has written a driver for this model, for kernel versions 1.1 and above. See the above information on the DE-600.

DE-650

Status: Semi-Supported, Driver Name: de650 (?)

Some people have been using this PCMCIA card for some time now with their notebooks. It is a basic 8390 design, much like a NE2000. The LinkSys PCMCIA card and the IC-Card Ethernet are supposedly DE-650 clones as well. Note that at present, this driver is *not* part of the standard kernel, and so you will have to do some patching.

See PCMCIA Support in this document, and if you can, have a look at:

Don's PCMCIA Stuff

5.15 DFI

DFINET-300 and DFINET-400

Status: Supported, Driver Name: ne (+8390)

These cards are now detected (as of 0.99pl15) thanks to Eberhard Moenkeberg `emoenke@gwdg.de` who noted that they use `DFI' in the first 3 bytes of the prom, instead of using `0x57` in bytes 14 and 15, which is what all the NE1000 and NE2000 cards use. (The 300 is an 8 bit pseudo NE1000 clone, and the 400 is a pseudo NE2000 clone.)

5.16 Digital / DEC

DEPCA, DE100/1, DE200/1/2, DE210, DE422

Status: Supported, Driver Name: depca

There is documentation included in the source file `depca.c', which includes info on how to use more than one of these cards in a machine. Note that the DE422 is an EISA card. These cards are all based on the AMD LANCE chip. See AMD LANCE for more info. A maximum of two of the ISA cards can be used, because they can only be set for `0x300` and `0x200` base I/O address. If you are intending to do this, please read the notes in the driver source file `depca.c` in the standard kernel source tree.

This driver will also work on Alpha CPU based machines, and there are various ioctl()s that the user can play with.

Digital EtherWorks 3 (DE203, DE204, DE205)

Status: Supported, Driver Name: ewrk3

These cards use a proprietary chip from DEC, as opposed to the LANCE chip used in the earlier cards like the DE200. These cards support both shared memory or programmed I/O, although you take about a 50%performance hit if you use PIO mode. The shared memory size can be set to 2kB, 32kB or 64kB, but only 2 and 32 have been tested with this driver. David says that the performance is virtually identical between the 2kB and 32kB mode. There is more information (including using the driver as a loadable module) at the top of the driver file `ewrk3.c` and also in `README.ewrk3`. Both of these files come with the standard kernel distribution. This driver has Alpha CPU support like depca.c does.

The standard driver has a number of interesting ioctl() calls that can be used to get or clear packet statistics, read/write the EEPROM, change the hardware address, and the like. Hackers can see the source code for more info on that one.

David has also written a configuration utility for this card (along the lines of the DOS program `NICSETUP.EXE`) along with other tools. These can be found on most Linux FTP sites in the directory `/pub/Linux/system/Network/management` -- look for the file `ewrk3tools-X.XX.tar.gz`.

DE425 EISA, DE434, DE435, DE500

Status: Supported, Driver Name: de4x5, tulip

These cards are based on the 21040 chip mentioned below. The DE500 uses the 21140 chip to provide 10/100Mbs ethernet connections. Have a read of the 21040 section below for extra info. There are also some compile-time options available for non-DEC cards using this driver. Have a look at `README.de4x5` for details.

All the Digital cards will autoprobe for their media (except, temporarily, the DE500 due to a patent issue).

This driver is also Alpha CPU ready and supports being loaded as a module. Users can access the driver internals through ioctl() calls - see the 'ewrk3' tools and the de4x5.c sources for information about how to do this.

DEC 21040, 21041, 2114x, Tulip

Status: Supported, Driver Name: de4x5, tulip

The DEC 21040 is a bus-mastering single chip ethernet solution from Digital, similar to AMD's PCnet chip. The 21040 is specifically designed for the PCI bus architecture. SMC's new EtherPower PCI card uses this chip.

You have a choice of *two* drivers for cards based on this chip. There is the DE425 driver discussed above, and the generic 21040 `tulip' driver.

Warning: Even though your card may be based upon this chip, *the drivers may not work for you*. David C. Davies writes:

``There are no guarantees that either `tulip.c' OR `de4x5.c' will run any DC2114x based card other than those they've been written to support. WHY?? You ask. Because there is a register, the General Purpose Register (CSR12) that (1) in the DC21140A is programmable by each vendor and they all do it differently (2) in the DC21142/3 this is now an SIA control register (a la DC21041). The only small ray of hope is that we can decode the SROM to help set up the driver. However, this is not a guaranteed solution since some vendors (e.g. SMC 9332 card) don't follow the Digital Semiconductor recommended SROM programming format."

In non-technical terms, this means that if you aren't sure that an unknown card with a DC2114x chip will work with the linux driver(s), then make sure you can return the card to the place of purchase *before* you pay for it.

The updated 21041 chip is also found in place of the 21040 on most of the later SMC EtherPower cards. The 21140 is for supporting 100Base-? and works with the Linux drivers for the 21040 chip. To use David's `de4x5` driver with non-DEC cards, have a look at `README.de4x5` for details.

Donald has used SMC EtherPower-10/100 cards to develop the `tulip' driver. Note that the driver that is in the standard kernel tree at the moment is not the most up to date version. If you are having trouble with this driver, you should get the newest version from Donald's ftp/WWW site.

Tulip Driver

The above URL also contains a (non-exhaustive) list of various cards/vendors that use the 21040 chip.

Also note that the tulip driver is still considered an *alpha* driver (see Alpha Drivers) at the moment, and should be treated as such. To use it, you will have to edit `arch/i386/config.in` and uncomment the line for `CONFIG_DEC_ELCP` support.

Donald has even set up a mailing list for tulip driver support announcements, etc. To join it just type:

```
echo subscribe | /bin/mail linux-tulip-request@cesdis.gsfc.nasa.gov
```

TOC

5.17 Farallon

Farallon sells EtherWave adaptors and transceivers. This device allows multiple 10baseT devices to be daisy-chained.

Farallon Etherwave

Status: Supported, Driver Name: 3c509

This is reported to be a 3c509 clone that includes the EtherWave transceiver. People have used these successfully with Linux and the present 3c509 driver. They are too expensive for general use, but are a great option for special cases. Hublet prices start at $125, and Etherwave adds $75-$100 to the price of the board -- worth it if you have pulled one wire too few, but not if you are two network drops short.

5.18 Fujitsu

Unlike many network chip manufacturers, Fujitsu have also made and sold some network cards based upon their chip.

Fujitsu FMV-181/182/183/184

Status: Supported, Driver Name: fmv18x

According to the driver, these cards are a straight forward Fujitsu MB86965 implementation, which would make them very similar to the Allied Telesis AT1700 cards.

5.19 Hewlett Packard

The 272** cards use programmed I/O, similar to the NE*000 boards, but the data transfer port can be `turned off' when you aren't accessing it, avoiding problems with autoprobing drivers.

Thanks to Glenn Talbott for helping clean up the confusion in this section regarding the version numbers of the HP hardware.

27245A

Status: Supported, Driver Name: hp (+8390)

8 Bit 8390 based 10BaseT, not recommended for all the 8 bit reasons. It was re-designed a couple years ago to be highly integrated which caused some changes in initialization timing which only affected testing programs, not LAN drivers. (The new card is not `ready' as soon after switching into and out of loopback mode.)

If you intend on using this driver as a loadable module you should probably see Using the Ethernet Drivers as Modules for module specific information.

TOC

HP EtherTwist, PC Lan+ (27247, 27252A)

Status: Supported, Driver Name: hp+ (+8390)

The HP PC Lan+ is different to the standard HP PC Lan card. This driver was added to the list of drivers in the standard kernel during the v1.1.x development cycle. It can be operated in either a PIO mode like a ne2000, or a shared memory mode like a wd8013.

The 47B is a 16 Bit 8390 based 10BaseT w/AUI, and the 52A is a 16 Bit 8390 based ThinLAN w/AUI. These cards have 32K onboard RAM for Tx/Rx packet buffering instead of the usual 16KB, and they both offer LAN connector autosense.

If you intend on using this driver as a loadable module you should probably see Using the Ethernet Drivers as Modules for module specific information.

HP-J2405A

Status: Supported, Driver Name: lance

These are lower priced, and slightly faster than the 27247/27252A, but are missing some features, such as AUI, ThinLAN connectivity, and boot PROM socket. This is a fairly generic LANCE design, but a minor design decision makes it incompatible with a generic `NE2100' driver. Special support for it (including reading the DMA channel from the board) is included thanks to information provided by HP's Glenn Talbott.

More technical information on LANCE based cards can be found in Notes on AMD...

HP-Vectra On Board Ethernet

Status: Supported, Driver Name: lance

The HP-Vectra has an AMD PCnet chip on the motherboard. DMA selection and chip numbering information can be found in AMD LANCE.

More technical information on LANCE based cards can be found in Notes on AMD...

HP 10/100 VG Any Lan Cards (27248B, J2573, J2577, J2585, J970, J973)

Status: Supported, Driver Name: hp100

This driver also supports some of the Compex VG products. Since the driver supports ISA, EISA and PCI cards, it is found under ISA cards when running `make config` on a kernel source.

HP NetServer 10/100TX PCI (D5013A)

Status: Supported, Driver Name: eepro100

Apparently these are just a rebadged Intel EtherExpress Pro 10/100B card. See the Intel section for more information.

5.20 IBM / International Business Machines

IBM Thinkpad 300

Status: Supported, Driver Name: znet

This is compatible with the Intel based Zenith Z-note. See Z-note for more info.

Supposedly this site has a comprehensive database of useful stuff for newer versions of the Thinkpad. I haven't checked it out myself yet.

Thinkpad-info

For those without a WWW browser handy, try `peipa.essex.ac.uk:/pub/tp750/`

IBM Credit Card Adaptor for Ethernet

Status: Semi-Supported, Driver Name: ? (distributed separately)

People have been using this PCMCIA card with Linux as well. Similar points apply, those being that you need a supported PCMCIA chipset on your notebook, and that you will have to patch the PCMCIA support into the standard kernel.

See PCMCIA Support in this document, and if you can, have a look at:

Don's PCMCIA Stuff

IBM Token Ring

Status: Semi-Supported, Driver Name: ibmtr

To support token ring requires more than only writing a device driver, it also requires writing the source routing routines for token ring. It is the source routing that would be the most time comsuming to write.

Peter De Schrijver has been spending some time on Token Ring lately. and has worked with IBM ISA and MCA token ring cards.

The present token ring code has been included into the first of the 1.3.x series kernels.

Peter says that it was originally tested on an MCA 16/4 Megabit Token Ring board, but it should work with other Tropic based boards.

TOC

5.21 ICL Ethernet Cards

ICL EtherTeam 16i/32

Status: Supported, Driver Name: eth16i

Mika Kuoppala (miku@pupu.elt.icl.fi) wrote this driver, and it was included into early 1.3.4x kernels. It uses the Fujitsu MB86965 chip that is also used on the at1700 cards.

5.22 Intel Ethernet Cards

Note that the naming of the various Intel cards is ambiguous and confusing at best. If in doubt, then check the `i8xxxx` number on the main chip on the card or for PCI cards, use the PCI information in the `/proc` directory and then compare that to the numbers listed here.

Ether Express

Status: Supported, Driver Name: eexpress

This card uses the intel i82586. Earlier versions of this driver (in v1.2 kernels) were classed as alpha-test, as it didn't work well for most people. The driver in the v2.0 kernel seems to work much better for those who have tried it, although the driver source still lists it as experimental and more problematic on faster machines.

The comments at the top of the driver source list some of the problems (and fixes!) associated with these cards. The slowdown hack of replacing all the `outb` with `outb_p` in the driver has been reported to avoid lockups for at least one user.

Ether Express PRO/10

Status: Supported, Driver Name: eepro

Bao Chau Ha has written a driver for these cards that has been included into early 1.3.x kernels. It may also work with some of the Compaq built-in ethernet systems that are based on the i82595 chip.

Ether Express PRO/10 PCI (EISA)

Status: Semi-Supported, Driver Name: ? (distributed separately)

John Stalba (stalba@ultranet.com) has written a driver for the PCI version. These cards use the PLX9036 PCI interface chip with the Intel i82596 LAN controller chip. If your card has the i82557 chip, then you *don't* have this card, but rather the version discussed next, and hence want the EEPro100 driver instead.

You can get the alpha driver for the PRO/10 PCI card, along with instructions on how to use it at:

EEPro10 Driver

If you have the EISA card, you will probably have to hack the driver a bit to account for the differen (PCI vs. EISA) detection mechanisms that are used in each case.

TOC

Ether Express PRO 10/100B

Status: Supported, Driver Name: eepro100

Note that this driver will *not* work with the older 100A cards. The chip numbers listed in the driver are i82557/i82558. For driver updates and/or driver support, have a look at:

EEPro-100B Page

To subscribe to the mailing list relating to this driver, do:

```
echo subscribe | /bin/mail linux-eepro100-request@cesdis.gsfc.nasa.gov
```

Apparently Donald had to sign a non-disclosure agreement that stated he could actually disclose the driver source code! How is that for sillyness on intel's part?

5.23 Kingston

Kingston make various cards, including NE2000+, AMD PCnet based cards, and DEC tulip based cards. Most of these cards should work fine with their respective driver. See Kingston Web Page

The KNE40 DEC 21041 tulip based card is reported to work fine with the generic tulip driver.

5.24 LinkSys

LinkSys make a handful of different NE2000 clones, some straight ISA cards, some ISA plug and play and some even ne2000-PCI clones based on one of the supported ne2000-PCI chipsets. There are just too many models to list here.

LinkSys are linux-friendly, with a linux specific WWW support page, and even have Linux printed on the boxes of some of their products. Have a look at:

```
http://www.linksys.com/support/solution/nos/linux.htm
```

LinkSys Etherfast 10/100 Cards.

Status: Supported, Driver Name: tulip

Note that with these cards there have been several `revisions' (i.e. different chipset used) all with the same card name. The 1st used the DEC chipset. The 2nd revision used the Lite-On PNIC 82c168 PCI Network Interface Controller, and support for this was merged into the standard tulip driver (as of version 0.83 and newer). More PNIC information is available at:

```
http://cesdis.gsfc.nasa.gov/linux/drivers/pnic.html
```

More information on the various versions of these cards can be found at the LinkSys WWW site mentioned above.

LinkSys Pocket Ethernet Adapter Plus (PEAEPP)

Status: Supported, Driver Name: de620

This is supposedly a DE-620 clone, and is reported to work well with that driver. See DE-620 for more information.

LinkSys PCMCIA Adaptor

Status: Supported, Driver Name: de650 (?)

This is supposed to be a re-badged DE-650. See DE-650 for more information.

5.25 Microdyne

Microdyne Exos 205T

Status: Semi-Supported, Driver Name: ?

Another i82586 based card. Dirk Niggemann `dirk-n@dircon.co.uk` has written a driver that he classes as ``pre-alpha'' that he would like people to test. Mail him for more details.

5.26 Mylex

Mylex can be reached at the following numbers, in case anyone wants to ask them anything.

```
MYLEX CORPORATION, Fremont
Sales:  800-77-MYLEX, (510) 796-6100
FAX:    (510) 745-8016.
```

They also have a web site: Mylex WWW Site

Mylex LNE390A, LNE390B

Status: Supported, Driver Name: lne390 (+8390)

These are fairly old EISA cards that make use of a shared memory implementation similar to the wd80x3. A driver for these cards is available in the current 2.1.x series of kernels. Ensure you set the shared memory address below 1MB or above the highest address of the physical RAM installed in the machine.

Mylex LNP101

Status: Supported, Driver Name: de4x5, tulip

This is a PCI card that is based on DEC's 21040 chip. It is selectable between 10BaseT, 10Base2 and 10Base5 output. The LNP101 card has been verified to work with the generic 21040 driver.

See the section on the 21040 chip (DEC 21040) for more information.

Mylex LNP104

Status: Semi-Supported, Driver Name: de4x5, tulip

The LNP104 uses the DEC 21050 chip to deliver *four* independent 10BaseT ports. It should work with recent 21040 drivers that know how to share IRQs, but nobody has reported trying it yet (that I am aware of).

5.27 Novell Ethernet, NExxxx and associated clones.

The prefix `NE' came from Novell Ethernet. Novell followed the cheapest NatSemi databook design and sold the manufacturing rights (spun off?) Eagle, just to get reasonably-priced ethercards into the market. (The now ubiquitous NE2000 card.)

NE1000, NE2000

Status: Supported, Driver Name: ne (+8390)

The ne2000 is now a generic name for a bare-bones design around the NatSemi 8390 chip. They use programmed I/O rather than shared memory, leading to easier installation but slightly lower performance and a few problems. Some of the more common problems that arise with NE2000 cards are listed in Problems with...

Some NE2000 clones use the National Semiconductor `AT/LANTic' 83905 chip, which offers a shared memory mode similar to the wd8013 and EEPROM software configuration. The shared memory mode will offer less CPU usage (i.e. more efficient) than the programmed I/O mode.

In general it is not a good idea to put a NE2000 clone at I/O address `0x300` because nearly *every* device driver probes there at boot. Some poor NE2000 clones don't take kindly to being prodded in the wrong areas, and will respond by locking your machine. Also `0x320` is bad because SCSI drivers probe into `0x330`.

Donald has written a NE2000 diagnostic program (ne2k.c) for all ne2000 cards. See Diagnostic Programs for more information.

If you intend on using this driver as a loadable module you should probably see Using the Ethernet

Drivers as Modules for module specific information.

TOC

NE2000-PCI (RealTek/Winbond/Compex)

Status: Supported, Driver Name: ne, ne2k-pci (+8390)

Yes, believe it or not, people are making PCI cards based on the more than ten year old interface design of the ne2000. At the moment nearly all of these cards are based on the RealTek 8029 chip, or the Winbond 89c940 chip. The Compex, KTI, VIA and Netvin cards apparently also use these chips, but have a different PCI ID.

The latest v2.0 kernel has support to automatically detect all these cards and use them. (If you are using a kernel v2.0.34 or older, you should upgrade to ensure your card will be detected.) There are now two drivers to choose from; the original ISA/PCI `ne.c` driver, and a relatively new PCI-only `ne2k-pci.c` driver.

To use the original ISA/PCI driver you have to say `Y' to the `Other ISA cards' option when running `make config` as you are actually using the same NE2000 driver as the ISA cards use. (That should also give you a hint that these cards aren't anywhere as intelligent as say a PCNet-PCI or DEC 21040 card...)

The newer PCI-only driver differs from the ISA/PCI driver in that all the support for old NE1000 8 bit cards has been removed and that data is moved to/from the card in bigger blocks, without any intervening pauses that the older ISA-NE2000's required for reliable operation. The result is a driver that is slightly smaller and slightly more efficient, but don't get too excited as the difference will not be obvious under normal use. (If you really wanted maximum efficiency/low CPU use, then a PCI-NE2000 is simply a very poor choice.) Driver updates and more information can be found at:

`http://cesdis.gsfc.nasa.gov/linux/drivers/ne2k-pci.html`

If you have a NE2000 PCI card that is *not* detected by the most current version of the driver, please contact the maintainer of the NE2000 driver as listed in `/usr/src/linux/MAINTAINERS` along with the output from a `cat /proc/pci` and `dmesg` so that support for your card can also be added to the driver.

Also note that various card makers have been known to put `NE2000 Compatible' stickers on their product boxes even when it is completely different (e.g. PCNet-PCI or RealTek 8139). If in doubt check the main chip number against this document.

NE-10/100

Status: Not Supported.

These are ISA 100Mbps cards based on the National Semiconductor DP83800 and DP83840 chips. There is currently no driver support, nor has anyone reported that they are working on a driver. Apparently documentation on the chip is unavailable with the exception of a single PDF file that doesn't give enough details for a driver.

TOC

NE1500, NE2100

Status: Supported, Driver Name: lance

These cards use the original 7990 LANCE chip from AMD and are supported using the Linux lance driver. Newer NE2100 clones use the updated PCnet/ISA chip from AMD.

Some earlier versions of the lance driver had problems with getting the IRQ line via autoIRQ from the original Novell/Eagle 7990 cards. Hopefully this is now fixed. If not, then specify the IRQ via LILO, and let us know that it still has problems.

DMA selection and chip numbering information can be found in AMD LANCE.

More technical information on LANCE based cards can be found in Notes on AMD...

NE/2 MCA

Status: Semi-Supported, Driver Name: ne2

There were a few NE2000 microchannel cards made by various companies. This driver, available in v2.2 kernels, will detect the following MCA cards: Novell Ethernet Adapter NE/2, Compex ENET-16 MC/P, and the Arco Ethernet Adapter AE/2.

NE3200

Status: Not Supported.

This old EISA card uses a 8MHz 80186 in conjunction with an i82586. Nobody is working on a driver for it, as there is no information available on the card, and no real demand for a driver either.

NE3210

Status: Supported, Driver Name: ne3210 (+8390)

This EISA card is completely different from the NE3200, as it uses a Nat Semi 8390 chip. The driver can be found in the v2.2 kernel source tree. Ensure you set the shared memory address below 1MB or above the highest address of the physical RAM installed in the machine.

NE5500

Status: Supported, Driver Name: pcnet32

These are just AMD PCnet-PCI cards ('970A) chips. More information on LANCE/PCnet based cards can be found in AMD LANCE.

5.28 Proteon

Proteon P1370-EA

Status: Supported, Driver Name: ne (+8390)

Apparently this is a NE2000 clone, and works fine with Linux.

Proteon P1670-EA

Status: Supported, Driver Name: de4x5, tulip

This is yet another PCI card that is based on DEC's Tulip chip. It has been reported to work fine with Linux.

See the section on the 21040 chip (DEC 21040) for more driver information.

5.29 Pure Data

PDUC8028, PDI8023

Status: Supported, Driver Name: wd (+8390)

The PureData PDUC8028 and PDI8023 series of cards are reported to work, thanks to special probe code contributed by Mike Jagdis `jaggy@purplet.demon.co.uk`. The support is integrated with the WD driver.

5.30 Racal-Interlan

Racal Interlan can be reached via WWW at `www.interlan.com`. I believe they were also known as MiCom-Interlan at one point in the past.

ES3210

Status: Semi-Supported, Driver Name: es3210

This is an EISA 8390 based shared memory card. An experimetal driver is shipped with v2.2 kernels and it is reported to work fine, but the EISA IRQ and shared memory address detection appears not to work with (at least) the early revision cards. (This problem is not unique to the Linux world either...) In that case, you have to supply them to the driver. For example, card at IRQ 5 and shared memory `0xd0000`, with a modular driver, add `options es3210 irq=5 mem=0xd0000` to `/etc/conf.modules`. Or with the driver compiled into the kernel, supply at boot `ether=5,0,0xd0000,eth0` The I/O base is automatically detected and hence a value of zero should be used.

TOC

NI5010

Status: Semi-Supported, Driver Name: ni5010

You used to have to go get the driver for these old 8 bit MiCom-Interlan cards separately, but now it is shipped with the v2.2 kernels as an experimental driver.

NI5210

Status: Semi-Supported, Driver Name: ni52

This card also uses one of the Intel chips. Michael Hipp has written a driver for this card. It is included in the standard kernel as an `alpha' driver. Michael would like to hear feedback from users that have this card. See Alpha Drivers for important information on using alpha-test ethernet drivers with Linux.

NI6510 (not EB)

Status: Semi-Supported, Driver Name: ni65

There is also a driver for the LANCE based NI6510, and it is also written by Michael Hipp. Again, it is also an `alpha' driver. For some reason, this card is not compatible with the generic LANCE driver. See Alpha Drivers for important information on using alpha-test ethernet drivers with Linux.

EtherBlaster (aka NI6510EB)

Status: Supported, Driver Name: lance

As of kernel 1.3.23, the generic LANCE driver had a check added to it for the `0x52, 0x44` NI6510EB specific signature. Others have reported that this signature is not the same for all NI6510EB cards however, which will cause the lance driver to not detect your card. If this happens to you, you can change the probe (at about line 322 in lance.c) to printk() out what the values are for your card and then use them instead of the `0x52, 0x44` defaults.

The cards should probably be run in `high-performance' mode and not in the NI6510 compatible mode when using the lance driver.

5.31 RealTek

RealTek RTL8002/8012 (AT-Lan-Tec) Pocket adaptor

Status: Supported, Driver Name: atp

This is a generic, low-cost OEM pocket adaptor being sold by AT-Lan-Tec, and (likely) a number of other suppliers. A driver for it is included in the standard kernel. Note that there is substantial information contained in the driver source file `atp.c'.

Note that the device name that you pass to `ifconfig` was *not* `eth0` but `atp0` for earlier versions of

this driver.

RealTek 8009

Status: Supported, Driver Name: ne (+8390)

This is an ISA NE2000 clone, and is reported to work fine with the linux NE2000 driver. The `rset8009.exe` program can be obtained from RealTek's WWW site at `http://www.realtek.com.tw` - or via ftp from the same site.

RealTek 8019

Status: Supported, Driver Name: ne (+8390)

This is a Plug and Pray version of the above. Use the DOS software to disable PnP and enable jumperless configuration; set the card to a sensible I/O address and IRQ and you should be ready to go. (If using the driver as a module, don't forget to add an `io=0xNNN` option to `/etc/conf.modules`). The `rset8019.exe` program can be obtained from RealTek's WWW site at `http://www.realtek.com.tw` - or via ftp from the same site.

RealTek 8029

Status: Supported, Driver Name: ne, ne2k-pci (+8390)

This is a PCI single chip implementation of a NE2000 clone. Various vendors are now selling cards with this chip. See NE2000-PCI for information on using any of these cards. Note that this is still a 10+ year old design just glued onto a PCI bus. Performance won't be staggeringly better than the equivalent ISA model.

RealTek 8129/8139

Status: Semi-Supported, Driver Name: rtl8139

Another PCI single chip ethernet solution from RealTek. A driver for cards based upon this chip was included in the v2.0.34 release of linux. You currently have to answer `Y' when asked if you want experimental drivers for v2.2 kernels to get access to this driver. For more information, see:

`http://cesdis.gsfc.nasa.gov/linux/drivers/rtl8139.html`

5.32 Sager

Sager NP943

Status: Semi-Supported, Driver Name: 3c501

This is just a 3c501 clone, with a different S.A. PROM prefix. I assume it is equally as brain dead as the original 3c501 as well. The driver checks for the NP943 I.D. and then just treats it as a 3c501 after that.

See 3Com 3c501 for all the reasons as to why you really don't want to use one of these cards.

5.33 Schneider & Koch

SK G16

Status: Supported, Driver Name: sk_g16

This driver was included into the v1.1 kernels, and it was written by PJD Weichmann and SWS Bern. It appears that the SK G16 is similar to the NI6510, in that it is based on the first edition LANCE chip (the 7990). Once again, it appears as though this card won't work with the generic LANCE driver.

5.34 SEEQ

SEEQ 8005

Status: Supported, Driver Name: seeq8005

This driver was included into early 1.3.x kernels, and was written by Hamish Coleman. There is little information about the card included in the driver, and hence little information to be put here. If you have a question, you are probably best off e-mailing hamish@zot.apana.org.au

5.35 SMC (Standard Microsystems Corp.)

The ethernet part of Western Digital was bought out by SMC many years ago when the wd8003 and wd8013 were the main product. Since then SMC has continued making 8390 based ISA cards (Elite16, Ultra, EtherEZ) and also added several PCI products to their range.

Contact information for SMC:

SMC / Standard Microsystems Corp., 80 Arkay Drive, Hauppage, New York, 11788, USA. Technical Support via phone: 800-992-4762 (USA) or 800-433-5345 (Canada) or 516-435-6250 (Other Countries). Literature requests: 800-SMC-4-YOU (USA) or 800-833-4-SMC (Canada) or 516-435-6255 (Other Countries). Technical Support via E-mail: `techsupt@ccmail.west.smc.com`. FTP Site: `ftp.smc.com`. WWW Site: SMC.

WD8003, SMC Elite

Status: Supported, Driver Name: wd (+8390)

These are the 8-bit versions of the card. The 8 bit 8003 is slightly less expensive, but only worth the savings for light use. Note that some of the non-EEPROM cards (clones with jumpers, or old *old* old wd8003 cards) have no way of reporting the IRQ line used. In this case, auto-irq is used, and if that fails, the driver silently assings IRQ 5. You can get the SMC setup/driver disks from SMC's ftp site. Note that some of the newer SMC `SuperDisk' programs will fail to detect the real old EEPROM-less cards. The

TOC

file `SMCDSK46.EXE` seems to be a good all-round choice. Also the jumper settings for all their card in an ASCII text file in the aforementioned archive. The latest (greatest?) version can be obtained fr `ftp.smc.com`.

As these are basically the same as their 16 bit counterparts (WD8013 / SMC Elite16), you should see the next section for more information.

WD8013, SMC Elite16

Status: Supported, Driver Name: wd (+8390)

Over the years the design has added more registers and an EEPROM. (The first wd8003 cards appeared about ten years ago!) Clones usually go by the `8013' name, and usually use a non-EEPROM (jumpered) design. Late model SMC cards will have the SMC 83c690 chip instead of the original Nat Semi DP8390 found on earlier cards. The shared memory design makes the cards a bit faster than PIO cards, especially with larger packets. More importantly, from the driver's point of view, it avoids a few bugs in the programmed-I/O mode of the 8390, allows safe multi-threaded access to the packet buffer, and it doesn't have a programmed-I/O data register that hangs your machine during warm-boot probes.

Non-EEPROM cards that can't just read the selected IRQ will attempt auto-irq, and if that fails, they will silently assign IRQ 10. (8 bit versions will assign IRQ 5)

Cards with a non standard amount of memory on board can have the memory size specified at boot (or as an option in `/etc/conf.modules` if using modules). The standard memory size is 8kB for an 8bit card and 16kB for a 16bit card. For example, the older WD8003EBT cards could be jumpered for 32kB memory. To make full use of that RAM, you would use something like (for I/O=0x280 and IRQ 9):

```
LILO: linux ether=9,0x280,0xd0000,0xd8000,eth0
```

Also see 8013 problems for some of the more common problems and frequently asked questions that pop up often.

If you intend on using this driver as a loadable module you should probably see Using the Ethernet Drivers as Modules for module specific information.

SMC Elite Ultra

Status: Supported, Driver Name: smc-ultra (+8390)

This ethercard is based on the 83c790 chip from SMC, which has a few new features over the 83c690. While it has a mode that is similar to the older SMC ethercards, it's not entirely compatible with the old WD80*3 drivers. However, in this mode it shares most of its code with the other 8390 drivers, while operating slightly faster than a WD8013 clone.

Since part of the Ultra *looks like* an 8013, the Ultra probe is supposed to find an Ultra before the wd8013 probe has a chance to mistakenly identify it.

TOC

Donald mentioned that it is possible to write a separate driver for the Ultra's `Altego' mode which all chaining transmits at the cost of inefficient use of receive buffers, but that will probably not happen

Bus-Master SCSI host adaptor users take note: In the manual that ships with Interactive UNIX, it mentions that a bug in the SMC Ultra will cause data corruption with SCSI disks being run from an aha-154X host adaptor. This will probably bite aha-154X compatible cards, such as the BusLogic boards, and the AMI-FastDisk SCSI host adaptors as well.

SMC has acknowledged the problem occurs with Interactive, and older Windows NT drivers. It is a hardware conflict with early revisions of the card that can be worked around in the driver design. The current Ultra driver protects against this by only enabling the shared memory during data transfers with the card. Make sure your kernel version is at least 1.1.84, or that the driver version reported at boot is at least `smc-ultra.c:v1.12` otherwise you are vulnerable.

If you intend on using this driver as a loadable module you should probably see Using the Ethernet Drivers as Modules for module specific information.

SMC Elite Ultra32 EISA

Status: Supported, Driver Name: smc-ultra32 (+8390)

This EISA card shares a lot in common with its ISA counterpart. A working (and stable) driver is included in both v2.0 and v2.2 kernels. Thanks go to Leonard Zubkoff for purchasing some of these cards so that linux support could be added for them.

SMC EtherEZ (8416)

Status: Supported, Driver Name: smc-ultra (+8390)

This card uses SMC's 83c795 chip and supports the Plug 'n Play specification. It also has an *SMC Ultra* compatible mode, which allows it to be used with the Linux Ultra driver. For best results, use the SMC supplied program (avail. from their www/ftp site) to disable PnP and configure it for shared memory mode. See the above information for notes on the Ultra driver.

For v1.2 kernels, the card had to be configured for shared memory operation. However v2.0 kernels can use the card in shared memory or programmed I/O mode. Shared memory mode will be slightly faster, and use less CPU resources as well.

SMC EtherPower PCI (8432)

Status: Supported, Driver Name: de4x5, tulip

NB: The EtherPower II is an entirely different card. See below! These cards are a basic DEC 21040 implementation, i.e. one big chip and a couple of transceivers. Donald has used one of these cards for his development of the generic 21040 driver (aka `tulip.c`). Thanks to Duke Kamstra, once again, for supplying a card to do development on.

Some of the later revisons of this card use the newer DEC 21041 chip, which may cause problems with

older versions of the tulip driver. If you have problems, make sure you are using the latest driver rele which may not yet be included in the current kernel source tree.

See DEC 21040 for more details on using one of these cards, and the current status of the driver.

Apparently, the latest revision of the card, the EtherPower-II uses the 9432 chip. It is unclear at the moment if this one will work with the present driver. As always, if unsure, check that you can return the card if it doesn't work with the linux driver *before* paying for the card.

SMC EtherPower II PCI (9432)

Status: Semi-Supported, Driver Name: epic100

These cards, based upon the SMC 83c170 chip, are entirely different than the Tulip based cards. A new driver has been included in kernels v2.0 and v2.2 to support these cards. For more details, see:

`http://cesdis.gsfc.nasa.gov/linux/drivers/epic100.html`

SMC 3008

Status: Not Supported.

These 8 bit cards are based on the Fujitsu MB86950, which is an ancient version of the MB86965 used in the Linux at1700 driver. Russ says that you could probably hack up a driver by looking at the at1700.c code and his DOS packet driver for the Tiara card (tiara.asm). They are not very common.

SMC 3016

Status: Not Supported.

These are 16bit I/O mapped 8390 cards, much similar to a generic NE2000 card. If you can get the specifications from SMC, then porting the NE2000 driver would probably be quite easy. They are not very common.

SMC-9000 / SMC 91c92/4

Status: Supported, Driver Name: smc9194

The SMC9000 is a VLB card based on the 91c92 chip. The 91c92 appears on a few other brand cards as well, but is fairly uncommon. Erik Stahlman (erik@vt.edu) has written this driver which is in v2.0 kernels, but not in the older v1.2 kernels. You may be able to drop the driver into a v1.2 kernel source tree with minimal difficulty.

SMC 91c100

Status: Semi-Supported, Driver Name: smc9194

The SMC 91c92 driver is supposed to work for cards based on this 100Base-T chip, but at the moment

this is unverified.

5.36 Texas Instruments

ThunderLAN

Status: Supported, Driver Name: tlan

This driver covers many Compaq built-in ethernet devices, including the NetFlex and Netelligent groups. It also supports the Olicom 2183, 2185, 2325 and 2326 products.

5.37 Thomas Conrad

Thomas Conrad TC-5048

This is yet another PCI card that is based on DEC's 21040 chip.

See the section on the 21040 chip (DEC 21040) for more information.

5.38 VIA

You probably won't see a VIA networking card, as VIA make several networking chips that are then used by others in the construction of an ethernet card. They have a WWW site at:

`http://www.via.com.tw/`

VIA 86C926 Amazon

Status: Supported, Driver Name: ne, ne2k-pci (+8390)

This controller chip is VIA's PCI-NE2000 offering. You can choose between the ISA/PCI `ne.c` driver or the PCI-only `ne2k-pci.c` driver. See the PCI-NE2000 section for more details.

VIA 86C100A Rhine II (and 3043 Rhine I)

Status Supported, Driver Name: via-rhine

This relatively new driver can be found in current 2.0 and 2.1 kernels. It is an improvement over the 86C926 NE2000 chip in that it supports bus master transfers, but strict 32 bit buffer alignment requirements limit the benefit gained from this. For more details and driver updates, see:

`http://cesdis.gsfc.nasa.gov/linux/drivers/via-rhine.html`

TOC

5.39 Western Digital

Please see SMC for information on SMC cards. (SMC bought out Western Digital's network card section many years ago.)

5.40 Winbond

Winbond don't really make and sell complete cards to the general public -- instead they make single chip ethernet solutions that other companies buy, stick onto a PCI board with their own name and then sell through retail stores.

Winbond 89c840

Status: Semi-Supported, Driver Name: winbond-840

This driver isn't currently shipped with the kernel, as it is in the testing phase. It is available at:

`http://cesdis.gsfc.nasa.gov/linux/drivers/test/winbond-840.c`

Winbond 89c940

Status: Supported, Driver Name: ne, ne2k-pci (+8390)

This chip is one of the two commonly found on the low price PCI ne2000 cards sold by lots of manufacturers. Note that this is still a 10+ year old design just glued onto a PCI bus. Performance won't be staggeringly better than the equivalent ISA model.

5.41 Xircom

For the longest time, Xircom wouldn't release the programming information required to write a driver, unless you signed your life away. Apparently enough linux users have pestered them for driver support (they claim to support all popular networking operating systems...) so that they have changed their policy to allow documentation to be released without having to sign a non-disclosure agreement. Some people have said they they will release the source code to the SCO driver, while others have been told that they are no longer providing information on `obsolete' products like the earlier PE models. If you are interested and want to check into this yourself, you can reach Xircom at 1-800-874-7875, 1-800-438-4526 or +1-818-878-7600.

Xircom PE1, PE2, PE3-10B*

Status: Not Supported.

Not to get your hopes up, but if you have one of these parallel port adaptors, you may be able to use it in the DOS emulator with the Xircom-supplied DOS drivers. You will have to allow DOSEMU access to your parallel port, and will probably have to play with SIG (DOSEMU's Silly Interrupt Generator).

TOC

Xircom PCMCIA Cards

Status: Semi-Supported, Driver Name: ????

Some of the Xircom PCMCIA card(s) have drivers that are available with David Hinds PCMCIA package. Check there for the most up to date indformation

5.42 Zenith

Z-Note

Status: Supported, Driver Name: znet

The built-in Z-Note network adaptor is based on the Intel i82593 using *two* DMA channels. There is an (alpha?) driver available in the present kernel version. As with all notebook and pocket adaptors, it is under the `Pocket and portable adaptors' section when running `make config`. Also note that the IBM ThinkPad 300 is compatible with the Z-Note.

5.43 Znyx

Znyx ZX342 (DEC 21040 based)

Status: Supported, Driver Name: de4x5, tulip

You have a choice of *two* drivers for cards based on this chip. There is the DE425 driver written by David, and the generic 21040 driver that Donald has written.

Note that as of 1.1.91, David has added a compile time option that may allow non-DEC cards (such as the Znyx cards) to work with this driver. Have a look at `README.de4x5` for details.

See DEC 21040 for more information on these cards, and the present driver situation.

5.44 Identifying an Unknown Card

Okay, so your uncle's cousin's neighbour's friend had a brother who found an old ISA ethernet card in the AT case he was using as a cage for his son's pet hampster. Somehow you ended up with the card and want to try and use it with linux, but nobody has a clue what the card is and there isn't any documentation.

First of all, look for any obvious model numbers that might give a clue. Any model number that contains 2000 will most likely be a NE2000 clone. Any cards with 8003 or 8013 on them somewhere will be Western/Digital WD80x3 cards or SMC Elite cards or clones of them.

TOC

Identifying the Network Interface Controller

Look for the biggest chip on the card. This will be the network controller (NIC) itself, and most can be identified by the part number. If you know which NIC is on the card, the following might be able to help you figure out what card it is.

Probably still the most common NIC is the National Semiconductor DP8390 aka NS32490 aka DP83901 aka DP83902 aka DP83905 aka DP83907. And those are just the ones made by National! Other companies such as Winbond and UMC make DP8390 and DP83905 clone parts, such as the Winbond 89c904 (DP83905 clone) and the UMC 9090. If the card has some form of 8390 on it, then chances are it is a ne1000 or ne2000 clone card. The second most common 8390 based card are wd80x3 cards and clones. Cards with a DP83905 can be configured to be an ne2000 *or* a wd8013. Never versions of the genuine wd80x3 and SMC Elite cards have an 83c690 in place of the original DP8390. The SMC Ultra cards have an 83c790, and use a slightly different driver than the wd80x3 cards. The SMC EtherEZ cards have an 83c795, and use the same driver as the SMC Ultra. All BNC cards based on some sort of 8390 or 8390 clone will usually have an 8392 (or 83c692, or ???392) 16 pin DIP chip very close to the BNC connector.

Another common NIC found on older cards is the Intel i82586. Cards having this NIC include the 3c505, 3c507, 3c523, Intel EtherExpress-ISA, Microdyne Exos-205T, and the Racal-Interlan NI5210.

The original AMD LANCE NIC was numbered AM7990, and newer revisions include the 79c960, 79c961, 79c965, 79c970, and 79c974. Most cards with one of the above will work with the Linux LANCE driver, with the exception of the old Racal-Interlan NI6510 cards that have their own driver.

Newer PCI cards having a DEC 21040, 21041, 21140, or similar number on the NIC should be able to use the linux tulip or de4x5 driver.

Other PCI cards having a big chip marked RTL8029 or 89C940 or 86C926 are ne2000 clone cards, and the ne driver in linux version v2.0 and up should automatically detect these cards at boot.

Identifying the Ethernet Address

Each ethernet card has its own six byte address that is unique to that card. The first three bytes of that address are the same for each card made by that particular manufacturer. For example all SMC cards start with `00:00:c0`. The last three are assigned by the manufacturer uniquely to each individual card as they are produced.

If your card has a sticker on it giving all six bits of its address, you can look up the vendor from the first three. However it is more common to see only the last three bytes printed onto a sticker attached to a socketed PROM, which tells you nothing.

You can determine which vendors have which assigned addresses from RFC-1340. Apparently there is a more up to date listing available in various places as well. Try a WWW or FTP search for `EtherNet-codes` or `Ethernet-codes` and you will find something.

Tips on Trying to Use an Unknown Card

If you are still not sure what the card is, but have at least narrowed it down some, then you can build a kernel with a whole bunch of drivers included, and see if any of them autodetect the card at boot.

If the kernel doesn't detect the card, it may be that the card is not configured to one of the addresses that the driver probes when looking for a card. In this case, you might want to try getting `scanport.tar.gz` from your local linux ftp site, and see if that can locate where your card is jumpered for. It scans ISA I/O space from `0x100` to `0x3ff` looking for devices that aren't registered in `/proc/ioports`. If it finds an unknown device starting at some particular address, you can then explicity point the ethernet probes at that address with an `ether=` boot argument.

If you manage to get the card detected, you can then usually figure out the unknown jumpers by changing them one at a time and seeing at what I/O base and IRQ that the card is detected at. The IRQ settings can also usually be determined by following the traces on the back of the card to where the jumpers are soldered through. Counting the `gold fingers' on the backside, from the end of the card with the metal bracket, you have IRQ 9, 7, 6, 5, 4, 3, 10, 11, 12, 15, 14 at fingers 4, 21, 22, 23, 24, 25, 34, 35, 36, 37, 38 respectively. Eight bit cards only have up to finger 31.

Jumpers that appear to do nothing usually are for selecting the memory address of an optional boot ROM. Other jumpers that are located near the BNC or RJ-45 or AUI connectors are usually to select the output media. These are also typically near the `black box' voltage converters marked YCL, Valor, or Fil-Mag.

A nice collection of jumper settings for various cards can be found at the following URL:

Ethercard Settings

5.45 Drivers for Non-Ethernet Devices

There are a few other drivers that are in the linux source that present an *ethernet-like* device to network programs, while not really being ethernet. These are briefly listed here for completeness.

`dummy.c` - The purpose of this driver is to provide a device to point a route through, but not to actually transmit packets.

`eql.c` - Load Equalizer, enslaves multiple devices (usually modems) and balances the Tx load across them while presenting a single device to the network programs.

`ibmtr.c` - IBM Token Ring, which is not really ethernet. Broken-Ring requires source routing and other uglies.

`loopback.c` - Loopback device, for which all packets from your machine and destined for your own machine go. It essentially just moves the packet off the Tx queue and onto the Rx queue.

`pi2.c` - Ottawa Amateur Radio Club PI and PI2 interface.

`plip.c` - Parallel Line Internet Protocol, allows two computers to send packets to each other over two joined parallel ports in a point-to-point fashion.

`ppp.c` - Point-to-Point Protocol (RFC1331), for the Transmission of Multi-protocol Datagrams over Point-to-Point Link (again usually modems).

`slip.c` - Serial Line Internet Protocol, allows two computers to send packets to each other over two joined serial ports (usually via modems) in a point-to-point fashion.

`tunnel.c` - Provides an IP tunnel through which you can tunnel network traffic transparently across subnets

`wavelan.c` - An Ethernet-like radio transceiver controlled by the Intel 82586 coprocessor which is used on other ethercards such as the Intel EtherExpress.

---

6. Cables, Coax, Twisted Pair

If you are starting a network from scratch, you will have to decide whether to use thin ethernet (RG58 co-ax cable with BNC connectors) or 10baseT (twisted pair telco-style cables with RJ-45 eight wire `phone' connectors). The old-fashioned thick ethernet, RG-5 cable with N connectors is obsolete and rarely seen anymore.

See Type of cable... for an introductory look at cables. Also note that the FAQ from *comp.dcom.lans.ethernet* has a lot of useful information on cables and such. FTP to rtfm.mit.edu and look in `/pub/usenet-by-hierarchy/` for the FAQ for that newsgroup.

6.1 Thin Ethernet (thinnet)

Thin ethernet cable is pretty inexpensive. If you are making your own cables solid-core RG58A is $0.27/m. and stranded RG58AU is $0.45/m. Twist-on BNC connectors are < $2 ea., and other misc. pieces are similarly inexpensive. It is essential that you properly terminate each end of the cable with 50 ohm terminators, so budget $2 ea. for a pair. It's also vital that your cable have no `stubs' -- the `T' connectors must be attached directly to the ethercards.

There are two main drawbacks to using thinnet. The first is that it is limited to 10Mb/sec - 100Mb/sec requires twisted pair. The second drawback is that if you have a big loop of machines connected together, and some bonehead breaks the loop by taking one cable off the side of his tee, the whole network goes down because it sees an infinite impedance (open circuit) instead of the required 50 ohm termination. Note that you can remove the tee piece from the card itself without killing the whole subnet, as long as you don't remove the cables from the tee itself. Of course this will disturb the machine that you pull the actual tee off of. 8-) And if you are doing a small network of two machines, you *still* need the tees and the 50 ohm terminators -- you *can't* just cable them together!

There are also some fancy cable systems which *look like* a single lead going to the card, but the lead is actually two runs of cable laying side-by-side covered by an outer sheath, giving the lead an oval shaped cross-section. At the turnaround point of the loop, a BNC connector is spliced in which connects to your card. So you have the equivalent of two runs of cable and a BNC T, but in this case, it is impossible for the user to remove a cable from one side of the T and disturb the network.

6.2 Twisted Pair

Twisted pair networks require active hubs, which start around $50, and the raw cable cost can actually be higher than thinnet. You can pretty much ignore claims that you can use your existing telephone wiring as it is a rare installation where that turns out to be the case.

On the other hand, all 100Mb/sec ethernet proposals use twisted pair, and most new business installations use twisted pair. Also, Russ Nelson adds that `New installations should use Category 5 wiring. Anything

else is a waste of your installer's time, as 100Base-whatever is going to require Cat 5.'

If you are only connecting two machines, it is possible to avoid using a hub, by swapping the Rx and Tx pairs (1-2 and 3-6).

If you hold the RJ-45 connector facing you (as if you were going to plug it into your mouth) with the lock tab on the top, then the pins are numbered 1 to 8 from left to right. The pin usage is as follows:

```
Pin Number              Assignment
----------              ----------
1                       Output Data (+)
2                       Output Data (-)
3                       Input Data (+)
4                       Reserved for Telephone use
5                       Reserved for Telephone use
6                       Input Data (-)
7                       Reserved for Telephone use
8                       Reserved for Telephone use
```

If you want to make a cable, the following should spell it out for you. Differential signal pairs must be on the same twisted pair to get the required minimal impedance/loss of a UTP cable. If you look at the above table, you will see that 1+2 and 3+6 are the two sets of differential signal pairs. Not 1+3 and 2+6 !!!!!! At 10MHz, with short lengths, you *may* get away with such errors, if it is only over a short length. Don't even think about it at 100MHz.

For a normal patch cord, with ends `A' and `B', you want straight through pin-to-pin mapping, with the input and output each using a pair of twisted wires (for impedance issues). That means 1A goes to 1B, 2A goes to 2B, 3A goes to 3B and 6A goes to 6B. The wires joining 1A-1B and 2A-2B must be a twisted pair. Also the wires joining 3A-3B and 6A-6B must be another twisted pair.

Now if you don't have a hub, and want to make a `null cable', what you want to do is make the input of `A' be the output of `B' and the output of `A' be the input of `B', without changing the polarity. Tha means connecting 1A to 3B (out+ A to in+ B) and 2A to 6B (out- A to in- B). These two wires must be a twisted pair. They carry what card/plug `A' considers output, and what is seen as input for card/plug `B'. Then connect 3A to 1B (in+ A to out+ B) and also connect 6A to 2B (in- A to out- B). These second two must also be a twisted pair. They carry what card/plug `A' considers input, and what card/plug `B' considers output.

So, if you consider a normal patch cord, chop one end off of it, swap the places of the Rx and Tx twisted pairs into the new plug, and crimp it down, you then have a `null' cable. Nothing complicated. You just want to feed the Tx signal of one card into the Rx of the second and vice versa.

Note that before 10BaseT was ratified as a standard, there existed other network formats using RJ-45 connectors, and the same wiring scheme as above. Examples are SynOptics's LattisNet, and AT&T's StarLAN. In some cases, (as with early 3C503 cards) you could set jumpers to get the card to talk to hubs of different types, but in most cases cards designed for these older types of networks will not work with standard 10BaseT networks/hubs. (Note that if the cards also have an AUI port, then there is no reason as to why you can't use that, combined with an AUI to 10BaseT transceiver.)

6.3 Thick Ethernet

Thick ethernet is mostly obsolete, and is usually used only to remain compatible with an existing implementation. You can stretch the rules and connect short spans of thick and thin ethernet together with a passive $3 N-to-BNC connector, and that's often the best solution to expanding an existing thicknet. A correct (but expensive) solution is to use a repeater in this case.

---

7. Software Configuration and Card Diagnostics

In most cases, if the configuration is done by software, and stored in an EEPROM, you will usually have to boot DOS, and use the vendor supplied DOS program to set the cards IRQ, I/O, mem_addr and whatnot. Besides, hopefully it is something you will only be setting once. If you don't have the DOS software for your card, try looking on the WWW site of your card manufacturer. If you don't know the site name, take a guess at it, i.e. `www.my_vendor.com' where `my_vendor' is the name of your card manufacturer. This works for SMC, 3Com, and many *many* other manufacturers.

There are some cards for which Linux versions of the config utils exist, and they are listed here. Donald has written a few small card diagnostic programs that run under Linux. Most of these are a result of debugging tools that he has created while writing the various drivers. Don't expect fancy menu-driven interfaces. You will have to read the source code to use most of these. Even if your particular card doesn't have a corresponding diagnostic, you can still get some information just by typing `cat /proc/net/dev` -- assuming that your card was at least detected at boot.

In either case, you will have to run most of these programs as root (to allow I/O to the ports) and you probably want to shut down the ethercard before doing so by typing `ifconfig eth0 down` first.

7.1 Configuration Programs for Ethernet Cards

WD80x3 Cards

For people with wd80x3 cards, there is the program `wdsetup` which can be found in `wdsetup-0.6a.tar.gz` on Linux ftp sites. It is not being actively maintained, and has not been updated for quite a while. If it works fine for you then great, if not, use the DOS version that you should have got with your card. If you don't have the DOS version, you will be glad to know that the SMC setup/driver disks are available at SMC's ftp site. Of course, you *have* to have an EEPROM card to use this utility. Old, *old* wd8003 cards, and some wd8013 clones use jumpers to set up the card instead.

Digital / DEC Cards

The Digital EtherWorks 3 card can be configured in a similar fashion to the DOS program `NICSETUP.EXE`. David C. Davies wrote this and other tools for the EtherWorks 3 in conjunction with the driver. Look on you local linux FTP site in the directory `/pub/linux/system/Network/management` for the file that is named `ewrk3tools-X.XX.tar.gz`.

NE2000+ or AT/LANTIC Cards

Some Nat Semi DP83905 implementations (such as the AT/LANTIC and the NE2000+) are software configurable. (Note that these cards can also emulate a wd8013 card!) You can get the file `/pub/linux/setup/atlantic.c` from Donald's ftp server, `cesdis.gsfc.nasa.gov` to configure this card. In addition, the configuration programs for the Kingston DP83905 cards seem to work with all cards, as they don't check for a vendor specific address before allowing you to use them. Follow the following URL: Kingston Software and get `20XX12.EXE` and `INFOSET.EXE`.

Be careful when configuring NE2000+ cards, as you can give them bad setting values which can cause problems. A typical example is accidentally enabling the boot ROM in the EEPROM (even if no ROM is installed) to a setting that conflicts with the VGA card. The result is a computer that just beeps at you when you turn it on and nothing appears on the screen.

You can typically recover from this by doing the following: Remove the card from the machine, and then boot and enter the CMOS setup. Change the `Display Adapter' to `Not Installed' and change the default boot drive to `A:' (your floppy drive). Also change the `Wait for F1 if any Error' to `Disabled'. This way, the computer should boot without user intervention. Now create a bootable DOS floppy (`format a: /s /u') and copy the program `default.exe` from the `20XX12.EXE` archive above onto that floppy. Then type `echo default > a:autoexec.bat` so that the program to set the card back to sane defaults will be run automatically when you boot from this floppy. Shut the machine off, re-install the ne2000+ card, insert your new boot floppy, and power it back up. It will still probably beep at you, but eventually you should see the floppy light come on as it boots from the floppy. Wait a minute or two for the floppy to stop, indicating that it has finished running the `default.exe` program, and then power down your computer. When you then turn it on again, you should hopefully have a working display again, allowing you to change your CMOS settings back, and to change the card's EEPROM settings back to the values you want.

Note that if you don't have DOS handy, you can do the whole method above with a linux boot disk that automatically runs Donald's `atlantic` program (with the right command line switches) instead of a DOS boot disk that automatically runs the `default.exe` program.

3Com Cards

The 3Com Etherlink III family of cards (i.e. 3c5x9) can be configured by using another config utility from Donald. You can get the file `/pub/linux/setup/3c5x9setup.c` from Donald's ftp server, `cesdis.gsfc.nasa.gov` to configure these cards. (Note that the DOS 3c5x9B config utility may have more options pertaining to the new ``B'' series of the Etherlink III family.)

7.2 Diagnostic Programs for Ethernet Cards

Any of the diagnostic programs that Donald has written can be obtained from this URL.

Ethercard Diagnostics

Allied Telesis AT1700 -- look for the file `/pub/linux/diag/at1700.c` on

cesdis.gsfc.nasa.gov.

Cabletron E21XX -- look for the file /pub/linux/diag/e21.c on cesdis.gsfc.nasa.gov.

HP PCLAN+ -- look for the file /pub/linux/diag/hp+.c on cesdis.gsfc.nasa.gov.

Intel EtherExpress -- look for the file /pub/linux/diag/eexpress.c on cesdis.gsfc.nasa.gov.

NE2000 cards -- look for the file /pub/linux/diag/ne2k.c on cesdis.gsfc.nasa.gov. There is also a PCI version for the now common NE2000-PCI clones.

RealTek (ATP) Pocket adaptor -- look for the file /pub/linux/diag/atp-diag.c on cesdis.gsfc.nasa.gov.

All Other Cards -- try typing cat /proc/net/dev and dmesg to see what useful info the kernel has on the card in question.

---

8. Technical Information

For those who want to understand a bit more about how the card works, or play with the present drivers, or even try to make up their own driver for a card that is presently unsupported, this information should be useful. If you do not fall into this category, then perhaps you will want to skip this section.

8.1 Programmed I/O vs. Shared Memory vs. DMA

If you can already send and receive back-to-back packets, you just can't put more bits over the wire. Every modern ethercard can receive back-to-back packets. The Linux DP8390 drivers (wd80x3, SMC-Ultra, 3c503, ne2000, etc) come pretty close to sending back-to-back packets (depending on the current interrupt latency) and the 3c509 and AT1500 hardware have no problem at all automatically sending back-to-back packets.

The ISA bus can do 5.3MB/sec (42Mb/sec), which sounds like more than enough for 10Mbps ethernet. In the case of the 100Mbps cards, you clearly need a faster bus to take advantage of the network bandwidth.

Programmed I/O (e.g. NE2000, 3c509)

Pro: Doesn't use any constrained system resources, just a few I/O registers, and has no 16M limit.

Con: Usually the slowest transfer rate, the CPU is waiting the whole time, and interleaved packet access is usually difficult to impossible.

Shared memory (e.g. WD80x3, SMC-Ultra, 3c503)

Pro: Simple, faster than programmed I/O, and allows random access to packets. Where possible, the linux drivers compute the checksum of incoming IP packets as they are copied off the card, resulting in a further reduction of CPU usage vs. an equivalent PIO card.

Con: Uses up memory space (a big one for DOS users, essentially a non-issue under Linux), and it still ties up the CPU.

Slave (normal) Direct Memory Access (e.g. none for Linux!)

Pro: Frees up the CPU during the actual data transfer.

Con: Checking boundary conditions, allocating contiguous buffers, and programming the DMA registers makes it the slowest of all techniques. It also uses up a scarce DMA channel, and requires aligned low memory buffers.

Bus Master Direct Memory Access (e.g. LANCE, DEC 21040)

Pro: Frees up the CPU during the data transfer, can string together buffers, can require little or no CPU time lost on the ISA bus. Most of the bus-mastering linux drivers now use a `copybreak' scheme where large packets are put directly into a kernel networking buffer by the card, and small packets are copied by the CPU which primes the cache for subsequent processing.

Con: (Only applicable to ISA bus cards) Requires low-memory buffers and a DMA channel for cards. Any bus-master will have problems with other bus-masters that are bus-hogs, such as some primitive SCSI adaptors. A few badly-designed motherboard chipsets have problems with bus-masters. And a reason for not using *any* type of DMA device is using a 486 processor designed for plug-in replacement of a 386: these processors must flush their cache with each DMA cycle. (This includes the Cx486DLC, Ti486DLC, Cx486SLC, Ti486SLC, etc.)

8.2 Writing a Driver

The only thing that one needs to use an ethernet card with Linux is the appropriate driver. For this, it is essential that the manufacturer will release the technical programming information to the general public without you (or anyone) having to sign your life away. A good guide for the likelihood of getting documentation (or, if you aren't writing code, the likelihood that someone else will write that driver you really, really need) is the availability of the Crynwr (nee Clarkson) packet driver. Russ Nelson runs this operation, and has been very helpful in supporting the development of drivers for Linux. *Net-surfers* can try this URL to look up Russ' software.

Russ Nelson's Packet Drivers

Given the documentation, you can write a driver for your card and use it for Linux (at least in theory). Keep in mind that some old hardware that was designed for XT type machines will not function very well in a multitasking environment such as Linux. Use of these will lead to major problems if your network sees a reasonable amount of traffic.

Most cards come with drivers for MS-DOS interfaces such as NDIS and ODI, but these are useless for Linux. Many people have suggested directly linking them in or automatic translation, but this is nearly impossible. The MS-DOS drivers expect to be in 16 bit mode and hook into `software interrupts', both incompatible with the Linux kernel. This incompatibility is actually a feature, as some Linux drivers are considerably better than their MS-DOS counterparts. The `8390' series drivers, for instance, use ping-pong transmit buffers, which are only now being introduced in the MS-DOS world.

(Ping-pong Tx buffers means using at least 2 max-size packet buffers for Tx packets. One is loaded while the card is transmitting the other. The second is then sent as soon as the first finished, and so on. In this way, most cards are able to continuously send back-to-back packets onto the wire.)

OK. So you have decided that you want to write a driver for the Foobar Ethernet card, as you have the programming information, and it hasn't been done yet. (...these are the two main requirements ;-) You should start with the skeleton network driver that is provided with the Linux kernel source tree. It can be found in the file /usr/src/linux/drivers/net/skeleton.c in all recent kernels. Also have a look at the Kernel Hackers Guide, at the following URL: KHG

8.3 Driver interface to the kernel

TOC

Here are some notes on the functions that you would have to write if creating a new driver. Reading this in conjunction with the above skeleton driver may help clear things up.

Probe

Called at boot to check for existence of card. Best if it can check un-obtrsively by reading from memory, etc. Can also read from I/O ports. Initial writing to I/O ports in a probe is *not good* as it may kill another device. Some device initialization is usually done here (allocating I/O space, IRQs,filling in the dev->??? fields etc.) You need to know what io ports/mem the card can be configured to, how to enable shared memory (if used) and how to select/enable interrupt generation, etc.

Interrupt handler

Called by the kernel when the card posts an interrupt. This has the job of determining why the card posted an interrupt, and acting accordingly. Usual interrupt conditions are data to be rec'd, transmit completed, error conditions being reported. You need to know any relevant interrupt status bits so that you can act accordingly.

Transmit function

Linked to dev->hard_start_xmit() and is called by the kernel when there is some data that the kernel wants to put out over the device. This puts the data onto the card and triggers the transmit. You need to know how to bundle the data and how to get it onto the card (shared memory copy, PIO transfer, DMA?) and in the right place on the card. Then you need to know how to tell the card to send the data down the wire, and (possibly) post an interrupt when done. When the hardware can't accept additional packets it should set the dev->tbusy flag. When additional room is available, usually during a transmit-complete interrupt, dev->tbusy should be cleared and the higher levels informed with `mark_bh(INET_BH)`.

Receive function

Called by the kernel interrupt handler when the card reports that there is data on the card. It pulls the data off the card, packages it into a sk_buff and lets the kernel know the data is there for it by doing a netif_rx(sk_buff). You need to know how to enable interrupt generation upon Rx of data, how to check any relevant Rx status bits, and how to get that data off the card (again sh mem, PIO, DMA, etc.)

Open function

linked to dev->open and called by the networking layers when somebody does `ifconfig eth0 up` - this puts the device on line and enables it for Rx/Tx of data. Any special initialization incantations that were not done in the probe sequence (enabling IRQ generation, etc.) would go in here.

Close function (optional)

This puts the card in a sane state when someone does `ifconfig eth0 down`. It should free the IRQs and DMA channels if the hardware permits, and turn off anything that will save power (like the transceiver).

Miscellaneous functions

Things like a reset function, so that if things go south, the driver can try resetting the card as a last ditch effort. Usually done when a Tx times out or similar. Also a function to read the statistics registers of the card if so equipped.

8.4 Technical information from 3Com

If you are interested in working on drivers for 3Com cards, you can get technical documentation from 3Com. Cameron has been kind enough to tell us how to go about it below:

3Com's Ethernet Adapters are documented for driver writers in our `Technical References' (TRs). These manuals describe the programmer interfaces to the boards but they don't talk about the diagnostics, installation programs, etc that end users can see.

The Network Adapter Division marketing department has the TRs to give away. To keep this program efficient, we centralized it in a thing called `CardFacts.' CardFacts is an automated phone system. You call it with a touch-tone phone and it faxes you stuff. To get a TR, call CardFacts at 408-727-7021. Ask it for Developer's Order Form, document number 9070. Have your fax number ready when you call. Fill out the order form and fax it to 408-764-5004. Manuals are shipped by Federal Express 2nd Day Service.

There are people here who think we are too free with the manuals, and they are looking for evidence that the system is too expensive, or takes too much time and effort. So far, 3Com customers have been really good about this, and there's no problem with the level of requests we've been getting. We need your continued cooperation and restraint to keep it that way.

8.5 Notes on AMD PCnet / LANCE Based cards

The AMD LANCE (Local Area Network Controller for Ethernet) was the original offering, and has since been replaced by the `PCnet-ISA' chip, otherwise known as the 79C960. Note that the name `LANCE' has stuck, and some people will refer to the new chip by the old name. Dave Roberts of the Network Products Division of AMD was kind enough to contribute the following information regarding this chip:

`Functionally, it is equivalent to a NE1500. The register set is identical to the old LANCE with the 1500/2100 architecture additions. Older 1500/2100 drivers will work on the PCnet-ISA. The NE1500 and NE2100 architecture is basically the same. Initially Novell called it the 2100, but then tried to distinguish between coax and 10BASE-T cards. Anything that was 10BASE-T only was to be numbered in the 1500 range. That's the only difference.

Many companies offer PCnet-ISA based products, including HP, Racal-Datacom, Allied Telesis, Boca

Research, Kingston Technology, etc. The cards are basically the same except that some manufacture have added `jumperless' features that allow the card to be configured in software. Most have not. Al offers a standard design package for a card that uses the PCnet-ISA and many manufacturers use our design without change. What this means is that anybody who wants to write drivers for most PCnet-ISA based cards can just get the data-sheet from AMD. Call our literature distribution center at (800)222-9323 and ask for the Am79C960, PCnet-ISA data sheet. It's free.

A quick way to understand whether the card is a `stock' card is to just look at it. If it's stock, it should just have one large chip on it, a crystal, a small IEEE address PROM, possibly a socket for a boot ROM, and a connector (1, 2, or 3, depending on the media options offered). Note that if it's a coax card, it will have some transceiver stuff built onto it as well, but that should be near the connector and away from the PCnet-ISA.'

A note to would-be card hackers is that different LANCE implementations do `restart' in different ways. Some pick up where they left off in the ring, and others start right from the beginning of the ring, as if just initialised.

8.6 Multicast and Promiscuous Mode

Another one of the things Donald has worked on is implementing multicast and promiscuous mode hooks. All of the *released* (i.e. **not** ALPHA) ISA drivers now support promiscuous mode.

Donald writes: `I'll start by discussing promiscuous mode, which is conceptually easy to implement. For most hardware you only have to set a register bit, and from then on you get every packet on the wire. Well, it's almost that easy; for some hardware you have to shut the board (potentially dropping a few packet), reconfigure it, and then re-enable the ethercard. OK, so that's easy, so I'll move on something that's not quite so obvious: Multicast. It can be done two ways:

1. Use promiscuous mode, and a packet filter like the Berkeley packet filter (BPF). The BPF is a pattern matching stack language, where you write a program that picks out the addresses you are interested in. Its advantage is that it's very general and programmable. Its disadvantage is that there is no general way for the kernel to avoid turning on promiscuous mode and running every packet on the wire through every registered packet filter. See The Berkeley Packet Filter for more info.
2. Using the built-in multicast filter that most etherchips have.

I guess I should list what a few ethercards/chips provide:

```
Chip/card  Promiscuous  Multicast filter
----------------------------------------
Seeq8001/3c501  Yes     Binary filter (1)
3Com/3c509      Yes     Binary filter (1)
8390            Yes     Autodin II six bit hash (2) (3)
LANCE           Yes     Autodin II six bit hash (2) (3)
i82586          Yes     Hidden Autodin II six bit hash (2) (4)
```

TOC

1. These cards claim to have a filter, but it's a simple yes/no `accept all multicast packets', or `ac no multicast packets'.
2. AUTODIN II is the standard ethernet CRC (checksum) polynomial. In this scheme multicast addresses are hashed and looked up in a hash table. If the corresponding bit is enabled, this packet is accepted. Ethernet packets are laid out so that the hardware to do this is trivial -- you just latch six (usually) bits from the CRC circuit (needed anyway for error checking) after the first six octets (the destination address), and use them as an index into the hash table (six bits -- a 64-bit table).
3. These chips use the six bit hash, and must have the table computed and loaded by the host. This means the kernel must include the CRC code.
4. The 82586 uses the six bit hash internally, but it computes the hash table itself from a list of multicast addresses to accept.

Note that none of these chips do perfect filtering, and we still need a middle-level module to do the final filtering. Also note that in every case we must keep a complete list of accepted multicast addresses to recompute the hash table when it changes.

8.7 The Berkeley Packet Filter (BPF)

The general idea of the developers is that the BPF functionality should not be provided by the kernel, but should be in a (hopefully little-used) compatibility library.

For those not in the know: BPF (the Berkeley Packet Filter) is an mechanism for specifying to the kernel networking layers what packets you are interested in. It's implemented as a specialized stack language interpreter built into a low level of the networking code. An application passes a program written in this language to the kernel, and the kernel runs the program on each incoming packet. If the kernel has multiple BPF applications, each program is run on each packet.

The problem is that it's difficult to deduce what kind of packets the application is really interested in from the packet filter program, so the general solution is to always run the filter. Imagine a program that registers a BPF program to pick up a low data-rate stream sent to a multicast address. Most ethernet cards have a hardware multicast address filter implemented as a 64 entry hash table that ignores most unwanted multicast packets, so the capability exists to make this a very inexpensive operation. But with the BPF the kernel must switch the interface to promiscuous mode, receive _all_ packets, and run them through this filter. This is work, BTW, that's very difficult to account back to the process requesting the packets.

---

9. Networking with a Laptop/Notebook Computer

There are several ways to put your laptop on a network. You can use the SLIP code (and run at serial line speeds); you can get a notebook with a supported PCMCIA slot built-in; you can get a laptop with a docking station and plug in an ISA ethercard; or you can use a parallel port Ethernet adapter.

9.1 Using SLIP

This is the cheapest solution, but by far the most difficult. Also, you will not get very high transmission rates. Since SLIP is not really related to ethernet cards, it will not be discussed further here. See the NET-2 Howto.

9.2 PCMCIA Support

Try and determine exactly what hardware you have (ie. card manufacturer, PCMCIA chip controller manufacturer) and then ask on the LAPTOPS channel. Regardless, don't expect things to be all that simple. Expect to have to fiddle around a bit, and patch kernels, etc. Maybe someday you will be able to type `make config' 8-)

At present, the two PCMCIA chipsets that are supported are the Databook TCIC/2 and the intel i82365.

There is a number of programs on tsx-11.mit.edu in /pub/linux/packages/laptops/ that you may find useful. These range from PCMCIA Ethercard drivers to programs that communicate with the PCMCIA controller chip. Note that these drivers are usually tied to a specific PCMCIA chip (ie. the intel 82365 or the TCIC/2)

For NE2000 compatible cards, some people have had success with just configuring the card under DOS, and then booting linux from the DOS command prompt via `loadlin`.

Things are looking up for Linux users that want PCMCIA support, as substantial progress is being made. Pioneering this effort is David Hinds. His latest PCMCIA support package can be obtained from:

PCMCIA Package

Look for a file like `pcmcia-cs-X.Y.Z.tgz` where X.Y.Z will be the latest version number. This is most likely uploaded to the `tsx-11.mit.edu` FTP site as well.

Note that Donald's PCMCIA enabler works as a user-level process, and David Hinds' is a kernel-level solution. You may be best served by David's package as it is much more widely used and under continuous development.

9.3 ISA Ethercard in the Docking Station.

Docking stations for laptops typically cost about $250 and provide two full-size ISA slots, two serial and one parallel port. Most docking stations are powered off of the laptop's batteries, and a few allow adding extra batteries in the docking station if you use short ISA cards. You can add an inexpensive ethercard and enjoy full-speed ethernet performance.

9.4 Pocket / parallel port adaptors.

The `pocket' ethernet adaptors may also fit your need. Note that the transfer speed will not be all that great (perhaps 200kB/s tops?) due to the limitations of the parallel port interface.

Also most tie you down with a wall-brick power supply. You can sometimes avoid the wall-brick with the adaptors by buying or making a cable that draws power from the laptop's keyboard port. (See keyboard power)

See DE-600 / DE-620 and RealTek for two supported pocket adaptors.

---

10. Miscellaneous.

Any other associated stuff that didn't fit in anywhere else gets dumped here. It may not be relevant, and it may not be of general interest but it is here anyway.

10.1 Passing Ethernet Arguments to the Kernel

Here are two generic kernel commands that can be passed to the kernel at boot time (`ether` and `reserve`). This can be done with LILO, loadlin, or any other booting utility that accepts optional arguments.

For example, if the command was `blah' and it expected 3 arguments (say 123, 456, and 789) then, with LILO, you would use:

```
LILO: linux blah=123,456,789
```

For more information on (and a complete list of) boot time arguments, please see the BootPrompt-HOWTO

The `ether` command

The `ether=` argument is used in conjunction with drivers that are directly built into the kernel. The `ether=` argument will have *absolutely no effect* on a modular driver. In its most generic form, it looks something like this:

```
ether=IRQ,BASE_ADDR,PARAM_1,PARAM_2,NAME
```

All arguments are optional. The first non-numeric argument is taken as the NAME.

IRQ: Obvious. An IRQ value of `0' (usually the default) means to autoIRQ. It's a historical accident that the IRQ setting is first rather than the base_addr -- this will be fixed whenever something else changes.

BASE_ADDR: Also obvious. A value of `0' (usually the default) means to probe a card-type-specific address list for an ethercard.

PARAM_1: It was orginally used as an override value for the memory start for a shared-memory ethercard, like the WD80*3. Some drivers use the low four bits of this value to set the debug message level. 0 -- default, 1-7 -- level 1..7, (7 is maximum verbosity) 8 -- level 0 (no messages). Also, the LANCE driver uses the low four bits of this value to select the DMA channel. Otherwise it uses auto-DMA.

PARAM_2: The 3c503 driver uses this to select between the internal and external transceivers. 0 -- default/internal, 1 -- AUI external. The Cabletron E21XX card also uses the low 4 bits of PARAM_2 to select the output media. Otherwise it detects automatically.

TOC

NAME: Selects the network device the values refer to. The standard kernel uses the names `eth0', `e `eth2' and `eth3' for bus-attached ethercards, and `atp0' for the parallel port `pocket' ethernet adaptor. arcnet driver uses `arc0' as its name. The default setting is for a single ethercard to be probed for as `eth0'. Multiple cards can only be enabled by explicitly setting up their base address using these LILO parameters. The 1.0 kernel has LANCE-based ethercards as a special case. LILO arguments are ignored, and LANCE cards are always assigned `eth<n>' names starting at `eth0'. Additional non-LANCE ethercards must be explicitly assigned to `eth<n+1>', and the usual `eth0' probe disabled with something like `ether=0,-1,eth0'. (Yes, this is bug.)

The `reserve` command

This next lilo command is used just like `ether=' above, ie. it is appended to the name of the boot select specified in lilo.conf

```
reserve=IO-base,extent{,IO-base,extent...}
```

In some machines it may be necessary to prevent device drivers from checking for devices (auto-probing) in a specific region. This may be because of poorly designed hardware that causes the boot to *freeze* (such as some ethercards), hardware that is mistakenly identified, hardware whose state is changed by an earlier probe, or merely hardware you don't want the kernel to initialize.

The `reserve` boot-time argument addresses this problem by specifying an I/O port region that shouldn't be probed. That region is reserved in the kernel's port registration table as if a device has already been found in that region. Note that this mechanism shouldn't be necessary on most machines. Only when there is a problem or special case would it be necessary to use this.

The I/O ports in the specified region are protected against device probes. This was put in to be used when some driver was hanging on a NE2000, or misidentifying some other device as its own. A correct device driver shouldn't probe a reserved region, unless another boot argument explicitly specifies that it do so. This implies that `reserve` will most often be used with some other boot argument. Hence if you specify a `reserve` region to protect a specific device, you must generally specify an explicit probe for that device. Most drivers ignore the port registration table if they are given an explicit address.

For example, the boot line

```
LILO: linux reserve=0x300,32 ether=0,0x300,eth0
```

keeps all device drivers except the ethercard drivers from probing 0x300-0x31f.

As usual with boot-time specifiers there is an 11 parameter limit, thus you can only specify 5 reserved regions per `reserve` keyword. Multiple `reserve` specifiers will work if you have an unusually complicated request.

10.2 Using the Ethernet Drivers as Modules

Most of the linux distributions now ship kernels that have very few drivers built-in. The drivers are instead supplied as a bunch of independent dynamically loadable modules. These modular drivers are typically loaded by the administrator with the `modprobe(8)` command, or in some cases they are automatically loaded by the kernel through `kerneld' (in 2.0) or `kmod' (in 2.1) which then calls

`modprobe`.

You particular distribution may offer nice graphical configuration tools for setting up ethernet modules. If possible you should try and use them first. The description that follows here gives information on what underlies any fancy configuration program, and what these programs change.

The information that controls what modules are to be used and what options are supplied to each module is usually stored in the file `/etc/conf.modules`. The two main options of interest (for ethernet cards) that will be used in this file are `alias` and `options`. The `modprobe` command consults this file for module information.

The actual modules themselves are typically stored in a directory named `/lib/modules/`uname -r`/net` where the `uname -r` command gives the kernel version (e.g. 2.0.34). You can look in there to see which module matches your card.

The first thing you need in your `conf.modules` file is something to tell `modprobe` what driver to use for the `eth0` (and `eth1` and...) network interface. You use the `alias` command for this. For example, if you have an ISA SMC EtherEZ card which uses the `smc-ultra.o` driver module, you need to `alias` this driver to `eth0` by adding the line:

```
alias eth0 smc-ultra
```

The other thing you may need is an `options` line indicating what options are to be used with a particular module (or module alias). Continuing with the above example, if you only used the single `alias` line with no `options` line, the kernel would warn you (see `dmesg`) that autoprobing for ISA cards is not a good idea. To get rid of this warning, you would add another line telling the module what I/O base the card is configured to, in this case say the hexidecimal address `0x280` for example.

```
options smc-ultra io=0x280
```

Most ISA modules accept parameters like `io=0x340` and `irq=12` on the `insmod` command line. It is *REQUIRED* or at least *STRONGLY ADVISED* that you supply these parameters to avoid probing for the card. Unlike PCI and EISA devices, there is no real safe way to do auto-probing for most ISA devices, and so it should be avoided when using drivers as modules.

A list of all the options that each module accepts can be found in the file:

`/usr/src/linux/Documentation/networking/net-modules.txt`

It is recommended that you read that to find out what options you can use for your particular card. Note that some modules support comma separated value lists for modules that have the capability to handle multiple devices from a single module, such as all the 8390 based drivers, and the PLIP driver. For exmple:

```
options 3c503 io=0x280,0x300,0x330,0x350 xcvr=0,1,0,1
```

TOC

The above would have the one module controlling four 3c503 cards, with card 2 and 4 using external transcievers. Don't put spaces around the `=' or commas.

Also note that a *busy* module can't be removed. That means that you will have to `ifconfig eth0 down` (shut down the ethernet card) before you can remove the module(s).

The command `lsmod` will show you what modules are loaded, whether they are in use, and `rmmod` will remove them.

10.3 Related Documentation

Much of this info came from saved postings from the comp.os.linux groups, which shows that it is a valuable resource of information. Other useful information came from a bunch of small files by Donald himself. Of course, if you are setting up an Ethernet card, then you will want to read the NET-2 Howto so that you can actually configure the software you will use. Also, if you fancy yourself as a bit of a hacker, you can always scrounge some additional info from the driver source files as well. There is usually a paragraph or two in there describing any important points before any actual code starts..

For those looking for information that is not specific in any way to Linux (i.e. what is 10BaseT, what is AUI, what does a hub do, etc.) I strongly recommend making use of the newsgroup *comp.dcom.lans.ethernet* and/or *comp.sys.ibm.pc.hardware.networking*. Newsgroup archives such as those at `dejanews.com` can also be an invaluable source of information. You can grab the newsgroup FAQ from RTFM (which holds all the newsgroup FAQs) at the following URL:

Usenet FAQs

You can also have a look at the `Ethernet-HomePage' so to speak, which is at the following URL:

Ethernet-HomePage

10.4 Disclaimer and Copyright

This document is *not* gospel. However, it is probably the most up to date info that you will be able to find. Nobody is responsible for what happens to your hardware but yourself. If your ethercard or any other hardware goes up in smoke (...nearly impossible!) we take no responsibility. ie. THE AUTHORS ARE NOT RESPONSIBLE FOR ANY DAMAGES INCURRED DUE TO ACTIONS TAKEN BASED ON THE INFORMATION INCLUDED IN THIS DOCUMENT.

This document is Copyright (c) 1993-1997 by Paul Gortmaker. Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copying, provided that this copyright notice is included exactly as in the original, and that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this document into another language, under

the above conditions for modified versions.

A hint to people considering doing a translation. First, translate the SGML source (available via FTP from the HowTo main site) so that you can then generate other output formats. Be sure to keep a copy of the original English SGML source that you translated from! When an updated HowTo is released, get the new SGML source for that version, and then a simple `diff -u old.sgml new.sgml` will show you exactly what has changed so that you can easily incorporate those changes into your translated SMGL source without having to re-read or re-translate everything.

If you are intending to incorporate this document into a published work, please make contact (via e-mail) so that you can be supplied with the most up to date information available. In the past, out of date versions of the Linux HowTo documents have been published, which caused the developers undue grief from being plagued with questions that were already answered in the up to date versions.

10.5 Closing

If you have found any glaring typos, or outdated info in this document, please send an e-mail. It is big, and it is easy to overlook stuff. If you have e-mailed about a change, and it hasn't been included in the next version, please don't hesitate to send it again, as it might have got lost amongst the usual sea of SPAM and junk mail I get.

Thanks!

Paul Gortmaker, `p_gortmaker@yahoo.com`

Linux Access HOWTO

Michael De La Rue, <access-howto@ed.ac.uk>

v2.11, 28 March 1997

The Linux Access HOWTO covers the use of adaptive technology with Linux, In particular, using adaptive technology to make Linux accessible to those who could not use it otherwise. It also covers areas where Linux can be used within more general adaptive technology solutions.

1. Introduction

- 1.1 Distribution Policy

2. Comparing Linux with other Operating Systems

- 2.1 General Comparison
- 2.2 Availability of Adaptive Technology
- 2.3 Inherent Usability

3. Visually Impaired

- 3.1 Seeing the Screen with Low Vision
- 3.2 Aids for Those Who Can't Use Visual Output
- 3.3 Beginning to Learn Linux
- 3.4 Braille Embossing

4. Hearing Problems

- 4.1 Visual Bells

TOC

5. Physical Problems

- 5.1 Unable to Use a Mouse/Pointer
- 5.2 Speech Recognition
- 5.3 Making the Keyboard Behave

6. General Programming Issues

- 6.1 Try to Make it Easy to Provide Multiple Interfaces
- 6.2 Make software configurable.
- 6.3 Test the Software on Users.
- 6.4 Make Output Distinct
- 6.5 Licenses

7. Other Information

- 7.1 Linux Documentation
- 7.2 Mailing Lists
- 7.3 WWW References
- 7.4 Suppliers
- 7.5 Manufacturers

8. Software Packages

- 8.1 Emacspeak
- 8.2 BRLTTY
- 8.3 Screen
- 8.4 Rsynth
- 8.5 xocr
- 8.6 xzoom
- 8.7 NFBtrans
- 8.8 UnWindows

TOC

9. Hardware

- 9.1 Braille terminals driven from Screen Memory
- 9.2 Software Driven Braille Terminals
- 9.3 Speech Synthesisers

10. Acknowledgements

---

1. Introduction

The aim of this document is to serve as an introduction to the technologies which are available to make Linux usable by people who, through some disability would otherwise have problems with it. In other words the target groups of the technologies are, the blind, the partially sighted, deaf and the physically disabled. As any other technologies or pieces of information are discovered they will be added.

The information here not just for these people (although that is probably the main aim) but also to allow developers of Linux to become aware of the difficulties involved here. Possibly the biggest problem is that, right now, very few of the developers of Linux are aware of the issues and various simple ways to make life simpler for implementors of these systems. This has, however, changed noticeably since the introduction of this document, and at least to a small extent because of this document, but also to a large extent due to the work of some dedicated developers, many of whom are mentioned in the document's Acknowledgements.

Please send any comments or extra information or offers of assistance to <access-howto@ed.ac.uk> This address might become a mailing list in future, or be automatically handed over to a future maintainer of the HOWTO, so please don't use it for personal email.

I don't have time to follow developments in all areas. I probably won't even read a mail until I have time to update this document. It's still gratefully received. If a mail is sent to the blind-list or the access-list, I *will* eventually read it and put any useful information into the document. Otherwise, please send a copy of anything interesting to the above email address.

Normal mail can be sent to

```
Linux Access HOWTO
23 Kingsborough Gardens
Glasgow G12 9NH
Scotland
U.K.
```

And will gradually make its way round the world to me. Email will be faster by weeks.

I can be personally contacted using <miked@ed.ac.uk>. Since I use mail filtering on all mail I receive, please use the other address except for personal email. This is most likely to lead to an appropriate response.

1.1 Distribution Policy


```
The ACCESS-HOWTO is copyrighted (c) 1996 Michael De La Rue
The ACCESS-HOWTO may be distributed, at your choice, under
either the terms of the GNU Public License version 2 or later
```

or the standard Linux Documentation project terms. These licenses should be available from where you got this document. Please note that since the LDP terms don't allow modification (other than translation), modified versions can be assumed to be distributed under the GPL.

2. Comparing Linux with other Operating Systems

2.1 General Comparison

The best place to find out about this is in such documents as the `Linux Info Sheet', `Linux Meta FAQ' and `Linux FAQ' (see Linux Documentation). Major reasons for a visually impaired person to use Linux would include it's inbuilt networking which gives full access to the Internet. More generally, users are attracted by the full development environment included. Also, unlike most other modern GUI environments, the graphical front end to Linux (X Windows) is clearly separated from the underlying environment and there is a complete set of modern programs such as World Wide Web browsers and fax software which work directly in the non graphical environment. This opens up the way to provide alternative access paths to the systems functionality; Emacspeak is a good example.

For other users, the comparison is probably less favourable and less clear. People with very specific and complex needs will find that the full development system included allows properly customised solutions. However, much of the software which exists on other systems is only just beginning to become available. More development is being done however in almost all directions.

2.2 Availability of Adaptive Technology

There is almost nothing commercial available *specifically* for Linux. There is a noticeable amount of free software which would be helpful in adaptation, for example, a free speech synthesiser and some free voice control software. There are also a number of free packages which provide good support for certain Braille terminals, for example.

2.3 Inherent Usability

Linux has the vast advantage over Windows that most of it's software is command line oriented. This is now changing and almost everything is now available with a graphical front end. However, because it is in origin a programmers operating system, line oriented programs are still being written covering almost all new areas of interest. For the physically disabled, this means that it is easy to build custom programs to suit their needs. For the visually impaired, this should make use with a speech synthesiser or Braille terminal easy and useful for the foreseeable future.

Linux's multiple virtual consoles system make it practical to use as a multi-tasking operating system by a visually impaired person working directly through Braille.

The windowing system used by Linux (X11) comes with many programming tools, and should be adaptable. However, in practice, the adaptive programs available up till now have been more primitive than those on the Macintosh or Windows. They are, however, completely free (as opposed to hundreds of pounds) and the quality is definitely improving.

In principle it should be possible to put together a complete, usable Linux system for a visually impaired person for about $500 (cheap & nasty PC + sound card). This compares with many thousands of dollars for other operating systems (screen reader software/ speech synthesiser hardware). I have yet to see this. I doubt it would work in practice because the software speech synthesisers available for Linux aren't yet sufficiently good. For a physically disabled person, the limitation will still be the expense of input hardware.

---

3. Visually Impaired

I'll use two general categories here. People who are partially sighted and need help seeing / deciphering / following the text and those who are unable to use any visual interface whatsoever.

3.1 Seeing the Screen with Low Vision

There are many different problems here. Often magnification can be helpful, but that's not the full story. Sometimes people can't track motion, sometimes people can't find the cursor unless it moves. This calls for a range of techniques, the majority of which are only just being added to X.

SVGATextMode

This program is useful for improving the visibility of the normal text screen that Linux provides. The normal screen that Linux provides shows 80 characters across by 25 vertically. This can be changed (and the quality of those characters improved) using SVGATextMode. The program allows full access to the possible modes of an SVGA graphics card. For example, the text can be made larger so that only 50 by 15 characters appear on the screen. There isn't any easy way to zoom in on sections of a screen, but you can resize when needed.

X Window System

For people who can see the screen there are a large number of ways of improving X. They don't add up to a coherent set of features yet, but if set up correctly could solve many problems.

Different Screen Resolutions

The X server can be set up with many different resolutions. A single key press can then change between them allowing difficult to read text to be seen.

In the file `/etc/XF86Config`, you have an entry in the Screen section with a line beginning with modes. If, for example, you set this to

```
    Modes        "1280x1024" "1024x768" "800x600" "640x480" "320x240"
```

with each mode set up correctly (which requires a reasonably good monitor for the highest resolution mode), you will be able to have four times screen magnification, switching between the different levels using

`Ctrl+Alt+Keypad-Plus` and `Ctrl+Alt+Keypad-Minus`

Moving the mouse around the screen will scroll you to different parts of the screen. For more details on how to set this up you should see the documentation which comes with the **XFree86** X server.

TOC

Screen Magnification

There are several known screen magnification programs, `xmag` which will magnify a portion of the screen as much as needed but is very primitive. Another one is `xzoom`. Previously I said that there had to be something better than `xmag`, well this is it. See section xzoom.

Another program which is available is `puff`. This is specifically oriented towards visually impaired users. It provides such features as a box around the pointer which makes it easier to locate. Other interesting features of `puff` are that, if correctly set up, it is able to select and magnify portions of the screen as they are updated. However, there seem to be interacations between `xpuff` and the window manager which could make it difficult to use. When used with my `fvwm` setup, it didn't respond at all to key presses. However using `twm` improved the situation.

The final program which I have seen working is `dynamag`. This again has some specific advantages such as the ability to select a specific area of the screen and monitor it, refreshing the magnified display at regular intervals between a few tenths of a second at twenty seconds. `dynamag` is part of the UnWindows distribution. See UnWindows for more details.

Change Screen Font

The screen fonts all properly written X software should be changeable. You can simply make it big enough for you to read. This is generally accomplished by putting a line the file `.Xdefaults` which should be in your home directory. By putting the correct lines in this you can change the fonts of your programs, for example

```
Emacs.font: -sony-fixed-medium-r-normal--16-150-75-75-c-80-iso8859-*
```

To see what fonts are available, use the program `xfontsel` under X.

There should be some way of changing things at a more fundamental level so that everything comes out with a magnified font. This could be done by renaming fonts, and by telling telling font generating programs to use a different level of scaling. If someone gets this to work properly, please send me the details of how you did it.

Cross Hair Cursors etc..

For people that have problems following cursors there are many things which can help;

- cross-hair cursors (horizontal and vertical lines from the edge of the screen)
- flashing cursors (flashes when you press a key)

No software I know of specifically provides a cross hair cursor. `puff`, mentioned in the previous section does however provide a flashing box around the cursor which can make it considerably easier to locate.

For now the best that can be done is to change the cursor bitmap. Make a bitmap file as you want it, and another one which is the same size, but completely black. Convert them to the XBM format and run

```
xsetroot -cursor cursorfile.xbm black-file.xbm
```

actually, if you understand masks, then the black-file doesn't have to be completely black, but start with it like that. The `.Xdefaults` file controls cursors used by actual applications. For much more information, please see the X Big Cursor mini-HOWTO, by Joerg Schneider <schneid@ira.uka.de>.

Audio

Provided that the user can hear, audio input can be very useful for making a more friendly and communicative computing environment. For a person with low vision, audio clues can be used to help locate the pointer (see UnWindows). For a console mode user using Emacspeak (see Emacspeak), the audio icons available will provide very many useful facilities.

Setting up Linux audio is covered in the Linux Sound HOWTO (see Linux Documentation). Once sound is set up, sounds can be played with the `play` command which is included with most versions of Linux. This is the way to use my version of UnWindows.

Producing Large Print

Using large print with Linux is quite easy. There are several techniques.

LaTeX / TeX

LaTeX is an extremely powerful document preparation system. It may be used to produce large print documents of almost any nature. Though somewhat complicated to learn, many documents are produced using LaTeX or the underlying typesetting program, TeX.

this will produce some reasonably large text

```
\font\magnifiedtenrm=cmr10 at 20pt  % setup a big font
\magnifiedtenrm
this is some large text
\bye
```

For more details, see the LaTeX book which is available in any computer book shop. There are also a large number of introductions available on the internet.

Outputting Large Text

Almost all Linux printing uses postscript, and Linux can drive almost any printer using it. I output large text teaching materials using a standard Epson dot matrix printer.

For users of X, there are various tools available which can produce large Text. These include `LyX`, and many commercial word processors.

3.2 Aids for Those Who Can't Use Visual Output

TOC

For someone who is completely unable to use a normal screen there are two alternatives Braille and Speech. Obviously for people who also have hearing loss, speech isn't always useful, so Braille will always be important.

If you can choose, which should you choose? This is a matter of `vigorous' debate. Speech is rapid to use, reasonably cheap and especially good for textual applications (e.g. reading a long document like this one). Problems include needing a quiet environment, possibly needing headphones to work without disturbing others and avoid being listened in on by them (not available for all speech synthesisers).

Braille is better for applications where precise layout is important (e.g. spreadsheets). Also can be somewhat more convenient if you want to check the beginning of a sentence when you get to the end. Braille is, however, much more expensive and slower for reading text. Obviously, the more you use Braille, the faster you get. Grade II Braille is difficult to learn, but is almost certainly worth it since it is much faster. This means that if you don't use Braille for a fair while you can never discover its full potential and decide for yourself. Anyway, enough said on this somewhat controversial topic.

based on original by James Bowden <jrbowden@bcs.org.uk>

Braille Terminals

Braille terminals are normally a line or two of Braille. Since these are at most 80 characters wide and normally 40 wide, they are somewhat limited. I know of two kinds

- Hardware driven Braille terminals.
- Software driven Braille terminals.

The first kind works only when the computer is in text mode and reads the screen memory directly. See section hardware driven Braille terminals.

The second kind of Braille terminal is similar, in many ways, to a normal terminal screen of the kind Linux supports automatically. Unfortunately, they need special software to make them usable.

There are two packages which help with these. The first, BRLTTY, works with several Braille display types and the authors are keen to support more as information becomes available. Currently BRLTTY supports Tieman B.V.'s CombiBraille series, Alva B.V.'s ABT3 series and Telesensory Systems Inc.'s PowerBraille and Navigator series displays. The use of Blazie Engineering's Braille Lite as a Braille display is discouraged, but support may be renewed on demand. See section Software Braille Terminals.

The other package I am aware of is Braille Enhanced Screen. This is designed to work on other UNIX systems as well as Linux. This should allow user access to a Braille terminal with many useful features such as the ability to run different programs in different `virtual terminals' at the same time.

Speech Synthesis

Speech Synthesisers take (normally) ASCII text and convert it into actual spoken output. It is possible to have these implemented as either hardware or software. Unfortunately, the free Linux speech synthesisers are, reportedly, not good enough to use as a sole means of output.

Hardware speech synthesisers are the alternative. The main one that I know of that works is DECtalk from Digital, driven by `emacspeak`. However, at this time (March 1997) a driver for the Doubletalk synthesiser has been announced. Using `emacspeak` full access to all of the facilities of Linux is fairly easy. This includes the normal use of the shell, a world wide web browser and many other similar features, such as email. Although, it only acts as a plain text reader (similar to IBM's one for the PC) when controling programs it doesn't understand, with those that it does, it can provide much more sophisticated control. See section Emacspeak for more information about `emacspeak`.

Handling Console Output

When it starts up, Linux at present puts all of its messages straight to the normal (visual) screen. This could be changed if anyone with a basic level of kernel programming ability wants to do it. This means that it is impossible for most Braille devices to get information about what Linux is doing before the operating system is completely working.

It is only at that stage that you can start the program that you need for access. If the `BRLTTY` program is used and run very early in the boot process, then from this stage on the messages on the screen can be read. Most hardware and software will still have to wait until the system is completely ready. This makes administering a Linux system difficult, but not impossible for a visually impaired person. Once the system is ready however, you can scroll back by pressing (on the default keyboard layout) Shift-PageUP.

There is one Braille system that can use the console directly, called the Braillex. This is designed to read directly from the screen memory. Unfortunately the normal scrolling of the terminal gets in the way of this. If you are using a Kernel newer than 1.3.75, just type `linux no-scroll` at the LILO prompt or configure LILO to do this automatically. If you have an earlier version of Linux, see section Screen Memory Braille Terminals

The other known useful thing to do is to use sounds to say when each stage of the boot process has been reached. (T.V. Raman suggestion)

Optical Character Recognition

There is a free Optical Character Recognition (OCR) program for Linux called `xocr`. In principle, if it is good enough, this program would allow visually impaired people to read normal books to some extent (accuracy of OCR is never high enough..). However, according to the documentation, this program needs training to recognise the particular font that it is going to use and I have no idea how good it is since I don't have the hardware to test it.

3.3 Beginning to Learn Linux

TOC

Beginning to learn Linux can seem difficult and daunting for someone who is either coming from no computing background or from a pure DOS background. Doing the following things may help:

- Learn to use Linux (or UNIX) on someone else's system before setting up your own.
- Initially control Linux from your own known speaking/Braille terminal. If you plan to use speech, you may want to learn `emacs` now. You can learn it as you go along though. See below
- If you come from an MS-DOS background, read the DOS2Linux Mini HOWTO for help with converting (see The Linux HOWTO Documents).

The Emacspeak HOWTO written by Jim Van Zandt (<jrv@vanzandt.mv.com>) covers this in much more detail (see The Linux HOWTO Documents).

If you are planning to use Emacspeak, you should know that Emacspeak does not attempt to teach Emacs, so in this sense, prior knowledge of Emacs would always be useful. This said, you certainly do not need to know much about Emacs to start using Emacspeak. In fact, once Emacspeak is installed and running, it provides a fluent interface to the rich set of online documentation including the info pages, and makes learning what you need a lot easier.

"In summary: starting to use Emacspeak takes little learning. Getting the full mileage out of Emacs and Emacspeak, especially if you intend using it as a replacement for X Windows as I do does involve eventually becoming familiar with a lot of the Emacs extensions; but this is an incremental process and does not need to be done in a day." - *T.V.Raman*

One other option which may be interesting are the RNIB training tapes which include one covering UNIX. These can be got from

```
RNIB
Customer Services
PO Box 173
Peterborough
Cambridgeshire PE2 6WS
Tel: 01345 023153 (probably only works in UK)
```

3.4 Braille Embossing

Linux should be the perfect platform to drive a Braille embosser from. There are many formatting tools which are aimed specifically at the fixed width device. A Braille embosser can just be connected to the serial port using the standard Linux printing mechanisms. For more info see the Linux Printing HOWTO.

There is a free software package which acts as a multi-lingual grade two translator available for Linux from the American ``National Federation for the Blind''. This is called NFBtrans. See section NFB translator for more details.

---

4. Hearing Problems

For the most part there is little problem using a computer for people with hearing problems. Almost all of the output is visual. There are some situations where sound output is used though. For these, the problem can sometimes be worked round by using visual output instead.

4.1 Visual Bells

By tradition, computers go `beep' when some program sends them a special code. This is generally used to get attention to the program and for little else. Most of the time, it's possible to replace this by making the entire screen (or terminal emulator) flash. How to do this is very variable though.

xterm (under X)

for xterm, you can either change the setting by pressing the middle mouse button while holding down the control key, or by putting a line with just `` `XTerm*visualBell: true' `` (not the quotes of course) in the file `.Xdefaults` in your home directory.

the console (otherwise)

The console is slightly more complex. Please see Alessandro Rubini's Visual Bell mini HOWTO for details on this. Available along with all the other Linux documentation (see section other Linux documents). Mostly the configuration has to be done on a per application basis, or by changing the Linux Kernel its self.

5. Physical Problems

Many of these problems have to be handled individually. The needs of the individual, the ways that they can generate input and other factors vary so much that all that this HOWTO can provide is a general set of pointers to useful software and expertise.

5.1 Unable to Use a Mouse/Pointer

Limited mobility can make it difficult to use a mouse. For some people a tracker ball can be a very good solution, but for others the only possible input device is a keyboard (or even something which simulates a keyboard). For normal use of Linux this shouldn't be a problem (but see the section Making the keyboard behave), but for users of X, this may cause major problems under some circumstances.

Fortunately, the `fvwm` window manager has been designed for use without a pointer and most things can be done using this. I actually do this myself when I lose my mouse (don't ask) or want to just keep typing. `fvwm` is included with all distributions of Linux that I know of. Actually using other programs will depend on their ability to accept key presses. Many X programs do this for all functions. Many don't. I sticky mouse keys, which are supposedly present in the current release of X should make this easier.

Unable to Use a Keyboard

People who are unable to use a keyboard normally can sometimes use one through a headstick or a mouthstick. This calls for special setup of the keyboard. Please see also the section Making the keyboard behave.

Other Input Hardware (X Windows System only)

For others, the keyboard cannot be used at all and only pointing devices are available. In this case, no solution is available under the standard Linux Console and X will have to be used. If the X-Input extension can be taught to use the device and the correct software for converting pointer input to characters can be found (I haven't seen it yet) then any pointing should be usable without a keyboard.

There are a number of devices worth considering for such input such as touch screens and eye pointers. Many of these will need a `device driver' written for them. This is not terribly difficult if the documentation is available, but requires someone with good C programming skills. Please see the *Linux Kernel Hackers guide* and other kernel reference materials for more information. Once this is set up, it should be possible to use these devices like a normal mouse.

Controlling Physical Hardware From Linux

The main group of interest here are the Linux Lab Project. Generally, much GPIB (a standard interface to scientific equipment, also known as the IEEE bus) hardware can be controlled. This potentially gives much potential for very ambitious accessibility projects. As far as I know none have yet been attempted.

5.2 Speech Recognition

Speech recognition is a very powerful tool for enabling computer use. There are two recognition systems that I know of for Linux, the first is `ears` which is described as ``recognition is not optimal. But it is fine for playing and will be improved", the second is `AbbotDemo` ``A speaker independent continuous speech recognition system" which may well be more interesting, though isn't available for commercial use without prior arrangement. See the Linux software map for details (see section other Linux documents).

5.3 Making the Keyboard Behave

X Window System.

The latest X server which comes with Linux can include many features which assist in input. This includes such features as StickKeys, MouseKeys, RepeatKeys, BounceKeys, SlowKeys, and TimeOut. These allow customisation of the keyboard to the needs of the user. These are provided as part of the `XKB>` extension in versions of X after version 6.1. To find out your version and see whether you have the extension installed, you can try.

```
xdpyinfo -queryExtensions
```

Getting Rid of Auto Repeat

To turn off key repeat on the Linux console run this command (I think it has to be run once per console; a good place to run it would be in your login files, `.profile` or `.login` in your home directory).

```
setterm -repeat off
```

To get rid of auto repeat on any X server, you can use the command

```
xset -r
```

which you could put into the file which get runs when you start using X (often `.xsession` or `.xinit` under some setups)

Both of these commands are worth looking at for more ways of changing behaviour of the console.

TOC

Macros / Much input, few key presses

Often in situations such as this, the biggest problem is speed of input. Here the most important thing to aim for is the most number of commands with the fewest key presses. For users of the shell (`bash` / `tcsh`) you should look at the manual page, in particular command and filename completion (press the tab key and bash tries to guess what should come next). For information on macros which provide sequences of commands for just one key press, have a look at the Keystroke HOWTO.

Sticky Keys

Sticky keys are a feature that allow someone who can only reliably press one button at a time to use a keyboard with all of the various modifier keys such as shift and control. These keys, instead of having to be held on at the same time as the other key instead become like the caps lock key and stay on while the other key is pressed. They may then either switch off or stay on for the next key depending on what is needed. For information about how to set this up please see the Linux Keyboard HOWTO, especially section `I can use only one finger to type with' (section 15 in the version I have) for more information on this. - Information from Toby Reed.

---

6. General Programming Issues

Many of the issues worth taking into account are the same when writing software which is designed to be helpful for access as when trying to follow good design.

6.1 Try to Make it Easy to Provide Multiple Interfaces

If your software is only usable through a graphical interface then it can be very hard to make it usable for someone who can't see. If it's only usable through a line oriented interface, then someone who can't type will have difficulties.

Provide keyboard shortcuts as well as the use of the normal X pointer (generally the mouse). You can almost certainly rely on the user being able to generate key presses to your application.

6.2 Make software configurable.

If it's easy to change fonts then people will be able to change to one they can read. If the colour scheme can be changed then people who are colour blind will be more likely to be able to use it. If fonts can be changed easily then the visually impaired will find your software more useful.

6.3 Test the Software on Users.

If you have a number of people use your software, each with different access problems then they will be more likely to point up specific problems. Obviously, this won't be practical for everybody, but you can always ask for feedback.

6.4 Make Output Distinct

Where possible, make it clear what different parts of your program are what. Format error messages in a specific way to identify them. Under X, make sure each pane of your window has a name so that any screen reader software can identify it.

6.5 Licenses

Some software for Linux (though none of the key programs) has license like `not for commercial use'. This could be quite bad for a person who starts using the software for their personal work and then possibly begins to be able to do work they otherwise couldn't with it. This could be something which

frees them from financial and other dependence on others people. Even if the author of the software is willing to make exceptions, it makes the user vulnerable both to changes of commercial conditions (if company buys up the rights) and to refusal from people they could work for (many companies are overly paranoid about licenses). It is much better to avoid this kind of licensing where possible. Protection from commercial abuse of software can be obtained through more specific licenses like the GNU Public License or Artistic License where needed.

---

7. Other Information

7.1 Linux Documentation

The Linux documentation is critical to the use of Linux and most of the documents mentioned here should be included in recent versions of Linux, from any source I know of.

If you want to get the documentation on the Internet, here are some example sites. These should be mirrored at most of the major FTP sites in the world.

- ftp.funet.fi (128.214.6.100) : /pub/OS/Linux/doc/
- tsx-11.mit.edu (18.172.1.2) : /pub/linux/docs/
- sunsite.unc.edu (152.2.22.81) : /pub/Linux/docs/

The Linux Info Sheet

A simple and effective explanation of what Linux is. This is one of the things that you should hand over when you want to explain why you want Linux and what it is good for.

The Linux Info Sheet is available on the World Wide Web from http://sunsite.unc.edu/mdw/HOWTO/INFO-SHEET.html and other mirrors.

The Linux Meta FAQ

A list of other information resources, much more complete than this one. The meta FAQ is available on the World Wide Web from http://sunsite.unc.edu/mdw/HOWTO/META-FAQ.html and other mirrors

The Linux Software Map

The list of software available for Linux on the Internet. Many of the packages listed here were found through this. The LSM is available in a searchable form from http://www.boutell.com/lsm/. It is also available in a single text file in all of the FTP sites mentioned in section Linux Documentation.

The Linux HOWTO documents

The HOWTO documents are the main documentation of Linux. This Access HOWTO is an example of one.

The home site for the Linux Documentation Project which produces this information is http://sunsite.unc.edu/mdw/linux.html. There are also many companies producing these in book form. Contact a local Linux supplier for more details.

The Linux HOWTO documents will be in the directory `HOWTO` in all of the FTP sites mentioned in section Linux Documentation.

The Linux FAQ

A list of `Frequently Asked Questions' with answers which should solve many common questions. The FAQ list is available from http://www.cl.cam.ac.uk/users/iwj10/linux-faq/ as well as all of the FTP sites mentioned in section Linux Documentation.

7.2 Mailing Lists

There are two lists that I know of covering these issues specifically for Linux. There are also others which it is worth researching which cover computer use more generally. Incidentally, if a mail is sent to these lists I *will* read it eventually and include any important information in the Access-HOWTO, so you don't need to send me a separate copy unless it's urgent in some way.

The Linux Access List

This is a general list covering Linux access issues. It is designed `to service the needs of users and developers of the Linux OS and software who are either disabled or want to help make Linux more accessible'. To subscribe send email to <majordomo@ssv1.union.utah.edu> and in the BODY (not the subject) of the email message put:

```
subscribe linux-access <your-email-address>
```

The Linux Blind List

This is a mailing list covering Linux use for blind users. There is also a list of important and useful software being gathered in the list's archive. To subscribe send mail to <blinux-list-request@redhat.com> with the `subject: help`. This list is now moderated.

7.3 WWW References

The World Wide Web is, by it's nature, very rapidly changing. If you are reading this document in an old version then some of these are likely to be out of date. The original version that I maintain on the WWW shouldn't go more than a month or two out of date, so refer to that please.

Linux Documentation is available from http://sunsite.unc.edu/mdw/linux.html

Linux Access On the Web http://www.tardis.ed.ac.uk/~mikedlr/access/ with all of the versions of the HOWTO in http://www.tardis.ed.ac.uk/~mikedlr/access/HOWTO/. Preferably, however, download from one of the main Linux FTP sites. If I get a vast amount of traffic I'll have to close down these pages and move them elsewhere.

TOC

The BLINUX Documentation and Development Project http://leb.net/blinux/. "The purpose of The BLINUX Documentation and Development Project is to serve as a catalyst which will both spur and speed the development of software and documentation which will enable the blind user to run his or her own Linux workstation."

Emacspeak WWW page http://cs.cornell.edu/home/raman/emacspeak/emacspeak.html

BRLTTY unofficial WWW page http://www.sf.co.kr/t.linux/new/brltty.html

Yahoo (one of the most major Internet catalogues) http://www.yahoo.com/Society_and_Culture/Disabilities/Adaptive_Technology/

The Linux Lab Project http://www.fu-berlin.de/~clausi/.

The BLYNX pages: Lynx Support Files Tailored For Blind and Visually Handicapped Users http://leb.net/blinux/blynx/.

7.4 Suppliers

This is a UK supplier for the Braillex.

```
Alphavision Limited
```

7.5 Manufacturers

Alphavision

I think that they are a manufacturer? RNIB only lists them as a supplier, but others say they make the Braillex.

```
Alphavision Ltd
Seymour House
Copyground Lane
High Wycombe
Bucks HP12 3HE
England
U.K.
```

Phone

+44 1494-530 555

Linux Supported Alphavision AT Products

- Braillex

Blazie Engineering

The Braille Lite was supported in the original version of BRLTTY. That support has now been discontinued. If you have one and want to use it with Linux then that may be possible by using this version of the software.

```
Blazie Engineering
105 East Jarrettsville Rd.
Forest Hill, MD 21050
U.S.A.
```

Phone

+1 (410) 893-9333

FAX

+1 (410) 836-5040

BBS

+1 (410) 893-8944

E-Mail

<info@blazie.com>

WWW

http://www.blazie.com/

Blazie AT Products

- Braille Lite (support discontinued)

Digital Equipment Corporation

```
Digital Equipment Corporation
P.O. Box CS2008
Nashua
NH 03061-2008
U.S.A
```

Order

+1 800-722-9332

TOC

Tech info

+1 800-722-9332

FAX

+1 603-884-5597

WWW

http://www.digital.com/

Linux Supported DEC AT Products

- DECTalk Express

Kommunikations-Technik Stolper GmbH

```
KTS Stolper GmbH
Herzenhaldenweg 10
73095 Albershausen
Germany
```

Phone

+49 7161 37023

Fax

+49 7161 32632

Linux Supported KTG AT Products

- Brailloterm

---

8. Software Packages

References in this section are taken directly from the Linux Software map which can be found in all standard places for Linux documentation and which lists almost all of the software available for Linux.

8.1 Emacspeak

Emacspeak is the software side of a speech interface to Linux. Any other character based program, such as a WWW browser, or `telnet` or another editor can potentially be used within `emacspeak`. The main difference between it and normal screen reader software for such operating systems as DOS is that it also has a load more extra features. It is based in the emacs text editor.

A text editor is generally just a program which allows you to change the contents of a file, for example, adding new information to a letter. Emacs is in fact far beyond a normal text editor, and so this package is much more useful than you might imagine. You can run any other program from within emacs, getting any output it generates to appear in the emacs terminal emulator.

The reason that emacs is a better environment for Emacspeak is that it can can understand the layout of the screen and can intelligently interpret the meaning of, for example, a calendar, which would just be a messy array of numbers otherwise. The originator of the package manages to look after his own Linux machine entirely, doing all of the administration from within emacs. He also uses it to control a wide variety of other machines and software directly from that machine.

Emacspeak is included within the Debian Linux distribution and is included as contributed software within the Slakware distribution. This means that it is available on many of the CDROM distributions of Linux. By the time this is published, the version included should be 5 or better, but at present I only have version 4 available for examination.

```
Begin3
Title:          emacspeak - a speech output interface to Emacs
Version:        4.0
Entered-date:   30MAY96
Description:    Emacspeak is the first full-fledged speech output
                system that will allow someone who cannot see to work
                directly on a UNIX system. (Until now, the only option
                available to visually impaired users has been to use a
                talking PC as a terminal.) Emacspeak is built on top
                of Emacs. Once you start emacs with emacspeak loaded,
                you get spoken feedback for everything you do. Your
                mileage will vary depending on how well you can use
                Emacs.  There is nothing that you cannot do inside
                Emacs:-)
Keywords:       handicap access visually impaired blind speech emacs
Author:         raman@adobe.com (T. V. Raman)
Maintained-by:  jrv@vanzandt.mv.com (Jim Van Zandt)
Primary-site:   sunsite.unc.edu apps/sound/speech
                124kB   emacspeak-4.0.tgz
Alternate-site:
Original-site:  http://www.cs.cornell.edu /pub/raman/emacspeak
                123kB   emacspeak.tar.gz/Info/People/raman/emacspeak/emacspeak.tar.gz
Platforms:      DECtalk Express or DEC Multivoice speech synthesizer,
                GNU FSF Emacs 19 (version 19.23 or later) and TCLX
                7.3B (Extended TCL).
Copying-policy: GPL
End
```

8.2 BRLTTY

This is a program for running a serial port Braille terminal. It has been widely tested and used, and supports a number of different kinds of hardware (see the Linux Software Map entry below).

The maintainer is, Nikhil Nair <nn201@cus.cam.ac.uk>. The other people working on it are Nicolas Pitre <nico@cam.org> and Stephane Doyon <doyons@jsp.umontreal.ca>. Send any comments to all of them.

The authors seem keen to get support in for more different devices, so if you have one you should consider contacting them. They will almost certainly need programming information for the device, so if you can contact your manufacturer and get that they are much more likely to be able to help you.

A brief feature list (from their README file) to get you interested

- Full implementation of the standard screen review facilities.
- A wide range of additional optional features, including blinking cursor and capital letters, screen freezing for leisurely review, attribute display to locate highlighted text, hypertext links, etc.
- `Intelligent' cursor routing. This allows easy movement of the cursor in text editors etc. without moving the hands from the Braille display.
- A cut & paste function. This is particularly useful for copying long filenames, complicated commands etc.
- An on-line help facility.
- Support for multiple Braille codes.
- Modular design allows relatively easy addition of drivers for other Braille displays, or even (hopefully) porting to other Unix-like platforms.

```
Begin3
Title:          BRLTTY - Access software for Unix for a blind person
                         using a soft Braille terminal
Version:        1.0.2, 17SEP96
Entered-date:   17SEP96
Description:    BRLTTY is a daemon which provides access to a Unix console
                for a blind person using a soft Braille display (see the
                README file for a full explanation).

                BRLTTY only works with text-mode applications.

                We hope that this system will be expanded to support
                other soft Braille displays, and possibly even other
                Unix-like platforms.
Keywords:       Braille console access visually impaired blind
Author:         nn201@cus.cam.ac.uk (Nikhil Nair)
                nico@cam.org (Nicolas Pitre)
                doyons@jsp.umontreal.ca (Stephane Doyon)
                jrbowden@bcs.org.uk (James Bowden)
Maintained-by:  nn201@cus.cam.ac.uk (Nikhil Nair)
Primary-site:   sunsite.unc.edu /pub/Linux/system/Access
                110kb brltty-1.0.2.tar.gz (includes the README file)
                  6kb brltty-1.0.2.README
                  1kb brltty-1.0.2.lsm
Platforms:      Linux (kernel 1.1.92 or later) running on a PC or DEC Alpha.
                Not X/graphics.
                Supported Braille displays (serial communication only):
                  - Tieman B.V.: CombiBraille 25/45/85;
                  - Alva B.V.: ABT3xx series;
                  - Telesensory Systems Inc.: PowerBraille 40 (not 65/80),
                    Navigator 20/40/80 (latest firmware version only?).
Copying-policy: GPL
```

```
End
```

8.3 Screen

Screen is a standard piece of software to allow many different programs to run at the same time on one terminal. It has been enhanced to support some Braille terminals (those from Telesensory) directly.

8.4 Rsynth

This is a speech synthesiser listed in the Linux Software Map. It doesn't apparently work well enough for use by a visually impaired person. Use hardware instead, or improve it.. a free speech synthesiser would be really really useful.

8.5 xocr

`xocr` is a package which implements optical character recognition for Linux. As with `Rsynth`, I don't think that this will be acceptable as a package for use as a sole means of input by a visually impaired person. I suspect that the algorithm used means that it will need to be watched over by someone who can check that it is reading correctly. I would love to be proved wrong.

8.6 xzoom

`xzoom` is a screen magnifier, in the same vein as `xmag`, but sufficiently better to be very useful to a visually impaired person. The main disadvantages of `xzoom` are that it can't magnify under itself, that some of the key controls aren't compatible with `fvwm`, the normal Linux window manager and that it's default configuration doesn't run over a network (this can be fixed at some expense to speed). Apart from that though, it's excellent. It does continuous magnification which allows you to, for example, scroll a document up and down, whilst keeping the section you are reading magnified. Alternatively, you can move a little box around the screen, magnifying the contents and letting you search for the area you want to see. `xzoom` is also available as an rpm from the normal RedHat sites, making it very easy to install for people using the rpm system (such as Redhat users).

```
Begin3
Title:          xzoom
Version:        0.1
Entered-date:   Mar 30 1996
Description:    xzoom can magnify (by integer value) rotate
                (by a multiple if 90 degrees) and mirror about
                the X or Y axes areas on X11 screen
                and display them in it's window.
Keywords:       X11 zoom magnify xmag
Author:         Itai Nahshon <nahshon@best.com>
Maintained-by:  Itai Nahshon <nahshon@best.com>
Primary-site:   sunsite.unc.edu
                probably in /pub/Linux/X11/xutils/xzoom-0.1.tgz
Platforms:      Linux+11. Support only for 8-bit depth.
                Tested only in Linux 1.3.* with the XSVGA 3.1.2
driver.
                                Needs the XSHM extension.
Copying-policy: Free
End
```

8.7 NFBtrans

`nfbtrans` is a multi-grade Braille translation program distributed by the National Federation for the Blind in the U.S.A. It is released for free in the hope that someone will improve it. Languages covered are USA English, UK English, Spanish, Russian, Esperanto, German, Biblical Hebrew and Biblical Greek, though others could be added just by writing a translation table. Also covered are some computer and math forms. I have managed to get it to compile under Linux, though, not having a Braille embosser

available at the present moment I have not been able to test it.

NFBtrans is available from ftp://nfb.org/ftp/nfb/braille/nfbtrans/. After downloading it, you will have to compile it.

Compiling NFBtrans on Linux

I have returned this patch to the maintainer of NFBtrans and he says that he has included it, so if you get a version later than 740, you probably won't have to do anything special. Just follow the instructions included in the package.

```
unzip -L NFBTR740.ZIP   #or whatever filename you have
mv makefile Makefile
```

Next save the following to a file (e.g. `patch-file`)

```
*** nfbpatch.c.orig     Tue Mar 12 11:37:28 1996
--- nfbpatch.c  Tue Mar 12 11:37:06 1996
***************
*** 185,190 ****
--- 185,193 ----
    return (finfo.st_size);
  }                /* filelength */

+ #ifndef linux
+ /* pretty safe to assume all linux has usleep I think ?? this should be
+ done properly anyway */
  #ifdef SYSVR4
  void usleep(usec)
    int usec;
***************
*** 195,200 ****
--- 198,204 ----
UKP  }                /* usleep */

  #endif
+ #endif

  void beep(count)
    int count;
```

and run

```
patch < patch-file
```

then type

```
make
```

and the program should compile.

8.8 UnWindows

UnWindows is a package of access utilities for X which provides many useful facilities for the visually impaired (not blind). It includes a screen magnifier and other customised utilities to help locate the pointer. UnWindows can be downloaded from ftp://ftp.cs.rpi.edu/pub/unwindows.

As it comes by default, the package will not work on Linux because it relies on special features of Suns. However, some of the utilities do work and I have managed to port most of the rest so this package may be interesting to some people. My port will either be incorporated back into the original or will be available in the BLINUX archives (see WWW references). The remaining utility

which doesn't yet work is the configuration utility.

In my version the programs, instead of generating sounds themselves, just call another program. The other program could for example be

```
play /usr/lib/games/xboing/sounds/ouch.au
```

Which would make the `xboing` ouch noise, for example it could do this as the pointer hit the left edge of the screen.

dynamag

`dynamag` is a screen magnification program. please see the section on Screen magnification (magnification). This program worked in the default distribution.

coloreyes

`coloreyes` makes it easy to find the pointer (mouse) location. It consists of a pair of eyes which always look in the direction of the pointer (like xeyes) and change color depending on how far away the mouse is (unlike xeyes). This doesn't work in the default distribution, but the test version, at the same location, seems to work.

border

`border` is a program which detects when the pointer (mouse) has moved to the edge of the screen and makes a sound according to which edge of the screen has been approached. The version which is available uses a SUN specific sound system. I have now changed this so that instead of that, it just runs a command, which could be any Linux sound program.

un-twm

The window manager is a special program which controls the location of all of the other windows (programs) displayed on the X screen. `un-twm` is a special version which will make a sound as the pointer enters different windows. The sound will depend on what window has been entered. The distributed version doesn't work on linux because, like `border` it relies on SUN audio facilities. Again I already have a special version which will be avaliable by the time you read this.

---

9. Hardware

9.1 Braille terminals driven from Screen Memory

These are Braille terminals that can read the screen memory directly in a normal text mode. It is possible to use it to work with Linux for almost all of the things that a seeing user can do on the console, including installation. However, it has a problem with the scrolling of the normal Linux kernel, so a kernel patch needs to be applied. See Patching the Kernel for Braillex and Brailloterm.

Braillex

The Braillex is a terminal which is designed to read directly from the Screen memory, thus getting round any problems with MS-DOS programs which don't behave strangely. If you could see it on screen, then this terminal should be able to display it in Braille. In Linux, unfortunately, screen handling is done differently from MS-DOS, so this has to be changed somewhat.

To get this terminal to work, you have to apply the patch given below in section Patching the Kernel. Once this is done, the Braillex becomes one of the most convenient ways to use Linux as it allows all of the information normally available to a seeing person to be read. Other terminals don't start working until the operating system has completely booted.

The Braillex is available with two arrangements of Braille cells (80x1 or 40x2) and there is a model, called the IB 2-D which also has a vertical bar to show information about all of the lines of the screen (using 4 programmable dots per screen line)

```
Price: 8,995  (pounds sterling) or 11495 UKP for 2-D
Manufacturer: Alphavision Limited (UK)
Suppliers: ????
```

Brailloterm

``What is Brailloterm?

It's a refreshable display Braille, made by KTS Kommunikations-Technik Stolper GmbH. It has 80 Braille cells in an unique line. Each cell has 8 dots that are combined (up/down) to represent a character. By default, Brailloterm shows me the line in which the screen cursor is. I can use some functions in Brailloterm to see any line in the screen." - *Jose Vilmar Estacio de Souza* <jvilmar@embratel.net.br>

Jose then goes on to say that the terminal can also use the serial port under DOS but that it needs a special program. I don't know if any of the ones for Linux would work.

As with Braillex, this needs a special patch to the kernel work properly. See section Patching the Kernel.

```
Price: about 23.000,- DM /  $ 15.000,
Manufacturer: Kommunikations-Technik Stolper GmbH
Suppliers: ????
```

Patching the Kernel for Braillex and Brailloterm

This probably also applies to any other terminals which read directly from screen memory to work under MS-DOS. Mail me to confirm any terminals that you find work. This does not apply and will actually lose some features for terminals driven using the BRLTTY software.

I am told this patch applies to all Kernels version 1.2.X. It should also work on all Kernel versions from 1.1.X to 1.3.72, with just a warning from patch (I've tested that the patch applies to 1.3.68 at least). **From 1.3.75 the patch is no longer needed**

because the Kernel can be configured not to scroll using `linux no-scroll' at the LILO prompt. See the Boot Prompt HOWTO for more details.

```
*** drivers/char/console.c~     Fri Mar 17 07:31:40 1995
--- drivers/char/console.c      Tue Mar  5 04:34:47 1996
***************
*** 601,605 ****
  static void scrup(int currcons, unsigned int t, unsigned int b)
  {
!       int hardscroll = 1;

        if (b > video_num_lines || t >= b)
--- 601,605 ----
  static void scrup(int currcons, unsigned int t, unsigned int b)
  {
!       int hardscroll = 0;

        if (b > video_num_lines || t >= b)
```

To apply it:

1. Save the above text to a file (say patch-file)
2. change to the drivers/char directory of your kernel sources
3. run

```
patch < patch-file
```

4. Compile your kernel as normal

Apply those patches and you should be able to use the Braille terminal as normal to read the Linux Console.

Put in words, the patch just means `change the 1 to a 0 in the first line of the function `scrup` which should be near line 603 in the file drivers/char/console.c'. The main thing about `patch` is that program understands this, and that it knows how to guess what to do when the Linux developers change things in that file.

If you want to use a more modern kernel with completely disabled scrolling, (instead of the boot prompt solution I already mentioned), please use the following patch. **This does not apply to kernels earlier than 1.3.75**.

```
*** console.c~  Fri Mar 15 04:01:45 1996
--- console.c   Thu Apr  4 13:29:48 1996
***************
*** 516,520 ****
  unsigned char has_wrapped;          /* all of videomem is data of fg_console */
  static unsigned char hardscroll_enabled;
! static unsigned char hardscroll_disabled_by_init = 0;

  void no_scroll(char *str, int *ints)
--- 516,520 ----
  unsigned char has_wrapped;          /* all of videomem is data of fg_console */
  static unsigned char hardscroll_enabled;
! static unsigned char hardscroll_disabled_by_init = 1;

  void no_scroll(char *str, int *ints)
```

9.2 Software Driven Braille Terminals

TOC

The principle of operation of these terminal is very close to that of a CRT terminal such as the vt100. They connect to the serial port and the computer has to run a program which sends them output. At present there are two known programs for Linux. BRLTTY, see section BRLTTY) and Braille enhanced screen.

Tieman B.V.

CombiBraille

This Braille terminal is supported by the BRLTTY software. It comes in three versions with 25, 45 or 85 Braille cells. The extra five cells over a standard display are used for status information.

```
Price: around 4600 UKP for the 45 cell model ...
Manufacturer: Tieman B.V.
Suppliers: Concept Systems, Nottingham, England (voice +44 115 925 5988)
```

Alva B.V.

The ABT3xx series is supported in BRLTTY. Only the ABT340 has been confirmed to work at this time. Please pass back information to the BRLTTY authors on other models.

```
Price: 20 cell - 2200 UKP; 40 cell 4500 UKP; 80 cell 8000 UKP
Manufacturer: Alva
Suppliers: Professional Vision Services LTD, Hertshire, England
           (+44 1462 677331)
```

Telesensory Systems Inc. displays

Because they have provided programming information to the developers, the Telesensory displays are supported both by BRLTTY and screen.

Powerbraille

There are three models the 40, the 65 and the 80. Only the 40 is known to be supported by BRLTTY.

```
Price: 20 cell - 2200 UKP; 40 cell 4500 UKP; 80 cell 8000 UKP
Manufacturer: Alva
Suppliers: Professional Vision Services LTD, Hertshire, England
           (+44 1462 677331)
```

Navigator

Again there are three models the 20, the 60 and the 80. Recent versions are all known to work with BRLTTY but whether earlier ones (with earlier firmware) also work has not been confirmed.

```
Price: 80 cell 7800 UKP
Manufacturer: Alva
Suppliers: Professional Vision Services LTD, Hertshire, England
           (+44 1462 677331)
```

TOC

Braille Lite

This is more a portable computer than a terminal. It could, however, be used with BRLTTY version 0.22 (but not newer versions) as if it was a normal Braille terminal. Unfortunately, many of the features available with the CombiBraille cannot be used with the Braille Lite. This means that it should be avoided for Linux use where possible.

```
Price: $3,395.00
Manufacturer: Blazie Engineering
```

9.3 Speech Synthesisers

Speech synthesisers normally connect to the serial port of a PC. Useful features include

- Braille labels on parts
- Many voices to allow different parts of document to be spoken differently
- Use with headphones (not available on all models)

The critical problem is that the quality of the speech. This is much more important to someone who is using the speech synthesiser as their main source of information than to someone who is just getting neat sounds out of a game. For this reason T.V. Raman seems to only recommend the DECTalk. Acceptable alternatives would be good.

DECTalk Express

This is a hardware speech synthesiser. It is recommended for use with Emacspeak and in fact the DECTalk range are the only speech synthesisers which work with that package at present. This synthesiser has every useful feature that I know about. The only disadvantage that I know of at present is price.

```
Price: $1195.00
Manufacturer: Digital Equipment Corporation

Suppliers: Many.  I'd like details of those with Specific Linux
        support / delivering international or otherwise of note only
        please.  Otherwise refer to local organisations.
        Digital themselves or the Emacspeak WWW pages.
```

Accent SA

This is a synthesiser made by Aicom Corporation. An effort has begun to write a driver for it however help is needed. Please see http://www.cyberspc.mb.ca/~astrope/speak.html if you think you can help.

SPO256-AL2 Speak and Spell chip.

Some interest has been expressed in using this chip in self built talking circuits. I'd be interested to know if anyone has found this useful. A software package `speak-0.2pl1.tar.gz` was produced by David Sugar <dyfet@tycho.com>. My suspicion, though, is that the quality of the output wouldn't be good enough for regular use.

---

10. Acknowledgements

Much of this document was created from various information sources on the Internet, many found from Yahoo and DEC's Alta Vista Search engine. Included in this was the documentation of most of the software packages mentioned in the text. Some information was also gleaned from the Royal National Institute for the Blind's helpsheets.

T.V. Raman, the author of Emacspeak has reliably contributed comments, information and text as well as putting me in touch with other people who he knew on the Internet.

Kenneth Albanowski <kjahds@kjahds.com> provided the patch needed for the Brailloterm and information about it.

Roland Dyroff of S.u.S.E. GmbH (Linux distributors and makers of S.u.S.E. Linux (English/German)) looked up KTS Stolper GmbH at my request and got some hardware details and information on the Brailloterm.

The most major and careful checks over of this document were done by James Bowden, <jrbowden@bcs.org.uk> and Nikhil Nair <nn201@cus.cam.ac.uk>, the BRLTTY authors who suggested a large number of corrections as well as extra information for some topics.

The contributors to the blinux and linux-access mailing lists have contributed to this document by providng information for me to read.

Mark E. Novak of the Trace R&D centre http://trace.wisc.edu/ pointed me in the direction of several packages of software and information which I had not seen before. He also made some comments on the structure of the document which I have partially taken into account and should probably do more about.

Other contributors include Nicolas Pitrie and Stephane Doyon.

A number of other people have contributed comments and information. Specific contributions are acknowledged within the document.

This version was specifically produced for RedHat's Dr. Linux book. This is because they provided warning of it's impending release to myself and other LDP authors. Their doing this is strongly appreciated since wrong or old information sits around much longer in a book than on the Internet.

No doubt you made a contribution and I haven't mentioned it. Don't worry, it was an accident. I'm sorry. Just tell me and I will add you to the next version.

Brief Introduction to Alpha Systems and Processors

Neal Crook, Digital Equipment (Editor: David Mosberger)

V0.11, 6 June 1997

This document is a brief overview of existing Alpha CPUs, chipsets and systems. It has something of a hardware bias, reflecting my own area of expertese. Although I am an employee of Digital Equipment Corporation, this is not an official statement by Digital and any opinions expressed are mine and not Digital's.

1. What is Alpha
2. What is Digital Semiconductor
3. Alpha CPUs
4. 21064 performance vs 21066 performance
5. A Few Notes On Clocking
6. The chip-sets
7. The Systems
8. Bytes and all that stuff

TOC

9. PALcode and all that stuff

10. Porting

11. More Information

12. References

---

1. What is Alpha

"Alpha" is the name given to Digital's 64-bit RISC architecture. The Alpha project in Digital began in mid-1989, with the goal of providing a high-performance migration path for VAX customers. This was not the first RISC architecture to be produced by Digital, but it was the first to reach the market. When Digital announced Alpha, in March 1992, it made the decision to enter the merchant semicondutor market by selling Alpha microprocessors.

Alpha is also sometimes referred to as Alpha AXP, for obscure and arcane reasons that aren't worth persuing. Suffice it to say that they are one and the same.

2. What is Digital Semiconductor

Digital Semiconductor (DS) is the business unit within Digital Equipment Corporation (Digital - we don't like the name DEC) that sells semiconductors on the merchant market. Digital's products include CPUs, support chipsets, PCI-PCI bridges and PCI peripheral chips for comms and multimedia.

3. Alpha CPUs

There are currently 2 generations of CPU core that implement the Alpha architecture:

- EV4
- EV5

Opinions differ as to what "EV" stands for (Editor's note: the true answer is of course "Electro Vlassic" [1]), but the number represents the first generation of Digital's CMOS technology that the core was implemented in. So, the EV4 was originally implemented in CMOS4. As time goes by, a CPU tends to get a mid-life performance kick by being optically shrunk into the next generation of CMOS process. EV45, then, is the EV4 core implemented in CMOS5 process. There is a big difference between shrinking a design into a particular technology and implementing it from scratch in that technology (but I don't want to go into that now). There are a few other wildcards in here: there is also a CMOS4S (optical shrink in CMOS4) and a CMOS5L.

True technophiles will be interested to know that CMOS4 is a 0.75 micron process, CMOS5 is a 0.5 micron process and CMOS6 is a 0.35 micron process.

To map these CPU cores to *chips* we get:

21064-150,166

EV4 (originally), EV4S (now)

21064-200

EV4S

21064A-233,275,300

EV45

21066

LCA4S (EV4 core, with EV4 FPU)

21066A-233

LCA45 (EV4 core, but with EV45 FPU)

21164-233,300,333

EV5

21164A-417

EV56

21264

EV6

TOC

The EV4 core is a dual-issue (it can issue 2 instructions per CPU clock) superpipelined core with int unit, floating point unit and branch prediction. It is fully bypassed and has 64-bit internal data paths and tightly coupled 8Kbyte caches, one each for Instruction and Data. The caches are write-through (they never get dirty).

The EV45 core has a couple of tweaks to the EV4 core: it has a slightly improved floating point unit, and 16KB caches, one each for Instruction and Data (it also has cache parity). (Editor's note: Neal Crook indicated in a separate mail that the changes to the floating point unit (FPU) improve the performance of the divider. The EV4 FPU divider takes 34 cycles for a single-precision divide and 63 cycles for a double-precision divide (non data-dependent). In constrast, the EV45 divider takes typically 19 cycles (34 cycles max) for single-precision and typically 29 cycles (63 cycles max) for a double-precision division (data-dependent).)

The EV5 core is a quad-issue core, also superpipelined, fully bypassed etc etc. It has tightly-coupled 8Kbyte caches, one each for I and D. These caches are write-through. It also has a tightly-coupled 96Kbyte on-chip second-level cache (the Scache) which is 3-way set associative and write-back (it can be dirty). The EV4->EV5 performance increase is better than just the increase achieved by clock speed improvements. As well as the bigger caches and quad issue, there are microarchitectural improvements to reduce producer/consumer latencies in some paths.

The EV56 core is fundamentally the same microarchitecture as the EV5, but it adds some new instructions for 8 and 16-bit loads and stores (see Section Bytes and all that stuff). These are primarily intended for use by device drivers. The EV56 core is implemented in CMOS6, which is a 2.0V process.

The 21064 was anounced in March 1992. It uses the EV4 core, with a 128-bit bus interface. The bus interface supports the 'easy' connection of an external second-level cache, with a block size of 256-bits (2 data beats on the bus). The Bcache timing is completely software configurable. The 21064 can also be configured to use a 64-bit external bus, (but I'm not sure if any shipping system uses this mode). The 21064 does not impose any policy on the Bcache, but it is usually configured as a write-back cache. The 21064 does contain hooks to allow external hardware to maintain cache coherence with the Bcache and internal caches, but this is hairy.

The 21066 uses the EV4 core and integrates a memory controller and PCI host bridge. To save pins, the memory controller has a 64-bit data bus (but the internal caches have a block size of 256 bits, just like the 21064, therefore a block fill takes 4 beats on the bus). The memory controller supports an external Bcache and external DRAMs. The timing of the Bcache and DRAMs is completely software configurable, and can be controlled to the resolution of the CPU clock period. Having a 4-beat process to fill a cache block isn't as bad as it sounds because the DRAM access is done in page mode. Unfortunately, the memory controller doesn't support any of the new esoteric DRAMs (SDRAM, EDO or BEDO) or synchronous cache RAMs. The PCI bus interface is fully rev2.0 compliant and runs at upto 33MHz.

The 21164 has a 128-bit data bus and supports split reads, with upto 2 reads outstanding at any time (this allows 100% data bus utilisation under best-case dream-on conditions, i.e., you can theoretically transfer 128-bits of data on every bus clock). The 21164 supports easy connection of an external 3-rd level cache (Bcache) and has all the hooks to allow external systems to maintain full cache coherence with all caches. Therefore, symmetric multiprocessor designs are 'easy'.

The 21164A was announced in October, 1995. It uses the EV56 core. It is nominally pin-compatible the 21164, but requires split power rails; all of the power pins that were +3.3V power on the 21164 h now been split into two groups; one group provided 2.0V power to the CPU core, the other group supplies 3.3V to the I/O cells. Unlike older implementations, the 21164 pins are not 5V-tolerant. The end result of this change is that 21164 systems are, in general, not upgradeable to the 21164A (though note that it would be relatively straightforward to design a 21164A system that could also accommodate a 21164). The 21164A also has a couple of new pins to support the new 8 and 16-bit loads and stores. It also improves the 21164 support for using synchronus SRAMs to implement the external Bcache.

---

4. 21064 performance vs 21066 performance

The 21064 and the 21066 have the same (EV4) CPU core. If the same program is run on a 21064 and a 21066, at the same CPU speed, then the difference in performance comes only as a result of system Bcache/memory bandwidth. Any code thread that has a high hit-rate on the *internal* caches will perform the same. There are 2 big performance killers:

1. Code that is write-intensive. Even though the 21064 and the 21066 have write buffers to swallow some of the delays, code that is write-intensive will be throttled by write bandwidth at the system bus. This arises because the on-chip caches are write-through.
2. Code that wants to treat floats as integers. The Alpha architecture does not allow register-register transfers from integer registers to floating point registers. Such a conversion has to be done via memory (And therefore, because the on-chip caches are write-through, via the Bcache). (Editor's note: it seems that both the EV4 and EV45 can perform the conversion through the primary data cache (Dcache), provided that the memory is cached already. In such a case, the store in the conversion sequence will update the Dcache and the subsequent load is, under certain circumstances, able to read the updated d-cache value, thus avoiding a costly roundtrip to the Bcache. In particular, it seems best to execute the stq/ldt or stt/ldq instructions back-to-back, which is somewhat counter-intuitive.)

If you make the same comparison between a 21064A and a 21066A, there is an additional factor due to the different Icache and Dcache sizes between the two chips.

Now, the 21164 solves both these problems: it achieve *much* higher system bus bandwidths (despite having the same number of signal pins - yes, I *know* it's got about twice as many pins as a 21064, but all those extra ones are power and ground! (yes, really!!)) and it has write-back caches. The only remaining problem is the answer to the question "how much does it cost?"

5. A Few Notes On Clocking

All of the current Alpha CPUs use high-speed clocks, because their microarchitectures have been designed as so-called short-tick designs. None of the sytem busses have to run at horrendous speeds as a result though:

- on the 21066(A), 21064(A), 21164 the off-chip cache (Bcache) timing is completely programmable, to the resolution of the CPU clock. For example, on a 275MHz CPU, the Bcache read access time can be controller with a resolution of 3.6ns
- on the 21066(A), the DRAM timing is completely programmable, to the resolution of the CPU clock (*not* the PCI clock, the CPU clock).
- on the 21064(A), 21164(A), the system bus frequency is a sub-multiple of the CPU clock frequency. Most of the 21064 motherboards use a 33MHz system bus clock.
- Systems that use the 21066 can run the PCI at any frequency relative to the CPU. Generally, the PCI runs at 33MHz.
- Systems that use the APECs chipset (see Section The chip-sets) always have their CPU system bus equal to their PCI bus frequency. This means that both busses tends to run at either 25MHz or 33MHz (since these are the frequencies that scale up to match the CPU frequencies). On APECs systems, the DRAM controller timings are software programmable in terms of the CPU system bus frequency

Aside: someone suggested that they were getting bad performance on a 21066 because the 21066 memory controller was only running at 33MHz. Actually, it's the superfast 21064A systems that have memory controllers that 'only' run at 33MHz.

6. The chip-sets

DS sells two CPU support chipsets. The 2107x chipset (aka APECS) is a 21064(A) support chiset. The 2117x chipset (aka ALCOR) is a 21164 support chipset. There will also be 2117xA chipset (aka ALCOR 2) as a 21164A support chipset.

Both chipsets provide memory controllers and PCI host bridges for their CPU. APECS provides a 32-bit PCI host bridge, ALCOR provides a 64-bit PCI host bridge which (in accordance with the requirements of the PCI spec) can support both 32-bit and 64-bit PCI devices.

APECS consists of 6, 208-pin chips (4, 32-bit data slices (DECADE), 1 system controller (COMANCHE), 1 PCI controller (EPIC)). It provides a DRAM controller (128-bit memory bus) and a PCI interface. It also does all the work to maintain memory coherence when a PCI device DMAs into (or out of) memory.

ALCOR consists of 5 chips (4, 64-bit data slices (Data Switch, DSW) - 208-pin PQFP and 1 control (Control, I/O Address, CIA) - a 383 pin plastic PGA). It provides a DRAM controller (256-bit memory bus) and a PCI interface. It also does all the work required to support an external Bcache and to maintain memory coherence when a PCI device DMAs into (or out of) memory.

There is no support chipset for the 21066, since the memory controller and PCI host bridge functionality are integrated onto the chip.

7. The Systems

The applications engineering group in DS produces example designs using the CPUs and support chipsets. These are typically PC-AT size motherboards, with all the functionality that you'd typically find on a high-end Pentium motherboard. Originally, these example designs were intended to be used as starting points for third-parties to produce motherboard designs from. These first-generation designs were called Evaluation Boards (EBs). As the amount of engineering required to build a motherboard has increased (due to higher-speed clocks and the need to meet RF emission and susceptibility regulations) the emphasis has shifted towards providing motherboards that are suitable for volume manufacture.

Digital's system groups have produced several generations of machines using Alpha processors. Some of these systems use support logic that is designed by the systems groups, and some use commodity chipsets from DS. In some cases, systems use a combination of both.

Various third-parties build systems using Alpha processors. Some of these companies design systems from scratch, and others use DS support chipsets, clone/modify DS example designs or simply package systems using build and tested boards from DS.

The EB64: Obsolete design using 21064 with memory controller implemented using programmable logic. I/O provided by using programmable logic to interface a 486<->ISA bridge chip. On-board Ethernet, SuperI/O (2S, 1P, FD), Ethernet and ISA. PC-AT size. Runs from standard PC power supply.

The EB64+: Uses 21064 or 21064A and APECs. Has ISA and PCI expansion (3 ISA, 2 PCI, one pair are on a shared slot). Supports 36-bit DRAM SIMs. ISA bus generated by Intel SaturnI/O PCI-ISA bridge. On-board SCSI (NCR 810 on PCI) Ethernet (Digital 21040), KBD, MOUSE (PS2 style), SuperI/O (2S, 1P, FD), RTC/NVRAM. Boot ROM is EPROM. PC-AT size. Runs from standard PC power supply.

The EB66: Uses 21066 or 21066A. I/O sub-system is identical to EB64+. Baby PC-AT size. Runs from standard PC power supply. The EB66 schematic was published as a marketing poster advertising the 21066 as "the first microprocessor in the world with embedded PCI" (for trivia fans: there are actually 2 versions of this poster - I drew the circuits and wrote the spiel for the first version, and some Americans mauled the spiel for the second version)

The EB164: Uses 21164 and ALCOR. Has ISA and PCI expansion (3 ISA slots, 2 64-bit PCI slots (one is shared with an ISA slot) and 2 32-bit PCI slots. Uses plus-in Bcache SIMMs. I/O sub-system provides SuperI/O (2S, 1P, FD), KBD, MOUSE (PS2 style), RTC/NVRAM. Boot ROM is Flash. PC-AT-sized motherboard. Requires power supply with 3.3V output.

The AlphaPC64 (aka Cabriolet): derived from EB64+ but now baby-AT with Flash boot ROM, no on-board SCSI or Ethernet. 3 ISA slots, 4 PCI slots (one pair are on a shared slot), uses plug-in Bcache SIMMs. Requires power supply with 3.3V output.

The AXPpci33 (aka NoName), is based on the EB66. This design is produced by Digital's Technical OEM (TOEM) group. It uses the 21066 processor running at 166MHz or 233MHz. It is a baby-AT size, and runs from a standard PC power supply. It has 5 ISA slots and 3 PCI slots (one pair are a shared slot).

There are 2 versions, with either PS/2 or large DIN connectors for the keyboard.

Other 21066-based motherboards: most if not all other 21066-based motherboards on the market are also based on EB66 - there's really not many system options when designing a 21066 system, because all the control is done on-chip.

Multia (aka the Universal Desktop Box): This is a very compact pedestal desktop system based on the 21066. It includes 2 PCMCIA sockets, 21030 (TGA) graphics, 21040 Ethernet and NCR 810 SCSI disk along with floppy, 2 serial ports and a parallel port. It has limited expansion capability (one PCI slot) due to its compact size. (There is some restriction on when you can use the PCI slot, can't remember what) (Note that 21066A-based and Pentium-based Multia's are also available).

DEC PC 150 AXP (aka Jensen): This is a very old Digital system - one of the first-generation Alpha systems. It is only mentioned here because a number of these systems seem to be available on the second-hand market. The Jensen is a floor-standing tower system which used a 150MHz 21064 (later versions used faster CPUs but I'm not sure what speeds). It used programmable logic to interface a 486 EISA I/O bridge to the CPU.

Other 21064(A) systems: There are 3 or 4 motherboard designs around (I'm not including Digital *systems* here) and all the ones I know of are derived from the EB64+ design. These include:

- EB64+ (some vendors package the board and sell it unmodified); AT form-factor.
- Aspen Systems motherboard: EB64+ derivative; baby-AT form-factor.
- Aspen Systems server board: many PCI slots (includes PCI bridge).
- AlphaPC64 (aka Cabriolet), baby AT form-factor.

Other 21164(A) systems: The only one I'm aware of that isn't simply an EB164 clone is a system made by DeskStation. That system is implemented using a memory and I/O controller proprietary to Desk Station. I don't know what their attitude towards Linux is.

---

8. Bytes and all that stuff

When the Alpha architecture was introduced, it was unique amongst RISC architectures for eschewing 8-bit and 16-bit loads and stores. It supported 32-bit and 64-bit loads and stores (longword and quadword, in Digital's nomenclature). The co-architects (Dick Sites, Rich Witek) justified this decision by citing the advantages:

1. Byte support in the cache and memory sub-system tends to slow down accesses for 32-bit and 64-bit quantities.
2. Byte support makes it hard to build high-speed error-correction circuitry into the cache/memory sub-system.

Alpha compensates by providing powerful instructions for manipulating bytes and byte groups within 64-bit registers. Standard benchmarks for string operations (e.g., some of the Byte benchmarks) show that Alpha performs very well on byte manipulation.

The absence of byte loads and stores impacts some software semaphores and impacts the design of I/O sub-systems. Digital's solution to the I/O problem is to use some low-order address lines to specify the data size during I/O transfers, and to decode these as byte enables. This so-called Sparse Addressing wastes address space and has the consequence that I/O space is non-contiguous (more on the intricacies of Sparse Addressing when I get around to writing it). Note that I/O space, in this context, refers to all system resources present on the PCI and therefore includes both PCI memory space and PCI I/O space.

With the 21164A introduction, the Alpha archtecture was ECO'd to include byte addressing. Executing these new instructions on an earlier CPU will cause an OPCDEC PALcode exception, so that the PALcode will handle the access. This will have a performance impact. The ramifications of this are that use of these new instructions (IMO) should be restricted to device drivers rather than applications code.

These new byte load and stores mean that future support chipsets will be able to support contiguous I/O space.

9. PALcode and all that stuff

This is a placeholder for a section explaining PALcode. I will write it if there is sufficient interest.

10. Porting

The ability of any Alpha-based machine to run Linux is really only limited by your ability to get information on the gory details of its innards. Since there are Linux ports for the E66, EB64+ and EB164 boards, all systems based on the 21066, 21064/APECS or 21164/ALCOR should run Linux with little or no modification. The major thing that is different between any of these motherboards is the way that they route interrupts. There are three sources of interrupts:

- on-board devices
- PCI devices
- ISA devices

All the systems use an Intel System I/O bridge (SIO) to act as a bridge between PCI and ISA (the main I/O bus is PCI, the ISA bus is a secondary bus used to support slow-speed and 'legacy' I/O devices). The SIO contains the traditional pair of daisy-chained 8259s.

Some systems (e.g., the Noname) route all of their interrupts through the SIO and thence to the CPU. Some systems have a separate interrupt controller and route all PCI interrupts plus the SIO interrupt (8259 output) through that, and all ISA interrupts through the SIO.

Other differences between the systems include:

- how many slots they have
- what on-board PCI devices they have
- whether they have Flash or EPROM

11. More Information

All of the DS evaluation boards and motherboard designs are license-free and the whole documentation kit for a design costs about \$50. That includes all the schematics, programmable parts sources, data sheets for CPU and support chipset. The doc kits are available from Digital Semiconductor distributors. I'm not suggesting that many people will want to rush out and buy this, but I do want to point out that the information is available.

Hope that was helpful. Comments/updates/suggestions for expansion to Neal Crook.

12. References

[1] Bill Hamburgen, Jeff Mogul, Brian Reid, Alan Eustace, Richard Swan, Mary Jo Doherty, and Joel Bartlett. *Characterization of Organic Illumination Systems*. DEC WRL, Technical Note 13, April 1989.

Assembly HOWTO

François-René Rideau fare@tunes.org

v0.4q, 22 June 1999

This is the Linux Assembly HOWTO. This document describes how to program in assembly using FREE programming tools, focusing on development for or from the Linux Operating System on i386 platforms. Included material may or may not be applicable to other hardware and/or software platforms. Contributions about these would be gladly accepted. keywords: assembly, assembler, free, macroprocessor, preprocessor, asm, inline asm, 32-bit, x86, i386, gas, as86, nasm

1. INTRODUCTION

- 1.1 Legal Blurp
- 1.2 Important Note
- 1.3 Foreword
- 1.4 History
- 1.5 Credits

2. DO YOU NEED ASSEMBLY?

- 2.1 Pros and Cons
- 2.2 How to NOT use Assembly

3. ASSEMBLERS

- 3.1 GCC Inline Assembly
- 3.2 GAS
- 3.3 GASP
- 3.4 NASM
- 3.5 AS86
- 3.6 OTHER ASSEMBLERS

4. METAPROGRAMMING/MACROPROCESSING

- 4.1 What's integrated into the above
- 4.2 External Filters

5. CALLING CONVENTIONS

- 5.1 Linux
- 5.2 DOS
- 5.3 Winblows and suches
- 5.4 Yer very own OS

6. TODO & POINTERS

---

1. INTRODUCTION

1.1 Legal Blurp

Copyright © 1996-1999 by François-René Rideau.

This document is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

1.2 Important Note

This is an interactively evolving document: you are especially invited to ask questions, to answer to questions, to correct given answers, to add new FAQ answers, to give pointers to other software, to point the current maintainer to bugs or deficiencies in the pages. If you're motivated, you could even *take over the maintenance of the HOWTO*. In one word, contribute!

To contribute, please contact whoever appears to maintain the Assembly-HOWTO. At the time of this writing, it's me, i.e. François-René Rideau.

However, it's been some time since I've been looking for a serious hacker to replace me as maintainer of this document. Disadvantages are you must spend some time updating and correcting the document, and learning the LDP publication tools. Advantages are you get some fame *and* you can receive complimentary copies of HOWTO compendiums.

1.3 Foreword

This document aims at answering frequently asked questions of people who program or want to program 32-bit x86 assembly using *free software*, particularly under the Linux operating system. It may also point to other documents about non-free, non-x86, or non-32-bit assemblers, though such is not its primary goal.

Because the main interest of assembly programming is to build to write the guts of operating systems, interpreters, compilers, and games, where a C compiler fails to provide the needed expressiveness (performance is more and more seldom an issue), we stress on development of such software.

How to use this document

This document contains answers to some frequently asked questions. At many places, Universal Resource Locators (URL) are given for some software or documentation repository. Please see that the most useful repositories are mirrored, and that by accessing a nearer mirror site, you relieve the whole

Internet from unneeded network traffic, while saving your own precious time. Particularly, there are large repositories all over the world, that mirror other popular repositories. You should learn and not what are those places near you (networkwise). Sometimes, the list of mirrors is listed in a file, or in a login message. Please heed the advice. Else, you should ask archie about the software you're looking for...

The most recent version for this documents sits in http://www.tunes.org/~fare/files/asm/Assembly-HOWTO.en.sgml but what's in Linux HOWTO repositories should be fairly up to date, too (I can't know): http://metalab.unc.edu/LDP/HOWTO/. A french translation of this HOWTO can be found around ftp://ftp.lip6.fr/pub/linux/french/HOWTO/.

Other related documents

- If you don't know what *free* software is, please do read *carefully* the GNU General Public License, which is used in a lot of free software, and is a model for most of their licenses. It generally comes in a file named `COPYING`, with a library version in a file named `COPYING.LIB`. Literature from the FSF (free software foundation) might help you, too.
- Particularly, the interesting kind of free software comes with sources that you can consult and correct, or sometimes even borrow from. Read your particular license carefully, and do comply to it.
- There is a FAQ for comp.lang.asm.x86 that answers generic questions about x86 assembly programming, and questions about some commercial assemblers in a 16-bit DOS environment. Some of it apply to free 32-bit asm programming, so you may want to read this FAQ... http://www2.dgsys.com/~raymoon/faq/asmfaq.zip
- FAQs and docs exist about programming on your favorite platform, whichever it is, that you should consult for platform-specific issues not directly related to programming in assembler.

1.4 History

Each version includes a few fixes and minor corrections, which needs not be repeatedly mentionned every time.

Version 0.1 23 Apr 1996

Francois-Rene "Faré" Rideau <fare@tunes.org> creates and publishes the first mini-HOWTO, because ``I'm sick of answering ever the same questions on comp.lang.asm.x86"

Version 0.2 4 May 1996

*

Version 0.3c 15 Jun 1996

*

Version 0.3f 17 Oct 1996

*

TOC

Version 0.3g 2 Nov 1996

Created the History. Added pointers in cross-compiling section. Added section about I/O programming under Linux (particularly video).

Version 0.3h 6 Nov 1996

more about cross-compiling -- See on sunsite: devel/msdos/

Version 0.3i 16 Nov 1996

NASM is getting pretty slick

Version 0.3j 24 Nov 1996

point to french translated version

Version 0.3k 19 Dec 1996

What? I had forgotten to point to terse???

Version 0.3l 11 Jan 1997

*

Version 0.4pre1 13 Jan 1997

text mini-HOWTO transformed into a full linuxdoc-sgml HOWTO, to see what the SGML tools are like.

Version 0.4 20 Jan 1997

first release of the HOWTO as such.

Version 0.4a 20 Jan 1997

CREDITS section added

Version 0.4b 3 Feb 1997

NASM moved: now is before AS86

Version 0.4c 9 Feb 1997

Added section "DO YOU NEED ASSEMBLY?"

Version 0.4d 28 Feb 1997

Vapor announce of a new Assembly-HOWTO maintainer.

Version 0.4e 13 Mar 1997

Release for DrLinux

Version 0.4f 20 Mar 1997

*

Version 0.4g 30 Mar 1997

*

Version 0.4h 19 Jun 1997

still more on "how not to use assembly"; updates on NASM, GAS.

Version 0.4i 17 July 1997

info on 16-bit mode access from Linux.

Version 0.4j 7 September 1997

*

Version 0.4k 19 October 1997

*

Version 0.4l 16 November 1997

release for LSL 6th edition.

Version 0.4m 23 March 1998

corrections about gcc invocation

Version 0.4o 1 December 1998

*

Version 0.4p 6 June 1999

clean up and updates.

Version 0.4q 22 June 1999

process argument passing (argc,argv,environ) in assembly.

This is yet another ``last release by Faré before new maintainer takes over". Only nobody knows who the new maintainer might be.

1.5 Credits

I would like to thanks the following persons, by order of appearance:

- Linus Torvalds for Linux
- Bruce Evans for bcc from which as86 is extracted
- Simon Tatham and Julian Hall for NASM
- Greg Hankins and now Tim Bynum for maintaining HOWTOs
- Raymond Moon for his FAQ
- Eric Dumas for his translation of the mini-HOWTO into french (sad thing for the original author to be french and write in english)
- Paul Anderson and Rahim Azizarab for helping me, if not for taking over the HOWTO.
- Marc Lehman for his insight on GCC invocation.

- Abhijit Menon-Sen for helping me figure out the process argument passing convention
- All the people who have contributed ideas, remarks, and moral support.

2. DO YOU NEED ASSEMBLY?

Well, I wouldn't want to interfere with what you're doing, but here are a few advice from hard-earned experience.

2.1 Pros and Cons

The advantages of Assembly

Assembly can express very low-level things:

- you can access machine-dependent registers and I/O.
- you can control the exact behavior of code in critical sections that might otherwise involve deadlock between multiple software threads or hardware devices.
- you can break the conventions of your usual compiler, which might allow some optimizations (like temporarily breaking rules about memory allocation, threading, calling conventions, etc).
- you can build interfaces between code fragments using incompatible such conventions (e.g. produced by different compilers, or separated by a low-level interface).
- you can get access to unusual programming modes of your processor (e.g. 16 bit mode to interface startup, firmware, or legacy code on Intel PCs)
- you can produce reasonably fast code for tight loops to cope with a bad non-optimizing compiler (but then, there are free optimizing compilers available!)
- you can produce code where (but only on CPUs with known instruction timings, which generally excludes all current ....
- you can produce hand-optimized code that's perfectly tuned for your particular hardware setup, though not to anyone else's.
- you can write some code for your new language's optimizing compiler (that's something few will ever do, and even they, not often).

The disadvantages of Assembly

Assembly is a very low-level language (the lowest above hand-coding the binary instruction patterns). This means

- it's long and tedious to write initially,
- it's very bug-prone,
- your bugs will be very difficult to chase,
- it's very difficult to understand and modify, i.e. to maintain.
- the result is very non-portable to other architectures, existing or future,

TOC

- your code will be optimized only for a certain implementation of a same architecture: for insta among Intel-compatible platforms, each CPU design and its variations (relative latency, throughput, and capacity, of processing units, caches, RAM, bus, disks, presence of FPU, MMX extensions, etc) implies potentially completely different optimization techniques. CPU designs already include Intel 386, 486, Pentium, PPro, Pentium II; Cyrix 5x86, 6x86; AMD K5, K6. New designs keep popping up, so don't expect either this listing or your code to be up-to-date.
- your code might also be unportable accross different OS platforms on the same architecture, by lack of proper tools. (well, GAS seems to work on all platforms; NASM seems to work or be workable on all intel platforms).
- you spend more time on a few details, and can't focus on small and large algorithmic design, that are known to bring the largest part of the speed up. [e.g. you might spend some time building very fast list/array manipulation primitives in assembly; only a hash table would have sped up your program much more; or, in another context, a binary tree; or some high-level structure distributed over a cluster of CPUs]
- a small change in algorithmic design might completely invalidate all your existing assembly code. So that either you're ready (and able) to rewrite it all, or you're tied to a particular algorithmic design;
- On code that ain't too far from what's in standard benchmarks, commercial optimizing compilers outperform hand-coded assembly (well, that's less true on the x86 architecture than on RISC architectures, and perhaps less true for widely available/free compilers; anyway, for typical C code, GCC is fairly good);
- And in any case, as says moderator John Levine on comp.compilers, ``compilers make it a lot easier to use complex data structures, and compilers don't get bored halfway through and generate reliably pretty good code." They will also *correctly* propagate code transformations throughout the whole (huge) program when optimizing code between procedures and module boundaries.

Assessment

All in all, you might find that though using assembly is sometimes needed, and might even be useful in a few cases where it is not, you'll want to:

- minimize the use of assembly code,
- encapsulate this code in well-defined interfaces
- have your assembly code automatically generated from patterns expressed in a higher-level language than assembly (e.g. GCC inline assembly macros).
- have automatic tools translate these programs into assembly code
- have this code be optimized if possible
- All of the above, i.e. write (an extension to) an optimizing compiler back-end.

Even in cases when Assembly is needed (e.g. OS development), you'll find that not so much of it is, and that the above principles hold.

See the sources for the Linux kernel about it: as little assembly as needed, resulting in a fast, reliable, portable, maintainable OS. Even a successful game like DOOM was almost massively written in C, with a tiny part only being written in assembly for speed up.

2.2 How to NOT use Assembly

TOC

General procedure to achieve efficient code

As says Charles Fiterman on comp.compilers about human vs computer-generated assembly code,

``The human should always win and here is why.

- First the human writes the whole thing in a high level language.
- Second he profiles it to find the hot spots where it spends its time.
- Third he has the compiler produce assembly for those small sections of code.
- Fourth he hand tunes them looking for tiny improvements over the machine generated code.

The human wins because he can use the machine."

Languages with optimizing compilers

Languages like ObjectiveCAML, SML, CommonLISP, Scheme, ADA, Pascal, C, C++, among others, all have free optimizing compilers that'll optimize the bulk of your programs, and often do better than hand-coded assembly even for tight loops, while allowing you to focus on higher-level details, and without forbidding you to grab a few percent of extra performance in the above-mentionned way, once you've reached a stable design. Of course, there are also commercial optimizing compilers for most of these languages, too!

Some languages have compilers that produce C code, which can be further optimized by a C compiler. LISP, Scheme, Perl, and many other are suches. Speed is fairly good.

General procedure to speed your code up

As for speeding code up, you should do it only for parts of a program that a profiling tool has consistently identified as being a performance bottleneck.

Hence, if you identify some code portion as being too slow, you should

- first try to use a better algorithm;
- then try to compile it rather than interpret it;
- then try to enable and tweak optimization from your compiler;
- then give the compiler hints about how to optimize (typing information in LISP; register usage with GCC; lots of options in most compilers, etc).
- then possibly fallback to assembly programming

Finally, before you end up writing assembly, you should inspect generated code, to check that the problem really is with bad code generation, as this might really not be the case: compiler-generated code might be better than what you'd have written, particularly on modern multi-pipelined architectures! Slow parts of a program might be intrinsically so. Biggest problems on modern architectures with fast processors are due to delays from memory access, cache-misses, TLB-misses, and page-faults; register optimization becomes useless, and you'll more profitably re-think data structures and threading to

achieve better locality in memory access. Perhaps a completely different approach to the problem mi help, then.

Inspecting compiler-generated code

There are many reasons to inspect compiler-generated assembly code. Here are what you'll do with such code:

- check whether generated code can be obviously enhanced with hand-coded assembly (or by tweaking compiler switches)
- when that's the case, start from generated code and modify it instead of starting from scratch
- more generally, use generated code as stubs to modify, which at least gets right the way your assembly routines interface to the external world
- track down bugs in your compiler (hopefully rarer)

The standard way to have assembly code be generated is to invoke your compiler with the `-S` flag. This works with most Unix compilers, including the GNU C Compiler (GCC), but YMMV. As for GCC, it will produce more understandable assembly code with the `-fverbose-asm` command-line option. Of course, if you want to get good assembly code, don't forget your usual optimization options and hints!

---

3. ASSEMBLERS

3.1 GCC Inline Assembly

The well-known GNU C/C++ Compiler (GCC), an optimizing 32-bit compiler at the heart of the GNU project, supports the x86 architecture quite well, and includes the ability to insert assembly code in C programs, in such a way that register allocation can be either specified or left to GCC. GCC works on most available platforms, notably Linux, *BSD, VSTa, OS/2, *DOS, Win*, etc.

Where to find GCC

The original GCC site is the GNU FTP site ftp://prep.ai.mit.edu/pub/gnu/gcc/ together with all the released application software from the GNU project. Linux-configured and precompiled versions can be found in ftp://metalab.unc.edu/pub/Linux/GCC/ There exists a lot of FTP mirrors of both sites. everywhere around the world, as well as CD-ROM copies.

GCC development has split in two branches recently. See more about the experimental version, egcs, at http://www.cygnus.com/egcs/

Sources adapted to your favorite OS, and binaries precompiled for it, should be found at your usual FTP sites.

For most popular DOS port of GCC is named DJGPP, and can be found in directories of such name in FTP sites. See:

http://www.delorie.com/djgpp/

There is also a port of GCC to OS/2 named EMX, that also works under DOS, and includes lots of unix-emulation library routines. See around the following site: ftp://ftp-os2.cdrom.com/pub/os2/emx09c/. Other URLs listed in previous versions of this HOWTO seem to be as dead as OS/2.

Where to find docs for GCC Inline Asm

The documentation of GCC includes documentation files in texinfo format. You can compile them with tex and print then result, or convert them to .info, and browse them with emacs, or convert them to .html, or nearly whatever you like. convert (with the right tools) to whatever you like, or just read as is. The .info files are generally found on any good installation for GCC.

The right section to look for is: `C Extensions::Extended Asm::`

Section `Invoking GCC::Submodel Options::i386 Options::` might help too. Particularly, it gives the i386 specific constraint names for registers: abcdSDB correspond to `%eax`, `%ebx`, `%ecx`, `%edx`, `%esi`, `%edi`, `%ebp` respectively (no letter for `%esp`).

TOC

The DJGPP Games resource (not only for game hackers) had this page specifically about assembly, but it's down. Its data have nonetheless been recovered on the DJGPP site, that contains a mine of other useful information: http://www.delorie.com/djgpp/doc/brennan/

GCC depends on GAS for assembling, and follow its syntax (see below); do mind that inline asm needs percent characters to be quoted so they be passed to GAS. See the section about GAS below.

Find *lots* of useful examples in the `linux/include/asm-i386/` subdirectory of the sources for the Linux kernel.

Invoking GCC to have it properly inline assembly code ?

Because assembly routines from the kernel headers (and most likely your own headers, if you try making your assembly programming as clean as it is in the linux kernel) are embedded in `extern inline` functions, GCC must be invoked with the `-O` flag (or `-O2`, `-O3`, etc), for these routines to be available. If not, your code may compile, but not link properly, since it will be looking for non-inlined `extern` functions in the libraries against which your program is being linked !!! Another way is to link against libraries that include fallback versions of the routines.

Inline assembly can be disabled with `-fno-asm`, which will have the compiler die when using extended inline asm syntax, or else generate calls to an external function named `asm()` that the linker can't resolve. To counter such flag, `-fasm` restores treatment of the `asm` keyword.

More generally, good compile flags for GCC on the x86 platform are

```
gcc -O2 -fomit-frame-pointer -W -Wall
```

`-O2` is the good optimization level in most cases. Optimizing besides it takes longer, and yields code that is a lot larger, but only a bit faster; such overoptimization might be useful for tight loops only (if any), which you may be doing in assembly anyway. In cases when you need really strong compiler optimization for a few files, do consider using up to `-O6`.

`-fomit-frame-pointer` allows generated code to skip the stupid frame pointer maintenance, which makes code smaller and faster, and frees a register for further optimizations. It precludes the easy use of debugging tools (`gdb`), but when you use these, you just don't care about size and speed anymore anyway.

`-W -Wall` enables all warnings and helps you catch obvious stupid errors.

You can add some cpu-specific `-m486` or such flag so that GCC will produce code that is more adapted to your precise computer. Note that EGCS (and perhaps GCC 2.8) have `-mpentium` and such flags, whereas GCC 2.7.x and older versions do not. A good choice of CPU-specific flags should be in the Linux kernel. Check the texinfo documentation of your current GCC installation for more.

`-m386` will help optimize for size, hence also for speed on computers whose memory is tight and/or loaded, since big programs cause swap, which more than counters any "optimization" intended by the larger code. In such settings, it might be useful to stop using C, and use instead a language that favors

TOC

code factorization, such as a functional language and/or FORTH, and use a bytecode- or wordcode-based implementation.

Note that you can vary code generation flags from file to file, so that performance-critical files use maximal optimization, whereas other files be optimized for size.

To optimize even more, option `-mregparm=2` and/or corresponding function attribute might help, but might pose lots of problems when linking to foreign code, *including the libc*. There are ways to correctly declare foreign functions so the right call sequences be generated, or you might want to recompile the foreign libraries to use the same register-based calling convention...

Note that you can add make these flags the default by editing file `/usr/lib/gcc-lib/i486-linux/2.7.2.3/specs` or wherever that is on your system (better not add -Wall there, though). The exact location of the GCC specs files on *your* system can be found by asking `gcc -v`.

3.2 GAS

GAS is the GNU Assembler, that GCC relies upon.

Where to find it

Find it at the same place where you found GCC, in a package named binutils.

What is this AT&T syntax

Because GAS was invented to support a 32-bit unix compiler, it uses standard ``AT&T'' syntax, which resembles a lot the syntax for standard m68k assemblers, and is standard in the UNIX world. This syntax is no worse, no better than the ``Intel'' syntax. It's just different. When you get used to it, you find it much more regular than the Intel syntax, though a bit boring.

Here are the major caveats about GAS syntax:

- Register names are prefixed with `%`, so that registers are `%eax`, `%dl` and suches instead of just `eax`, `dl`, etc. This makes it possible to include external C symbols directly in assembly source, without any risk of confusion, or any need for ugly underscore prefixes.
- The order of operands is source(s) first, and destination last, as opposed to the intel convention of destination first and sources last. Hence, what in intel syntax is `mov ax,dx` (move contents of register `dx` into register `ax`) will be in att syntax `mov %dx, %ax`.
- The operand length is specified as a suffix to the instruction name. The suffix is `b` for (8-bit) byte, `w` for (16-bit) word, and `l` for (32-bit) long. For instance, the correct syntax for the above instruction would have been `movw %dx,%ax`. However, gas does not require strict att syntax, so the suffix is optional when length can be guessed from register operands, and else defaults to 32-bit (with a warning).
- Immediate operands are marked with a `$` prefix, as in `addl $5,%eax` (add immediate long value 5 to register `%eax`).

TOC

- No prefix to an operand indicates it is a memory-address; hence `movl $foo,%eax` puts the *address* of variable `foo` in register `%eax`, but `movl foo,%eax` puts the *contents* of variable `foo` in register `%eax`.
- Indexing or indirection is done by enclosing the index register or indirection memory cell address in parentheses, as in `testb $0x80,17(%ebp)` (test the high bit of the byte value at offset 17 from the cell pointed to by `%ebp`).

A program exists to help you convert programs from TASM syntax to AT&T syntax. See ftp://x2ftp.oulu.fi/pub/msdos/programming/convert/ta2asv08.zip. (Since the original x2ftp site is closing, use a mirror site). There also exists a program for the reverse conversion: http://www.multimania.com/placr/a2i.html.

GAS has comprehensive documentation in TeXinfo format, which comes at least with the source distribution. Browse extracted .info pages with Emacs or whatever. There used to be a file named gas.doc or as.doc around the GAS source package, but it was merged into the TeXinfo docs. Of course, in case of doubt, the ultimate documentation is the sources themselves! A section that will particularly interest you is `Machine Dependencies::i386-Dependent::`

Again, the sources for Linux (the OS kernel), come in as good examples; see under linux/arch/i386, the following files: `kernel/*.S, boot/compressed/*.S, mathemu/*.S`

If you are writing kind of a language, a thread package, etc you might as well see how other languages (OCaml, gforth, etc), or thread packages (QuickThreads, MIT pthreads, LinuxThreads, etc), or whatever, do it.

Finally, just compiling a C program to assembly might show you the syntax for the kind of instructions you want. See section Do you need Assembly? above.

Limited 16-bit mode

GAS is a 32-bit assembler, meant to support a 32-bit compiler. It currently has only limited support for 16-bit mode, which consists in prepending the 32-bit prefixes to instructions, so you write 32-bit code that runs in 16-bit mode on a 32 bit CPU. In both modes, it supports 16-bit register usage, but what is unsupported is 16-bit addressing. Use the directive `.code16` and `.code32` to switch between modes. Note that an inline assembly statement `asm(".code16\n")` will allow GCC to produce 32-bit code that'll run in real mode!

I've been told that most code needed to fully support 16-bit mode programming was added to GAS by Bryan Ford (please confirm?), but at least, it doesn't show up in any of the distribution I tried, up to binutils-2.8.1.x ... more info on this subject would be welcome.

A cheap solution is to define macros (see below) that somehow produce the binary encoding (with `.byte`) for just the 16-bit mode instructions you need (almost nothing if you use code16 as above, and can safely assume the code will run on a 32-bit capable x86 CPU). To find the proper encoding, you can get inspiration from the sources of 16-bit capable assemblers for the encoding.

TOC

3.3 GASP

GASP is the GAS Preprocessor. It adds macros and some nice syntax to GAS.

Where to find GASP

GASP comes together with GAS in the GNU binutils archive.

How it works

It works as a filter, much like cpp and the like. I have no idea on details, but it comes with its own texinfo documentation, so just browse them (in .info), print them, grok them. GAS with GASP looks like a regular macro-assembler to me.

3.4 NASM

The Netwide Assembler project is producing yet another i386 assembler, written in C, that should be modular enough to eventually support all known syntaxes and object formats.

Where to find NASM

http://www.cryogen.com/Nasm

Binary release on your usual metalab mirror in `devel/lang/asm/` Should also be available as .rpm or .deb in your usual RedHat/Debian distributions' contrib.

What it does

At the time this HOWTO is written, version 0.98 of NASM is just out.

The syntax is Intel-style. Some macroprocessing support is integrated.

Supported object file formats are `bin`, `aout`, `coff`, `elf`, `as86`, (DOS) `obj`, `win32`, (their own format) `rdf`.

NASM can be used as a backend for the free LCC compiler (support files included).

Surely NASM evolves too fast for this HOWTO to be kept up to date. Unless you're using BCC as a 16-bit compiler (which is out of scope of this 32-bit HOWTO), you should definitely use NASM instead of say AS86 or MASM, because it is actively supported online, and runs on all platforms.

Note: NASM also comes with a disassembler, NDISASM.

Its hand-written parser makes it much faster than GAS, though of course, it doesn't support three bazillion different architectures. For the x86 target, it should be the assembler of choice...

3.5 AS86

TOC

AS86 is a 80x86 assembler, both 16-bit and 32-bit, part of Bruce Evans' C Compiler (BCC). It has mostly Intel-syntax, though it differs slightly as for addressing modes.

Where to get AS86

A completely outdated version of AS86 is distributed by HJLu just to compile the Linux kernel, in a package named bin86 (current version 0.4), available in any Linux GCC repository. But I advise no one to use it for anything else but compiling Linux. This version supports only a hacked minix object file format, which is not supported by the GNU binutils or anything, and it has a few bugs in 32-bit mode, so you really should better keep it only for compiling Linux.

The most recent versions by Bruce Evans (bde@zeta.org.au) are published together with the FreeBSD distribution. Well, they were: I could not find the sources from distribution 2.1 on :(Hence, I put the sources at my place: http://www.tunes.org/~fare/files/asm/bcc-95.3.12.src.tgz

The Linux/8086 (aka ELKS) project is somehow maintaining bcc (though I don't think they included the 32-bit patches). See around http://www.linux.org.uk/ELKS-Home/ (or http://www.elks.ecs.soton.ac.uk) and ftp://linux.mit.edu/pub/linux/ELKS/. I haven't followed these developments, and would appreciate a reader contributing on this topic.

Among other things, these more recent versions, unlike HJLu's, supports Linux GNU a.out format, so you can link you code to Linux programs, and/or use the usual tools from the GNU binutils package to manipulate your data. This version can co-exist without any harm with the previous one (see according question below).

BCC from 12 march 1995 and earlier version has a misfeature that makes all segment pushing/popping 16-bit, which is quite annoying when programming in 32-bit mode. I wrote a patch at a time when the TUNES Project used as86: http://www.tunes.org/~fare/files/asm/as86.bcc.patch.gz. Bruce Evans accepted this patch, but since as far as I know he hasn't published a new release of bcc, the ones to ask about integrating it (if not done yet) are the ELKS developers.

How to invoke the assembler?

Here's the GNU Makefile entry for using bcc to transform `.s` asm into both GNU a.out `.o` object and `.l` listing:

```
%.o %.l:        %.s
        bcc -3 -G -c -A-d -A-l -A$*.l -o $*.o $<
```

Remove the `%.l`, `-A-l`, and `-A$*.l`, if you don't want any listing. If you want something else than GNU a.out, you can see the docs of bcc about the other supported formats, and/or use the objcopy utility from the GNU binutils package.

TOC

Where to find docs

The docs are what is included in the bcc package. I salvaged the man pages that used to be available from the FreeBSD site at http://www.tunes.org/~fare/files/asm/bcc-95.3.12.src.tgz. Maybe ELKS developers know better. When in doubt, the sources themselves are often a good docs: it's not very well commented, but the programming style is straightforward. You might try to see how as86 is used in ELKS or Tunes 0.0.0.25...

What if I can't compile Linux anymore with this new version ?

Linus is buried alive in mail, and since HJLu (official bin86 maintainer) chose to write hacks around an obsolete version of as86 instead of building clean code around the latest version, I don't think my patch for compiling Linux with a modern as86 has any chance to be accepted if resubmitted. Now, this shouldn't matter: just keep your as86 from the bin86 package in /usr/bin, and let bcc install the good as86 as /usr/local/libexec/i386/bcc/as where it should be. You never need explicitly call this ``good'' as86, because bcc does everything right, including conversion to Linux a.out, when invoked with the right options; so assemble files exclusively with bcc as a frontend, not directly with as86.

3.6 OTHER ASSEMBLERS

These are other, non-regular, options, in case the previous didn't satisfy you (why?), that I don't recommend in the usual (?) case, but that could prove quite useful if the assembler must be integrated in the software you're designing (i.e. an OS or development environment).

Win32Forth assembler

Win32Forth is a *free* 32-bit ANS FORTH system that successfully runs under Win32s, Win95, Win/NT. It includes a free 32-bit assembler (either prefix or postfix syntax) integrated into the reflective FORTH language. Macro processing is done with the full power of the reflective language FORTH; however, the only supported input and output contexts is Win32For itself (no dumping of .obj file, but you could add that feature yourself, of course). Find it at ftp://ftp.forth.org/pub/Forth/Compilers/native/windows/Win32For/.

Terse

Terse is a programming tool that provides *THE* most compact assembler syntax for the x86 family! However, it is evil proprietary software. It is said that there was a project for a free clone somewhere, that was abandonned after worthless pretenses that the syntax would be owned by the original author. Thus, if you're looking for a nifty programming project related to assembly hacking, I invite you to develop a terse-syntax frontend to NASM, if you like that syntax.

Non-free and/or Non-32bit x86 assemblers.

TOC

You may find more about them, together with the basics of x86 assembly programming, in Raymond Moon's FAQ for comp.lang.asm.x86: http://www2.dgsys.com/~raymoon/faq/asmfaq.zip.

Note that all DOS-based assemblers should work inside the Linux DOS Emulator, as well as other similar emulators, so that if you already own one, you can still use it inside a real OS. Recent DOS-based assemblers also support COFF and/or other object file formats that are supported by the GNU BFD library, so that you can use them together with your free 32-bit tools, perhaps using GNU objcopy (part of the binutils) as a conversion filter.

---

4. METAPROGRAMMING/MACROPROCESSING

Assembly programming is a bore, but for critical parts of programs.

You should use the appropriate tool for the right task, so don't choose assembly when it's not fit; C, OCAML, perl, Scheme, might be a better choice for most of your programming.

However, there are cases when these tools do not give a fine enough control on the machine, and assembly is useful or needed. In those case, you'll appreciate a system of macroprocessing and metaprogramming that'll allow recurring patterns to be factored each into a one indefinitely reusable definition, which allows safer programming, automatic propagation of pattern modification, etc. A ``plain'' assembler is often not enough, even when one is doing only small routines to link with C.

4.1 What's integrated into the above

Yes I know this section does not contain much useful up-to-date information. Feel free to contribute what you discover the hard way...

GCC

GCC allows (and requires) you to specify register constraints in your ``inline assembly'' code, so the optimizer always know about it; thus, inline assembly code is really made of patterns, not forcibly exact code.

Thus, you can make put your assembly into CPP macros, and inline C functions, so anyone can use it in as any C function/macro. Inline functions resemble macros very much, but are sometimes cleaner to use. Beware that in all those cases, code will be duplicated, so only local labels (of `1:` style) should be defined in that asm code. However, a macro would allow the name for a non local defined label to be passed as a parameter (or else, you should use additional meta-programming methods). Also, note that propagating inline asm code will spread potential bugs in them; so watch out doubly for register constraints in such inline asm code.

Lastly, the C language itself may be considered as a good abstraction to assembly programming, which relieves you from most of the trouble of assembling.

GAS

GAS has some macro capability included, as detailed in the texinfo docs. Moreover, while GCC recognizes .s files as raw assembly to send to GAS, it also recognizes .S files as files to pipe through CPP before to feed them to GAS. Again and again, see Linux sources for examples.

TOC

GASP

It adds all the usual macroassembly tricks to GAS. See its texinfo docs.

NASM

NASM has some macro support, too. See according docs. If you have some bright idea, you might wanna contact the authors, as they are actively developing it. Meanwhile, see about external filters below.

AS86

It has some simple macro support, but I couldn't find docs. Now the sources are very straightforward, so if you're interested, you should understand them easily. If you need more than the basics, you should use an external filter (see below).

OTHER ASSEMBLERS

- Win32FORTH: CODE and END-CODE are normal that do not switch from interpretation mode to compilation mode, so you have access to the full power of FORTH while assembling.
- TUNES: it doesn't work yet, but the Scheme language is a real high-level language that allows arbitrary meta-programming.

4.2 External Filters

Whatever is the macro support from your assembler, or whatever language you use (even C !), if the language is not expressive enough to you, you can have files passed through an external filter with a Makefile rule like that:

```
%.s:    %.S other_dependencies
        $(FILTER) $(FILTER_OPTIONS) < $< > $@
```

CPP

CPP is truely not very expressive, but it's enough for easy things, it's standard, and called transparently by GCC.

As an example of its limitations, you can't declare objects so that destructors are automatically called at the end of the declaring block; you don't have diversions or scoping, etc.

CPP comes with any C compiler. However, considering how mediocre it is, stay away from it if by chance you can make it without C,

M4

M4 gives you the full power of macroprocessing, with a Turing equivalent language, recursion, regular expressions, etc. You can do with it everything that CPP cannot.

See macro4th (this4th) or the Tunes 0.0.0.25 sources as examples of advanced macroprogramming using m4.

However, its disfunctional quoting and unquoting semantics force you to use explicit continuation-passing tail-recursive macro style if you want to do *advanced* macro programming (which is remindful of TeX -- BTW, has anyone tried to use TeX as a macroprocessor for anything else than typesetting ?). This is NOT worse than CPP that does not allow quoting and recursion anyway.

The right version of m4 to get is GNU m4 1.4 (or later if exists), which has the most features and the least bugs or limitations of all. m4 is designed to be slow for anything but the simplest uses, which might still be ok for most assembly programming (you're not writing million-lines assembly programs, are you?).

Macroprocessing with yer own filter

You can write your own simple macro-expansion filter with the usual tools: perl, awk, sed, etc. That's quick to do, and you control everything. But of course, any power in macroprocessing must be earned the hard way.

Metaprogramming

Instead of using an external filter that expands macros, one way to do things is to write programs that write part or all of other programs.

For instance, you could use a program outputing source code

- to generate sine/cosine/whatever lookup tables,
- to extract a source-form representation of a binary file,
- to compile your bitmaps into fast display routines,
- to extract documentation, initialization/finalization code, description tables, as well as normal code from the same source files,
- to have customized assembly code, generated from a perl/shell/scheme script that does arbitrary processing,
- to propagate data defined at one point only into several cross-referencing tables and code chunks.
- etc.

Think about it!

Backends from compilers

Compilers like GCC, SML/NJ, Objective CAML, MIT-Scheme, CMUCL, etc, do have their own generic assembler backend, which you might choose to use, if you intend to generate code semi-automatically from the according languages, or from a language you hack: rather than write great assembly code, you may instead modify a compiler so that it dumps great assembly code!

The New-Jersey Machine-Code Toolkit

There is a project, using the programming language Icon (with an experimental ML version), to build a basis for producing assembly-manipulating code. See around http://www.cs.virginia.edu/~nr/toolkit/

TUNES

The TUNES Project for a Free Reflective Computing System is developping its own assembler as an extension to the Scheme language, as part of its development process. It doesn't run at all yet, though help is welcome.

The assembler manipulates abstract syntax trees, so it could equally serve as the basis for a assembly syntax translator, a disassembler, a common assembler/compiler back-end, etc. Also, the full power of a real language, Scheme, make it unchallenged as for macroprocessing/metaprograming.

---

5. CALLING CONVENTIONS

5.1 Linux

Linking to GCC

That's the preferred way. Check GCC docs and examples from Linux kernel `.S` files that go through gas (not those that go through as86).

32-bit arguments are pushed down stack in reverse syntactic order (hence accessed/popped in the right order), above the 32-bit near return address. `%ebp`, `%esi`, `%edi`, `%ebx` are callee-saved, other registers are caller-saved; `%eax` is to hold the result, or `%edx:%eax` for 64-bit results.

FP stack: I'm not sure, but I think it's result in `st(0)`, whole stack caller-saved.

Note that GCC has options to modify the calling conventions by reserving registers, having arguments in registers, not assuming the FPU, etc. Check the i386 .info pages.

Beware that you must then declare the `cdecl` or `regparm(0)` attribute for a function that will follow standard GCC calling conventions. See in the GCC info pages the section: `C Extensions::Extended Asm::`. See also how Linux defines its asmlinkage macro...

ELF vs a.out problems

Some C compilers prepend an underscore before every symbol, while others do not.

Particularly, Linux a.out GCC does such prepending, while Linux ELF GCC does not.

If you need cope with both behaviors at once, see how existing packages do. For instance, get an old Linux source tree, the Elk, qthreads, or OCAML...

You can also override the implicit C->asm renaming by inserting statements like

```
void foo asm("bar") (void);
```

to be sure that the C function foo will be called really bar in assembly.

Note that the utility `objcopy`, from the `binutils` package, should allow you to transform your a.out objects into ELF objects, and perhaps the contrary too, in some cases. More generally, it will do lots of file format conversions.

TOC

Direct Linux syscalls

This is specifically *NOT* recommended, because the conventions change from time to time or from kernel flavor to kernel flavor (cf L4Linux), plus it's not portable, it's a burden to write, it's redundant with the libc effort, AND it precludes fixes and extensions that are made to the libc, like, for instance the `zlibc` package, that does on-the-fly transparent decompression of gzip-compressed files. The standard, recommended way to call Linux system services is, and will stay, to go through the libc.

Shared objects should keep your stuff small. And if you really want smaller binaries, do use `#!` stuff, with the interpreter having all the overhead you want to keep out of your binaries.

Now, if for some reason, you don't want to link to the libc, go get the libc and understand how it works! After all, you're pretending to replace it, ain't you? You might also take a look at how my eforth 1.0e does it.

The sources for Linux come in handy, too, particularly the asm/unistd.h header file, that describes how to do system calls...

Basically, you issue an `int $0x80`, with the `__NR_`syscallname number (from `asm/unistd.h`) in `%eax`, and parameters (up to five) in `%ebx`, `%ecx`, `%edx`, `%esi`, `%edi` respectively. Result is returned in `%eax`, with a negative result being an error whose opposite is what libc would put in errno. The user-stack is not touched, so you needn't have a valid one when doing a syscall.

As for the invocation arguments passed to a process upon startup, the general principle is that the stack originally contains the number of arguments argc, then the list of pointers that constitute *argv, then a null-terminated sequence of null-terminated variable=value strings. For more details, read the sources of C startup code from your libc (crt0.S or crt1.S), the sources of eforth 1.0e, or those of the linux kernel (exec.c et binfmt_*.c in linux/fs/).

Hardware I/O under Linux

If you want to do direct I/O under Linux, either it's something very simple that needn't OS arbitration, and you should see the `IO-Port-Programming` mini-HOWTO; or it needs a kernel device driver, and you should try to learn more about kernel hacking, device driver development, kernel modules, etc, for which there are other excellent HOWTOs and documents from the LDP.

Particularly, if what you want is Graphics programming, then do join one of the GGI or XFree86 projects.

Some people have even done better, writing small and robust XFree86 drivers in an interpreted domain-specific language, GAL, and achieving the efficiency of hand C-written drivers through partial evaluation (drivers not only not in asm, but not even in C!). The problem is that the partial evaluator they used to achieve efficiency is not itself free software. Any taker for a replacement?

Anyway, in all these cases, you'll be better off using GCC inline assembly with the macros from linux/asm/*.h than writing full assembly source files.

Accessing 16-bit drivers from Linux/i386

Such thing is theoretically possible (proof: see how DOSEMU can selectively grant hardware port access to programs), and I've heard rumors that someone somewhere did actually do it (in the PCI driver? Some VESA access stuff? ISA PnP? dunno). If you have some more precise information on that, you'll be most welcome. Anyway, good places to look for more information are the Linux kernel sources, DOSEMU sources (and other programs in the DOSEMU repository), and sources for various low-level programs under Linux... (perhaps GGI if it supports VESA).

Basically, you must either use 16-bit protected mode or vm86 mode.

The first is simpler to setup, but only works with well-behaved code that won't do any kind of segment arithmetics or absolute segment addressing (particularly addressing segment 0), unless by chance it happens that all segments used can be setup in advance in the LDT.

The later allows for more "compatibility" with vanilla 16-bit environments, but requires more complicated handling.

In both cases, before you can jump to 16-bit code, you must

- mmap any absolute address used in the 16-bit code (such as ROM, video buffers, DMA targets, and memory-mapped I/O) from /dev/mem to your process' address space,
- setup the LDT and/or vm86 mode monitor.
- grab proper I/O permissions from the kernel (see the above section)

Again, carefully read the source for the stuff contributed to the DOSEMU project, particularly these mini-emulators for running ELKS and/or simple .COM programs under Linux/i386.

5.2 DOS

Most DOS extenders come with some interface to DOS services. Read their docs about that, but often, they just simulate `int $0x21` and such, so you do ``as if'' you were in real mode (I doubt they have more than stubs and extend things to work with 32-bit operands; they most likely will just reflect the interrupt into the real-mode or vm86 handler).

Docs about DPMI and such (and much more) can be found on ftp://x2ftp.oulu.fi/pub/msdos/programming/ (again, the original x2ftp site is closing, so use a mirror site).

DJGPP comes with its own (limited) glibc derivative/subset/replacement, too.

It is possible to cross-compile from Linux to DOS, see the devel/msdos/ directory of your local FTP mirror for metalab.unc.edu Also see the MOSS dos-extender from the Flux project from university of Utah.

Other documents and FAQs are more DOS-centered. We do not recommend DOS development.

5.3 Winblows and suches

Hey, this document covers only free software. Ring me when Winblows becomes free, or when there are free dev tools for it!

Well, after all there are: Cygnus Solutions has developped the cygwin32.dll library, for GNU programs to run on MacroShit platforms. Thus, you can use GCC, GAS, all the GNU tools, and many other Unix applications. Have a look around their homepage. I (Faré) don't intend to expand on Losedoze programming, but I'm sure you can find lots of documents about it everywhere...

5.4 Yer very own OS

Control being what attract many programmers to assembly, want of OS development is often what leads to or stems from assembly hacking. Note that any system that allows self-development could be qualified an "OS" even though it might run "on top" of an underlying system that multitasking or I/O (much like Linux over Mach or OpenGenera over Unix), etc. Hence, for easier debugging purpose, you might like to develop your ``OS" first as a process running on top of Linux (despite the slowness), then use the Flux OS kit (which grants use of Linux and BSD drivers in yer own OS) to make it standalone. When your OS is stable, it's still time to write your own hardware drivers if you really love that.

This HOWTO will not itself cover topics such as Boot loader code & getting into 32-bit mode, Handling Interrupts, The basics about intel ``protected mode" or ``V86/R86" braindeadness, defining your object format and calling conventions. The main place where to find reliable information about that all is source code of existing OSes and bootloaders. Lots of pointers lie in the following WWW page: http://www.tunes.org/Review/OSes.html

---

6. TODO & POINTERS

- find someone who has got some time to takeover the maintenance
- fill incomplete sections
- add more pointers to software and docs
- add simple examples from real life to illustrate the syntax, power, and limitations of each proposed solution.
- ask people to help with this HOWTO
- perhaps give a few words for assembly on other architectures than i386?
- A few pointers (in addition to those already in the rest of the HOWTO)
  - 80x86 CPU family references: intel manuals; bugs.
  - ftp.luth.se mirrors the hornet and x2ftp former archives of msdos assembly coding stuff.
  - A few starting points on the web about assembly programming: Jannes Faber's; QZX's; JanW's; this one (?)
  - Fun stuff: CoreWars, a fun way to learn assembly in general.
  - USENET: comp.lang.asm.x86; alt.os.assembly.
- And of course, do use your usual Internet Search Tools to look for more information, and tell me anything interesting you find!

Author's .sig:

```
## Faré | VN: Đ£ng-Vû Bân   | Join the TUNES project!  http://www.tunes.org/ ##
## FR: François-René Rideau |    TUNES is a Useful, Not Expedient System     ##
## Reflection&Cybernethics  | Project for a Free Reflective Computing System ##
```

Bash Prompt HOWTO

Giles Orr, giles@interlog.com

v0.60, 07 January 1999

Creating and controlling terminal and xterm prompts is discussed, including incorporating standard escape sequences to give username, current working directory, time, etc. Further suggestions are made on how to modify xterm title bars, use external functions to provide prompt information, and how to use ANSI colours.

1. Introduction and Administrivia

- 1.1 Requirements
- 1.2 How To Use This Document
- 1.3 Translations
- 1.4 Problems
- 1.5 Send Me Comments and Suggestions
- 1.6 Credits
- 1.7 Copyright and Disclaimer

2. Bash and Bash Prompts

- 2.1 What is Bash?
- 2.2 What can tweaking your Bash Prompt do for you?
- 2.3 Why bother?
- 2.4 The First Step
- 2.5 Bash Prompt Escape Sequences
- 2.6 Setting the PS? Strings Permanently

TOC

3. External Commands

- 3.1 PROMPT_COMMAND
- 3.2 External Commands in the Prompt
- 3.3 What to Put in Your Prompt
- 3.4 Bash Environment and Functions

4. Xterm Title Bar Manipulations

5. ANSI Escape Sequences: Colours and Cursor Movement

- 5.1 Colours
- 5.2 Cursor Movement
- 5.3 Moving the Cursor With tput

6. Special Characters: Octal Escape Sequences

7. The Bash Prompt Package

- 7.1 Availability
- 7.2 Changing the Xterm Font

8. Loading a Different Prompt

- 8.1 Loading a Different Prompt, Later
- 8.2 Loading a Different Prompt, Immediately

9. Loading Prompt Colours Dynamically

- 9.1 A "Proof of Concept" Example

10. Example Prompts

- 10.1 A "Lightweight" Prompt
- 10.2 Elite from Bashprompt Themes
- 10.3 A "Power User" Prompt
- 10.4 A Prompt the Width of Your Term
- 10.5 The Elegant Useless Clock Prompt

1. Introduction and Administrivia

1.1 Requirements

You will need Bash. The default version on almost all Linux distributions is 1.14.7 (as of this writing, November 98), which is a well known and reliable shell. Bash is now available in version 2.0+: I've been using Bash 2.0 for a while now, but almost all code presented here should work under 1.14.7. If I'm aware of a problem, I'll mention it. You can check your Bash version by typing `echo $BASH_VERSION` at the prompt. On my machine, it responds with `2.02.1(1)-release`.

Shell programming experience would be good, but isn't essential: the more you know, the more complex prompts you'll be able to create. I assume a basic knowledge of shell programming and Unix utilities as I go through this tutorial. However, my own shell programming skills are limited, so I give a lot of examples and explanation that may appear unnecessary to an experienced shell programmer.

1.2 How To Use This Document

I include a lot of examples and explanatory text. Different parts will be of varying usefulness to different people. This has grown long enough that reading it straight through would be difficult - just read the sections you need, backtrack as necessary.

1.3 Translations

As I write (6 January 99), translations are under weigh to both Japanese (Akira Endo, akendo@t3.rim.or.jp) and German (Thomas Keil, thomas@h-preissler.de). Many thanks to both of them! URLs will be included when they become available.

1.4 Problems

This is a list of problems I've noticed while programming prompts. Don't start reading here, and don't let this list discourage you - these are mostly quite minor details. Just check back if you run into anything odd.

- Many Bash features (such as math within $(()) among others) are compile time options. If you're using a binary distribution such as comes with an standard Linux distribution, all such features should be compiled in. But if you're working on someone else's system, this is worth keeping in mind if something you expected to work doesn't. Some notes about this in *Learning the Bash Shell*, p.260-262.
- The terminal screen manager "screen" doesn't always get along with ANSI colours. I'm not a screen expert, unfortunately. My current version of screen (a very recent one) seems to work well

in all cases, but I've seen occasions where screen reduced all prompt colours to the standard foreground colour in X terminals. This doesn't appear to be a problem on the console.

- Xdefaults files can override colours. Look in /.Xdefaults for lines referring to XTerm*background and XTerm*foreground (or possibly XTerm*Background and XTerm*Foreground).
- One of the prompts mentioned in this document uses the output of "jobs" - as discussed at that time, "jobs" output to a pipe is broken in Bash 2.02.
- ANSI cursor movement escape sequences aren't all implemented in all X terminals. That's discussed in its own section.
- Some nice looking pseudo-graphics can be created by using a VGA font rather than standard Linux fonts. Unfortunately, these effects look awful if you don't use a VGA font, and there's no way to detect within a term what kind of font it's using.
- Bash 2.0+ is out, and it incorporates some new features, and changes some behaviour. Things that work under 1.14.7 don't necessarily work the same under 2.0+, or vice versa.

1.5 Send Me Comments and Suggestions

This is a learning experience for me. I've come to know a fair bit about what can be done to create interesting and useful Bash Prompts, but I need your input to correct and improve this document. I've tried to check suggestions I make against different versions of Bash (mostly 2.02, which I use, and 1.14.7, which is in wide use), but let me know of any incompatibilities you find.

The latest version of this document should always be available at http://www.interlog.com/~giles/bashprompt.html. Please check this out, and feel free to e-mail me at giles@interlog.com with suggestions.

I use the Linux Documentation Project HOWTOs almost exclusively in the HTML format, so when I convert this from SGML, HTML is the only format I check thoroughly. If there are problems with other formats, I may not know about them, and I'd appreciate a note about them.

1.6 Credits

In producing this document, I have borrowed heavily from the work of the Bashprompt project at http://bash.current.nu/. Other sources used include the *xterm Title mini-HOWTO* by Ric Lister, available at http://sunsite.unc.edu/LDP/HOWTO/mini/Xterm-Title.html, *Ansi Prompts* by Keebler, available at http://www.ncal.verio.com/~keebler/ansi.html, *How to make a Bash Prompt Theme* by Stephen Webb, available at http://bash.current.nu/bash/HOWTO.html, and *X ANSI Fonts* by Stumpy, available at http://home.earthlink.net/~us5zahns/enl/ansifont.html.

Also of immense help were several conversations and e-mails from Dan, a co-worker at Georgia College & State University, whose knowledge of Unix far exceeds mine. He's given me several excellent suggestions, and ideas of his have led to some interesting prompts.

Three books that have been very useful while programming prompts are *Linux in a Nutshell* by Jessica Heckman Perry (O'Reilly, 1997), *Learning the Bash Shell* by Cameron Newham and Bill Rosenblatt

(O'Reilly, 2nd. ed., 1998) and *Unix Shell Programming* by Lowell Jay Arthur (Wiley, 1986. This is t first edition, the fourth came out in 1997).

1.7 Copyright and Disclaimer

This document is copyright 1998-1999 by Giles Orr. You are encouraged to redistribute it. You may not modify this document (see the section on contacting me: I have so far been incorporating all changes recommended by readers). Please contact me if you're interested in doing a translation: that's one modification I can live with.

This document is available for free, and, while I have done the best I can to make it accurate and up to date, I take no responsibility for any problems you may encounter resulting from the use of this document.

---

2. Bash and Bash Prompts

2.1 What is Bash?

Descended from the Bourne Shell, Bash is a GNU product, the "**B**ourne **A**gain **SH**ell." It's the standard command line interface on most Linux machines. It excels at interactivity, supporting command line editing, completion, and recall. It also supports configurable prompts - most people realize this, but don't know how much can be done.

2.2 What can tweaking your Bash Prompt do for you?

Most Linux systems have a default prompt in one colour (usually gray) that tells you your user name, the name of the machine you're working on, and some indication of your current working directory. This is all useful information, but you can do much more with the prompt: all sorts of information can be displayed (tty number, time, date, load, number of users, uptime ...) and the prompt can use ANSI colours, either to make it look interesting, or to make certain information stand out. You can also manipulate the title bar of an Xterm to reflect some of this information.

2.3 Why bother?

Beyond looking cool, it's often useful to keep track of system information. One idea that I know appeals to some people is that it makes it possible to put prompts on different machines in different colours. If you have several Xterms open on several different machines, or if you tend to forget what machine you're working on and delete the wrong files, you'll find this a great way to remember what machine you're on.

2.4 The First Step

The appearance of the prompt is governed by the shell variable PS1. Command continuations are indicated by the PS2 string, which can be modified in exactly the same ways discussed here - since controlling it is exactly the same, and it isn't as "interesting," I'll mostly be modifying the PS1 string. (There are also PS3 and PS4 strings. These are never seen by the average user - see the Bash man page if you're interested in their purpose.) To change the way the prompt looks, you change the PS1 variable. For experimentation purposes, you can enter the PS1 strings directly at the prompt, and see the results immediately (this only affects your current session, and the changes go away when you log out). If you want to make a change to the prompt permanent, modify your /.bashrc file, and add the new definition of PS1 there. If you have root permissions, you can look in /etc/profile and modify the "**PS1=**" line. Be aware that on some distributions (RedHat 5.1 at least) that the /etc/bashrc overrides the setting of the PS1 and PS2 strings.

Before we get started, it's important to remember that the PS1 string is stored in the environment like any other environment variable. If you modify it at the command line, your prompt will change. Before you make any changes, you can save your current prompt to another environment variable:

```
[giles@nikola giles]$ SAVE=$PS1
[giles@nikola giles]$
```

The simplest prompt would be a single character, such as:

```
[giles@nikola giles]$ PS1=$
$ls
bin   mail
$
```

This demonstrates the best way to experiment with basic prompts, entering them at the command line. Notice that the text entered by the user appears immediately after the prompt: I prefer to use

```
$PS1="$ "
$ ls
bin   mail
$
```

which forces a space after the prompt, making it more readable. To restore your original prompt, just call up the variable you stored:

```
$ PS1=$SAVE
[giles@nikola giles]$
```

2.5 Bash Prompt Escape Sequences

There are a lot of escape sequences offered by the Bash shell for insertion in the prompt. From the Bash 2.02 man page:

```
When executing interactively, bash  displays  the  primary
prompt  PS1  when  it  is ready to read a command, and the
secondary prompt PS2 when it needs more input to  complete
a  command.   Bash  allows these prompt strings to be cus-
tomized by inserting a number of backslash-escaped special
characters that are decoded as follows:
       \a     an ASCII bell character (07)
       \d     the  date  in  "Weekday  Month  Date" format
              (e.g., "Tue May 26")
       \e     an ASCII escape character (033)
       \h     the hostname up to the first `.'
       \H     the hostname
       \n     newline
       \r     carriage return
       \s     the name of the shell, the  basename  of  $0
              (the portion following the final slash)
       \t     the current time in 24-hour HH:MM:SS format
       \T     the current time in 12-hour HH:MM:SS format
```

```
\@      the current time in 12-hour am/pm format
\u      the username of the current user
\v      the version of bash (e.g., 2.00)
\V      the  release  of  bash, version + patchlevel
        (e.g., 2.00.0)
\w      the current working directory
\W      the basename of the current  working  direc-
        tory
\!      the history number of this command
\#      the command number of this command
\$      if  the effective UID is 0, a #, otherwise a
        $
\nnn    the character  corresponding  to  the  octal
        number nnn
\\      a backslash
\[      begin a sequence of non-printing characters,
        which could be used to embed a terminal con-
        trol sequence into the prompt
\]      end a sequence of non-printing characters
```

Continuing where we left off:

```
[giles@nikola giles]$ PS1="\u@\h \W> "
giles@nikola giles> ls
bin   mail
giles@nikola giles>
```

This is similar to the default on most Linux distributions. I wanted a slightly different appearance, so I changed this to:

```
giles@nikola giles> PS1="[\t][\u@\h:\w]\$ "
[21:52:01][giles@nikola:~]$ ls
bin   mail
[21:52:15][giles@nikola:~]$
```

2.6 Setting the PS? Strings Permanently

Various people and distributions set their PS? strings in different places. The most common places are /etc/profile, /etc/bashrc, /.bash_profile, and /.bashrc . Johan Kullstam (johan19@idt.net) writes:

```
the PS1 string should be set in .bashrc.  this is because
non-interactive bashes go out of their way to unset PS1.  the bash man
page tells how the presence or absence of PS1 is a good way of knowing
whether one is in an interactive vs non-interactive (ie script) bash
session.

the way i realized this is that startx is a bash script.  what this
```

TOC

```
means is, startx will wipe out your prompt.  when you set PS1 in
.profile (or .bash_profile), login at console, fire up X via startx
your PS1 gets nuked in the process leaving you with the default
prompt.

one workaround is to launch xterms and rxvts with the -ls option to
force them to read .profile.  but any time a shell is called via a
non-interactive shell-script middleman PS1 is lost.  system(3) uses sh
-c which if sh is bash will kill PS1.  a better way is to place the
PS1 definition in .bashrc.  this is read every time bash starts and is
where interactive things - eg PS1 should go.

therefore it should be stressed that PS1=..blah.. should be in .bashrc
and not .profile.
```

I tried to duplicate the problem he explains, and encountered a different one: my PROMPT_COMMAND variable (which will be introduced later) was blown away. My knowledge in this area is somewhat shaky, so I'm going to go with what Johan says.

---

3. External Commands

3.1 PROMPT_COMMAND

Bash provides another environment variable called **PROMPT_COMMAND**. The contents of this variable are executed as a regular Bash command just before Bash displays a prompt.

```
[21:55:01][giles@nikola:~] PS1="[\u@\h:\w]\$ "
[giles@nikola:~] PROMPT_COMMAND="date +%H%M"
2155
[giles@nikola:~] d
bin   mail
2156
[giles@nikola:~]
```

What happened above was that I changed PS1 to no longer include the `\t` escape sequence, so the time was no longer a part of the prompt. Then I used `date +%H%M` to display the time in a format I like better. But it appears on a different line than the prompt. Tidying this up using `echo -n ...` as shown below works with Bash 2.0+, but appears not to work with Bash 1.14.7: apparently the prompt is drawn in a different way, and the following method results in overlapping text.

```
2156
[giles@nikola:~] PROMPT_COMMAND="echo -n [$(date +%H%M)]"
[2156][giles@nikola:~]$
[2156][giles@nikola:~]$ d
bin   mail
[2157][giles@nikola:~]$ unset PROMPT_COMMAND
[giles@nikola:~]
```

`echo -n ...` controls the output of the date command and supress the trailing newline, allowing the prompt to appear all on one line. At the end, I used the `unset` command to remove the PROMPT_COMMAND environment variable.

Note that I use the $(<command>) convention for command substitution: that is,

```
$(date +%H%M)
```

means "substitute the output from the **date +%H%M** command here." This works in Bash 2.0+. In some older versions of Bash, prior to 1.14.7, you may need to use backquotes (`` `date +%H%M` ``). Backquotes can be used in Bash 2.0+, but are being phased out in favor of $(), which nests better. I will continue to use this convention throughout this document. If you're using an earlier version of Bash, you can usually substitute backquotes where you see $(). If the command substitution is escaped (ie. \$(command)), then

use backslashes to escape BOTH your backquotes (ie. \'command\').

TOC

3.2 External Commands in the Prompt

You can use the output of regular Linux commands directly in the prompt as well. Obviously, you don't want to insert a lot of material, or it will create a large prompt. You also want to use a **fast** command, because it's going to be executed every time your prompt appears on the screen, and delays in the appearance of your prompt while you're working can be very annoying. (Unlike the previous example that this closely resembles, this does work with Bash 1.14.7.)

```
[21:58:33][giles@nikola:~]$ PS1="[\$(date +%H%M)][\u@\h:\w]\$ "
[2159][giles@nikola:~]$ ls
bin   mail
[2200][giles@nikola:~]$
```

It's important to notice the backslash before the dollar sign of the command substitution. Without it, the external command is executed exactly once: when the PS1 string is read into the environment. For this prompt, that would mean that it would display the same time no matter how long the prompt was used. The backslash protects the contents of $() from immediate shell interpretation, so "date" is called every time a prompt is generated.

Linux comes with a lot of small utility programs like **date**, **grep**, or **wc** that allow you to manipulate data. If you find yourself trying to create complex combinations of these programs within a prompt, it may be easier to make a shell script of your own, and call it from the prompt. Escape sequences are often required in bash shell scripts to ensure that shell variables are expanded at the correct time (as seen above with the date command): this is raised to another level within the prompt PS1 line, and avoiding it by creating shell scripts is a good idea.

An example of a small shell script used within a prompt is given below:

```
#!/bin/bash
#     lsbytesum - sum the number of bytes in a directory listing
TotalBytes=0
for Bytes in $(ls -l | grep "^-" | cut -c30-41)
do
    let TotalBytes=$TotalBytes+$Bytes
done
TotalMeg=$(echo -e "scale=3 \n$TotalBytes/1048576 \nquit" | bc)
echo -n "$TotalMeg"
```

I have at times kept this both as a function (much more efficient - unfortunately, explaining functions in detail is beyond the scope of this document), or as a shell script in my /bin directory, which is on my path. Used in a prompt:

```
[2158][giles@nikola:~]$ PS1="[\u@\h:\w (\$(lsbytesum) Mb)]\$ "
[giles@nikola:~ (0 Mb)]$ cd /bin
[giles@nikola:/bin (4.498 Mb)]$
```

TOC

3.3 What to Put in Your Prompt

You'll find I put username, machine name, time, and current directory name in most of my prompts. With the exception of the time, these are very standard items to find in a prompt, and time is probably the next most common addition. But what you include is entirely a matter of personal taste. Here are examples from people I know to help give you ideas.

Dan's prompt is minimal but very effective, particularly for the way he works.

```
[giles@nikola:~]$ cur_tty=$(tty | sed -e "s/.*tty\(.*\)/\1/")
[giles@nikola:~]$ echo $cur_tty
p4
[giles@nikola:~]$ PS1="\!,$cur_tty,\$?\$ "
1095,p4,0$
```

Dan doesn't like that having the current working directory can resize the prompt drastically as you move through the directory tree, so he keeps track of that in his head (or types "pwd"). He learned Unix with csh and tcsh, so he uses his command history extensively (something many of us weened on Bash do not do), so the first item in the prompt is the history number. The second item is the significant characters of the tty (the output of "tty" is cropped with sed), an item that can be useful to "screen" users. The third item is the exit value of the last command/pipeline (note that this is rendered useless by any command executed within the prompt - you could work around that by capturing it to a variable and playing it back, though). Finally, the "\$" is a dollar sign for a regular user, and switches to a hash mark ("#") if the user is root.

Torben Fjerdingstad wrote to tell me that he often suspends jobs, and then forgets about them, so he uses his prompt to remind him of suspended jobs:

```
[giles@nikola:~]$ function jobcount {
> jobs|wc -l| awk '{print $1}'
> }
[giles@nikola:~]$ export PS1='\W[`jobcount`]# '
giles[0]# man ls &
[1] 4150

[1]+  Stopped (tty output)    man ls
giles[1]#
```

Torben uses **awk** to trim the whitespace from the output of wc, while I would have used sed or tr - not because they're better, but because I'm more familiar with them. There are probably other ways as well. Torben also surrounds his PS1 string in single quotes, which prevent Bash from immediately interpreting the backquotes, so he doesn't have to escape them as I have mentioned.

NOTE: There is a known bug in Bash 2.02 that causes the **jobs** command (a shell builtin) to return nothing to a pipe. If you try the above under Bash 2.02, you will always get a "0" back regardless of how many jobs you have suspended. Chet Ramey, one of the maintainers of Bash, tells me that this will be fixed for v2.03.

3.4 Bash Environment and Functions

As mentioned earlier, PS1, PS2, PS3, PS4, and PROMPT_COMMAND are all stored in the Bash environment. For those of us coming from a DOS background, the idea of tossing big hunks of code into the environment is horrifying, because that DOS environment was small, and didn't exactly grow well. There are probably practical limits on what you can and should put in the environment, but I don't know what they are, and we're probably talking a couple of orders of magnitude larger than what DOS users are used to. As Dan put it:

"In my interactive shell I have 62 aliases and 25 functions. My rule of thumb is that if I need something solely for interactive use and can handily write it in bash I make it a shell function (assuming it can't be easily expressed as an alias). If these people are worried about memory they don't need to be using bash. Bash is one of the largest programs I run on my linux box (outside of Oracle). Run top sometime and press 'M' to sort by memory - see how close bash is to the top of the list. Heck, it's bigger than sendmail! Tell 'em to go get ash or something."

I guess he was using console only the day he tried that: running X and X apps, I have a lot of stuff larger than Bash. But the idea is the same: the environment is something to be used, and don't worry about overfilling it.

I risk censure by Unix gurus when I say this (for the crime of over-simplification), but functions are basically small shell scripts that are loaded into the environment for the purpose of efficiency. Quoting Dan again: "Shell functions are about as efficient as they can be. It is the approximate equivalent of sourcing a bash/bourne shell script save that no file I/O need be done as the function is already in memory. The shell functions are typically loaded from [.bashrc or .bash_profile] depending on whether you want them only in the initial shell or in subshells as well. Contrast this with running a shell script: Your shell forks, the child does an exec, potentially the path is searched, the kernel opens the file and examines enough bytes to determine how to run the file, in the case of a shell script a shell must be started with the name of the script as its argument, the shell then opens the file, reads it and executes the statements. Compared to a shell function, everything other than executing the statments can be considered unnecessary overhead."

---

4. Xterm Title Bar Manipulations

Non-printing escape sequences can be used to produce interesting effects in prompts. To use these escape sequences, you need to enclose them in **\[** and **\]**, telling Bash to ignore this material while calculating the size of the prompt. Failing to include these delimiters results in line editing code placing the cursor in the wrong place because it doesn't know the actual size of the prompt. Escape sequences must also be preceded by **\033[** in Bash prior to version 2, or by either **\033[** or **\e[** in later versions.

If you try to change the title bar of your Xterm with your prompt when you're at the console, you'll produce garbage in your prompt. To avoid this, test the TERM environment variable to tell if your prompt is going to be in an Xterm.

```
function proml
{
case $TERM in
    xterm*)
        local TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        local TITLEBAR=''
        ;;
esac

PS1="${TITLEBAR}\
[\$(date +%H%M)]\
[\u@\h:\w]\
\$ "
PS2='> '
PS4='+ '
}
```

This is a function that can be incorporated into /.bashrc. The function name could then be called to execute the function. The function, like the PS1 string, is stored in the environment. Once the PS1 string is set by the function, you can remove the function from the environment with `unset proml`. Since the prompt can't change from being in an Xterm to being at the console, the TERM variable isn't tested every time the prompt is generated. I used continuation markers (backslashes) in the definition of the prompt, to allow it to be continued on multiple lines. This improves readability, making it easier to modify and debug.

I define this as a function because this is how the Bashprompt package (discussed later in this document) deals with prompts: it's not the only way to do it, but it works well. As the prompts you use become more complex, it becomes more and more cumbersome to type them in at the prompt, and more practical to make them into some sort of text file. In this case, to test this at the prompt, save the above as a text file called "proml". You can work with it as follows:

```
[giles@nikola:/bin (4.498 Mb)]$ cd          -> Go where you want to save the prompt
[giles@nikola:~ (0 Mb)]$ vi proml           -> Edit the prompt file
...                                         -> Enter the text given above
[giles@nikola:~ (0 Mb)]$ source proml       -> Read the prompt function
[giles@nikola:~ (0 Mb)]$ proml              -> Execute the prompt function
```

The first step in creating this prompt is to test if the shell we're starting is an xterm or not: if it is, the shell variable (${TITLEBAR}) is defined. It consists of the appropriate escape sequences, and **\u@\h:\w**, which puts <user>@<machine>:<working directory> in the Xterm title bar. This is particularily useful with minimized Xterms, making them more rapidly identifiable. The other material in this prompt should be familiar from previous prompts we've created.

The only drawback to manipulating the Xterm title bar like this occurs when you log into a system on which you haven't set up the

title bar hack: the Xterm will continue to show the information from the previous system that had the title bar hack in place.

5. ANSI Escape Sequences: Colours and Cursor Movement

5.1 Colours

As mentioned before, non-printing escape sequences have to be enclosed in **\[\033[** and **\]**. For colour escape sequences, they should also be followed by a lowercase **m**.

If you try out the following prompts in an xterm and find that you aren't seeing the colours named, check out your /.Xdefaults file (and possibly its bretheren) for lines like "XTerm*Foreground: BlanchedAlmond". This can be commented out by placing an exclamation mark ("!") in front of it. Of course, this will also be dependent on what terminal emulator you're using. This is the likeliest place that your term foreground colours would be overridden.

To include blue text in the prompt:

```
PS1="\[\033[34m\][\$(date +%H%M)][\u@\h:\w]$ "
```

The problem with this prompt is that the blue colour that starts with the 34 colour code is never switched back to the regular colour, so any text you type after the prompt is still in the colour of the prompt. This is also a dark shade of blue, so combining it with the **bold** code might help:

```
PS1="\[\033[1;34m\][\$(date +%H%M)][\u@\h:\w]$\[\033[0m\] "
```

The prompt is now in light blue, and it ends by switching the colour back to nothing (whatever foreground colour you had previously).

Here are the rest of the colour equivalences:

```
Black       0;30     Dark Gray     1;30
Blue        0;34     Light Blue    1;34
Green       0;32     Light Green   1;32
Cyan        0;36     Light Cyan    1;36
Red         0;31     Light Red     1;31
Purple      0;35     Light Purple  1;35
Brown       0;33     Yellow        1;33
Light Gray  0;37     White         1;37
```

You can also set background colours by using 44 for Blue background, 41 for a Red background, etc. There are no bold background colours. Combinations can be used, like Light Red text on a Blue background: **\[\033[44;1;31m\]**, although setting the colours separately seems to work better (ie. **\[\033[44m\]\[\033[1;31m\]**). Other codes available include 4: Underscore, 5: Blink, 7: Inverse, and 8: Concealed.

Aside: Many people (myself included) object strongly to the "blink" attribute. Fortunately, it doesn't work in any terminal emulators that I'm aware of - but it will still work on the console. And, if you were wondering (as I did) "What use is a 'Concealed' attribute?!" - I saw it used in an example shell script (not a prompt) to allow someone to type in a password without it being echoed to the screen.

Based on a prompt called "elite2" in the Bashprompt package (which I have modified to work better on a standard console, rather than with the special xterm fonts required to view the original properly), this is a prompt I've used a lot:

```
function elite
{

local GRAY="\[\033[1;30m\]"
local LIGHT_GRAY="\[\033[0;37m\]"
local CYAN="\[\033[0;36m\]"
local LIGHT_CYAN="\[\033[1;36m\]"

case $TERM in
    xterm*)
        local TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        local TITLEBAR=""
        ;;
esac

local GRAD1=$(tty|cut -d/ -f3)
PS1="$TITLEBAR\
$GRAY-$CYAN-$LIGHT_CYAN(\
$CYAN\u$GRAY@$CYAN\h\
$LIGHT_CYAN)$CYAN-$LIGHT_CYAN(\
$CYAN\#$GRAY/$CYAN$GRAD1\
$LIGHT_CYAN)$CYAN-$LIGHT_CYAN(\
$CYAN\$(date +%H%M)$GRAY/$CYAN\$(date +%d-%b-%y)\
$LIGHT_CYAN)$CYAN-$GRAY-\
$LIGHT_GRAY\n\
$GRAY-$CYAN-$LIGHT_CYAN(\
$CYAN\$$GRAY:$CYAN\w\
$LIGHT_CYAN)$CYAN-$GRAY-$LIGHT_GRAY "
PS2="$LIGHT_CYAN-$CYAN-$GRAY-$LIGHT_GRAY "
}
```

I define the colours as temporary shell variables in the name of readability. It's easier to work with. The "GRAD1" variable is a check to determine what terminal you're on. Like the test to determine if you're working in an Xterm, it only needs to be done once. The prompt you see look like this, except in colour:

```
--(giles@nikola)-(75/ttyp7)-(1908/12-Oct-98)--
--($:~/tmp)--
```

To help myself remember what colours are available, I wrote the following script which echoes all the colours to screen:

```
#!/bin/bash
#
#   This file echoes a bunch of colour codes to the terminal to demonstrate
#   what's available.  Each line is one colour on black and gray
#   backgrounds, with the code in the middle.  Verified to work on white,
#   black, and green BGs (2 Dec 98).
#
echo "  On Light Gray:        On Black:"
echo -e "\033[47m\033[1;37m  White        \033[0m\
 1;37m \
\033[40m\033[1;37m  White        \033[0m"
echo -e "\033[47m\033[37m  Light Gray   \033[0m\
   37m \
\033[40m\033[37m  Light Gray   \033[0m"
echo -e "\033[47m\033[1;30m  Gray         \033[0m\
 1;30m \
\033[40m\033[1;30m  Gray         \033[0m"
echo -e "\033[47m\033[30m  Black        \033[0m\
   30m \
\033[40m\033[30m  Black        \033[0m"
echo -e "\033[47m\033[31m  Red          \033[0m\
   31m \
\033[40m\033[31m  Red          \033[0m"
echo -e "\033[47m\033[1;31m  Light Red    \033[0m\
 1;31m \
\033[40m\033[1;31m  Light Red    \033[0m"
echo -e "\033[47m\033[32m  Green        \033[0m\
   32m \
\033[40m\033[32m  Green        \033[0m"
echo -e "\033[47m\033[1;32m  Light Green  \033[0m\
 1;32m \
\033[40m\033[1;32m  Light Green  \033[0m"
echo -e "\033[47m\033[33m  Brown        \033[0m\
   33m \
\033[40m\033[33m  Brown        \033[0m"
echo -e "\033[47m\033[1;33m  Yellow       \033[0m\
 1;33m \
\033[40m\033[1;33m  Yellow       \033[0m"
echo -e "\033[47m\033[34m  Blue         \033[0m\
   34m \
\033[40m\033[34m  Blue         \033[0m"
echo -e "\033[47m\033[1;34m  Light Blue   \033[0m\
 1;34m \
\033[40m\033[1;34m  Light Blue   \033[0m"
echo -e "\033[47m\033[35m  Purple       \033[0m\
   35m \
\033[40m\033[35m  Purple       \033[0m"
```

TOC

```
echo -e "\033[47m\033[1;35m  Pink         \033[0m\
 1;35m \
\033[40m\033[1;35m  Pink         \033[0m"
echo -e "\033[47m\033[36m  Cyan         \033[0m\
   36m \
\033[40m\033[36m  Cyan         \033[0m"
echo -e "\033[47m\033[1;36m  Light Cyan   \033[0m\
 1;36m \
\033[40m\033[1;36m  Light Cyan   \033[0m"
```

5.2 Cursor Movement

ANSI escape sequences allow you to move the cursor around the screen at will. This is more useful for full screen user interfaces generated by shell scripts, but can also be used in prompts. The movement escape sequences are as follows:

```
    - Position the Cursor:
      \033[<L>;<C>H
      puts the cursor at line L and column C.
    - Move the cursor up N lines:
      \033[<N>A
    - Move the cursor down N lines:
      \033[<N>B
    - Move the cursor forward N columns:
      \033[<N>C
    - Move the cursor backward N columns:
      \033[<N>D

    - Save cursor position:
      \033[s
    - Restore cursor position:
      \033[u
```

The latter two codes are NOT honoured by many terminal emulators. The only ones that I'm aware of that do are xterm and nxterm - even though the majority of terminal emulators are based on xterm code. As far as I can tell, rxvt, kvt, xiterm, and Eterm do not support this. They are supported on the console.

Try putting in the following line of code at the prompt (it's a little clearer what it does if the prompt is several lines down the terminal when you put this in): `echo -en "\033[7A\033[1;35m BASH \033[7B\033[6D"` This should move the cursor seven lines up screen, print the word " BASH ", and then return to where it started to produce a normal prompt. This isn't a prompt: it's just a demonstration of moving the cursor on screen, using colour to emphasize what has been done.

Save this in a file called "clock":

```
#!/bin/bash
```

```
function prompt_command {
let prompt_x=$COLUMNS-5
}

PROMPT_COMMAND=prompt_command

function clock {
local       BLUE="\[\033[0;34m\]"
local        RED="\[\033[0;31m\]"
local  LIGHT_RED="\[\033[1;31m\]"
local      WHITE="\[\033[1;37m\]"
local  NO_COLOUR="\[\033[0m\]"
case $TERM in
    xterm*)
        TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        TITLEBAR=""
        ;;
esac

PS1="${TITLEBAR}\
\[\033[s\033[1;\$(echo -n \${prompt_x})H\]\
$BLUE[$LIGHT_RED\$(date +%H%M)$BLUE]\[\033[u\033[1A\]
$BLUE[$LIGHT_RED\u@\h:\w$BLUE]\
$WHITE\$$NO_COLOUR "
PS2='> '
PS4='+ '
}
```

This prompt is fairly plain, except that it keeps a 24 hour clock in the upper right corner of the terminal (even if the terminal is resized). This will NOT work on the terminal emulators that I mentioned that don't accept the save and restore cursor position codes. If you try to run this prompt in any of those terminal emulators, the clock will appear correctly, but the prompt will be trapped on the second line of the terminal.

See also The Elegant Useless Clock Prompt for a more extensive use of these codes.

5.3 Moving the Cursor With tput

As with so many things in Unix, there is more than one way to achieve the same ends. A utility called "tput" can also be used to move the cursor around the screen, or get back information about the status of the terminal. "tput" for cursor positioning is less flexible than ANSI escape sequences: you can only move the cursor to an absolute position, you can't move it relative to its current position. I don't use "tput," so I'm not going to explain it in detail. Type "man tput" and you'll know as much as I do.

---

6. Special Characters: Octal Escape Sequences

Outside of the characters that you can type on your keyboard, there are a lot of other characters you can print on your screen. I've created a script to allow you to check out what the font you're using has available for you. The main command you need to use to utilise these characters is "echo -e". The "-e" switch tells echo to enable interpretation of backslash-escaped characters. What you see when you look at octal 200-400 will be very different with a VGA font from what you will see with a standard Linux font. Be warned that some of these escape sequences have odd effects on your terminal, and I haven't tried to prevent them from doing whatever they do. The linedraw and block characters (which many of us became familiar with with Word Perfect) that are used heavily by the Bashprompt project are between octal 260 and 337.

```
#!/bin/bash

#   Script: escgen

function usage {
   echo -e "\033[1;34mescgen\033[0m <lower_octal_value> [<higher_octal_value>]"
   echo "   Octal escape sequence generator: print all octal escape sequences"
   echo "   between the lower value and the upper value.  If a second value"
   echo "   isn't supplied, print eight characters."
   echo "   1998 - Giles Orr, no warranty."
   exit 1
}

if [ "$#" -eq "0" ]
then
   echo -e "\033[1;31mPlease supply one or two values.\033[0m"
   usage
fi
let lower_val=${1}
if [ "$#" -eq "1" ]
then
   #   If they don't supply a closing value, give them eight characters.
   upper_val=$(echo -e "obase=8 \n ibase=8 \n $lower_val+10 \n quit" | bc)
else
   let upper_val=${2}
fi
if [ "$#" -gt "2" ]
then
   echo -e "\033[1;31mPlease supply two values.\033[0m"
   echo
   usage
fi
if [ "${lower_val}" -gt "${upper_val}" ]
then
   echo -e "\033[1;31m${lower_val} is larger than ${upper_val}."
```

```
   echo
   usage
fi
if [ "${upper_val}" -gt "777" ]
   then
   echo -e "\033[1;31mValues cannot exceed 777.\033[0m"
   echo
   usage
fi

let i=$lower_val
let line_count=1
let limit=$upper_val
while [ "$i" -lt "$limit" ]
do
   octal_escape="\\$i"
   echo -en "$i:'$octal_escape' "
   if [ "$line_count" -gt "7" ]
   then
      echo
      #   Put a hard return in.
      let line_count=0
   fi
   let i=$(echo -e "obase=8 \n ibase=8 \n $i+1 \n quit" | bc)
   let line_count=$line_count+1
done
echo
```

You can also use **xfd** to display all the characters in an X font, with the command "xfd -fn <fontname>". Clicking on any given character will give you lots of information about that character, including its octal value. The script given above will be useful on the console, and if you aren't sure of the current font name.

---

7. The Bash Prompt Package

7.1 Availability

The Bash Prompt package is available at http://bash.current.nu, and is the work of several people, co-ordinated by Rob Current (aka BadLandZ). The package is an early beta, but offers a simple way of using multiple prompts (or themes), allowing you to set prompts for login shells, and for subshells (ie. putting PS1 strings in /.bash_profile and /.bashrc). Most of the themes use the extended VGA character set, so they look bad unless they're used with VGA fonts (which aren't the default on most systems).

7.2 Changing the Xterm Font

To use some of the most attractive prompts in the Bash Prompt package, you need to get and install fonts that support the character sets expected by the prompts. These are referred to as "VGA Fonts," but I'm unclear on the distinction between them and the fonts Linux usually ships with - although clearly they support different character sets. Standard Xterm fonts support an extended alphabet, including a lot of letters with accents. In VGA fonts, this material is replaced by graphical characters - blocks, dots, lines. If anyone can explain this in more detail, e-mail me and I'll include the explanation here.

Getting and installing these fonts is a somewhat involved process. First, retrieve the font(s). Next, ensure they're .pcf or .pcf.gz files. If they're .bdf files, investigate the "bdftopcf" command (ie. read the man page). Drop the .pcf or .pcf.gz files into the /usr/X11R6/lib/X11/fonts/misc dir (this is the correct directory for RedHat 5.1 and Slackware 3.4, it may be different on other distributions). "cd" to that directory, and run the "mkfontdir" command. Then run "xset fp rehash". Sometimes it's a good idea to go into the fonts.alias file in the same directory, and create shorter alias names for the fonts.

To use the new fonts, you start your Xterm program of choice with the appropriate command to your Xterm, which can be found either in the man page or by using the "--help" parameter on the command line. Popular terms would be used as follows:

```
xterm -font <fontname>
```

OR

```
xterm -fn <fontname> -fb <fontname-bold>
Eterm -f <fontname>
rxvt -fn <fontname>
```

VGA fonts are available from *Stumpy's ANSI Fonts* page at http://home.earthlink.net/~us5zahns/enl/ansifont.html (which I have borrowed from extensively while writing this).

8. Loading a Different Prompt

8.1 Loading a Different Prompt, Later

The explanations in this HOWTO have shown how to make PS1 environment variables, or how to incorporate those PS1 and PS2 strings into functions that could be called by /.bashrc or as a theme by the bashprompt package.

Using the bashprompt package, you would type `bashprompt -i` to see a list of available themes. To set the prompt in future login shells (primarily the console, but also telnet and Xterms, depending on how your Xterms are set up), you would type `bashprompt -l themename`. bashprompt then modifies your /.bash_profile to call the requested theme when it starts. To set the prompt in future subshells (usually Xterms, rxvt, etc.), you type `bashprompt -s themename`, and bashprompt modifies your /.bashrc file to call the appropriate theme at startup.

See also Setting the PS? Strings Permanently for Johan Kullstam's note regarding the importance of putting the PS? strings in /.bashrc .

8.2 Loading a Different Prompt, Immediately

You can change the prompt in your current terminal (using the example "elite" function above) by typing "`source elite`" followed by "`elite`" (assuming that the elite function file is the working directory). This is somewhat cumbersome, and leaves you with an extra function (elite) in your environment space - if you want to clean up the environment, you would have to type "`unset elite`" as well. This would seem like an ideal candidate for a small shell script, but a script doesn't work here because the script cannot change the environment of your current shell: it can only change the environment of the subshell it runs in. As soon as the script stops, the subshell goes away, and the changes the script made to the environment are gone. What **can** change environment variables of your current shell are environment functions. The bashprompt package puts a function called "callbashprompt" into your environment, and, while they don't document it, it can be called to load any bashprompt theme on the fly. It looks in the theme directory it installed (the theme you're calling has to be there), sources the function you asked for, loads the function, and then unsets the function, thus keeping your environment uncluttered. "callbashprompt" wasn't intended to be used this way, and has no error checking, but if you keep that in mind, it works quite well.

9. Loading Prompt Colours Dynamically

9.1 A "Proof of Concept" Example

This is a "proof of concept" more than an attractive prompt: changing colours within the prompt dynamically. In this example, the colour of the host name changes depending on the load (as a warning).

```
#!/bin/bash
#   "hostloadcolour" - 17 October 98, by Giles
#
#   The idea here is to change the colour of the host name in the prompt,
#   depending on a threshold load value.

# THRESHOLD_LOAD is the value of the one minute load (multiplied
# by one hundred) at which you want
# the prompt to change from COLOUR_LOW to COLOUR_HIGH
THRESHOLD_LOAD=200
COLOUR_LOW='1;34'
          # light blue
COLOUR_HIGH='1;31'
           # light red

function prompt_command {
ONE=$(uptime | sed -e "s/.*load average: \(.*\...\), \(.*\...\), \(.*\...\)/\1/" -e
"s/ //g")
#   Apparently, "scale" in bc doesn't apply to multiplication, but does
#   apply to division.
ONEHUNDRED=$(echo -e "scale=0 \n $ONE/0.01 \nquit \n" | bc)
if [ $ONEHUNDRED -gt $THRESHOLD_LOAD ]
then
    HOST_COLOUR=$COLOUR_HIGH
        # Light Red
else
    HOST_COLOUR=$COLOUR_LOW
        # Light Blue
fi
}

function hostloadcolour {

PROMPT_COMMAND=prompt_command
PS1="[$(date +%H%M)][\u@\[\033[\$(echo -n \$HOST_COLOUR)m\]\h\[\033[0;37m\]:\w]$ "
}
```

Using your favorite editor, save this to a file named "hostloadcolour". If you have the Bashprompt package installed, this will work as a theme. If you don't, type `source hostloadcolour` and then `hostloadcolour`. Either way, "prompt_command" becomes a function in your environment. If you examine the code, you will notice that the colours

($COLOUR_HIGH and $COLOUR_LOW) are set using only a partial colour code, ie. "1;34" instead of "\[\033[1;34r which I would have preferred. I have been unable to get it to work with the complete code. Please let me know if you manage this.

TOC

10. Example Prompts

10.1 A "Lightweight" Prompt

```
function proml {
local BLUE="\[\033[0;34m\]"
local RED="\[\033[0;31m\]"
local LIGHT_RED="\[\033[1;31m\]"
local WHITE="\[\033[1;37m\]"
local NO_COLOUR="\[\033[0m\]"
case $TERM in
    xterm*)
        TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        TITLEBAR=""
        ;;
esac

PS1="${TITLEBAR}\
$BLUE[$RED\$(date +%H%M)$BLUE]\
$BLUE[$LIGHT_RED\u@\h:\w$BLUE]\
$WHITE\$$NO_COLOUR "
PS2='> '
PS4='+ '
}
```

10.2 Elite from Bashprompt Themes

Note that this requires a VGA font.

```
# Created by KrON from windowmaker on IRC
# Changed by Spidey 08/06
function elite {
PS1="\[\033[31m\]\332\304\[\033[34m\](\[\033[31m\]\u\[\033[34m\]@\[\033[31m\]\h\
\[\033[34m\])\[\033[31m\]-\[\033[34m\](\[\033[31m\]\$(date +%I:%M%P)\
\[\033[34m\]-:-\[\033[31m\]\$(date +%m)\[\033[34m\033[31m\]/\$(date +%d)\
\[\033[34m\])\[\033[31m\]\304-\[\033[34m]\\371\[\033[31m\]-\371\371\
\[\033[34m\]\372\n\[\033[31m\]\300\304\[\033[34m\](\[\033[31m\]\W\[\033[34m\])\
\[\033[31m\]\304\371\[\033[34m\]\372\[\033[00m\]"
PS2="> "
}
```

10.3 A "Power User" Prompt

TOC

I actually do use this prompt, but it results in noticeable delays in the appearance of the prompt on a single-user PII-400 so I wouldn't recommend using it on a multi-user P-100 or anything ... Look at it for ideas, rather than as a practical prompt.

```
#!/bin/bash
#-----------------------------------------------------------------------
#       POWER USER PROMPT "pprom2"
#-----------------------------------------------------------------------
#
#   Created August 98, Last Modified 9 November 98 by Giles
#
#   Problem: when load is going down, it says "1.35down-.08", get rid
#   of the negative

function prompt_command
{
#   Create TotalMeg variable: sum of visible file sizes in current directory
local TotalBytes=0
for Bytes in $(ls -l | grep "^-" | cut -c30-41)
do
    let TotalBytes=$TotalBytes+$Bytes
done
TotalMeg=$(echo -e "scale=3 \nx=$TotalBytes/1048576\n if (x<1) {print \"0\"} \n print
x \nquit" | bc)

#      This is used to calculate the differential in load values
#      provided by the "uptime" command.  "uptime" gives load
#      averages at 1, 5, and 15 minute marks.
#
local one=$(uptime | sed -e "s/.*load average: \(.*\...\), \(.*\...\),
\(.*\...\)/\1/" -e "s/ //g")
local five=$(uptime | sed -e "s/.*load average: \(.*\...\), \(.*\...\),
\(.*\...\).*/\2/" -e "s/ //g")
local diff1_5=$(echo -e "scale = scale ($one) \nx=$one - $five\n if (x>0) {print
\"up\"} else {print \"down\"}\n print x \nquit \n" | bc)
loaddiff="$(echo -n "${one}${diff1_5}")"

#   Count visible files:
let files=$(ls -l | grep "^-" | wc -l | tr -d " ")
let hiddenfiles=$(ls -l -d .* | grep "^-" | wc -l | tr -d " ")
let executables=$(ls -l | grep ^-..x | wc -l | tr -d " ")
let directories=$(ls -l | grep "^d" | wc -l | tr -d " ")
let hiddendirectories=$(ls -l -d .* | grep "^d" | wc -l | tr -d " ")-2
let linktemp=$(ls -l | grep "^l" | wc -l | tr -d " ")
if [ "$linktemp" -eq "0" ]
then
    links=""
else
    links=" ${linktemp}l"
fi
unset linktemp
```

TOC

```
let devicetemp=$(ls -l | grep "^[bc]" | wc -l | tr -d " ")
if [ "$devicetemp" -eq "0" ]
then
    devices=""
else
    devices=" ${devicetemp}bc"
fi
unset devicetemp

}

PROMPT_COMMAND=prompt_command

function pprom2 {

local        BLUE="\[\033[0;34m\]"
local  LIGHT_GRAY="\[\033[0;37m\]"
local LIGHT_GREEN="\[\033[1;32m\]"
local  LIGHT_BLUE="\[\033[1;34m\]"
local  LIGHT_CYAN="\[\033[1;36m\]"
local      YELLOW="\[\033[1;33m\]"
local       WHITE="\[\033[1;37m\]"
local         RED="\[\033[0;31m\]"
local   NO_COLOUR="\[\033[0m\]"

case $TERM in
    xterm*)
        TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        TITLEBAR=""
        ;;
esac

PS1="$TITLEBAR\
$BLUE[$RED\$(date +%H%M)$BLUE]\
$BLUE[$RED\u@\h$BLUE]\
$BLUE[\
$LIGHT_GRAY\${files}.\${hiddenfiles}-\
$LIGHT_GREEN\${executables}x \
$LIGHT_GRAY(\${TotalMeg}Mb) \
$LIGHT_BLUE\${directories}.\
\${hiddendirectories}d\
$LIGHT_CYAN\${links}\
$YELLOW\${devices}\
$BLUE]\
$BLUE[${WHITE}\${loaddiff}$BLUE]\
$BLUE[\
$WHITE\$(ps ax | wc -l | sed -e \"s: ::g\")proc\
$BLUE]\
\n\
$BLUE[$RED\$PWD$BLUE]\
$WHITE\$\
\
$NO_COLOUR "
```

```
PS2='> '
PS4='+ '
}
```


10.4 A Prompt the Width of Your Term

A friend complained that he didn't like having a prompt that kept changing size because it had $PWD in it, so I wrote this prompt that adjusts its size to exactly the width of your term, with the working directory on the top line of two.

```
#!/bin/bash

#   termwide prompt
#      by Giles - created 2 November 98
#
#   The idea here is to have the upper line of this two line prompt
#   always be the width of your term.  Do this by calculating the
#   width of the text elements, and putting in fill as appropriate
#   or right-truncating $PWD.
#

function prompt_command {

TERMWIDTH=${COLUMNS}

#   Calculate the width of the prompt:

hostnam=$(echo -n $HOSTNAME | sed -e "s/[\.].*//")
#   "whoami" and "pwd" include a trailing newline
usernam=$(whoami)
let usersize=$(echo -n $usernam | wc -c | tr -d " ")
newPWD="${PWD}"
let pwdsize=$(echo -n ${newPWD} | wc -c | tr -d " ")
#   Add all the accessories below ...
let promptsize=$(echo -n "--(${usernam}@${hostnam})---(${PWD})--" \
                 | wc -c | tr -d " ")
let fillsize=${TERMWIDTH}-${promptsize}
fill=""
while [ "$fillsize" -gt "0" ]
do
   fill="${fill}-"
   let fillsize=${fillsize}-1
done

if [ "$fillsize" -lt "0" ]
then
   let cut=3-${fillsize}
   sedvar=""
   while [ "$cut" -gt "0" ]
   do
      sedvar="${sedvar}."
      let cut=${cut}-1
```

```
   done
   newPWD="...$(echo -n $PWD | sed -e "s/\(^${sedvar}\)\(.*\)/\2/")"
fi
}

PROMPT_COMMAND=prompt_command

function termwide {

local GRAY="\[\033[1;30m\]"
local LIGHT_GRAY="\[\033[0;37m\]"
local WHITE="\[\033[1;37m\]"
local NO_COLOUR="\[\033[0m\]"

local LIGHT_BLUE="\[\033[1;34m\]"
local YELLOW="\[\033[1;33m\]"

case $TERM in
    xterm*)
        TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        TITLEBAR=""
        ;;
esac

PS1="$TITLEBAR\
$YELLOW-$LIGHT_BLUE-(\
$YELLOW\${usernam}$LIGHT_BLUE@$YELLOW\${hostnam}\
${LIGHT_BLUE})-${YELLOW}-\${fill}${LIGHT_BLUE}-(\
$YELLOW\${newPWD}\
$LIGHT_BLUE)-$YELLOW-\
\n\
$YELLOW-$LIGHT_BLUE-(\
$YELLOW\$(date +%H%M)$LIGHT_BLUE:$YELLOW\$(date \"+%a,%d %b %y\")\
$LIGHT_BLUE:$WHITE\$$LIGHT_BLUE)-\
$YELLOW-\
$NO_COLOUR "

PS2="$LIGHT_BLUE-$YELLOW-$YELLOW-$NO_COLOUR "

}
```

10.5 The Elegant Useless Clock Prompt

This is probably the single most attractive (and useless) prompt I've ever created. Because many X terminal emulators don't implement cursor position save and restore, the alternative when putting a clock in the upper right corner is to anchor the cursor at the bottom of the terminal. This builds on the idea of the "termwide" prompt above, drawing a line up the right side of the screen from the prompt to the clock. A VGA font is required.

Note: There is an odd substitution in here, that may not print properly being translated from SGML to other formats: I had to substitute the screen character for \304 - I would normally have just included the sequence "\304", but it was necessary to make this substitution in this case.

TOC

```
#!/bin/bash

#   This prompt requires a VGA font.  The prompt is anchored at the bottom
#   of the terminal, fills the width of the terminal, and draws a line up
#   the right side of the terminal to attach itself to a clock in the upper
#   right corner of the terminal.

function prompt_command {
#   Calculate the width of the prompt:
hostnam=$(echo -n $HOSTNAME | sed -e "s/[\.].*//")
#   "whoami" and "pwd" include a trailing newline
usernam=$(whoami)
newPWD="${PWD}"
#   Add all the accessories below ...
let promptsize=$(echo -n "--(${usernam}@${hostnam})---(${PWD})-----" \
                 | wc -c | tr -d " ")
#   Figure out how much to add between user@host and PWD (or how much to
#   remove from PWD)
let fillsize=${COLUMNS}-${promptsize}
fill=""
#   Make the filler if prompt isn't as wide as the terminal:
while [ "$fillsize" -gt "0" ]
do
   fill="${fill}Ä"
   # The A with the umlaut over it (it will appear as a long dash if
   # you're using a VGA font) is \304, but I cut and pasted it in
   # because Bash will only do one substitution - which in this case is
   # putting $fill in the prompt.
   let fillsize=${fillsize}-1
done
#   Right-truncate PWD if the prompt is going to be wider than the terminal:
if [ "$fillsize" -lt "0" ]
then
   let cutt=3-${fillsize}
   sedvar=""
   while [ "$cutt" -gt "0" ]
   do
      sedvar="${sedvar}."
      let cutt=${cutt}-1
   done
   newPWD="...$(echo -n $PWD | sed -e "s/\(^${sedvar}\)\(.*\)/\2/")"
fi
#
#   Create the clock and the bar that runs up the right side of the term
#
local LIGHT_BLUE="\033[1;34m"
local     YELLOW="\033[1;33m"
#   Position the cursor to print the clock:
echo -en "\033[2;$((${COLUMNS}-9))H"
echo -en "$LIGHT_BLUE($YELLOW$(date +%H%M)$LIGHT_BLUE)\304$YELLOW\304\304\277"
local i=${LINES}
echo -en "\033[2;${COLUMNS}H"
#   Print vertical dashes down the side of the terminal:
```

TOC

```
while [ $i -ge 4 ]
do
   echo -en "\033[$(($i-1));${COLUMNS}H\263"
   let i=$i-1
done

let prompt_line=${LINES}-1
#   This is needed because doing \${LINES} inside a Bash mathematical
#   expression (ie. $(())) doesn't seem to work.
}

PROMPT_COMMAND=prompt_command

function clock3 {
local LIGHT_BLUE="\[\033[1;34m\]"
local     YELLOW="\[\033[1;33m\]"
local      WHITE="\[\033[1;37m\]"
local LIGHT_GRAY="\[\033[0;37m\]"
local  NO_COLOUR="\[\033[0m\]"

case $TERM in
    xterm*)
        TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        TITLEBAR=""
        ;;
esac

PS1="$TITLEBAR\
\[\033[\${prompt_line};0H\]
$YELLOW\332$LIGHT_BLUE\304(\
$YELLOW\${usernam}$LIGHT_BLUE@$YELLOW\${hostnam}\
${LIGHT_BLUE})\304${YELLOW}\304\${fill}${LIGHT_BLUE}\304(\
$YELLOW\${newPWD}\
$LIGHT_BLUE)\304$YELLOW\304\304\304\331\
\n\
$YELLOW\300$LIGHT_BLUE\304(\
$YELLOW\$(date \"+%a,%d %b %y\")\
$LIGHT_BLUE:$WHITE\$$LIGHT_BLUE)\304\
$YELLOW\304\
$LIGHT_GRAY "

PS2="$LIGHT_BLUE\304$YELLOW\304$YELLOW\304$NO_COLOUR "

}
```

The Belgian HOWTO

by Dag Wieërs, dag@mind.be

v1.0.11, 11 September 1999

This document describes how to tweak Linux for Belgian users and lists Linux user groups, businesses and other resources in Belgium.

1. Introduction

- 1.1 Why this document?
- 1.2 New versions of this document
- 1.3 Contributions
- 1.4 Feedback

2. Configuration

- 2.1 Keyboard setup
- 2.2 Display and applications
- 2.3 Time zone
- 2.4 Locale support
- 2.5 HOWTO translations
- 2.6 Native language support and codepages

3. Setting up Internet (Belgian ISP information)

- 3.1 Dialing in with your modem (PPP)
- 3.2 Dialing in with your ISDN-modem
- 3.3 Cablemodem (DHCP)
- 3.4 ADSL

4. Belgian Linux user groups

- 4.1 ALLIN Linux User Group
- 4.2 Antwerp Linux User Group (ALUG)
- 4.3 Brussels Linux User Group (BLUG)
- 4.4 HCC Limburgse Linux Gebruikers Groep (HLLGG)
- 4.5 HCC Linux Gebruikers Groep Antwerpen
- 4.6 HCC Linux Gebruikers Groep Leuven
- 4.7 Independent Group of Unix-Alikes and Network Activists (IGUANA)
- 4.8 Infogroep (IGWE)
- 4.9 Leuvense Linux Users (L2U)
- 4.10 Limburgse Linux User Group (LILUG)
- 4.11 Linux Belgian Team
- 4.12 Linux Gebruikers Oost-Vlaanderen (LGOVL)
- 4.13 Linux User Group De Bruxelles (BxLUG)
- 4.14 Linux User Group De Namur
- 4.15 Linux User Group Mons
- 4.16 Linux User Movement Underground Mad Belgian Aliens (LUMUMBA)
- 4.17 linuxbe.org
- 4.18 Local Linux Support Group Gent (LLSGG)
- 4.19 Open Technology Assembly Linux Special Interest Group
- 4.20 Student Information Networking (SIN)
- 4.21 TINA
- 4.22 Unix Lovers Yield Student Services & Internet Support (ULYSSIS)
- 4.23 Zeus WPI - Werkgroep Informatica RUG

5. Belgian Linux businesses

- 5.1 aDOC Services
- 5.2 Better Access N.V.
- 5.3 D. Connect
- 5.4 Double Barrel Consultancy and Productions
- 5.5 grmbl productions
- 5.6 LIFE

- 5.7 Phidani Software SPRL
- 5.8 Spier BVBA
- 5.9 VirgoPlus sprl

6. Belgian Linux resources

- 6.1 Websites
- 6.2 HOWTO
- 6.3 FTP
- 6.4 Usenet
- 6.5 Mailinglists
- 6.6 IRC

---

1. Introduction

1.1 Why this document?

Since there was no information anywhere on the net for Belgian linux users, I started to collect stuff that I thought could be important. This HOWTO aims to be the definitive list of all Belgian-related Linux information. If you think not, you have no excuse to complain and certainly not to not contribute. ;)

1.2 New versions of this document

If you need to know more about the Linux Documentation Project or about Linux HOWTO's, feel free to contact the supervisor Tim Bynum linux-howto@metalab.unc.edu.

Tim Bynum will post the listing to several national and international newsgroups on a monthly basis. In addition, the Belgian HOWTO can be found on the World Wide Web at http://dag.mind.be/howto/ New versions of the Belgian HOWTO are always placed at this site first, so please be sure to check if the copy you are reading is still up to date!

1.3 Contributions

A few people mailed me their suggestions and improvements, thanks go to:

- Wim Vandeputte wvdeputte@reptile.rug.ac.be
- Pablo Saratxaga srtxg@ping.be
- Christophe Lambin clambin@skynet.be
- Geert Uytterhoeven geert@cs.kuleuven.ac.be

Many thanks go to Ivo Clarysse soggie@iguana.be who started something similar earlier (http://linbel.iguana.be/) and his project helped me a lot to get this document online.

1.4 Feedback

If some information seems to be wrong, deceptive or missing, I'd appreciate if you mailed me the improvements. Since I'm just human this document isn't bug-free, but your contribution can and will make a difference.

To add yourself to the Linux user groups or businesses, collect all necessary information and mail it to me.

2. Configuration

2.1 Keyboard setup

Linux provides 2 ways to set up your keyboard. At the console you can use `loadkeys` and under XFree86 you can use `xmodmap`.

Console

To use the keytable for a Belgian keyboard you can use

```
loadkeys be2-latin1
```

or

```
loadkeys be-latin1
```

The only difference between the two is that be2-latin1 add support for twosuperior and threesuperior (keycode 41) and grave (keycode 43).

Usually `loadkeys` is started at boottime from the scripts located somewhere in */etc/rc.d* Different distributions handle it differentely. But most distributions have a simple program (like `kbdconfig` in RedHat or `yast` in SuSE) to change the scripts.

XFree86

To set your keyboard properly under XFree86 you've got more choices. By running `Xconfigurator` (or a similar program) a config-file for XFree86 is created (with some standard content). This file is called *XF86Config* and is usually located in */etc* or */etc/X11*

Make sure (and change otherwise) that something similar to this is given under the `Keyboard Section`.

```
Section "Keyboard"
        Protocol        "Standard"
        XkbRules        "xfree86"
        XkbModel        "pc101"
        XkbLayout       "be"
EndSection
```

If you own a 'microsoft' keyboard (or the less-known penguin keyboards ;p) you can enable those extra 3 keys by changing the `"pc101"` into `"pc104"`. (I use the extra keys as meta-keys in my windowmanager so some shortcuts doesn't conflict with the internal ones of my wm).

To get the AltGr key to work under XFree86 simply add in the `Keyboard Section:`

```
RightAlt        ModeShift
```

To get some key-combinations working under X11, you might want to change in file */usr/X11R6/lib/X11/xkb/symbols/be*:

```
key <AD11> { [ dead_circumflex,       diaeresis      ],
                [     bracketleft                      ]       };
key <AC11> { [          ugrave,         percent      ],
                [      quoteright                      ]       };
key <BKSL> { [              mu,        sterling      ],
                [       quoteleft                      ]       };
key <AB10> { [           equal,            plus      ],
                [      asciitilde                      ]       };
```

by

```
key <AD11> { [ dead_circumflex,  dead_diaeresis      ],
                [     bracketleft                      ]       };
key <AC11> { [          ugrave,         percent      ],
                [      dead_acute                      ]       };
key <BKSL> { [              mu,        sterling      ],
                [      dead_grave                      ]       };
key <AB10> { [           equal,            plus      ],
                [      dead_tilde                      ]       };
```

Some people prefer to use a 'compose'-key to enter their special characters, like:

```
<compose>'e => é
<compose>/o => ø
<compose>c, => ç
<compose>co => ©
<compose>ss => ß
```

Solution:

```
xmodmap -e "keycode xx = Multi_key"
```

or

```
xmodmap -e "keysym yy = Multi_key"
```

and 'xx' is the keycode and 'yy' the kesym of the key you choose to be the 'compose'-key. (use `xev` to get these

values)

Alternatively you can add

```
keycode xx = Multi_key
```

to your */.xmodmaprc*. This way it happens automagically.

2.2 Display and applications

Some applications have to be compiled as 8-bit clean to work well with the European characterset. Others like to be told in advance.

Using the ISO-8859-1 font in console

You can load the ISO-8859-1 font by typing the following in console:

```
setfont lat1u-16.psf
mapscrn trivial
echo -ne '\033(K'
```

To be able to use ë, è, é or ç in console, you might want to add these to your */.inputrc*:

```
set meta-flag on
set convert-meta off
set output-meta on
```

(this must be done for every user, if you want this to be default for all users you can either add this file to */etc/skel/* or you could add these lines to */etc/inputrc* and add the following line to */etc/bashrc*,

```
export INPUTRC=/etc/inputrc
```

)

Note: this is only useful to programs that use readline (like bash) Also set the following environment variables:

```
LC_CTYPE=iso-8859-1
SYSFONT=lat1u-16.psf
SYSTERM=linux
```

TOC

Support for the Euro symbol

Since Europe had to invent a new character, the Euro, there's a new standard. It's called ISO-8859-15 (aka Latin-9 or Latin-0). To add support you need both a new font and new keymaps. They're included with the EURO-package and can be found at: ftp://ftp.freshmeat.net/pub/euro-patch/. Soon this will be part of every major distribution.

The changes in the keymap-file consist of the following lines:

```
keycode  18 = +e        +E      currency
keycode  46 = +c        +C      cent
```

And the following environment-variable should be set:

```
SYSFONT=lat0-16.psf
```

to make sure you're using the right font.

2.3 Time zone

Since Belgium is located in the Central European Time zone (aka MET) which (in the winter) is equivalent to the Greenwhich Mean Time plus 1 (GMT+1), you can simply link */usr/lib/zoneinfo/localtime/MET* to */etc/localtime* symbolicly like:

```
ln -sf /usr/lib/zoneinfo/MET /etc/localtime
```

or

```
ln -sf /usr/lib/zoneinfo/Europe/Brussels /etc/localtime
```

This automagically sets Daylight Savings (which is GMT+2 in the summer).

Note: that different distributions have different paths, zoneinfo can be located in */usr/share* or such...

Note2: RedHat uses a tool called `timeconfig`, SuSE uses `yast`

To change the CMOS clock and then synchronize the system time with the CMOS clock, do something like this:

```
hwclock --set --date="Feb 25 03:38"
clock -u -s
```

If your clock is set to local time (which is discouraged but if you have other OS's installed a must), you can do:

```
hwclock --set --date="Feb 25 04:38"
clock -s
```

[Noticed the 1 hour difference ? ;p]

2.4 Locale support

Not much programs support locale yet, if you want to know more about locale or how to write your own programs to support it, check: [put the damn url here ;p]

Be aware that locale makes it harder to resolve problems as less people can help you !! You might understand a problem when it's in technical English, but what if it's translated in some obscure French ?!? Some obviously won't understand both. ;p

Dutch

To enable support for the Dutch locale in Belgium on a system with locale support you just have to set the following environment variables:

```
LANG=nl
LANGUAGE=nl_BE
```

Try `nl_BE.ISO_8859-1` if `nl_BE` does not work.

French

If you want French locale support in Belgium, set the following environment variabels:

```
LANG=fr
LANGUAGE=fr_BE
```

Try `fr_BE.ISO_8859-1` if `fr_BE` does not work.

German

If you want instead German locale support in Belgium, set the following environment variabels:

```
LANG=de
LANGUAGE=de_BE
```

Try `de_BE.ISO_8859-1` if `de_BE` does not work.

Walloon

For Walloon locale support you can find files and information at: http://www.ping.be/linux/walon/. To use it set:

```
LANG=wa
```

```
LANGUAGE=wa_BE:fr_BE
```

Try `wa_BE.ISO_8859-1` if `wa_BE` does not work.

2.5 HOWTO translations

As Dutch is less wide-spread, it is the least supported. But work is being done to resolve this ;) (and maybe *you* can help ?!?) French and German translations obviously are already taken care of.

Dutch

A succesfull initiative is making progress, but there's still a lot to do. If you want to help, start translating and send it there. Check: http://nl.linux.org/doc/HOWTO/.

French

French translations of HOWTO's can be found at:
ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/translations/french/

German

German translations of HOWTO's can be found at:
ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/translations/german/

2.6 Native language support and codepages

These options add support for the Belgian characterset and codepages. If you'd like to have these, compile a kernel with these options put on. They only apply to some filesystems (FAT-based, ISO9660, ...).

```
< > Codepage 850 (Europe)
< > NLS ISO 8859-1  (Latin 1; Western European Languages)
< > NLS ISO 8859-15 (Latin 9; Western European Languages with Euro)
```

Located in `Filesystems` > `Native Language Support`

If your kernel is a modular kernel that came with your distribution (=you never compiled a kernel before), then you might want to type:

```
insmod nls_cp850
insmod nls_iso8859_1
insmod nls_iso8859_15
```

3. Setting up Internet (Belgian ISP information)

3.1 Dialing in with your modem (PPP)

To configure PPP correctly, you should check the PPP-HOWTO. I try to explain it briefly, but if you have any questions or problems, read the HOWTO very carefully.

Configuring the modem

To configure your modem, you can alter the initstring of your modem. To learn more about your modem, check Ask Mr. Modem. Beware, if you have a winmodem you will not get it to work. More information about winmodems is at: http://linmodems.org/

Installing pppd

Now, you need to have the ppp-package installed (check this by typing `pppd --version`), make sure you have a recent one. Then you have to make sure you've got these files:

/usr/local/bin/ppp

```
        #!/bin/sh
        case $1 in
                on|start)
                TELEPHONE='555-1212'    # The telephone number for the connection
                ACCOUNT='your_login'    # The account name for logon (as in 'George
Burns')
                PASSWORD='your_pass'    # The password for this account (and 'Gracie
Allen')
                INITSTRING='AT'         # Modem initstring
                IPLOCAL=0.0.0.0         # Local IP address if known. Dynamic =
0.0.0.0
                IPREMOTE=0.0.0.0        # Remote IP address if desired. Normally
0.0.0.0
                DEVICE=/dev/ttyS1       # Serial Device com1=ttyS0, com2=ttyS1,...
                SPEED=57600             # 19200, 38400 or 57600 (don't try something
different)
                export TELEPHONE ACCOUNT PASSWORD INITSTRING
                exec /usr/sbin/pppd $DEVICE $SPEED $IPLOCAL:$IPREMOTE \
                        user $ACCOUNT connect /etc/ppp/ppp-on-dialer
                ;;
                off|stop)
                if [ -r /var/run/ppp0.pid ]; then
                        kill -INT `cat /var/run/ppp0.pid`
                        if [ ! "$?" = "0" ]; then
                                rm -f /var/run/ppp0.pid
                                echo "ERROR: Removed stale pid file"
                                exit 1
                        fi
                        exit 0
```

```
        fi
        echo "ERROR: PPP link is not active on ppp0"
        exit 1
        ;;
        info|status)
        /usr/sbin/pppstats
        ;;
        *)
        echo "Usage: ppp on|off|info|start|stop|status";
        ;;
esac
```

/etc/ppp/ppp-on-dialer

```
#!/bin/sh
exec /usr/sbin/chat -v                                  \
        TIMEOUT         3                               \
        ABORT           '\nBUSY\r'                      \
        ABORT           '\nNO ANSWER\r'                 \
        ABORT           '\nNO CARRIER\r'                \
        ABORT           '\nNO DIALTONE\r'               \
        ABORT           '\nRINGING\r\n\r\nRINGING\r'    \
        ''              "\r$INITSTRING\r"               \
        'OK-+++\c-OK'   'ATH0'                          \
        TIMEOUT         30                              \
        OK              "ATDT$TELEPHONE"                \
        CONNECT         ''                              \
        ogin:--ogin:    "$ACCOUNT"                      \
        ord:            "$PASSWORD"
```

You can remove the `-v` once everything looks normal.

The pppd command uses */etc/ppp/options* to list its options, change these options to whatever fits. These defaults normally work in most cases.

```
asyncmap 20a0000
crtscts
debug
default-mru
defaultroute
detach
escape 11,13,ff
hide-password
ipcp-accept-local
ipcp-accept-remote
lcp-echo-failure 4
lcp-echo-interval 400
modem
mtu 1500
netmask 255.255.255.0
noipdefault
passive
```

```
#idle 300
#kdebug 0
#lock
#-vj
```

Once dialing in works like a charm, you can leave `debug` out.

pap-secrets / chap-secrets

If your ISP requires CHAP or PAP, you need one of these files with their proper username and password:

/etc/ppp/pap-secrets or */etc/ppp/chap-secrets*

```
#client         server  secret          IP
your_login      *       your_pass       *
```

Network

To configure your nameservers, change */etc/resolv.conf*. If dialing in is succesfull, but you can't get it to connect to URL's: this is probably the cause.

```
domain domain_of_your_isp
nameserver your_primary_nameserver
nameserver your_secondary_nameserver
```

Dial + Hangup

Then if */usr/local/bin* is in your `PATH`, you can easily dial in with:

```
ppp on
```

and hangup with:

```
ppp off
```

and to get some statistics, you do:

```
ppp status
```

ISP-dependant information

Digibel:

Web:

http://www.digibel.org/

Eunet:

Authentication:

text-based

Options:

proxyarp

Web:

http://www.eunet.be/

Euronet (Wannadoo):

Authentication:

text-based

Web:

http://www.euronet.be/

Online:

Authentication:

text-based

Options:

noccp

Web:

http://www.online.be/

Ping:

Authentication:

text-based

Web:

http://www.ping.be/

Planet Internet:

Authentication:

CHAP

Web:

http://www.planetinternet.be/

Skynet (Belgacom):

Authentication:

PAP

Options:

defaultroute

Web:

http://www.skynet.be/

UUNet:

Authentication:

PAP

Web:

http://www.uunet.be/

XS4ALL:

Authentication:

PAP

Web:

http://www.xs4all.be/

3.2 Dialing in with your ISDN-modem

Contributed by Christophe Lambin clambin@skynet.be

This section does not aim to be a definitive guide on ISDN for Linux. For a more detailed discussion on the topic, see Paul Slootman's ISDN4Linux HOWTO and ISDN4Linux-FAQ.

Updating the kernel

If you're using a 2.2 kernel, it's recommended to get an updated version of the ISDN code. You can retrieve the CVS at ftp://ftp.suse.com/pub/isdn4linux/. For more information, check: http://www.brisse.dk/site/linux/docs/isdn.htm

Warning: if you've recently purchased an Eicon Diva 2.0 PCI (eg, via Belgacom), there's a good chance you've actually got a Diva 2.01.In this case, you must get a version dated 1 July 1999 or later !

Once the ISDN is in place, you can start up the driver. Eg:

```
modprobe hisax type=11 protocol=2
```

Refer to the Readme's for the right parameters and values for your card.

ISDN4Linux toolkit

Next, you'll need the ISDN4Linux toolkit. You can retrieve the latest version of the toolkit at: ftp://ftp.suse.com/pub/isdn4linux/

Configuring the interface

With the tools installed and configured, write a script to configure the interface, to be used for the ISDN connection. As always in Linux, there's no one correct way of doing this. I've put these in a script

/etc/rc.d/rc.isdn:

```
#!/bin/sh
MSNREMOTE='555-1212'    # Phone number of ISP
MSNLOCAL='555-1313'     # my number, without 0, with areacode
ACCOUNT='george'        # The account for logon (as 'George Burns')
IPLOCAL=10.0.0.2        # my fixed IP (use 10.0.0.2 if no fixed)
IPREMOTE=0.0.0.0        # IP number of ISP
INTERFACE=ippp0
/sbin/modprobe hisax type=11 protocol=2
/sbin/isdnctrl verbose 3 system on
/sbin/isdnctrl addif $IF
/sbin/isdnctrl secure $IF on
/sbin/isdnctrl addphone $IF out $MSNREMOTE
/sbin/isdnctrl eaz $IF $MSNLOCAL
```

```
        /sbin/isdnctrl huptimeout $IF 300
        /sbin/isdnctrl l2_prot $IF hdlc
        /sbin/isdnctrl l3_prot $IF trans
        /sbin/isdnctrl encap $IF syncppp
#       /sbin/isdnctrl dialmode $IF auto
        /sbin/ifconfig $IF $IPLOCAL pointopoint $IPREMOTE -arp -broadcast
        /sbin/ipppd /dev/ippp0 user $ACCOUNT $IPLOCAL:$IPREMOTE
```

To start this at boot time, make it executable and append the following to */etc/rc.d/rc.local*:

```
        if [ -x /etc/rc.d/rc.isdn ]; then
                . /etc/rc.d/rc.isdn
        fi
```

The ipppd command gets its parameters passed through a file, */etc/ppp/ioptions*:

```
        -ac
        -bsdcomp
        debug
        defaultroute
        ipcp-accept-local
        ipcp-accept-remote
        mru 1524
        mtu 1500
        noipdefault
        -pc
        useifip
        -vj
        -vjccomp

        #idle 360
        #persistent
```

Warning: do NOT specify +pap or +chap in this file. This specifies the authentication that ipppd should use for an INCOMING client. If you were to use this to connect to your ISP, ipppd would wait for the ISP to authenticate itself using the specified protocol.

Finally, create */etc/ppp/ip-down.local* to handle the shutdown of the interface:

```
        #!/bin/sh
        /sbin/ifconfig $1 down
        sleep 1
        /sbin/ifconfig $1 10.0.0.2 pointopoint
```

Note: this is to handle some problems with routes on shutdowns. Anyone know of a clean(er) solution ?

pap-secrets / chap-secrets

If your ISP uses PAP or CHAP, create a file */etc/ppp/pap-secrets* or */etc/ppp/chap-secrets*. Its format is:

```
        #client         server  secret          IP
```

```
your_login      *       your_pass       *
```

Network

Not really related to ISDN, but a lot of people forget this step (too eager to try out all the stuff they typed in above ? :-)).

Anyway, you need to configure the DNS, by creating a file */etc/resolv.conf*:

```
domain your_domain_of_your_isp
nameserver your_primary_nameserver
nameserver your_secondary_nameserver
```

Dial + Hangup

Finally, you can write a little wrapper to start and stop the ISDN connection. I've put this as */usr/local/bin/isdn*:

```
#!/bin/bash
case $1 in
        on|start)
        /sbin/isdnctrl dial ippp0
        ;;
        off|stop)
        /sbin/isdnctrl hangup ippp0
        ;;
        info|status)
        /sbin/isdnctrl list ippp0
        ;;
        *)
        echo "Usage: isdn on|off|info|start|stop|status"
        ;;
esac
```

Then if */usr/local/bin* is in your `PATH`, you can easily dial in with:

```
isdn on
```

and hangup with:

```
isdn off
```

3.3 Cablemodem (DHCP)

DHCP was designed to make life easier, and most of the times it does ;) More information to set up your DHCP client can be found at: http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html which describes everything you need to know. (along with the information below)

Lots of people want to connect a whole network to their cablemodem, that's not a problem. You'll need IP-Masquerading (http://metalab.unc.edu/LDP/HOWTO/mini/IP-Masquerade.html) and make sure the interface (that speaks to the cablemodem) has the right MAC-address !! Another solution (in case of problems) is to reset your cablemodem.

TOC

Chello (TVD):

Web:

http://www.chello.be/

More information:

http://foobar.starlab.net/~soggie/tvd_linux/

Pandora (Telenet):

Automatic proxy:

http://pac.pandora.be:8080

http proxy:

export http_proxy="http://proxy.pandora.be:8080/"

Web:

http://www.pandora.be/

3.4 ADSL

Turboline (Belgacom):

Some things:

using DANA

Web:

http://www.turboline.be/

More information:

http://foobar.starlab.net/~soggie/turboline/

---

4. Belgian Linux user groups

More information can be found in the User Group HOWTO located at: http://www.ntlug.org/~kclark/ A complete list of Linux User Groups is at: http://www.ssc.com/linux/glue/ and at: http://limestone.uoregon.edu/woven/lugww/

4.1 ALLIN Linux User Group

```
Mechelbaan 260
B-3130 Begijnendijk
```

Email:

allin@linuxfreak.com

Web:

http://come.to/allin/

Contact:

Paul Terweduwe paul@faes.be

Meetingplace:

'T Paviljoen in Heist Op Den Berg

Last modified:

13 April 1999.

4.2 Antwerp Linux User Group (ALUG)

```
Van Luppenstraat 70
B-2018 Antwerp
```

Phone:

+32 3 218 63 54

Email:

fv.alug@advalvas.be

TOC

Web:

http://sunsite.belnet.be/alug/

Contact:

Patrick Op de Beeck poe@glo.be

Meetings:

Almost every friday at 19h, see homepage first !

Meetingplace:

Hof van Rieth, Molenlei 68, Mortsel / Cultureel Centrum Berchem, Driekoningenstraat 126

Last modified:

8 March 1999.

4.3 Brussels Linux User Group (BLUG)

```
Brussels Linux User Group (BLUG)
```

Email:

johanb@linux.rave.org

Web:

http://linux.rave.org/

Last modified:

24 February 1999.

4.4 HCC Limburgse Linux Gebruikers Groep (HLLGG)

```
HCC Limburgse Linux Gebruikers Groep
```

Email:

Linuxgg@hotmail.com

Web:

http://www.uunet.be/hcc/Limburg/Linuxgg

Contact:

Andy Peeters & Bruno Tony

Last modified:

31 May 1999.

4.5 HCC Linux Gebruikers Groep Antwerpen

```
Armand Verachtert
Turnhoutsebaan 421
2140 Borgerhout
```

Web:

http://users.glo.be/~poe/index.html

Last modified:

25 February 1999.

4.6 HCC Linux Gebruikers Groep Leuven

```
Diegemstraat 61
B-1930 Zaventem
```

Email:

Yvo.Dries@village.uunet.be

Web:

http://www.uunet.be/hcc/

Contact:

Yvo Dries Yvo.Dries@village.uunet.be

Last modified:

2 August 1999.

4.7 Independent Group of Unix-Alikes and Network Activists (IGUANA)

```
Milcampslaan 101
```

```
B-1040 Schaarbeek
```

Phone:

+32 2 657 77 36

Fax:

+32 2 657 77 38

Email:

info@iguana.be

Web:

http://www.iguana.be/

Last modified:

5 March 1999.

4.8 Infogroep (IGWE)

```
Pleinlaan 2, 5F218 (building F)
B-1050 Brussels
```

Phone:

+32 2 629 33 56

Fax:

+32 2 629 33 89

Email:

ig@igwe.vub.ac.be

Web:

http://igwe.vub.ac.be/

Last modified:

6 July 1999.

4.9 Leuvense Linux Users (L2U)

```
Leuvense Linux Users (L2U)
```

Email:

darkcompanion@village.uunet.be

Web:

http://gallery.uunet.be/darkcompanion/linux/

Last modified:

23 February 1999.

4.10 Limburgse Linux User Group (LILUG)

```
Limburgse Linux User Group (LILUG)
```

Email:

frank.eersels@ping.be

Web:

http://lilug.linuxatwork.at/

Contact:

Frank Eersels frank.eersels@ping.be

Last modified:

6 May 1999.

4.11 Linux Belgian Team

```
Rue d'Aineffe 4
B-4317 Borlez Faimes
```

Email:

manu@rtfm.be

Web:

http://linux.rtfm.be/

Contact:

Alexandre J.D. Dulaunoy adulau@rtfm.be

Last modified:

25 February 1999.

4.12 Linux Gebruikers Oost-Vlaanderen (LGOVL)

```
Linux Gebruikers Oost-Vlaanderen
```

Email:

dennis@lgovl.freeservers.com

Web:

http://lgovl.freeservers.com/

Last modified:

25 February 1999.

4.13 Linux User Group De Bruxelles (BxLUG)

```
Linux User Group De Bruxelles (BxLUG)
```

Email:

bxl_information@linuxbe.org

Web:

http://bxlug.linuxbe.org/

Contact:

Raphael Bauduin bxl_information@linuxbe.org

Last modified:

17 July 1999.

4.14 Linux User Group De Namur

```
Linux User Group De Namur
```

Email:

namurlugs@linuxbe.org

Web:

http://namur.linuxbe.org/

Last modified:

17 July 1999.

4.15 Linux User Group Mons

```
Linux User Group Mons
```

Email:

webmaster@linuxwallonie.org

Web:

http://www.linuxwallonie.org/

Last modified:

26 May 1999.

4.16 Linux User Movement Underground Mad Belgian Aliens (LUMUMBA)

```
Universiteitslaan 1, Filii Lamberti
3590 Diepenbeek
```

Email:

helpdesk@lumumba.luc.ac.be

Web:

http://lumumba.luc.ac.be/

Last modified:

25 February 1999.

4.17 linuxbe.org

```
linuxbe.org
```

Email:

info@lists.linuxbe.org

Web:

http://linuxbe.org/

Contact:

Cedric Gavage niddle@linuxbe.org

Last modified:

5 March 1999.

4.18 Local Linux Support Group Gent (LLSGG)

```
Wim Vandeputte
PB 144
B-9000 Gent 12
```

Phone:

+32 9 240 29 17

Fax:

+32 9 240 34 39

Email:

wvdputte@reptile.rug.ac.be

Web:

http://llsgg.rug.ac.be/

Contact:

Wim Vandeputte wvdputte@reptile.rug.ac.be

Last modified:

4 March 1999.

4.19 Open Technology Assembly Linux Special Interest Group

```
Kruipstraat 14
B-1850 Grimbergen
```

Phone:

+32 11 27 58 98

Fax:

+32 11 27 03 89

Email:

info@ota.be

Web:

http://www.ota.be/sigs/linux/

Contact:

Jan Vanhercke jan.vanhercke@c-cure.be

Last modified:

4 March 1999.

4.20 Student Information Networking (SIN)

```
Kleinhoefstraat 4
B-2440 Geel
```

Email:

sin@sin.khk.be

Web:

http://www.sin.khk.be/

Last modified:

12 June 1999.

4.21 TINA

Email:

vraagtina@digibel.org

Web:

http://tina.digibel.org/

Contact:

Roman Pauwels art@ping.be

Last modified:

3 September 1999.

4.22 Unix Lovers Yield Student Services & Internet Support (ULYSSIS)

```
Blijde Inkomsstraat 21, lokaal 01.17
B-3000 Leuven
```

Phone:

+32 16 32 48 61

Email:

ulyssis@ulyssis.student.kuleuven.ac.be

Web:

http://www.ulyssis.student.kuleuven.ac.be/org/ulyssis/

Last modified:

25 February 1999.

4.23 Zeus WPI - Werkgroep Informatica RUG

```
Krijgslaan 281 S9
B-9000 Gent
```

Phone:

+32 09 264 47 51

Email:

info@zeus.rug.ac.be

Web:

http://www.zeus.rug.ac.be/

Comment:

Restricted to RUG students

Last modified:

22 June 1999.

5. Belgian Linux businesses

The Linux Documentation Project contains a worldwide list of Linux consultants, Linux Consultants HOWTO, and can be found at: http://consult.cyrius.com/.

5.1 aDOC Services

```
Avenue du Pesage, 31-33
B-1050 Brussels
```

Phone:

+32 2 646 00 76

Email:

sales@adoc-services.com

Web:

http://www.adoc-services.com/

Last modified:

25 February 1999.

5.2 Better Access N.V.

```
Geldenaakse Vest 6
3000 Leuven
```

Phone:

+32 16 29 80 45

Fax:

+32 16 29 80 46

Email:

info@ba.be

Web:

TOC

http://www.ba.be/

Type of support:

After an onsite installation, we mainly support our customers with SSH remote administration. Most problems are reported by E-mail or by the inhouse-developped webbased supportsystem. This doesn't mean that once in a while a supportcall comes in.

Special expertise:

Security, systemadministration and setups, security (firewalling, tigerteaming, VPN, etc.) Networkdesign, wireless networking, Troubleshooting, product development, etc. Please visit http://www.ba.be for more info.

Last modified:

7 March 1999.

5.3 D. Connect

```
Av. Van Goidtsnoven 33
B-1180 Uccle
```

Email:

webmaster@dconnect.be

Web:

http://www.dconnect.be/

Last modified:

25 February 1999.

5.4 Double Barrel Consultancy and Productions

```
Sportstraat 28
B-9000 Gent
```

Phone:

+32 9 222 77 64

Fax:

+32 9 222 49 76

Email:

TOC

mvergall@double-barrel.be

Web:

http://www.double-barrel.be/

Contact:

Michael C. Vergallen mvergall@double-barrel.be

Type of support:

phone, email remote network admin, inhouse, onsite.

Special expertise:

installation, system admin. internet and intranet connectivity, sendmail, firewalls, proxys, nameservers, troubleshooting software & hardware.

Last modified:

5 March 1999.

5.5 grmbl productions

```
Korte Vuldersstraat
B-8000 Brugge
```

Phone:

+32 50 344366

Email:

info@grmbl.com

Web:

http://www.grmbl.com/

Last modified:

2 March 1999.

5.6 LIFE

```
Mgr. Vanwaeyenberglaan 6/10
B-3000 Leuven
```

Phone:

TOC

+32 477 57 50 71

Email:

info@life.be

Web:

http://www.life.be/

Contact:

Jasper Nuyens jasper@life.be

Type of support:

LIFE is the first company in Europe to provide 24/7 professional support for Linux in Europe. LIFE also provides server and desktop installation services, educational services and fixed price development. Exclusive on the Linux operating system.

Special expertise:

Kernel driver development, database interconnectivity (VPN, wireless networking), security auditing (firewall, VPN, ethical hacking), e-commerce. See our website for more.

Last modified:

20 April 1999.

5.7 Phidani Software SPRL

```
Rue de l'autonomie 1
B-1070 Brussels
```

Email:

info@phidani.be

Web:

http://www.phidani.be/

Last modified:

25 February 1999.

5.8 Spier BVBA

```
Knaptandstraat 96-98
B-9100  Sint-Niklaas
```

Email:

info@spier,be

Web:

http://www.spier.be/

Last modified:

02 March 1999.

5.9 VirgoPlus sprl

```
Rue Dartois 1
B-4000 Liege
```

Phone:

+32 4 253 00 59

Fax:

+32 4 253 00 49

Email:

contact@virgoplus.com

Web:

http://www.virgoplus.com/

Contact:

Bruno Mairlot bruno@virgoplus.com

Type of support:

By phone, email, or remote networked administration, inhouse, onsite

Special expertise:

Installation, configuration, administration of Linux, internet connection, intranet developpment, firewalls, samba, netatalk, firewalls, proxys, DNS, web server (Apache), mod_perl, MySQL, Gui interface developpment (GTK+). We have acquired a special expertise with the Red Hat distributions.

Last modified:

28 June 1999.

6. Belgian Linux resources

6.1 Websites

Linux in Belgium

- Linux in Belgium
- Belgian HOWTO

Dutch Linux-sites

- http://nl.linux.org/doc/
- Nederlandstalige Linux Documentatie Project
- Linux Support Website
- Beginnen met Linux
- Installatie Handleiding RedHat 5.2

French Linux-sites

- linuxbe.org

Mirrors in Belgium

- Freshmeat at Chello (TVD)
- Linuxberg at Wallonie Internet
- Linuxberg at Online
- Linuxberg at Pandora (Telenet)
- Linux Gazette at Belnet

6.2 HOWTO

The most important collection of Linux information on the net. Please if you encounter erroneous information in one of them, do contact the author. Only that way it will be corrected and we all stand together ;)

TOC

Web

- HOWTO's at KULeuven
- HOWTO's at SIN

FTP

- HOWTO's at Belnet

6.3 FTP

Most of the times, it doesn't matter where you got it from, as long as it works. But to save you some time and save us some bandwidth, it's much better to get it near you ;) Naturally I forgot some, please send your additions or corrections.

Kernel-sources

- Official Mirror kernel-sources at Chello (TVD)
- Mirror kernel-sources at linux.be.eu.org
- Mirror kernel-sources at Digibel

Distributions

- Red Hat
  - Official Mirror at Belnet
  - Mirror at Chello (TVD)
  - Mirror at linux.be.eu.org
  - Mirror at Pandora (Telenet)
  - Mirror at Digibel
- Debian
  - Official Mirror at KULeuven
  - Official Mirror at spnet
  - Mirror at EUnet
  - Mirror at Skynet
  - Mirror at Pandora (Telenet)
  - Mirror at Interpac
- Slackware
  - Official Mirror at Interpac

TOC

    - Mirror at Skynet
    - Mirror at Globe
- SuSE
    - Mirror at Pandora (Telenet)
- Mandrake
    - Mirror at Belnet

Software

- Mirrors at Belnet
- Mirrors at Chello (TVD)
- Mirrors at linux.be.eu.org
- Mirrors at Pandora (Telenet)
- Mirrors at Digibel

6.4 Usenet

- news:be.comp.os.linux at WebWatch

6.5 Mailinglists

Linux user groups

- BLUG Mailinglist
- L2U Mailinglist
- LILUG Mailinglist
- LGOVL Mailinglist

6.6 IRC

The fastest medium to get support is IRC, but beware. Because IRC is rather anonymous it's also easy to fool people.

On IRCnet there's a channel called #linux.be that is closely related to bcol and the Belgian Linux-scene. The channel is still small, but it surely will grow in time. To get help in English, there's only one place #linux.

Linux Benchmarking HOWTO

by André D. Balsa, andrewbalsa@usa.net

v0.12, 15 August 1997

The Linux Benchmarking HOWTO discusses some issues associated with the benchmarking of Linux systems and presents a basic benchmarking toolkit, as well as an associated form, which enable one to produce significant benchmarking information in a couple of hours. Perhaps it will also help diminish the amount of useless articles in comp.os.linux.hardware...

1. Introduction

- 1.1 Why is benchmarking so important ?
- 1.2 Invalid benchmarking considerations

2. Benchmarking procedures and interpretation of results

- 2.1 Understanding benchmarking choices
- 2.2 Standard benchmarks available for Linux
- 2.3 Links and references

3. The Linux Benchmarking Toolkit (LBT)

- 3.1 Rationale
- 3.2 Benchmark selection
- 3.3 Test duration
- 3.4 Comments
- 3.5 Possible improvements
- 3.6 LBT Report Form
- 3.7 Network performance tests

- 3.8 SMP tests

4. Example run and results

5. Pitfalls and caveats of benchmarking

- 5.1 Comparing apples and oranges
- 5.2 Incomplete information
- 5.3 Proprietary hardware/software
- 5.4 Relevance

6. FAQ

7. Copyright, acknowledgments and miscellaneous

- 7.1 How this document was produced
- 7.2 Copyright
- 7.3 New versions of this document
- 7.4 Feedback
- 7.5 Acknowledgments
- 7.6 Disclaimer
- 7.7 Trademarks

1. Introduction

"What we cannot speak about we must pass over in silence."

Ludwig Wittgenstein (1889-1951), Austrian philosopher

Benchmarking means **measuring** the speed with which a computer system will execute a computing task, in a way that will allow comparison between different hard/software combinations. It **does not** involve user-friendliness, aesthetic or ergonomic considerations or any other subjective judgment.

Benchmarking is a tedious, repetitive task, and takes attention to details. Very often the results are not what one would expect, and subject to interpretation (which actually may be the most important part of a benchmarking procedure).

Finally, benchmarking deals with facts and figures, not opinion or approximation.

1.1 Why is benchmarking so important ?

Apart from the reasons pointed out in the BogoMips Mini-HOWTO (section 7, paragraph 2), one occasionally is confronted with a limited budget and/or minimum performance requirements while putting together a Linux box. In other words, when confronted with the following questions:

- How do I maximize performance within a given budget ?
- How do I minimize costs for a required minimum performance level ?
- How do I obtain the best performance/cost ratio (within a given budget or given performance requirements)?

one will have to examine, compare and/or produce benchmarks. Minimizing costs with no performance requirements usually involves putting together a machine with leftover parts (that old 386SX-16 box lying around in the garage will do fine) and does not require benchmarks, and maximizing performance with no cost ceiling is not a realistic situation (unless one is willing to put a Cray box in his/her living room - the leather-covered power supplies around it look nice, don't they ?).

Benchmarking per se is senseless, a waste of time and money; it is only meaningful as part of a decision process, i.e. if one has to make a choice between two or more alternatives.

Usually another parameter in the decision process is **cost**, but it could be availability, service, reliability, strategic considerations or any other rational, measurable characteristic of a computer system. When comparing the performance of different Linux kernel versions, for example, **stability** is almost always more important than speed.

1.2 Invalid benchmarking considerations

TOC

Very often read in newsgroups and mailing lists, unfortunately:

1. Reputation of manufacturer (unmeasurable and meaningless).
2. Market share of manufacturer (meaningless and irrelevant).
3. Irrational parameters (for example, superstition or prejudice: would you buy a processor labeled 131313ZAP and painted pink ?)
4. Perceived value (meaningless, unmeasurable and irrational).
5. Amount of marketing hype: this one is the worst, I guess. I personally am fed up with the "XXX inside" or "kkkkkws compatible" logos (now the "aaaaaPowered" has joined the band - what next ?). IMHO, the billions of dollars spent on such campaigns would be better used by research teams on the design of new, faster, (cheaper :-) bug-free processors. No amount of marketing hype will remove a floating-point bug in the FPU of the brand-new processor you just plugged in your motherboard, but an exchange against a redesigned processor will.
6. "You get what you pay for" opinions are just that: opinions. Give me the facts, please.

---

2. Benchmarking procedures and interpretation of results

A few semi-obvious recommendations:

1. First and foremost, **identify your benchmarking goals**. What is it you are exactly trying to benchmark ? In what way will the benchmarking process help later in your decision making ? How much time and resources are you willing to put into your benchmarking effort ?
2. **Use standard tools**. Use a current, stable kernel version, standard, current gcc and libc and a standard benchmark. In short, use the LBT (see below).
3. Give a **complete description** of your setup (see the LBT report form below).
4. Try to **isolate a single variable**. Comparative benchmarking is more informative than "absolute" benchmarking. **I cannot stress this enough.**
5. **Verify your results**. Run your benchmarks a few times and verify the variations in your results, if any. Unexplained variations will invalidate your results.
6. If you think your benchmarking effort produced meaningful information, **share it** with the Linux community in a **precise** and **concise** way.
7. Please **forget about BogoMips**. I promise myself I shall someday implement a very fast ASIC with the BogoMips loop wired in. Then we shall see what we shall see !

2.1 Understanding benchmarking choices

Synthetic vs. applications benchmarks

Before spending any amount of time on benchmarking chores, a basic choice must be made between "synthetic" benchmarks and "applications" benchmarks.

Synthetic benchmarks are specifically designed to measure the performance of individual components of a computer system, usually by exercising the chosen component to its maximum capacity. An example of a well-known synthetic benchmark is the **Whetstone** suite, originally programmed in 1972 by Harold Curnow in FORTRAN (or was that ALGOL ?) and still in widespread use nowadays. The Whestone suite will measure the floating-point performance of a CPU.

The main critic that can be made to synthetic benchmarks is that they do not represent a computer system's performance in real-life situations. Take for example the Whetstone suite: the main loop is very short and will easily fit in the primary cache of a CPU, keeping the FPU pipeline constantly filled and so exercising the FPU to its maximum speed. We cannot really criticize the Whetstone suite if we remember it was programmed 25 years ago (its design dates even earlier than that !), but we must make sure we interpret its results with care, when it comes to benchmarking modern microprocessors.

Another very important point to note about synthetic benchmarks is that, ideally, they should tell us

TOC

something about a **specific** aspect of the system being tested, independently of all other aspects: a synthetic benchmark for Ethernet card I/O throughput should result in the same or similar figures whether it is run on a 386SX-16 with 4 MBytes of RAM or a Pentium 200 MMX with 64 MBytes of RAM. Otherwise, the test will be measuring the overall performance of the CPU/Motherboard/Bus/Ethernet card/Memory subsystem/DMA combination: not very useful since the variation in CPU will cause a greater impact than the change in Ethernet network card (this of course assumes we are using the same kernel/driver combination, which could cause an even greater variation)!

Finally, a very common mistake is to average various synthetic benchmarks and claim that such an average is a good representation of real-life performance for any given system.

Here is a comment on FPU benchmarks quoted with permission from the Cyrix Corp. Web site:

> *"A Floating Point Unit (FPU) accelerates software designed to use floating point mathematics : typically CAD programs, spreadsheets, 3D games and design applications. However, today's most popular PC applications make use of both floating point and integer instructions. As a result, Cyrix chose to emphasize "parallelism" in the design of the 6x86 processor to speed up software that intermixes these two instruction types.*
>
> *The x86 floating point exception model allows integer instructions to issue and complete while a floating point instruction is executing. In contrast, a second floating point instruction cannot begin execution while a previous floating point instruction is executing. To remove the performance limitation created by the floating point exception model, the 6x86 can speculatively issue up to four floating point instructions to the on-chip FPU while continuing to issue and execute integer instructions. As an example, in a code sequence of two floating point instructions (FLTs) followed by six integer instructions (INTs) followed by two FLTs, the 6x86 processor can issue all ten instructions to the appropriate execution units prior to completion of the first FLT. If none of the instructions fault (the typical case), execution continues with both the integer and floating point units completing instructions in parallel. If one of the FLTs faults (the atypical case), the speculative execution capability of the 6x86 allows the processor state to be restored in such a way that it is compatible with the x86 floating point exception model.*
>
> *Examination of benchmark tests reveals that synthetic floating point benchmarks use a pure floating point-only code stream not found in real-world applications. This type of benchmark does not take advantage of the speculative execution capability of the 6x86 processor. Cyrix believes that non-synthetic benchmarks based on real-world applications better reflect the actual performance users will achieve. Real-world applications contain intermixed integer and floating point instructions and therefore benefit from the 6x86 speculative execution capability."*

So, the recent trend in benchmarking is to choose common applications and use them to test the performance of complete computer systems. For example, **SPEC**, the non-profit corporation that designed the well-known SPECINT and SPECFP synthetic benchmark suites, has launched a project for a new applications benchmark suite. But then again, it is very unlikely that such commercial benchmarks will ever include any Linux code.

Summarizing, synthetic benchmarks are valid as long as you understand their purposes and limitations. Applications benchmarks will better reflect a computer system's performance, but none are available for

Linux.

High-level vs. low-level benchmarks

Low-level benchmarks will directly measure the performance of the hardware: CPU clock, DRAM and cache SRAM cycle times, hard disk average access time, latency, track-to-track stepping time, etc... This can be useful in case you bought a system and are wondering what components it was built with, but a better way to check these figures would be to open the case, list whatever part numbers you can find and somehow obtain the data sheet for each part (usually on the Web).

Another use for low-level benchmarks is to check that a kernel driver was correctly configured for a specific piece of hardware: if you have the data sheet for the component, you can compare the results of the low-level benchmarks to the theoretical, printed specs.

High-level benchmarks are more concerned with the performance of the hardware/driver/OS combination for a specific aspect of a microcomputer system, for example file I/O performance, or even for a specific hardware/driver/OS/application performance, e.g. an Apache benchmark on different microcomputer systems.

Of course, all low-level benchmarks are synthetic. High-level benchmarks may be synthetic or applications benchmarks.

2.2 Standard benchmarks available for Linux

IMHO a simple test that anyone can do while upgrading any component in his/her Linux box is to launch a kernel compile before and after the hard/software upgrade and compare compilation times. If all other conditions are kept equal then the test is valid as a measure of compilation performance and one can be confident to say that:

> "Changing A to B led to an improvement of x % in the compile time of the Linux kernel under such and such conditions".

No more, no less !

Since kernel compilation is a very usual task under Linux, and since it exercises most functions that get exercised by normal benchmarks (except floating-point performance), it constitutes a rather good **individual** test. In most cases, however, results from such a test cannot be reproduced by other Linux users because of variations in hard/software configurations and so this kind of test cannot be used as a "yardstick" to compare dissimilar systems (unless we all agree on a standard kernel to compile - see below).

Unfortunately, there are no Linux-specific benchmarking tools, except perhaps the Byte Linux Benchmarks which are a slightly modified version of the Byte Unix Benchmarks dating back from May 1991 (Linux mods by Jon Tombs, original authors Ben Smith, Rick Grehan and Tom Yager).

There is a central Web site for the Byte Linux Benchmarks.

An improved, updated version of the Byte Unix Benchmarks was put together by David C. Niemi. It is called UnixBench 4.01 to avoid confusion with earlier versions. Here is what David wrote about his

mods:

> *"The original and slightly modified BYTE Unix benchmarks are broken in quite a number of ways which make them an unusually unreliable indicator of system performance. I intentionally made my "index" values look a lot different to avoid confusion with the old benchmarks."*

David has setup a majordomo mailing list for discussion of benchmarking on Linux and competing OSs. Join with "subscribe bench" sent in the body of a message to majordomo@wauug.erols.com. The Washington Area Unix User Group is also in the process of setting up a Web site for Linux benchmarks.

Also recently, Uwe F. Mayer, mayer@math.vanderbilt.edu ported the BYTE Bytemark suite to Linux. This is a modern suite carefully put together by Rick Grehan at BYTE Magazine to test the CPU, FPU and memory system performance of modern microcomputer systems (these are strictly processor-performance oriented benchmarks, no I/O or system performance is taken into account).

Uwe has also put together a Web site with a database of test results for his version of the Linux BYTEmark benchmarks.

While searching for synthetic benchmarks for Linux, you will notice that sunsite.unc.edu carries few benchmarking tools. To test the relative speed of X servers and graphics cards, the xbench-0.2 suite by Claus Gittinger is available from sunsite.unc.edu, ftp.x.org and other sites. Xfree86.org refuses (wisely) to carry or recommend any benchmarks.

The XFree86-benchmarks Survey is a Web site with a database of x-bench results.

For pure disk I/O throughput, the hdparm program (included with most distributions, otherwise available from sunsite.unc.edu) will measure transfer rates if called with the -t and -T switches.

There are many other tools freely available on the Internet to test various performance aspects of your Linux box.

2.3 Links and references

The comp.benchmarks.faq by Dave Sill is the standard reference for benchmarking. It is not Linux specific, but recommended reading for anybody serious about benchmarking. It is available from a number of FTP and web sites and lists **56 different benchmarks**, with links to FTP or Web sites that carry them. Some of the benchmarks listed are commercial (SPEC for example), though.

I will not go through each one of the benchmarks mentionned in the comp.benchmarks.faq, but there is at least one low-level suite which I would like to comment on: the lmbench suite, by Larry McVoy. Quoting David C. Niemi:

> *"Linus and David Miller use this a lot because it does some useful low-level measurements and can also measure network throughput and latency if you have 2 boxes to test with. But it does not attempt to come up with anything like an overall "figure of merit"..."*

A rather complete FTP site for **freely** available benchmarks was put together by Alfred Aburto. The Whetstone suite used in the LBT can be found at this site.

There is a **multipart FAQ by Eugene Miya** that gets posted regularly to comp.benchmarks; it is an excellent reference.

3. The Linux Benchmarking Toolkit (LBT)

I will propose a basic benchmarking toolkit for Linux. This is a preliminary version of a comprehensive Linux Benchmarking Toolkit, to be expanded and improved. Take it for what it's worth, i.e. as a proposal. If you don't think it is a valid test suite, feel free to email me your critics and I will be glad to make the changes and improve it if I can. Before getting into an argument, however, read this HOWTO and the mentionned references: informed criticism is welcomed, empty criticism is not.

3.1 Rationale

This is just common sense:

1. It should not take a whole day to run. When it comes to comparative benchmarking (various runs), nobody wants to spend days trying to figure out the fastest setup for a given system. Ideally, the entire benchmark set should take about 15 minutes to complete on an average machine.
2. All source code for the software used must be freely available on the Net, for obvious reasons.
3. Benchmarks should provide simple figures reflecting the measured performance.
4. There should be a mix of synthetic benchmarks and application benchmarks (with separate results, of course).
5. Each **synthetic** benchmarks should exercise a particular subsystem to its maximum capacity.
6. Results of **synthetic** benchmarks should **not** be averaged into a single figure of merit (that defeats the whole idea behind synthetic benchmarks, with considerable loss of information).
7. Applications benchmarks should consist of commonly executed tasks on Linux systems.

3.2 Benchmark selection

I have selected five different benchmark suites, trying as much as possible to avoid overlap in the tests:

1. Kernel 2.0.0 (default configuration) compilation using gcc.
2. Whetstone version 10/03/97 (latest version by Roy Longbottom).
3. xbench-0.2 (with fast execution parameters).
4. UnixBench benchmarks version 4.01 (partial results).
5. BYTE Magazine's BYTEmark benchmarks beta release 2 (partial results).

For tests 4 and 5, "(partial results)" means that not all results produced by these benchmarks are considered.

3.3 Test duration

1. Kernel 2.0.0 compilation: 5 - 30 minutes, depending on the **real** performance of your system.
2. Whetstone: 100 seconds.
3. Xbench-0.2: < 1 hour.
4. UnixBench benchmarks version 4.01: approx. 15 minutes.

5. BYTE Magazine's BYTEmark benchmarks: approx. 10 minutes.

TOC

3.4 Comments

Kernel 2.0.0 compilation:

- **What:** it is the only application benchmark in the LBT.
- The code is widely available (i.e. I finally found some use for my old Linux CD-ROMs).
- Most linuxers recompile the kernel quite often, so it is a significant measure of overall performance.
- The kernel is large and gcc uses a large chunk of memory: attenuates L2 cache size bias with small tests.
- It does frequent I/O to disk.
- Test procedure: get a pristine 2.0.0 source, compile with default options (make config, press Enter repeatedly). The reported time should be the time spent on compilation i.e. after you type make zImage, **not** including make dep, make clean. Note that the default target architecture for the kernel is the i386, so if compiled on another architecture, gcc too should be set to cross-compile, with i386 as the target architecture.
- **Results:** compilation time in minutes and seconds (please don't report fractions of seconds).

Whetstone:

- **What:** measures pure floating point performance with a short, tight loop. The source (in C) is quite readable and it is very easy to see which floating-point operations are involved.
- Shortest test in the LBT :-).
- It's an "Old Classic" test: comparable figures are available, its flaws and shortcomings are well known.
- Test procedure: the newest C source should be obtained from Aburto's site. Compile and run in double precision mode. Specify gcc and -O2 as precompiler and precompiler options, and define POSIX 1 to specify machine type.
- **Results:** a floating-point performance figure in MWIPS.

Xbench-0.2:

- **What:** measures X server performance.
- The xStones measure provided by xbench is a weighted average of several tests indexed to an old Sun station with a single-bit-depth display. Hmmm... it is questionable as a test of modern X servers, but it's still the best tool I have found.
- Test procedure: compile with -O2. We specify a few options for a shorter run: `./xbench -timegoal 3 > results/name_of_your_linux_box.out`. To get the xStones rating, we must run an awk script; the simplest way is to type `make summary.ms`. Check the summary.ms file: the xStone rating for your system is in the last column of the line with your machine name specified during the test.
- **Results:** an X performance figure in xStones.
- Note: this test, as it stands, is outdated. It should be re-coded.

UnixBench version 4.01:

- **What:** measures overall Unix performance. This test will exercice the file I/O and kernel multitasking performance.

- I have discarded all arithmetic test results, keeping only the system-related test results.
- Test procedure: make with -O2. Execute with `./Run -1` (run each test once). You will find the results in the ./results/report file. Calculate the geometric mean of the EXECL THROUGHPUT, FILECOPY 1, 2, 3, PIPE THROUGHPUT, PIPE-BASED CONTEXT SWITCHING, PROCESS CREATION, SHELL SCRIPTS and SYSTEM CALL OVERHEAD indexes.
- **Results:** a system index.

BYTE Magazine's BYTEmark benchmarks:

- **What:** provides a good measure of CPU performance. Here is an excerpt from the documentation: *"These benchmarks are meant to expose the theoretical upper limit of the CPU, FPU, and memory architecture of a system. They cannot measure video, disk, or network throughput (those are the domains of a different set of benchmarks). You should, therefore, use the results of these tests as part, not all, of any evaluation of a system."*
- I have discarded the FPU test results since the Whetstone test is just as representative of FPU performance.
- I have split the integer tests in two groups: those more representative of memory-cache-CPU performance and the CPU integer tests.
- Test procedure: make with -O2. Run the test with `./nbench > myresults.dat` or similar. Then, from myresults.dat, calculate geometric mean of STRING SORT, ASSIGNMENT and BITFIELD test indexes; this is the **memory index**; calculate the geometric mean of NUMERIC SORT, IDEA, HUFFMAN and FP EMULATION test indexes; this is the **integer index**.
- **Results:** a memory index and an integer index calculated as explained above.

3.5 Possible improvements

The ideal benchmark suite would run in a few minutes, with synthetic benchmarks testing every subsystem separately and applications benchmarks providing results for different applications. It would also automatically generate a complete report and eventually email the report to a central database on the Web.

We are not really interested in portability here, but it should at least run on all recent (> 2.0.0) versions and flavours (i386, Alpha, Sparc...) of Linux.

If anybody has any idea about benchmarking network performance in a simple, easy and reliable way, with a short (less than 30 minutes to setup and run) test, please contact me.

3.6 LBT Report Form

Besides the tests, the benchmarking procedure would not be complete without a form describing the setup, so here it is (following the guidelines from comp.benchmarks.faq):

```
LINUX BENCHMARKING TOOLKIT REPORT FORM
```

```
CPU
==
Vendor:
```

TOC

```
Model:
Core clock:
Motherboard vendor:
Mbd. model:
Mbd. chipset:
Bus type:
Bus clock:
Cache total:
Cache type/speed:
SMP (number of processors):
```

```
RAM
====
Total:
Type:
Speed:
```

```
Disk
====
Vendor:
Model:
Size:
Interface:
Driver/Settings:
```

```
Video board
===========
Vendor:
Model:
Bus:
Video RAM type:
Video RAM total:
X server vendor:
X server version:
X server chipset choice:
Resolution/vert. refresh rate:
Color depth:
```

```
Kernel
=====
Version:
Swap size:
```

TOC

```
gcc
===
Version:
Options:
libc version:
```

```
Test notes
==========
```

```
RESULTS
========
Linux kernel 2.0.0 Compilation Time: (minutes and seconds)
Whetstones: results are in MWIPS.
Xbench: results are in xstones.
Unixbench Benchmarks 4.01 system INDEX:
BYTEmark integer INDEX:
BYTEmark memory INDEX:
```

```
Comments*
=========
* This field is included for possible interpretations of the results, and as
such, it is optional. It could be the most significant part of your report,
though, specially if you are doing comparative benchmarking.
```

3.7 Network performance tests

Testing network performance is a challenging task since it involves at least two machines, a server and a client machine, hence twice the time to setup and many more variables to control, etc... On an ethernet network, I guess your best bet would be the ttcp package. (to be expanded)

3.8 SMP tests

SMP tests are another challenge, and any benchmark specifically designed for SMP testing will have a hard time proving itself valid in real-life settings, since algorithms that can take advantage of SMP are hard to come by. It seems later versions of the Linux kernel (> 2.1.30 or around that) will do "fine-grained" multiprocessing, but I have no more information than that for the moment.

According to David Niemi, *" ... shell8* [part of the Unixbench 4.01 benchmaks]*does a good job at comparing similar hardware/OS in SMP and UP modes."*

---

4. Example run and results

The LBT was run on my home machine, a Pentium-class Linux box that I put together myself and that I used to write this HOWTO. Here is the LBT Report Form for this system:

```
LINUX BENCHMARKING TOOLKIT REPORT FORM

CPU

==

Vendor: Cyrix/IBM

Model: 6x86L P166+

Core clock: 133 MHz

Motherboard vendor: Elite Computer Systems (ECS)

Mbd. model: P5VX-Be

Mbd. chipset: Intel VX

Bus type: PCI

Bus clock: 33 MHz

Cache total: 256 KB

Cache type/speed: Pipeline burst 6 ns

SMP (number of processors): 1

RAM

====
```

TOC

```
Total: 32 MB

Type: EDO SIMMs

Speed: 60 ns

Disk

====

Vendor: IBM

Model: IBM-DAQA-33240

Size: 3.2 GB

Interface: EIDE

Driver/Settings: Bus Master DMA mode 2

Video board

===========

Vendor: Generic S3

Model: Trio64-V2

Bus: PCI

Video RAM type: EDO DRAM

Video RAM total: 2 MB

X server vendor: XFree86

X server version: 3.3

X server chipset choice: S3 accelerated

Resolution/vert. refresh rate: 1152x864 @ 70 Hz
```

Color depth: 16 bits

Kernel

=====

Version: 2.0.29

Swap size: 64 MB

gcc

===

Version: 2.7.2.1

Options: -O2

libc version: 5.4.23

Test notes

==========

Very light load. The above tests were run with some of the special Cyrix/IBM 6x86 features enabled with the setx86 program: fast ADS, fast IORT, Enable DTE, fast LOOP, fast Lin. VidMem.

RESULTS

========

Linux kernel 2.0.0 Compilation Time: 7m12s

Whetstones: 38.169 MWIPS.

Xbench: 97243 xStones.

BYTE Unix Benchmarks 4.01 system INDEX: 58.43

TOC

BYTEmark integer INDEX: 1.50

BYTEmark memory INDEX: 2.50

Comments

=========

This is a very stable system with homogeneous performance, ideal for home use and/or Linux development. I will report results with a 6x86MX processor as soon as I can get my hands on one!

---

5. Pitfalls and caveats of benchmarking

After putting together this HOWTO I began to understand why the words "pitfalls" and "caveats" are so often associated with benchmarking...

5.1 Comparing apples and oranges

Or should I say Apples and PCs ? This is so obvious and such an old dispute that I won't go into any details. I doubt the time it takes to load Word on a Mac compared to an average Pentium is a real measure of anything. Likewise booting Linux and Windows NT, etc... Try as much as possible to compare identical machines with a single modification.

5.2 Incomplete information

A single example will illustrate this very common mistake. One often reads in comp.os.linux.hardware the following or similar statement: "I just plugged in processor XYZ running at nnn MHz and now compiling the linux kernel only takes i minutes" (adjust XYZ, nnn and i as required). This is irritating, because no other information is given, i.e. we don't even know the amount of RAM, size of swap, other tasks running simultaneously, kernel version, modules selected, hard disk type, gcc version, etc... I recommend you use the LBT Report Form, which at least provides a standard information framework.

5.3 Proprietary hardware/software

A well-known processor manufacturer once published results of benchmarks produced by a special, customized version of gcc. Ethical considerations apart, those results were meaningless, since 100% of the Linux community would go on using the standard version of gcc. The same goes for proprietary hardware. Benchmarking is much more useful when it deals with off-the-shelf hardware and free (in the GNU/GPL sense) software.

5.4 Relevance

We are talking Linux, right ? So we should forget about benchmarks produced on other operating systems (this is a special case of the "Comparing apples and oranges" pitfall above). Also, if one is going to benchmark Web server performance, **do not** quote FPU performance and other irrelevant information. In such cases, less is more. Also, you do **not** need to mention the age of your cat, your mood while benchmarking, etc..

6. FAQ

Q1.

Is there any single figure of merit for Linux systems ?

A:

No, thankfully nobody has yet come up with a Lhinuxstone (tm) measurement. And if there was one, it would not make much sense: Linux systems are used for many different tasks, from heavily loaded Web servers to graphics workstations for individual use. No single figure of merit can describe the performance of a Linux system under such different situations.

Q2.

Then, how about a dozen figures summarizing the performance of diverse Linux systems ?

A:

That would be the ideal situation. I would like to see that come true. Anybody volunteers for a **Linux Benchmarking Project** ? With a Web site and an on-line, complete, well-designed reports database ?

Q3.

... BogoMips ... ?

A:

BogoMips has nothing to do with the performance of your system. Check the BogoMips Mini-HOWTO.

Q4.

What is the "best" benchmark for Linux ?

A:

It all depends on which performance aspect of a Linux system one wants to measure. There are different benchmarks to measure the network (Ethernet sustained transfer rates), file server (NFS), disk I/O, FPU, integer, graphics, 3D, processor-memory bandwidth, CAD performance, transaction time, SQL performance, Web server performance, real-time performance, CD-ROM performance, Quake performance (!), etc ... AFAIK no bechmark suite exists for Linux that supports all these tests.

Q5.

What is the fastest processor under Linux ?

A:

Fastest at what task ? If one is heavily number-crunching oriented, a very high clock rate Alpha

TOC

(600 MHz and going) should be faster than anything else, since Alphas have been designed fo kind of performance. If, on the other hand, one wants to put together a very fast news server, i probable that the choice of a fast hard disk subsystem and lots of RAM will result in higher performance improvements than a change of processor, for the same amount of $.

Q6.

Let me rephrase the last question, then: is there a processor that is fastest for general purpose applications ?

A:

This is a tricky question but it takes a very simple answer: **NO**. One can always design a faster system even for general purpose applications, independent of the processor. Usually, all other things being equal, higher clock rates will result in higher performance systems (and more headaches too). Taking out an old 100 MHz Pentium from an (usually not) upgradable motherboard, and plugging in the 200 MHz version, one should feel the extra "hummph". Of course, with only 16 MBytes of RAM, the same investment would have been more wisely spent on extra SIMMs...

Q7.

So clock rates influence the performance of a system ?

A:

For most tasks except for NOP empty loops (BTW these get removed by modern optimizing compilers), an increase in clock rate will not give you a linear increase in performance. Very small processor intensive programs that will fit entirely in the primary cache inside the processor (the L1 cache, usually 8 or 16 K) will have a performance increase equivalent to the clock rate increase, but most "true" programs are much larger than that, have loops that do not fit in the L1 cache, share the L2 (external) cache with other processes, depend on external components and will give much smaller performance increases. This is because the L1 cache runs at the same clock rate as the processor, whereas most L2 caches and all other subsystems (DRAM, for example) will run asynchronously at lower clock rates.

Q8.

OK, then, one last question on that matter: which is the processor with the best price/performance ratio for general purpose Linux use ?

A:

Defining "general purpose Linux use" in not an easy thing ! For any particular application, there is always a processor with THE BEST price/performance ratio at any given time, but it changes rather frequently as manufacturers release new processors, so answering Processor XYZ running at n MHz would be a snapshot answer. However, the price of the processor is insignificant when compared to the price of the whole system one will be putting together. So, really, the question should be how can one maximize the price/performance ratio for a given system ? And the answer to that question depends heavily on the minimum performance requirements and/or maximum cost established for the configuration being considered. Sometimes, off-the-shelf hardware will not meet minimum performance requirements and expensive RISC systems will be the only

alternative. For home use, I recommend a balanced, homogeneous system for overall perform (now go figure what I mean by balanced and homogeneous :-); the choice of a processor is an important decision , but no more than choosing hard disk type and capacity, amount of RAM, video card, etc...

Q9.

What is a "significant" increase in performance ?

A:

I would say that anything under 1% is not significant (could be described as "marginal"). We, humans, will hardly perceive the difference between two systems with a 5 % difference in response time. Of course some hard-core benchmarkers are not humans and will tell you that, when comparing systems with 65.9 and 66.5 performance indexes, the later is "definitely faster".

Q10.

How do I obtain "significant" increases in performance at the lowest cost ?

A:

Since most source code is available for Linux, careful examination and algorithmic redesign of key subroutines could yield order-of-magnitude increases in performance in some cases. If one is dealing with a commercial project and does not wish to delve deeply in C source code a **Linux consultant should be called in**. See the Consultants-HOWTO.

---

7. Copyright, acknowledgments and miscellaneous

7.1 How this document was produced

The first step was reading section 4 "Writing and submitting a HOWTO" of the HOWTO Index by Tim Bynum.

I knew absolutely nothing about SGML or LaTeX, but was tempted to use an automated documentation generation package after reading the various comments about SGML-Tools. However, inserting tags manually in a document reminds me of the days I hand-assembled a 512 byte monitor program for a now defunct 8-bit microprocessor, so I got hold of the LyX sources, compiled it, and used its LinuxDoc mode. Highly recommended combination: **LyX and SGML-Tools**.

7.2 Copyright

The Linux Benchmarking HOWTO is copyright (C) 1997 by André D. Balsa. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu via email.

7.3 New versions of this document

New versions of the Linux Benchmarking-HOWTO will be placed on sunsite.unc.edu and mirror sites. There are other formats, such as a Postscript and dvi version in the other-formats directory. The Linux Benchmarking-HOWTO is also available for WWW clients such as Grail, a Web browser written in Python. It will also be posted regularly to comp.os.linux.answers.

7.4 Feedback

Suggestions, corrections, additions wanted. Contributors wanted and acknowledged. Flames not wanted.

I can always be reached at andrewbalsa@usa.net.

7.5 Acknowledgments

David Niemi, the author of the Unixbench suite, has proved to be an endless source of information and (valid) criticism.

I also want to thank Greg Hankins one of the main contributors to the SGML-tools package, Linus Torvalds and the entire Linux community. This HOWTO is my way of giving back.

7.6 Disclaimer

Your mileage may, and will, vary. Be aware that benchmarking is a touchy subject and a great time-and-energy consuming activity.

7.7 Trademarks

Pentium and Windows NT are trademarks of Intel and Microsoft Corporations respectively.

BYTE and BYTEmark are trademarks of McGraw-Hill, Inc.

Cyrix and 6x86 are trademarks of Cyrix Corporation.

Linux is not a trademark, hopefully never will be.

Beowulf HOWTO

Jacek Radajewski and Douglas Eadline

v1.1.1, 22 November 1998

This document introduces the Beowulf Supercomputer architecture and provides background information on parallel programming, including links to other more specific documents, and web pages.

1. Preamble

- 1.1 Disclaimer
- 1.2 Copyright
- 1.3 About this HOWTO
- 1.4 About the authors
- 1.5 Acknowledgements

2. Introduction

- 2.1 Who should read this HOWTO ?
- 2.2 What is a Beowulf ?
- 2.3 Classification

3. Architecture Overview

- 3.1 What does it look like ?
- 3.2 How to utilise the other nodes ?
- 3.3 How does Beowulf differ from a COW ?

4. System Design

- 4.1 A brief background on parallel computing.
- 4.2 The methods of parallel computing

- 4.3 Architectures for parallel computing
- 4.4 Suitability
- 4.5 Writing and porting parallel software

5. Beowulf Resources

- 5.1 Starting Points
- 5.2 Documentation
- 5.3 Papers
- 5.4 Software
- 5.5 Beowulf Machines
- 5.6 Other Interesting Sites
- 5.7 History

6. Source code

- 6.1 sum.c
- 6.2 sigmasqrt.c
- 6.3 prun.sh

---

1. Preamble

1.1 Disclaimer

We will not accept any responsibility for any incorrect information within this document, nor for any damage it might cause when applied.

1.2 Copyright

Copyright © 1997 - 1998 Jacek Radajewski and Douglas Eadline. Permission to distribute and modify this document is granted under the GNU General Public Licence.

1.3 About this HOWTO

Jacek Radajewski started work on this document in November 1997 and was soon joined by Douglas Eadline. Over a few months the Beowulf HOWTO grew into a large document, and in August 1998 it was split into three documents: Beowulf HOWTO, Beowulf Architecture Design HOWTO, and the Beowulf Installation and Administration HOWTO. Version 1.0.0 of the Beowulf HOWTO was released to the Linux Documentation Project on 11 November 1998. We hope that this is only the beginning of what will become a complete Beowulf Documentation Project.

1.4 About the authors

- Jacek Radajewski works as a Network Manager, and is studying for an honors degree in computer science at the University of Southern Queensland, Australia. Jacek's first contact with Linux was in 1995 and it was love at first sight. Jacek built his first Beowulf cluster in May 1997 and has been playing with the technology ever since, always trying to find new and better ways of setting things up. You can contact Jacek by sending e-mail to jacek@usq.edu.au
- Douglas Eadline, Ph.D. is President and Principal Scientist at Paralogic, Inc., Bethlehem, PA, USA. Trained as Physical/Analytical Chemist, he has been involved with computers since 1978 when he built his first single board computer for use with chemical instrumentation. Dr. Eadline's interests now include Linux, Beowulf clusters, and parallel algorithms. Dr. Eadline can be contacted by sending email to deadline@plogic.com

1.5 Acknowledgements

The writing of the Beowulf HOWTO was a long proces and is finally complete, thanks to many individuals. I would like to thank the following people for their help and contribution to this HOWTO.

- Becky for her love, support, and understanding.
- Tom Sterling, Don Becker, and other people at NASA who started the Beowulf project.
- Thanh Tran-Cong and the Faculty of Engineering and Surveying for making the *topcat* Beowulf machine available for experiments.
- My supervisor Christopher Vance for many great ideas.
- My friend Russell Waldron for great programming ideas, his general interest in the project, and support.
- My friend David Smith for proof reading this document.
- Many other people on the Beowulf mailing list who provided me with feedback and ideas.
- All the people who are responsible for the Linux operating system and all the other free software packages used on *topcat* and other Beowulf machines.

2. Introduction

As the performance of commodity computer and network hardware increase, and their prices decrease, it becomes more and more practical to build parallel computational systems from off-the-shelf components, rather than buying CPU time on very expensive Supercomputers. In fact, the price per performance ratio of a Beowulf type machine is between three to ten times better than that for traditional supercomputers. Beowulf architecture scales well, it is easy to construct and you only pay for the hardware as most of the software is free.

2.1 Who should read this HOWTO ?

This HOWTO is designed for a person with at least some exposure to the Linux operating system. Knowledge of Beowulf technology or understanding of more complex operating system and networking concepts is not essential, but some exposure to parallel computing would be advantageous (after all you must have some reason to read this document). This HOWTO will not answer all possible questions you might have about Beowulf, but hopefully will give you ideas and guide you in the right direction. The purpose of this HOWTO is to provide background information, links and references to more advanced documents.

2.2 What is a Beowulf ?

Famed was this Beowulf: far flew the boast of him, son of Scyld, in the Scandian lands. So becomes it a youth to quit him well with his father's friends, by fee and gift, that to aid him, aged, in after days, come warriors willing, should war draw nigh, liegemen loyal: by lauded deeds shall an earl have honor in every clan. Beowulf is the earliest surviving epic poem written in English. It is a story about a hero of great strength and courage who defeted a monster called Grendel. See History to find out more about the Beowulf hero.

There are probably as many Beowulf definitions as there are people who build or use Beowulf Supercomputer facilities. Some claim that one can call their system Beowulf only if it is built in the same way as the NASA's original machine. Others go to the other extreme and call Beowulf any system of workstations running parallel code. My definition of Beowulf fits somewhere between the two views described above, and is based on many postings to the Beowulf mailing list:

Beowulf is a multi computer architecture which can be used for parallel computations. It is a system which usually consists of one server node, and one or more client nodes connected together via Ethernet or some other network. It is a system built using commodity hardware components, like any PC capable of running Linux, standard Ethernet adapters, and switches. It does not contain any custom hardware components and is trivially reproducible. Beowulf also uses commodity software like the Linux operating system, Parallel Virtual Machine (PVM) and Message Passing Interface (MPI). The server node controls the whole cluster and serves files to the client nodes. It is also the cluster's console and

TOC

gateway to the outside world. Large Beowulf machines might have more than one server node, and possibly other nodes dedicated to particular tasks, for example consoles or monitoring stations. In most cases client nodes in a Beowulf system are dumb, the dumber the better. Nodes are configured and controlled by the server node, and do only what they are told to do. In a disk-less client configuration, client nodes don't even know their IP address or name until the server tells them what it is. One of the main differences between Beowulf and a Cluster of Workstations (COW) is the fact that Beowulf behaves more like a single machine rather than many workstations. In most cases client nodes do not have keyboards or monitors, and are accessed only via remote login or possibly serial terminal. Beowulf nodes can be thought of as a CPU + memory package which can be plugged in to the cluster, just like a CPU or memory module can be plugged into a motherboard.

Beowulf is not a special software package, new network topology or the latest kernel hack. Beowulf is a technology of clustering Linux computers to form a parallel, virtual supercomputer. Although there are many software packages such as kernel modifications, PVM and MPI libraries, and configuration tools which make the Beowulf architecture faster, easier to configure, and much more usable, one can build a Beowulf class machine using standard Linux distribution without any additional software. If you have two networked Linux computers which share at least the `/home` file system via NFS, and trust each other to execute remote shells (rsh), then it could be argued that you have a simple, two node Beowulf machine.

2.3 Classification

Beowulf systems have been constructed from a variety of parts. For the sake of performance some non-commodity components (i.e. produced by a single manufacturer) have been employed. In order to account for the different types of systems and to make discussions about machines a bit easier, we propose the following simple classification scheme:

CLASS I BEOWULF:

This class of machines built entirely from commodity "off-the-shelf" parts. We shall use the "Computer Shopper" certification test to define commodity "off-the-shelf" parts. (Computer Shopper is a 1 inch thick monthly magazine/catalog of PC systems and components.) The test is as follows:

A CLASS I Beowulf is a machine that can be assembled from parts found in at least 3 nationally/globally circulated advertising catalogs.

The advantages of a CLASS I system are:

- hardware is available form multiple sources (low prices, easy maintenance)
- no reliance on a single hardware vendor
- driver support from Linux commodity
- usually based on standards (SCSI, Ethernet, etc.)

The disadvantages of a CLASS I system are:

- best performance may require CLASS II hardware

CLASS II BEOWULF

TOC

A CLASS II Beowulf is simply any machine that does not pass the Computer Shopper certification t
This is not a bad thing. Indeed, it is merely a classification of the machine.

The advantages of a CLASS II system are:

- Performance can be quite good!

The disadvantages of a CLASS II system are:

- driver support may vary
- reliance on single hardware vendor
- may be more expensive than CLASS I systems.

One CLASS is not necessarily better than the other. It all depends on your needs and budget. This classification system is only intended to make discussions about Beowulf systems a bit more succinct. The "System Design" section may help determine what kind of system is best suited for your needs.

---

3. Architecture Overview

3.1 What does it look like ?

I think that the best way of describing the Beowulf supercomputer architecture is to use an example which is very similar to the actual Beowulf, but familiar to most system administrators. The example that is closest to a Beowulf machine is a Unix computer laboratory with a server and a number of clients. To be more specific I'll use the DEC Alpha undergraduate computer laboratory at the Faculty of Sciences, USQ as the example. The server computer is called *beldin* and the client machines are called *scilab01*, *scilab02*, *scilab03*, up to *scilab20*. All clients have a local copy of the Digital Unix 4.0 operating system installed, but get the user file space (`/home`) and `/usr/local` from the server via NFS (Network File System). Each client has an entry for the server and all the other clients in its `/etc/hosts.equiv` file, so all clients can execute a remote shell (rsh) to all others. The server machine is a NIS server for the whole laboratory, so account information is the same across all the machines. A person can sit at the *scilab02* console, login, and have the same environment as if he logged onto the server or *scilab15*. The reason all the clients have the same look and feel is that the operating system is installed and configured in the same way on all machines, and both the user's `/home` and `/usr/local` areas are physically on the server and accessed by the clients via NFS. For more information on NIS and NFS please read the NIS and NFS HOWTOs.

3.2 How to utilise the other nodes ?

Now that we have some idea about the system architecture, let us take a look at how we can utilise the available CPU cycles of the machines in the computer laboratory. Any person can logon to any of the machines, and run a program in their home directory, but they can also spawn the same job on a different machine simply by executing remote shell. For example, assume that we want to calculate the sum of the square roots of all integers between 1 and 10 inclusive. We write a simple program called `sigmasqrt` (please see source code) which does exactly that. To calculate the sum of the square roots of numbers from 1 to 10 we execute :

```
[jacek@beldin sigmasqrt]$ time ./sigmasqrt 1 10
22.468278

real    0m0.029s
user    0m0.001s
sys     0m0.024s
```

The `time` command allows us to check the wall-clock (the elapsed time) of running this job. As we can see, this example took only a small fraction of a second (0.029 sec) to execute, but what if I want to add the square root of integers from 1 to 1 000 000 000 ? Let us try this, and again calculate the wall-clock time.

```
[jacek@beldin sigmasqrt]$ time ./sigmasqrt 1 1000000000
21081851083600.559000

real    16m45.937s
user    16m43.527s
sys     0m0.108s
```

This time, the execution time of the program is considerably longer. The obvious question to ask is what can we do to speed up the execution time of the job? How can we change the way the job is running to minimize the wall-clock time of running this job? The obvious answer is to split the job into a number of sub-jobs and to run these sub-jobs in parallel on all computers. We could split one big addition task into 20 parts, calculating one range of square roots and adding them on each node. When all nodes finish the calculation and return their results, the 20 numbers could be added together to obtain the final solution. Before we run this job we will make a named pipe which will be used by all processes to write their results.

```
[jacek@beldin sigmasqrt]$ mkfifo output
[jacek@beldin sigmasqrt]$ ./prun.sh & time cat output | ./sum
[1] 5085
21081851083600.941000
[1]+  Done                    ./prun.sh

real    0m58.539s
user    0m0.061s
sys     0m0.206s
```

This time we get about 58.5 seconds. This is the time from starting the job until all the nodes have finished their computations and written their results into the pipe. The time does not include the final addition of the twenty numbers, but this time is a very small fraction of a second and can be ignored. We can see that there is a significant improvement in running this job in parallel. In fact the parallel job ran about 17 times faster, which is very reasonable for a 20 fold increase in the number of CPUs. The purpose of the above example is to illustrate the simplest method of parallelising concurrent code. In practice such simple examples are rare and different techniques (PVM and PMI APIs) are used to achieve the parallelism.

3.3 How does Beowulf differ from a COW ?

The computer laboratory described above is a perfect example of a Cluster of Workstations (COW). So what is so special about Beowulf, and how is it different from a COW? The truth is that there is not much difference, but Beowulf does have few unique characteristics. First of all, in most cases client nodes in a Beowulf cluster do not have keyboards, mice, video cards nor monitors. All access to the client nodes is done via remote connections from the server node, dedicated console node, or a serial console. Because there is no need for client nodes to access machines outside the cluster, nor for machines outside the cluster to access client nodes directly, it is a common practice for the client nodes to use private IP

addresses like the 10.0.0.0/8 or 192.168.0.0/16 address ranges (RFC 1918 http://www.alternic.net/rfcs/1900/rfc1918.txt.html). Usually the only machine that is also connected to the outside world using a second network card is the server node. The most common ways of using the system is to access the server's console directly, or either telnet or remote login to the server node from personal workstation. Once on the server node, users can edit and compile their code, and also spawn jobs on all nodes in the cluster. In most cases COWs are used for parallel computations at night, and over weekends when people do not actually use the workstations for every day work, thus utilising idle CPU cycles. Beowulf on the other hand is a machine usually dedicated to parallel computing, and optimised for this purpose. Beowulf also gives better price/performance ratio as it is built from off-the-shelf components and runs mainly free software. Beowulf has also more single system image features which help the users to see the Beowulf cluster as a single computing workstation.

---

4. System Design

Before you purchase any hardware, it may be a good idea to consider the design of your system. There are basically two hardware issues involved with design of a Beowulf system: the type of nodes or computers you are going to use; and way you connect the computer nodes. There is one software issue that may effect your hardware decisions; the communication library or API. A more detailed discussion of hardware and communication software is provided later in this document.

While the number of choices is not large, there are some important design decisions that must be made when constructing a Beowulf systems. Because the science (or art) of "parallel computing" has many different interpretations, an introduction is provided below. If you do not like to read background material, you may skip this section, but it is advised that you read section Suitability before you make you final hardware decisions.

4.1 A brief background on parallel computing.

This section provides background on parallel computing concepts. It is NOT an exhaustive or complete description of parallel computing science and technology. It is a brief description of the issues that may be important to a Beowulf designer and user.

As you design and build your Beowulf, many of these issues described below will become important in your decision process. Due to its component nature, a Beowulf Supercomputer requires that we consider many factors carefully because they are now under our control. In general, it is not all that difficult to understand the issues involved with parallel computing. Indeed, once the issues are understood, your expectations will be more realistic and success will be more likely. Unlike the "sequential world" where processor speed is considered the single most important factor, processor speed in the "parallel world" is just one of several factors that will determine overall system performance and efficiency.

4.2 The methods of parallel computing

Parallel computing can take many forms. From a user's perspective, it is important to consider the advantages and disadvantages of each methodology. The following section attempts to provide some perspective on the methods of parallel computing and indicate where the Beowulf machine falls on this continuum.

Why more than one CPU?

Answering this question is important. Using 8 CPUs to run your word processor sounds a little like "over-kill" -- and it is. What about a web server, a database, a rendering program, or a project scheduler? Maybe extra CPUs would help. What about a complex simulation, a fluid dynamics code, or a data mining application. Extra CPUs definitely help in these situations. Indeed, multiple CPUs are being used

to solve more and more problems.

The next question usually is: "Why do I need two or four CPUs, I will just wait for the 986 turbo-hyper chip." There are several reasons:

1. Due to the use of multi-tasking Operating Systems, it is possible to do several things at once. This is a natural "parallelism" that is easily exploited by more than one low cost CPU.
2. Processor speeds have been doubling every 18 months, but what about RAM speeds or hard disk speeds? Unfortunately, these speeds are not increasing as fast as the CPU speeds. Keep in mind most applications require "out of cache memory access" and hard disk access. Doing things in parallel is one way to get around some of these limitations.
3. Predictions indicate that processor speeds will not continue to double every 18 months after the year 2005. There are some very serious obstacles to overcome in order to maintain this trend.
4. Depending on the application, parallel computing can speed things up by any where from 2 to 500 times faster (in some cases even faster). Such performance is not available using a single processor. Even supercomputers that at one time used very fast custom processors are now built from multiple "commodity- off-the-shelf" CPUs.

If you need speed - either due to a compute bound problem and/or an I/O bound problem, parallel is worth considering. Because parallel computing is implemented in a variety of ways, solving your problem in parallel will require some very important decisions to be made. These decisions may dramatically effect portability, performance, and cost of your application.

Before we get technical, let's look take a look at a real "parallel computing problem" using an example with which we are familiar - waiting in long lines at a store.

The Parallel Computing Store

Consider a big store with 8 cash registers grouped together in the front of the store. Assume each cash register/cashier is a CPU and each customer is a computer program. The size of the computer program (amount of work) is the size of each customer's order. The following analogies can be used to illustrate parallel computing concepts.

Single-tasking Operating System

One cash register open (is in use) and must process each customer one at a time.

Computer Example: MS DOS

Multi-tasking Operating System:

One cash register open, but now we process only a part of each order at a time, move to the next person and process some of their order. Everyone "seems" to be moving through the line together, but if no one else is in the line, you will get through the line faster.

Computer Example: UNIX, NT using a single CPU

TOC

Multitasking Operating Systems with Multiple CPUs:

Now we open several cash registers in the store. Each order can be processed by a separate cash register and the line can move much faster. This is called SMP - Symmetric Multi-processing. Although there are extra cash registers open, you will still never get through the line any faster than just you and a single cash register.

Computer Example: UNIX and NT with multiple CPUs

Threads on a Multitasking Operating Systems extra CPUs

If you "break-up" the items in your order, you might be able to move through the line faster by using several cash registers at one time. First, we must assume you have a large amount of goods, because the time you invest "breaking up your order" must be regained by using multiple cash registers. In theory, you should be able to move through the line "n" times faster than before*; where "n" is the number of cash registers. When the cashiers need to get sub- totals, they can exchange information quickly by looking and talking to all the other "local" cash registers. They can even snoop around the other cash registers to find information they need to work faster. There is a limit, however, as to how many cash registers the store can effectively locate in any one place.

Amdals law will also limit the application speed-up to the slowest sequential portion of the program.

Computer Example: UNIX or NT with extra CPU on the same motherboard running multi-threaded programs.

Sending Messages on Multitasking Operating Systems with extra CPUs:

In order to improve performance, the store adds 8 cash registers at the back of the store. Because the new cash registers are far away from the front cash registers, the cashiers must call on the phone to send their sub-totals to the front of the store. This distance adds extra overhead (time) to communication between cashiers, but if communication is minimized, it is not a problem. If you have a really big order, one that requires all the cash registers, then as before your speed can be improved by using all cash registers at the same time, the extra overhead must be considered. In some cases, the store may have single cash registers (or islands of cash registers) located all over the store - each cash register (or island) must communicate by phone. Since all the cashiers working the cash registers can talk to each other by phone, it does not matter too much where they are.

Computer Example: One or several copies of UNIX or NT with extra CPUs on the same or different motherboard communicating through messages.

The above scenarios, although not exact, are a good representation of constraints placed on parallel systems. Unlike a single CPU (or cash register) communication is an issue.

4.3 Architectures for parallel computing

TOC

The common methods and architectures of parallel computing are presented below. While this description is by no means exhaustive, it is enough to understand the basic issues involved with Beowulf design.

Hardware Architectures

There are basically two ways parallel computer hardware is put together:

1. Local memory machines that communicate by messages (Beowulf Clusters)
2. Shared memory machines that communicate through memory (SMP machines)

A typical Beowulf is a collection of single CPU machines connected using fast Ethernet and is, therefore, a local memory machine. A 4 way SMP box is a shared memory machine and can be used for parallel computing - parallel applications communicate using shared memory. Just as in the computer store analogy, local memory machines (individual cash registers) can be scaled up to large numbers of CPUs, while the number of CPUs shared memory machines (the number of cash registers you can place in one spot) can have is limited due to memory contention.

It is possible, however, to connect many shared memory machines to create a "hybrid" shared memory machine. These hybrid machines "look" like a single large SMP machine to the user and are often called NUMA (non uniform memory access) machines because the global memory seen by the programmer and shared by all the CPUs can have different latencies. At some level, however, a NUMA machine must "pass messages" between local shared memory pools.

It is also possible to connect SMP machines as local memory compute nodes. Typical CLASS I motherboards have either 2 or 4 CPUs and are often used as a means to reduce the overall system cost. The Linux internal scheduler determines how these CPUs get shared. The user cannot (at this point) assign a specific task to a specific SMP processor. The user can however, start two independent processes or a threaded processes and expect to see a performance increase over a single CPU system.

Software API Architectures

There basically two ways to "express" concurrency in a program:

1. Using Messages sent between processors
2. Using operating system Threads

Other methods do exist, but these are the two most widely used. It is important to remember that the expression of concurrency is not necessary controlled by the underlying hardware. Both Messages and Threads can be implemented on SMP, NUMA-SMP, and clusters - although as explained below efficiently and portability are important issues.

TOC

Messages

Historically, messages passing technology reflected the design of early local memory parallel computers. Messages require copying data while Threads use data in place. The latency and speed at which messages can be copied are the limiting factor with message passing models. A Message is quite simple: some data and a destination processor. Common message passing APIs are PVM or MPI. Message passing can be efficiently implemented using Threads and Messages work well both on SMP machine and between clusters of machines. The advantage to using messages on an SMP machine, as opposed to Threads, is that if you decided to use clusters in the future it is easy to add machines or scale your application.

Threads

Operating system Threads were developed because shared memory SMP (symmetrical multiprocessing) designs allowed very fast shared memory communication and synchronization between concurrent parts of a program. Threads work well on SMP systems because communication is through shared memory. For this reason the user must isolate local data from global data, otherwise programs will not work properly. In contrast to messages, a large amount of copying can be eliminated with threads because the data is shared between processes (threads). Linux supports POSIX threads. The problem with threads is that it is difficult to extend them beyond one SMP machine and because data is shared between CPUs, cache coherence issues can contribute to overhead. Extending threads beyond the SMP boundary efficiently requires NUMA technology which is expensive and not natively supported by Linux. Implementing threads on top of messages has been done ((http://syntron.com/ptools/ptools_pg.htm)), but Threads are often inefficient when implemented using messages.

The following can be stated about performance:

	SMP machine performance	cluster of machines performance	scalability
messages	good	best	best
threads	best	poor*	poor*

* requires expensive NUMA technology.

Application Architecture

In order to run an application in parallel on multiple CPUs, it must be explicitly broken in to concurrent parts. A standard single CPU application will run no faster than a single CPU application on multiple processors. There are some tools and compilers that can break up programs, but parallelizing codes is not a "plug and play" operation. Depending on the application, parallelizing code can be easy, extremely difficult, or in some cases impossible due to algorithm dependencies.

Before the software issues can be addressed the concept of Suitability needs to be introduced.

4.4 Suitability

TOC

Most questions about parallel computing have the same answer:

"It all depends upon the application."

Before we jump into the issues, there is one very important distinction that needs to be made - the difference between CONCURRENT and PARALLEL. For the sake of this discussion we will define these two concepts as follows:

CONCURRENT parts of a program are those that can be computed independently.

PARALLEL parts of a program are those CONCURRENT parts that are executed on separate processing elements at the same time.

The distinction is very important, because CONCURRENCY is a property of the program and efficient PARALLELISM is a property of the machine. Ideally, PARALLEL execution should result in faster performance. The limiting factor in parallel performance is the communication speed and latency between compute nodes. (Latency also exists with threaded SMP applications due to cache coherency.) Many of the common parallel benchmarks are highly parallel and communication and latency are not the bottle neck. This type of problem can be called "obviously parallel". Other applications are not so simple and executing CONCURRENT parts of the program in PARALLEL may actually cause the program to run slower, thus offsetting any performance gains in other CONCURRENT parts of the program. In simple terms, the cost of communication time must pay for the savings in computation time, otherwise the PARALLEL execution of the CONCURRENT part is inefficient.

The task of the programmer is to determining what CONCURRENT parts of the program SHOULD be executed in PARALLEL and what parts SHOULD NOT. The answer to this will determine the EFFICIENCY of application. The following graph summarizes the situation for the programmer:

```
          | *
          | *
          | *
% of      | *
appli-    |  *
cations   |  *
          |  *
          |  *
          |     *
          |      *
          |       *
          |          ****
          |               ****
          |                    ********************
```

```
+-----------------------------------
 communication time/processing time
```

In a perfect parallel computer, the ratio of communication/processing would be equal and anything that is CONCURRENT could be implemented in PARALLEL. Unfortunately, Real parallel computers, including shared memory machines, are subject to the effects described in this graph. When designing a Beowulf, the user may want to keep this graph in mind because parallel efficiency depends upon ratio of communication time and processing time for A SPECIFIC PARALLEL COMPUTER. Applications may be portable between parallel computers, but there is no guarantee they will be efficient on a different platform.

IN GENERAL, THERE IS NO SUCH THING AS A PORTABLE AND EFFICIENT PARALLEL PROGRAM

There is yet another consequence to the above graph. Since efficiency depends upon the comm./process. ratio, just changing one component of the ratio does not necessary mean a specific application will perform faster. A change in processor speed, while keeping the communication speed that same may have non- intuitive effects on your program. For example, doubling or tripling the CPU speed, while keeping the communication speed the same, may now make some previously efficient PARALLEL portions of your program, more efficient if they were executed SEQUENTIALLY. That is, it may now be faster to run the previously PARALLEL parts as SEQUENTIAL. Furthermore, running inefficient parts in parallel will actually keep your application from reaching its maximum speed. Thus, by adding faster processor, you may actually slowed down your application (you are keeping the new CPU from running at its maximum speed for that application)

UPGRADING TO A FASTER CPU MAY ACTUALLY SLOW DOWN YOUR APPLICATION

So, in conclusion, to know whether or not you can use a parallel hardware environment, you need to have some insight into the suitability of a particular machine to your application. You need to look at a lot of issues including CPU speeds, compiler, message passing API, network, etc. Please note, just profiling an application, does not give the whole story. You may identify a computationally heavy portion of your program, but you do not know the communication cost for this portion. It may be that for a given system, the communication cost as do not make parallelizing this code efficient.

A final note about a common misconception. It is often stated that "a program is PARALLELIZED", but in reality only the CONCURRENT parts of the program have been located. For all the reasons given above, the program is not PARALLELIZED. Efficient PARALLELIZATION is a property of the machine.

4.5 Writing and porting parallel software

Once you decide that you need parallel computing and would like to design and build a Beowulf, a few moments considering your application with respect to the previous discussion may be a good idea.

In general there are two things you can do:

1. Go ahead and construct a CLASS I Beowulf and then "fit" your application to it. Or run existing parallel applications that you know work on your Beowulf (but beware of the portability and

efficiently issues mentioned above)

2. Look at the applications you need to run on your Beowulf and make some estimations as to the type of hardware and software you need.

In either case, at some point you will need to look at the efficiency issues. In general, there are three things you need to do:

1. Determine concurrent parts of your program
2. Estimate parallel efficiently
3. Describing the concurrent parts of your program

Let's look at these one at a time.

Determine concurrent parts of your program

This step is often considered "parallelizing your program". Parallelization decisions will be made in step 2. In this step, you need to determine data dependencies.

>From a practical standpoint, applications may exhibit two types of concurrency: compute (number crunching) and I/O (database). Although in many cases compute and I/O concurrency are orthogonal, there are application that require both. There are tools available that can perform concurrency analysis on existing applications. Most of these tools are designed for FORTRAN. There are two reasons FORTRAN is used: historically most number crunching applications were written in FORTRAN and it is easier to analyze. If no tools are available, then this step can be some what difficult for existing applications.

Estimate parallel efficiency

Without the help of tools, this step may require trial and error tests or just a plain old educated guess. If you have a specific application in mind, try to determine if it is CPU limited (compute bound) or hard disk limited (I/O bound). The requirements of your Beowulf may be quite different depending upon your needs. For example, a compute bound problem may need a few very fast CPUs and high speed low latency network, while an I/O bound problem may work better with more slower CPUs and fast Ethernet.

This recommendation often comes as a surprise to most people because, the standard assumption is that faster processor are always better. While this is true if your have an unlimited budget, real systems may have cost constraints that should be maximized. For I/O bound problems, there is a little known rule (called the Eadline-Dedkov Law) that is quite helpful:

For two given parallel computers with the same cumulative CPU performance index, the one which has slower processors (and a probably correspondingly slower interprocessor communication network) will have better performance for I/O-dominant applications.

While the proof of this rule is beyond the scope of this document, you find it interesting to download the paper *Performance Considerations for I/O-Dominant Applications on Parallel Computers* (Postscript format 109K) (ftp://www.plogic.com/pub/papers/exs-pap6.ps)

Once you have determined what type of concurrency you have in your application, you will need to estimate how efficient it will be in parallel. See Section Software for a description of Software tools.

In the absence of tools, you may try to guess your way through this step. If a compute bound loop measured in minutes and the data can be transferred in seconds, then it might be a good candidate for parallelization. But remember, if you take a 16 minute loop and break it into 32 parts, and your data transfers require several seconds per part, then things are going to get tight. You will reach a point of diminishing returns.

Describing the concurrent parts of your program

There are several ways to describe concurrent parts of your program:

1. Explicit parallel execution
2. Implicit parallel execution

The major difference between the two is that explicit parallelism is determined by the user where implicit parallelism is determined by the compiler.

Explicit Methods

These are basically method where the user must modify source code specifically for a parallel computer. The user must either add messages using PVM or MPI or add threads using POSIX threads. (Keep in mind however, threads can not move between SMP motherboards).

Explicit methods tend to be the most difficult to implement and debug. Users typically embed explicit function calls in standard FORTRAN 77 or C/C++ source code. The MPI library has added some functions to make some standard parallel methods easier to implement (i.e. scatter/gather functions). In addition, it is also possible to use standard libraries that have been written for parallel computers. Keep in mind, however, the portability vs. efficiently trade-off)

For historical reasons, most number crunching codes are written in FORTRAN. For this reasons, FORTRAN has the largest amount of support (tools, libraries, etc.) for parallel computing. Many programmers now use C or re- write existing FORTRAN applications in C with the notion the C will allow faster execution. While this may be true as C is the closest thing to a universal machine code, it has some major drawbacks. The use of pointers in C makes determining data dependencies extremely difficult. Automatic analysis of pointers is extremely difficult. If you have an existing FORTRAN program and think that you might want to parallelize it in the future - DO NOT CONVERT IT TO C!

Implicit Methods

Implicit methods are those where the user gives up some (or all) of the parallelization decisions to the compiler. Examples are FORTRAN 90, High Performance FORTRAN (HPF), Bulk Synchronous Parallel (BSP), and a whole collection of other methods that are under development.

Implicit methods require the user to provide some information about the concurrent nature of their application, but the compiler will then make many decisions about how to execute this concurrency in parallel. These methods provide some level of portability and efficiency, but there is still no "best way" to describe a concurrent problem for a parallel computer.

---

5. Beowulf Resources

5.1 Starting Points

- Beowulf mailing list. To subscribe send mail to beowulf-request@cesdis.gsfc.nasa.gov with the word *subscribe* in the message body.
- Beowulf Homepage http://www.beowulf.org
- Extreme Linux http://www.extremelinux.org
- Extreme Linux Software from Red Hat http://www.redhat.com/extreme

5.2 Documentation

- The latest version of the Beowulf HOWTO http://www.sci.usq.edu.au/staff/jacek/beowulf.
- Building a Beowulf System http://www.cacr.caltech.edu/beowulf/tutorial/building.html
- Jacek's Beowulf Links http://www.sci.usq.edu.au/staff/jacek/beowulf.
- Beowulf Installation and Administration HOWTO (DRAFT) http://www.sci.usq.edu.au/staff/jacek/beowulf.
- Linux Parallel Processing HOWTO http://yara.ecn.purdue.edu/~pplinux/PPHOWTO/pphowto.html

5.3 Papers

- Chance Reschke, Thomas Sterling, Daniel Ridge, Daniel Savarese, Donald Becker, and Phillip Merkey *A Design Study of Alternative Network Topologies for the Beowulf Parallel Workstation*. Proceedings Fifth IEEE International Symposium on High Performance Distributed Computing, 1996. http://www.beowulf.org/papers/HPDC96/hpdc96.html
- Daniel Ridge, Donald Becker, Phillip Merkey, Thomas Sterling Becker, and Phillip Merkey. *Harnessing the Power of Parallelism in a Pile-of-PCs*. Proceedings, IEEE Aerospace, 1997. http://www.beowulf.org/papers/AA97/aa97.ps
- Thomas Sterling, Donald J. Becker, Daniel Savarese, Michael R. Berry, and Chance Res. *Achieving a Balanced Low-Cost Architecture for Mass Storage Management through Multiple Fast Ethernet Channels on the Beowulf Parallel Workstation*. Proceedings, International Parallel Processing Symposium, 1996. http://www.beowulf.org/papers/IPPS96/ipps96.html
- Donald J. Becker, Thomas Sterling, Daniel Savarese, Bruce Fryxell, Kevin Olson. *Communication Overhead for Space Science Applications on the Beowulf Parallel Workstation*. Proceedings,High Performance and Distributed Computing, 1995.

TOC

http://www.beowulf.org/papers/HPDC95/hpdc95.html

- Donald J. Becker, Thomas Sterling, Daniel Savarese, John E. Dorband, Udaya A. Ranawak, Charles V. Packer. *BEOWULF: A PARALLEL WORKSTATION FOR SCIENTIFIC COMPUTATION*. Proceedings, International Conference on Parallel Processing, 95. http://www.beowulf.org/papers/ICPP95/icpp95.html
- Papers at the Beowulf site http://www.beowulf.org/papers/papers.html

5.4 Software

- PVM - Parallel Virtual Machine http://www.epm.ornl.gov/pvm/pvm_home.html
- LAM/MPI (Local Area Multicomputer / Message Passing Interface http://www.mpi.nd.edu/lam
- BERT77 - FORTRAN conversion tool http://www.plogic.com/bert.html
- Beowulf software from Beowulf Project Page http://beowulf.gsfc.nasa.gov/software/software.html
- Jacek's Beowulf-utils ftp://ftp.sci.usq.edu.au/pub/jacek/beowulf-utils
- bWatch - cluster monitoring tool http://www.sci.usq.edu.au/staff/jacek/bWatch

5.5 Beowulf Machines

- Avalon consists of 140 Alpha processors, 36 GB of RAM, and is probably the fastest Beowulf machine, cruising at 47.7 Gflops and ranking 114th on the Top 500 list. http://swift.lanl.gov/avalon/
- Megalon-A Massively PArallel CompuTer Resource (MPACTR) consists of 14, quad CPU Pentium Pro 200 nodes, and 14 GB of RAM. http://megalon.ca.sandia.gov/description.html
- theHIVE - Highly-parallel Integrated Virtual Environment is another fast Beowulf Supercomputer. theHIVE is a 64 node, 128 CPU machine with the total of 4 GB RAM. http://newton.gsfc.nasa.gov/thehive/
- Topcat is a much smaller machine and consists of 16 CPUs and 1.2 GB RAM. http://www.sci.usq.edu.au/staff/jacek/topcat
- MAGI cluster - this is a very interesting site with many good links. http://noel.feld.cvut.cz/magi/

5.6 Other Interesting Sites

- SMP Linux http://www.linux.org.uk/SMP/title.html
- Paralogic - Buy a Beowulf http://www.plogic.com

5.7 History

- Legends - Beowulf http://legends.dm.net/beowulf/index.html
- The Adventures of Beowulf http://www.lnstar.com/literature/beowulf/beowulf.html

6. Source code

6.1 sum.c

```
/* Jacek Radajewski jacek@usq.edu.au */
/* 21/08/1998 */

#include <stdio.h>
#include <math.h>

int main (void) {

  double result = 0.0;
  double number = 0.0;
  char string[80];


  while (scanf("%s", string) != EOF) {

    number = atof(string);
    result = result + number;
  }

  printf("%lf\n", result);

  return 0;

}
```

6.2 sigmasqrt.c

```
/* Jacek Radajewski jacek@usq.edu.au */
/* 21/08/1998 */

#include <stdio.h>
#include <math.h>

int main (int argc, char** argv) {
```

TOC

```
  long number1, number2, counter;
  double result;

  if (argc < 3) {
    printf ("usage : %s number1 number2\n",argv[0]);
    exit(1);
  } else {
    number1 = atol (argv[1]);
    number2 = atol (argv[2]);
    result = 0.0;
  }

  for (counter = number1; counter <= number2; counter++) {
    result = result + sqrt((double)counter);
  }

  printf("%lf\n", result);

  return 0;

}
```

6.3 prun.sh

```
#!/bin/bash
# Jacek Radajewski jacek@usq.edu.au
# 21/08/1998

export SIGMASQRT=/home/staff/jacek/beowulf/HOWTO/example1/sigmasqrt

# $OUTPUT must be a named pipe
# mkfifo output

export OUTPUT=/home/staff/jacek/beowulf/HOWTO/example1/output

rsh scilab01 $SIGMASQRT         1  50000000 > $OUTPUT < /dev/null&
rsh scilab02 $SIGMASQRT  50000001 100000000 > $OUTPUT < /dev/null&
rsh scilab03 $SIGMASQRT 100000001 150000000 > $OUTPUT < /dev/null&
rsh scilab04 $SIGMASQRT 150000001 200000000 > $OUTPUT < /dev/null&
rsh scilab05 $SIGMASQRT 200000001 250000000 > $OUTPUT < /dev/null&
rsh scilab06 $SIGMASQRT 250000001 300000000 > $OUTPUT < /dev/null&
rsh scilab07 $SIGMASQRT 300000001 350000000 > $OUTPUT < /dev/null&
rsh scilab08 $SIGMASQRT 350000001 400000000 > $OUTPUT < /dev/null&
```

TOC

```
rsh scilab09 $SIGMASQRT 400000001 450000000 > $OUTPUT < /dev/null&
rsh scilab10 $SIGMASQRT 450000001 500000000 > $OUTPUT < /dev/null&
rsh scilab11 $SIGMASQRT 500000001 550000000 > $OUTPUT < /dev/null&
rsh scilab12 $SIGMASQRT 550000001 600000000 > $OUTPUT < /dev/null&
rsh scilab13 $SIGMASQRT 600000001 650000000 > $OUTPUT < /dev/null&
rsh scilab14 $SIGMASQRT 650000001 700000000 > $OUTPUT < /dev/null&
rsh scilab15 $SIGMASQRT 700000001 750000000 > $OUTPUT < /dev/null&
rsh scilab16 $SIGMASQRT 750000001 800000000 > $OUTPUT < /dev/null&
rsh scilab17 $SIGMASQRT 800000001 850000000 > $OUTPUT < /dev/null&
rsh scilab18 $SIGMASQRT 850000001 900000000 > $OUTPUT < /dev/null&
rsh scilab19 $SIGMASQRT 900000001 950000000 > $OUTPUT < /dev/null&
rsh scilab20 $SIGMASQRT 950000001 1000000000 > $OUTPUT < /dev/null&
```

The Linux BootPrompt-HowTo

by Paul Gortmaker.

v1.2, May 15, 1999

This is the BootPrompt-Howto, which is a compilation of all the possible boot time arguments that can be passed to the Linux kernel at boot time. This includes all kernel and device parameters. A discussion of how the kernel sorts boot time arguments, along with an overview of some of the popular software used to boot Linux kernels is also included.

1. Introduction

- 1.1 Disclaimer and Copyright
- 1.2 Intended Audience and Applicability
- 1.3 Related Documentation
- 1.4 The Linux Newsgroups
- 1.5 New Versions of this Document

2. Overview of Boot Prompt Arguments

- 2.1 LILO (LInux LOader)
- 2.2 LoadLin
- 2.3 The ``rdev'' utility
- 2.4 How the Kernel Sorts the Arguments
- 2.5 Setting Environment Variables.
- 2.6 Passing Arguments to the `init' program

3. General Non-Device Specific Boot Args

- 3.1 Root Filesystem options
- 3.2 Options Relating to RAM Disk Management
- 3.3 Boot Arguments Related to Memory Handling

- 3.4 Boot Arguments for NFS Root Filesystem
- 3.5 Other Misc. Kernel Boot Arguments

4. Boot Arguments to Control PCI Bus Behaviour (`pci=')

- 4.1 The `pci=bios' and `pci=nobios' Arguments
- 4.2 The `pci=conf1' and `pci=conf2' Arguments
- 4.3 The `pci=io=' Argument
- 4.4 The `pci=nopeer' Argument
- 4.5 The `pci=nosort' Argument
- 4.6 The `pci=off' Argument
- 4.7 The `pci=reverse' Argument

5. Boot Arguments for Video Frame Buffer Drivers

- 5.1 The `video=map:...' Argument
- 5.2 The `video=scrollback:...' Argument
- 5.3 The `video=vc:...' Argument

6. Boot Arguments for SCSI Peripherals.

- 6.1 Arguments for Mid-level Drivers
- 6.2 Arguments for SCSI Host Adapters
- 6.3 SCSI Host Adapters that don't Accept Boot Args

7. Hard Disks

- 7.1 IDE Disk/CD-ROM Driver Parameters
- 7.2 Standard ST-506 Disk Driver Options (`hd=')
- 7.3 XT Disk Driver Options (`xd=')

TOC

8. CD-ROMs (Non-SCSI/ATAPI/IDE)

- 8.1 The Aztech Interface (`aztcd=')
- 8.2 The CDU-31A and CDU-33A Sony Interface (`cdu31a=')
- 8.3 The CDU-535 Sony Interface (`sonycd535=')
- 8.4 The GoldStar Interface (`gscd=')
- 8.5 The ISP16 Interface (`isp16=')
- 8.6 The Mitsumi Standard Interface (`mcd=')
- 8.7 The Mitsumi XA/MultiSession Interface (`mcdx=')
- 8.8 The Optics Storage Interface (`optcd=')
- 8.9 The Phillips CM206 Interface (`cm206=')
- 8.10 The Sanyo Interface (`sjcd=')
- 8.11 The SoundBlaster Pro Interface (`sbpcd=')

9. Serial and ISDN Drivers

- 9.1 The ICN ISDN driver (`icn=')
- 9.2 The PCBIT ISDN driver (`pcbit=')
- 9.3 The Teles ISDN driver (`teles=')
- 9.4 The DigiBoard Driver (`digi=')
- 9.5 The RISCom/8 Multiport Serial Driver (`riscom8=')
- 9.6 The Baycom Serial/Parallel Radio Modem (`baycom=')

10. Other Hardware Devices

- 10.1 Ethernet Devices (`ether=')
- 10.2 The Floppy Disk Driver (`floppy=')
- 10.3 The Sound Driver (`sound=')
- 10.4 The Bus Mouse Driver (`bmouse=')
- 10.5 The MS Bus Mouse Driver (`msmouse=')
- 10.6 The Printer Driver (`lp=')

11. Copying, Translations, Closing, etc.

- 11.1 Copyright and Disclaimer
- 11.2 Closing

1. Introduction

The kernel has a limited capability to accept information at boot in the form of a `command line', similar to an argument list you would give to a program. In general this is used to supply the kernel with information about hardware parameters that the kernel would not be able to determine on its own, or to avoid/override the values that the kernel would otherwise detect.

However, if you just copy a kernel image directly to a floppy, (e.g. `cp zImage /dev/fd0`) then you are not given a chance to specify any arguments to that kernel. So most Linux users will use software like *LILO* or *loadlin* that takes care of handing these arguments to the kernel, and then booting it.

This present revision covers kernels up to and including v2.2.9. Some features that are unique to development/testing kernels up to v2.3.2 are also documented.

The BootPrompt-Howto is by:

Paul Gortmaker, `p_gortmaker@yahoo.com`

1.1 Disclaimer and Copyright

This document is Copyright (c) 1995-1999 by Paul Gortmaker. Please see the Disclaimer and Copying information at the end of this document (copyright) for information about redistribution of this document and the usual `we are not responsible for what you manage to break...' type legal stuff.

1.2 Intended Audience and Applicability

Most Linux users should never have to even look at this document. Linux does an exceptionally good job at detecting most hardware and picking reasonable default settings for most parameters. The information in this document is aimed at users who might want to change some of the default settings to optimize the kernel to their particular machine, or to a user who has `rolled their own' kernel to support a not so common piece of hardware for which automatic detection is currently not available.

IMPORTANT NOTE: Driver related boot prompt arguments only apply to hardware drivers that are compiled directly into the kernel. They have *no effect* on drivers that are loaded as modules. Most Linux distributions come with a basic `bare-bones' kernel, and the drivers are small modules that are loaded after the kernel has initialized. If you are unsure if you are using modules then look at `man depmod` and `man modprobe` along with the contents of your `/etc/conf.modules`.

TOC

1.3 Related Documentation

The most up-to-date documentation will always be the kernel source itself. Hold on! Don't get scared. You don't need to know any programming to read the comments in the source files. For example, if you were looking for what arguments could be passed to the AHA1542 SCSI driver, then you would go to the `linux/drivers/scsi` directory, and look at the file `aha1542.c` -- and within the first 100 lines, you would find a plain english description of the boot time arguments that the 1542 driver accepts.

The `linux` directory is usually found in `/usr/src/` for most distributions. All references in this document to files that come with the kernel will have their pathname abbreviated to start with `linux` - you will have to append the `/usr/src/` or whatever is appropriate for your system. (If you can't find the file in question, then make use of the `find` and `locate` commands.)

The next best thing will be any documentation files that are distributed with the kernel itself. There are now quite a few of these, and most of them can be found in the directory `linux/Documentation` and subdirectories from there. Sometimes there will be `README.foo` files that can be found in the related driver directory (e.g. `linux/drivers/???/`, where examples of `???` could be `scsi`, `char`, or `net`).

If you have figured out what boot-args you intend to use, and now want to know how to get that information to the kernel, then look at the documentation that comes with the software that you use to boot the kernel (e.g. LILO or loadlin). A brief overview is given below, but it is no substitute for the documentation that comes with the booting software.

1.4 The Linux Newsgroups

If you have questions about passing boot arguments to the kernel, please check this document first. If this and the related documentation mentioned above does not answer your question(s) then you can try the Linux newsgroups. General questions on how to configure your system should be directed to the comp.os.linux.setup newsgroup. We ask that you *please* respect this general guideline for content, and don't cross-post your request to other groups.

Of course you should try checking the group before blindly posting your question, as it may even be a Frequently Asked Question (a FAQ). A quick browse of the Linux FAQ before posting is a *good* idea. You should be able to find the FAQ somewhere close to where you found this document. If it is not a FAQ then use newsgroup archives, such as those at `http://www.dejanews.com` to quickly search years worth of postings for your topic. Chances are someone else has already asked (and another person answered!) the question that you now have.

1.5 New Versions of this Document

New versions of this document can be retrieved via anonymous FTP from most Linux FTP sites in the directory `/pub/Linux/docs/HOWTO/`. Updates will be made as new information and/or drivers becomes available. If this copy that you are presently reading is more than six months old, then you should probably check to see if a newer copy exists. I would recommend viewing this via a WWW

browser or in the Postscript/dvi format. Both of these contain cross-references that are lost in a simpl plain text version.

If you want to get the official copy, here is URL.

BootPrompt-HOWTO

---

2. Overview of Boot Prompt Arguments

This section gives some examples of software that can be used to pass kernel boot-time arguments to the kernel itself. It also gives you an idea of how the arguments are processed, what limitations there are on the boot args, and how they filter down to each appropriate device that they are intended for.

It is *important* to note that spaces should *not* be used in a boot argument, but only between separate arguments. A list of values that are for a single argument are to be separated with a comma between the values, and again without any spaces. See the following examples below.

```
ether=9,0x300,0xd0000,0xd4000,eth0  root=/dev/hda1            *RIGHT*
ether = 9, 0x300, 0xd0000, 0xd4000, eth0  root = /dev/hda1    *WRONG*
```

Once the Linux kernel is up and running, one can view the command line arguments that were in place at boot by simply typing `cat /proc/cmdline` at a shell prompt.

2.1 LILO (LInux LOader)

The LILO program (LInux LOader) written by Werner Almesberger is the most commonly used. It has the ability to boot various kernels, and stores the configuration information in a plain text file. Most distributions ship with LILO as the default boot-loader. LILO can boot DOS, OS/2, Linux, FreeBSD, etc. without any difficulties, and is quite flexible.

A typical configuration will have LILO stop and print `LILO:` shortly after you turn on your computer. It will then wait for a few seconds for any optional input from the user, and failing that it will then boot the default system. Typical system labels that people use in the LILO configuration files are `linux` and `backup` and `msdos`. If you want to type in a boot argument, you type it in here, after typing in the system label that you want LILO to boot from, as shown in the example below.

```
LILO: linux root=/dev/hda1
```

LILO comes with excellent documentation, and for the purposes of boot args discussed here, the LILO `append=` command is of significant importance when one wants to add a boot time argument as a permanent addition to the LILO config file. You simply add something like `append = "foo=bar"` to the `/etc/lilo.conf` file. It can either be added at the top of the config file, making it apply to all sections, or to a single system section by adding it inside an `image=` section. Please see the LILO documentation for a more complete description.

2.2 LoadLin

The other commonly used Linux loader is `LoadLin' which is a DOS program that has the capability to launch a Linux kernel from the DOS prompt (with boot-args) assuming that certain resources are available. This is good for

people that use DOS and want to launch into Linux from DOS.

It is also very useful if you have certain hardware which relies on the supplied DOS driver to put the hardware into a known state. A common example is `SoundBlaster Compatible' sound cards that require the DOS driver to set a few proprietary registers to put the card into a SB compatible mode. Booting DOS with the supplied driver, and then loading Linux from the DOS prompt with `LOADLIN.EXE` avoids the reset of the card that happens if one rebooted instead. Thus the card is left in a SB compatible mode and hence is useable under Linux.

There are also other programs that can be used to boot Linux. For a complete list, please look at the programs available on your local Linux ftp mirror, under `system/Linux-boot/`.

2.3 The ``rdev'' utility

There are a few of the kernel boot parameters that have their default values stored in various bytes in the kernel image itself. There is a utility called `rdev` that is installed on most systems that knows where these values are, and how to change them. It can also change things that have no kernel boot argument equivalent, such as the default video mode used.

The rdev utility is usually also aliased to swapdev, ramsize, vidmode and rootflags. These are the five things that rdev can change, those being the root device, the swap device, the RAM disk parameters, the default video mode, and the readonly/readwrite setting of root device.

More information on `rdev` can be found by typing `rdev -h` or by reading the supplied man page (`man rdev`).

2.4 How the Kernel Sorts the Arguments

Most of the boot args take the form of:

```
name[=value_1][,value_2]...[,value_11]
```

where `name' is a unique keyword that is used to identify what part of the kernel the associated values (if any) are to be given to. Multiple boot args are just a space separated list of the above format. Note the limit of 11 is real, as the present code only handles 11 comma separated parameters per keyword. (However, you can re-use the same keyword with up to an additional 11 parameters in unusually complicated situations, assuming the setup function supports it.) Also note that the kernel splits the list into a maximum of ten integer arguments, and a following string, so you can't really supply 11 integers unless you convert the 11th arg from a string to an int in the driver itself.

Most of the sorting goes on in `linux/init/main.c`. First, the kernel checks to see if the argument is any of the special arguments `root=', `ro', `rw', or `debug'. The meaning of these special arguments is described further on in the document.

Then it walks a list of setup functions (contained in the `bootsetups` array) to see if the specified argument string (such as `foo') has been associated with a setup function (`foo_setup()`) for a particular device or part of the kernel. If you passed the kernel the line `foo=3,4,5,6,bar` then the kernel would search the `bootsetups` array to see if `foo' was registered. If it was, then it would call the setup function associated with `foo' (`foo_setup()`) and hand it the integer arguments 3, 4, 5 and 6 as given on the kernel command line, and also hand it the string argument `bar`.

TOC

2.5 Setting Environment Variables.

Anything of the form `foo=bar' that is not accepted as a setup function as described above is then interpreted as an environment variable to be set. An example would be to use `TERM=vt100` or `BOOT_IMAGE=vmlinuz.bak` as a boot argument. These environment variables are typically tested for in the initialization scripts to enable or disable a wide range of things.

2.6 Passing Arguments to the `init' program

Any remaining arguments that were not picked up by the kernel and were not interpreted as environment variables are then passed onto process one, which is usually the `init` program. The most common argument that is passed to the `init` process is the word *single* which instructs `init` to boot the computer in single user mode, and not launch all the usual daemons. Check the manual page for the version of `init` installed on your system to see what arguments it accepts.

---

3. General Non-Device Specific Boot Args

These are the boot arguments that are not related to any specific device or peripheral. They are instead related to certain internal kernel parameters, such as memory handling, ramdisk handling, root file system handling and others.

3.1 Root Filesystem options

The following options all pertain to how the kernel selects and handles the root filesystem.

The `root=' Argument

This argument tells the kernel what device is to be used as the root filesystem while booting. The default of this setting is the value of the root device of the system that the kernel was built on. For example, if the kernel in question was built on a system that used `/dev/hda1' as the root partition, then the default root device would be `/dev/hda1'. To override this default value, and select the second floppy drive as the root device, one would use `root=/dev/fd1'.

Valid root devices are any of the following devices:

(1) /dev/hdaN to /dev/hddN, which is partition N on ST-506 compatible disk `a to d'.

(2) /dev/sdaN to /dev/sdeN, which is partition N on SCSI compatible disk `a to e'.

(3) /dev/xdaN to /dev/xdbN, which is partition N on XT compatible disk `a to b'.

(4) /dev/fdN, which is floppy disk drive number N. Having N=0 would be the DOS `A:' drive, and N=1 would be `B:'.

(5) /dev/nfs, which is not really a device, but rather a flag to tell the kernel to get the root fs via the network.

The more awkward and less portable numeric specification of the above possible disk devices in major/minor format is also accepted. (e.g. /dev/sda3 is major 8, minor 3, so you could use `root=0x803` as an alternative.)

This is one of the few kernel boot arguments that has its default stored in the kernel image, and which can thus be altered with the `rdev` utility.

The `ro' Argument

When the kernel boots, it needs a root filesystem to read basic things off of. This is the root filesystem that is mounted at boot. However, if the root filesystem is mounted with write access, you can not reliably check the filesystem integrity with half-written files in progress. The `ro' option tells the kernel to mount the root filesystem as `readonly' so that any filesystem consistency check programs (fsck) can safely assume that there are no half-written files in progress while performing the check. No programs or processes can write to files on the filesystem in question until it is `remounted' as read/write capable.

This is one of the few kernel boot arguments that has its default stored in the kernel image, and which can thus be altered with the `rdev` utility.

TOC

The `rw' Argument

This is the exact opposite of the above, in that it tells the kernel to mount the root filesystem as read/write. The default is to mount the root filesystem as read/write anyway. Do not run any `fsck' type programs on a filesystem that is mounted read/write.

The same value stored in the image file mentioned above is also used for this parameter, accessible via `rdev`.

3.2 Options Relating to RAM Disk Management

The following options all relate to how the kernel handles the RAM disk device, which is usually used for bootstrapping machines during the install phase, or for machines with modular drivers that need to be installed to access the root filesystem.

The `ramdisk_start=' Argument

To allow a kernel image to reside on a floppy disk along with a compressed ramdisk image, the `ramdisk_start=<offset>' command was added. The kernel can't be included into the compressed ramdisk filesystem image, because it needs to be stored starting at block zero so that the BIOS can load the bootsector and then the kernel can bootstrap itself to get going.

Note: If you are using an uncompressed ramdisk image, then the kernel can be a part of the filesystem image that is being loaded into the ramdisk, and the floppy can be booted with LILO, or the two can be separate as is done for the compressed images.

If you are using a two-disk boot/root setup (kernel on disk 1, ramdisk image on disk 2) then the ramdisk would start at block zero, and an offset of zero would be used. Since this is the default value, you would not need to actually use the command at all.

The `load_ramdisk=' Argument

This parameter tells the kernel whether it is to try to load a ramdisk image or not. Specifying `load_ramdisk=1' will tell the kernel to load a floppy into the ramdisk. The default value is zero, meaning that the kernel should not try to load a ramdisk.

Please see the file `linux/Documentation/ramdisk.txt` for a complete description of the new boot time arguments, and how to use them. A description of how this parameter can be set and stored in the kernel image via `rdev' is also described.

The `prompt_ramdisk=' Argument

This parameter tells the kernel whether or not to give you a prompt asking you to insert the floppy containing the ramdisk image. In a single floppy configuration the ramdisk image is on the same floppy as the kernel that just finished loading/booting and so a prompt is not needed. In this case one can use `prompt_ramdisk=0'. In a two floppy configuration, you will need the chance to switch disks, and thus `prompt_ramdisk=1' can be used. Since this is the default value, it doesn't really need to be specified. ((Historical note: Sneaky people used to use the `vga=ask' LILO option to temporarily pause the boot process and allow a chance to switch from boot to root floppy.)

Please see the file `linux/Documentation/ramdisk.txt` for a complete description of the new boot time

arguments, and how to use them. A description of how this parameter can be set and stored in the kernel image via `rdev' is also described.

The `ramdisk_size=' Argument

While it is true that the ramdisk grows dynamically as required, there is an upper bound on its size so that it doesn't consume all available RAM and leave you in a mess. The default is 4096 (i.e. 4MB) which should be large enough for most needs. You can override the default to a bigger or smaller size with this boot argument.

Please see the file `linux/Documentation/ramdisk.txt` for a complete description of the new boot time arguments, and how to use them. A description of how this parameter can be set and stored in the kernel image via `rdev' is also described.

The `ramdisk=' Argument (obsolete)

(NOTE: This argument is obsolete, and should not be used except on kernels v1.3.47 and older. The commands that should be used for the ramdisk device are documented above.)

This specifies the size in kB of the RAM disk device. For example, if one wished to have a root filesystem on a 1.44MB floppy loaded into the RAM disk device, they would use:

```
ramdisk=1440
```

This is one of the few kernel boot arguments that has its default stored in the kernel image, and which can thus be altered with the `rdev` utility.

The `noinitrd' (initial RAM disk) Argument

The v2.x and newer kernels have a feature where the root filesystem can be initially a RAM disk, and the kernel executes `/linuxrc` on that RAM image. This feature is typically used to allow loading of modules needed to mount the real root filesystem (e.g. load the SCSI driver modules stored in the RAM disk image, and then mount the real root filesystem on a SCSI disk.)

The actual `noinitrd' argument determines what happens to the initrd data after the kernel has booted. When specified, instead of converting it to a RAM disk, it is accessible via `/dev/initrd`, which can be read once before the RAM is released back to the system. For full details on using the initial RAM disk, please consult `linux/Documentation/initrd.txt`. In addition, the most recent versions of `LILO` and `LOADLIN` should have additional useful information.

3.3 Boot Arguments Related to Memory Handling

The following arguments alter how Linux detects or handles the physical and virtual memory of your system.

The `mem=' Argument

This argument has two purposes: The original purpose was to specify the amount of installed memory (or a value less than that if you wanted to limit the amount of memory available to linux). The second (and hardly used) purpose is to specify `mem=nopentium` which tells the Linux kernel to not use the 4MB page table performance

TOC

feature.

The original BIOS call defined in the PC specification that returns the amount of installed memory was only designed to be able to report up to 64MB. (Yes, another lack of foresight, just like the 1024 cylinder disks... sigh.) Linux uses this BIOS call at boot to determine how much memory is installed. If you have more than 64MB of RAM installed, you can use this boot argument to tell Linux how much memory you have. Here is a quote from Linus on the usage of the `mem=` parameter.

``The kernel will accept any `mem=xx' parameter you give it, and if it turns out that you lied to it, it will crash horribly sooner or later. The parameter indicates the highest addressable RAM address, so `mem=0x1000000' means you have 16MB of memory, for example. For a 96MB machine this would be `mem=0x6000000'. If you tell Linux that it has more memory than it actually does have, bad things will happen: maybe not at once, but surely eventually."

Note that the argument does not have to be in hex, and the suffixes `k' and `M' (case insensitive) can be used to specify kilobytes and Megabytes, respectively. (A `k' will cause a 10 bit shift on your value, and a `M' will cause a 20 bit shift.) A typical example for a 128MB machine would be "`mem=128m`".

The `swap=' Argument

This allows the user to tune some of the virtual memory (VM) parameters that are related to swapping to disk. It accepts the following eight parameters:

```
MAX_PAGE_AGE
PAGE_ADVANCE
PAGE_DECLINE
PAGE_INITIAL_AGE
AGE_CLUSTER_FRACT
AGE_CLUSTER_MIN
PAGEOUT_WEIGHT
BUFFEROUT_WEIGHT
```

Interested hackers are advised to have a read of `linux/mm/swap.c` and also make note of the goodies in `/proc/sys/vm`. Kernels come with some useful documentation on this in the `linux/Documentation/vm/` directory.

The `buff=' Argument

Similar to the `swap=' argument, this allows the user to tune some of the parameters related to buffer memory management. It accepts the following six parameters:

```
MAX_BUFF_AGE
BUFF_ADVANCE
BUFF_DECLINE
BUFF_INITIAL_AGE
BUFFEROUT_WEIGHT
BUFFERMEM_GRACE
```

TOC

Interested hackers are advised to have a read of `linux/mm/swap.c` and also make note of the goodies in `/proc/sys/vm`. Kernels come with some useful documentation on this in the `linux/Documentation/vm/` directory.

3.4 Boot Arguments for NFS Root Filesystem

Linux supports systems such as diskless workstations via having their root filesystem as NFS (Network FileSystem). These arguments are used to tell the diskless workstation which machine it is to get its system from. Also note that the argument `root=/dev/nfs` is required. Detailed information on using an NFS root fs is in the file `linux/Documentation/nfsroot.txt`. You should read that file, as the following is only a quick summary taken directly from that file.

The `nfsroot=' Argument

This argument tells the kernel which machine, what directory and what NFS options to use for the root filesystem. The form of the argument is as follows:

```
nfsroot=[<server-ip>:]<root-dir>[,<nfs-options>]
```

If the nfsroot parameter is not given on the command line, the default `/tftpboot/%s' will be used. The other options are as follows:

<server-ip> -- Specifies the IP address of the NFS server. If this field is not given, the default address as determined by the nfsaddrs variable (see below) is used. One use of this parameter is for example to allow using different servers for RARP and NFS. Usually you can leave this blank.

<root-dir> -- Name of the directory on the server to mount as root. If there is a `%s' token in the string, the token will be replaced by the ASCII-representation of the client's IP address.

<nfs-options> -- Standard NFS options. All options are separated by commas. If the options field is not given, the following defaults will be used:

```
port            = as given by server portmap daemon
rsize           = 1024
wsize           = 1024
timeo           = 7
retrans         = 3
acregmin        = 3
acregmax        = 60
acdirmin        = 30
acdirmax        = 60
flags           = hard, nointr, noposix, cto, ac
```

TOC

The `nfsaddrs=' Argument

This boot argument sets up the various network interface addresses that are required to communicate over the network. If this argument is not given, then the kernel tries to use RARP and/or BOOTP to figure out these parameters. The form is as follows:

```
nfsaddrs=<my-ip>:<serv-ip>:<gw-ip>:<netmask>:<name>:<dev>:<auto>
```

<my-ip> -- IP address of the client. If empty, the address will either be determined by RARP or BOOTP. What protocol is used de- pends on what has been enabled during kernel configuration and on the <auto> parameter. If this parameter is not empty, neither RARP nor BOOTP will be used.

<serv-ip> -- IP address of the NFS server. If RARP is used to determine the client address and this parameter is NOT empty only replies from the specified server are accepted. To use different RARP and NFS server, specify your RARP server here (or leave it blank), and specify your NFS server in the nfsroot parameter (see above). If this entry is blank the address of the server is used which answered the RARP or BOOTP request.

<gw-ip> -- IP address of a gateway if the server in on a different subnet. If this entry is empty no gateway is used and the server is assumed to be on the local network, unless a value has been received by BOOTP.

<netmask> -- Netmask for local network interface. If this is empty, the netmask is derived from the client IP address, unless a value has been received by BOOTP.

<name> -- Name of the client. If empty, the client IP address is used in ASCII-notation, or the value received by BOOTP.

<dev> -- Name of network device to use. If this is empty, all devices are used for RARP requests, and the first one found for BOOTP. For NFS the device is used on which either RARP or BOOTP replies have been received. If you only have one device you can safely leave this blank.

<auto> -- Method to use for autoconfiguration. If this is either `rarp' or `bootp' the specified protocol is being used. If the value is `both' or empty, both protocols are used so far as they have been enabled during kernel configuration Using 'none' means no autoconfiguration. In this case you have to specify all necessary values in the fields before.

The <auto> parameter can appear alone as the value to the nfsaddrs parameter (without all the `:' characters before) in which case autoconfiguration is used. However, the `none' value is not available in that case.

3.5 Other Misc. Kernel Boot Arguments

These various boot arguments let the user tune certain internal kernel parameters.

The `debug' Argument

The kernel communicates important (and not-so important) messages to the operator via the `printk()` function. If the message is considered important, the `printk()` function will put a copy on the present console as well as handing it off to the `klogd()` facility so that it gets logged to disk. The reason for printing important messages to the console as well as logging them to disk is because under unfortunate circumstances (e.g. a disk failure) the message won't make it to disk and will be lost.

TOC

The threshold for what is and what isn't considered important is set by the `console_loglevel` variable default is to log anything more important than `DEBUG` (level 7) to the console. (These levels are defined in the include file `kernel.h`) Specifying `debug` as a boot argument will set the console loglevel to 10, so that *all* kernel messages appear on the console.

The console loglevel can usually also be set at run time via an option to the `klogd()` program. Check the man page for the version installed on your system to see how to do this.

The `init=' Argument

The kernel defaults to starting the `init' program at boot, which then takes care of setting up the computer for users via launching getty programs, running `rc' scripts and the like. The kernel first looks for `/sbin/init`, then `/etc/init` (depreciated), and as a last resort, it will try to use `/bin/sh` (possibly on `/etc/rc`). If for example, your init program got corrupted and thus stopped you from being able to boot, you could simply use the boot prompt `init=/bin/sh` which would drop you directly into a shell at boot, allowing you to replace the corrupted program.

The `kbd-reset' Argument

Normally on i386 based machines, the Linux kernel does not reset the keyboard controller at boot, since the BIOS is supposed to do this. But as usual, not all machines do what they should. Supplying this option may help if you are having problems with your keyboard behaviour. It simply forces a reset at initialization time. (Some have argued that this should be the default behaviour anyways).

The `maxcpus=' Argument

The number given with this argument limits the maximum number of CPUs activated in SMP mode. Using a value of 0 is equivalent to the `nosmp` option.

The `mca-pentium' Argument

The IBM model 95 Microchannel machines seem to lock up on the test that Linux usually does to detect the type of math chip coupling. Since all Pentium chips have a built in math processor, this test (and the lock up problem) can be avoided by using this boot option.

The `md=' Argument

If your root filesystem is on a Multiple Device then you can use this (assuming you compiled in boot support) to tell the kernel the multiple device layout. The format (from the file `linux/Documentation/md.txt`) is:

```
md=md_device_num,raid_level,chunk_size_factor,fault_level,dev0,dev1,...,devN
```

Where `md_device_num` is the number of the md device, i.e. 0 means md0, 1 means md1, etc. For `raid_level`, use -1 for linear mode and 0 for striped mode. Other modes are currently unsupported. The `chunk_size_factor` is for raid-0 and raid-1 only and sets the chunk size as PAGE_SIZE shifted left the specified amount. The `fault_level` is only for raid-1 and sets the maximum fault number to the specified number. (Currently unsupported due to lack of boot support for raid1.) The `dev0-devN` are a commaseparated list of the devices that make up the individual md device: e.g. `/dev/hda1,/dev/hdc1,/dev/sda1`

The `no387' Argument

Some i387 coprocessor chips have bugs that show up when used in 32 bit protected mode. For example, some of the early ULSI-387 chips would cause solid lockups while performing floating point calculations, apparently due to a bug in the FRSAV/FRRESTOR instructions. Using the `no387' boot argument causes Linux to ignore the math coprocessor even if you have one. Of course you must then have your kernel compiled with math emulation support! This may also be useful if you have one of those *really* old 386 machines that could use an 80287 FPU, as Linux can't use an 80287.

The `no-hlt' Argument

The i386 (and successors thereof) family of CPUs have a `hlt' instruction which tells the CPU that nothing is going to happen until an external device (keyboard, modem, disk, etc.) calls upon the CPU to do a task. This allows the CPU to enter a `low-power' mode where it sits like a zombie until an external device wakes it up (usually via an interrupt). Some of the early i486DX-100 chips had a problem with the `hlt' instruction, in that they couldn't reliably return to operating mode after this instruction was used. Using the `no-hlt' instruction tells Linux to just run an infinite loop when there is nothing else to do, and to *not* halt your CPU when there is no activity. This allows people with these broken chips to use Linux, although they would be well advised to seek a replacement through a warranty where possible.

The `no-scroll' Argument

Using this argument at boot disables scrolling features that make it difficult to use Braille terminals.

The `noapic' Argument

Using this option tells a SMP kernel to not use some of the advanced features of the interrupt controller on multi processor machines. See `linux/Documentation/IO-APIC.txt` for more information.

The `nosmp' Argument

Use of this option will tell a SMP kernel on a SMP machine to operate single processor. Typically only used for debugging and determining if a particular problem is SMP related.

The `panic=' Argument

In the unlikely event of a kernel panic (i.e. an internal error that has been detected by the kernel, and which the kernel decides is serious enough to moan loudly and then halt everything), the default behaviour is to just sit there until someone comes along and notices the panic message on the screen and reboots the machine. However if a machine is running unattended in an isolated location it may be desirable for it to automatically reset itself so that the machine comes back on line. For example, using `panic=30` at boot would cause the kernel to try and reboot itself 30 seconds after the kernel panic happened. A value of zero gives the default behaviour, which is to wait forever.

Note that this timeout value can also be read and set via the `/proc/sys/kernel/panic` sysctl interface.

TOC

The `pci=' Argument

The `pirq=' Argument

Using this option tells a SMP kernel information on the PCI slot versus IRQ settings for SMP motherboards which are unknown (or known to be blacklisted). See `linux/Documentation/IO-APIC.txt` for more information.

The `profile=' Argument

Kernel developers can enable an option that allows them to profile how and where the kernel is spending its CPU cycles in an effort to maximize efficiency and performance. This option lets you set the profile shift count at boot. Typically it is set to two. You can also compile your kernel with profiling enabled by default. In either case, you need a tool such as `readprofile.c` that can make use of the `/proc/profile` output.

The `reboot=' Argument

This option controls the type of reboot that Linux will do when it resets the computer (typically via `/sbin/init` handling a Control-Alt-Delete). The default as of v2.0 kernels is to do a `cold' reboot (i.e. full reset, BIOS does memory check, etc.) instead of a `warm' reboot (i.e. no full reset, no memory check). It was changed to be cold by default since that tends to work on cheap/broken hardware that fails to reboot when a warm reboot is requested. To get the old behaviour (i.e. warm reboots) use `reboot=w` or in fact any word that starts with `w` will work.

Why would you bother? Some disk controllers with cache memory on board can sense a warm reboot, and flush any cached data to disk. Upon a cold boot, the card may be reset and the write-back data in your cache card's memory is lost. Others have reported systems that take a long time to go through the memory check, and/or SCSI BIOSes that take longer to initialize on a cold boot as a good reason to use warm reboots.

The `reserve=' Argument

This is used to *protect* I/O port regions from probes. The form of the command is:

```
reserve=iobase,extent[,iobase,extent]...
```

In some machines it may be necessary to prevent device drivers from checking for devices (auto-probing) in a specific region. This may be because of poorly designed hardware that causes the boot to *freeze* (such as some ethercards), hardware that is mistakenly identified, hardware whose state is changed by an earlier probe, or merely hardware you don't want the kernel to initialize.

The `reserve` boot-time argument addresses this problem by specifying an I/O port region that shouldn't be probed. That region is reserved in the kernel's port registration table as if a device has already been found in that region (with the name `reserved`). Note that this mechanism shouldn't be necessary on most machines. Only when there is a problem or special case would it be necessary to use this.

The I/O ports in the specified region are protected against device probes that do a `check_region()` prior to probing blindly into a region of I/O space. This was put in to be used when some driver was hanging on a NE2000, or misidentifying some other device as its own. A correct device driver shouldn't probe a reserved region, unless another boot argument explicitly specifies that it do so. This implies that `reserve` will most often be used with some other boot argument. Hence if you specify a `reserve` region to protect a specific device, you must generally specify an explicit probe for that device. Most drivers ignore the port registration table if they are

given an explicit address.

For example, the boot line

```
reserve=0x300,32  blah=0x300
```

keeps all device drivers except the driver for `blah' from probing `0x300-0x31f`.

As usual with boot-time specifiers there is an 11 parameter limit, thus you can only specify 5 reserved regions per `reserve` keyword. Multiple `reserve` specifiers will work if you have an unusually complicated request.

The `vga=' Argument

Note that this is not really a boot argument. It is an option that is interpreted by LILO and not by the kernel like all the other boot arguments are. However its use has become so common that it deserves a mention here. It can also be set via using `rdev -v` or equivalently `vidmode` on the vmlinuz file. This allows the setup code to use the video BIOS to change the default display mode before actually booting the Linux kernel. Typical modes are 80x50, 132x44 and so on. The best way to use this option is to start with `vga=ask` which will prompt you with a list of various modes that you can use with your video adapter before booting the kernel. Once you have the number from the above list that you want to use, you can later put it in place of the `ask'. For more information, please see the file `linux/Documentation/svga.txt` that comes with all recent kernel versions.

Note that newer kernels (v2.1 and up) have the setup code that changes the video mode as an option, listed as `Video mode selection support` so you need to enable this option if you want to use this feature.

4. Boot Arguments to Control PCI Bus Behaviour (`pci=')

The `pci=' argument (not avail. in v2.0 kernels) can be used to change the behaviour of PCI bus device probing and device behaviour. Firstly the file `linux/drivers/pci/pci.c` checks for architecture independent `pci=` options. The remaining allowed arguments are handled in `linux/arch/???/kernel/bios32.c` and are listed below for ???=i386.

4.1 The `pci=bios' and `pci=nobios' Arguments

These are used to set/clear the flag indicating that the PCI probing is to take place via the PCI BIOS. The default is to use the BIOS.

4.2 The `pci=conf1' and `pci=conf2' Arguments

If PCI direct mode is enabled, the use of these enables either configuration Type 1 or Type 2. These implicitly clear the PCI BIOS probe flag (i.e. `pci=nobios') too.

4.3 The `pci=io=' Argument

If you get a message like `PCI: Unassigned IO` space for.../ then you may need to supply an I/O value with this option. From the source:

``Several BIOS'es forget to assign addresses to I/O ranges. We try to fix it here, expecting there are free addresses starting with `0x5800`. Ugly, but until we come with better resource management, it's the only simple solution."

4.4 The `pci=nopeer' Argument

This disables the default peer bridge fixup, which according to the source does the following:

``In case there are peer host bridges, scan bus behind each of them. Although several sources claim that the host bridges should have header type 1 and be assigned a bus number as for PCI2PCI bridges, the reality doesn't pass this test and the bus number is usually set by BIOS to the first free value."

TOC

4.5 The `pci=nosort' Argument

Using this argument instructs the kernel to not sort the PCI devices during the probing phase.

4.6 The `pci=off' Argument

Using this option disables all PCI bus probing. Any device drivers that make use of PCI functions to find and initialize hardware will most likely fail to work.

4.7 The `pci=reverse' Argument

This option will reverse the ordering of the PCI devices on that PCI bus.

---

5. Boot Arguments for Video Frame Buffer Drivers

The `video=' argument (not avail. in v2.0 kernels) is used when the frame buffer device abstraction layer is built into the kernel. If that sounds complicated, well it isn't really too bad. It basically means that instead of having a different video program (the X11R6 server) for each brand of video card (e.g. XF86_S3, XF86_SVGA, ...), the kernel would have a built in driver available for each video card and export a single interface for the video program so that only one X11R6 server (XF86_FBDev) would be required. This is similar to how networking is now - the kernel has drivers available for each brand of network card and exports a single network interface so that just one version of a network program (like Netscape) will work for all systems, regardless of the underlying brand of network card.

The typical format of this argument is `video=name:option1,option2,...` where `name` is the name of a generic option or of a frame buffer driver. The `video=` option is passed from `linux/init/main.c` into `linux/drivers/video/fbmem.c` for further processing. Here it is checked for some generic options before trying to match to a known driver name. Once a driver name match is made, the comma separated option list is then passed into that particular driver for final processing. The list of valid driver names can be found by reading down the `fb_drivers` array in the file `fbmem.c` mentioned above.

Information on the options that each driver supports will eventually be found in `linux/Documentation/fb/` but currently (v2.2) only a few are described there. Unfortunately the number of video drivers and the number of options for each one is content for another document itself and hence too much to list here.

If there is no Documentation file for your card, you will have to get the option information directly from the driver. Go to `linux/drivers/video/` and look in the appropriate `???fb.c` file (the ??? will be based on the card name). In there, search for a function with `_setup` in its name and you should see what options the driver tries to match, such as `font` or `mode` or...

5.1 The `video=map:...' Argument

This option is used to set/override the console to frame buffer device mapping. A comma separated list of numbers sets the mapping, with the value of option N taken to be the frame buffer device number for console N.

5.2 The `video=scrollback:...' Argument

A number after the colon will set the size of memory allocated for the scrollback buffer. (Use Shift and Page Up or Page Down keys to scroll.) A suffix of `k' or `K' after the number will indicate that the number is to be interpreted as kilobytes instead of bytes.

5.3 The `video=vc:...' Argument

A number, or a range of numbers (e.g. `video=vc:2-5`) will specify the first, or the first and last frame buffer virtual console(s). The use of this option also has the effect of setting the frame buffer console to *not* be the default console.

6. Boot Arguments for SCSI Peripherals.

This section contains the descriptions of the boot args that are used for passing information about the installed SCSI host adapters, and SCSI devices.

6.1 Arguments for Mid-level Drivers

The mid level drivers handle things like disks, CD-ROMs and tapes without getting into host adapter specifics.

Maximum Probed LUNs (`max_scsi_luns=')

Each SCSI device can have a number of `sub-devices' contained within itself. The most common example is any of the SCSI CD-ROMs that handle more than one disk at a time. Each CD is addressed as a `Logical Unit Number' (LUN) of that particular device. But most devices, such as hard disks, tape drives and such are only one device, and will be assigned to LUN zero.

The problem arises with single LUN devices with bad firmware. Some poorly designed SCSI devices (old and unfortunately new) can not handle being probed for LUNs not equal to zero. They will respond by locking up, and possibly taking the whole SCSI bus down with them.

The kernel has a configuration option that allows you to set the maximum number of probed LUNs. The default is to only probe LUN zero, to avoid the problem described above.

To specify the number of probed LUNs at boot, one enters `max_scsi_luns=n' as a boot arg, where n is a number between one and eight. To avoid problems as described above, one would use n=1 to avoid upsetting such broken devices

SCSI Logging (`scsi_logging=')

Supplying a non-zero value to this boot argument turns on logging of all SCSI events (error, scan, mlqueue, mlcomplete, llqueue, llcomplete, hlqueue, hlcomplete). Note that better control of which events are logged can be obtained via the `/proc/scsi/scsi` interface if you aren't interested in the events that take place at boot before the `/proc/` filesystem is accessible.

Parameters for the SCSI Tape Driver (`st=')

Some boot time configuration of the SCSI tape driver can be achieved by using the following:

```
st=buf_size[,write_threshold[,max_bufs]]
```

The first two numbers are specified in units of kB. The default `buf_size` is 32kB, and the maximum size that can be specified is a ridiculous 16384kB. The `write_threshold` is the value at which the buffer is committed to tape, with a default value of 30kB. The maximum number of buffers varies with the number of drives detected, and has a default of two. An example usage would be:

```
st=32,30,2
```

Full details can be found in the `README.st` file that is in the `scsi` directory of the kernel source tree.

6.2 Arguments for SCSI Host Adapters

General notation for this section:

`iobase` -- the first I/O port that the SCSI host occupies. These are specified in hexidecimal notation, and usually lie in the range from `0x200` to `0x3ff`.

`irq` -- the hardware interrupt that the card is configured to use. Valid values will be dependent on the card in question, but will usually be 5, 7, 9, 10, 11, 12, and 15. The other values are usually used for common peripherals like IDE hard disks, floppies, serial ports, etc.

`dma` -- the DMA (Direct Memory Access) channel that the card uses. Typically only applies to bus-mastering cards. PCI and VLB cards are native bus-masters, and do not require and ISA DMA channel.

`scsi-id` -- the ID that the host adapter uses to identify itself on the SCSI bus. Only some host adapters allow you to change this value, as most have it permanently specified internally. The usual default value is seven, but the Seagate and Future Domain TMC-950 boards use six.

`parity` -- whether the SCSI host adapter expects the attached devices to supply a parity value with all information exchanges. Specifying a one indicates parity checking is enabled, and a zero disables parity checking. Again, not all adapters will support selection of parity behaviour as a boot argument.

Adaptec aha151x, aha152x, aic6260, aic6360, SB16-SCSI (`aha152x=')

The aha numbers refer to cards and the aic numbers refer to the actual SCSI chip on these type of cards, including the Soundblaster-16 SCSI.

The probe code for these SCSI hosts looks for an installed BIOS, and if none is present, the probe will not find your card. Then you will have to use a boot argument of the form:

```
aha152x=iobase[,irq[,scsi-id[,reconnect[,parity]]]]
```

Note that if the driver was compiled with debugging enabled, a sixth value can be specified to set the

TOC

debug level.

All the parameters are as described at the top of this section, and the `reconnect` value will allow device disconnect/reconnect if a non-zero value is used. An example usage is as follows:

```
aha152x=0x340,11,7,1
```

Note that the parameters must be specified in order, meaning that if you want to specify a parity setting, then you will have to specify an iobase, irq, scsi-id and reconnect value as well.

Adaptec aha154x (`aha1542=')

These are the aha154x series cards. The aha1542 series cards have an i82077 floppy controller onboard, while the aha1540 series cards do not. These are busmastering cards, and have parameters to set the ``fairness'' that is used to share the bus with other devices. The boot argument looks like the following.

```
aha1542=iobase[,buson,busoff[,dmaspeed]]
```

Valid `iobase` values are usually one of: `0x130, 0x134, 0x230, 0x234, 0x330, 0x334`. Clone cards may permit other values.

The `buson, busoff` values refer to the number of microseconds that the card dominates the ISA bus. The defaults are 11us on, and 4us off, so that other cards (such as an ISA LANCE Ethernet card) have a chance to get access to the ISA bus.

The `dmaspeed` value refers to the rate (in MB/s) at which the DMA (Direct Memory Access) transfers proceed at. The default is 5MB/s. Newer revision cards allow you to select this value as part of the soft-configuration, older cards use jumpers. You can use values up to 10MB/s assuming that your motherboard is capable of handling it. Experiment with caution if using values over 5MB/s.

Adaptec aha274x, aha284x, aic7xxx (`aic7xxx=')

These boards can accept an argument of the form:

```
aic7xxx=extended,no_reset
```

The `extended` value, if non-zero, indicates that extended translation for large disks is enabled. The `no_reset` value, if non-zero, tells the driver not to reset the SCSI bus when setting up the host adaptor at boot.

TOC

AdvanSys SCSI Host Adaptors (`advansys=')

The AdvanSys driver can accept up to four i/o addresses that will be probed for an AdvanSys SCSI card. Note that these values (if used) do not effect EISA or PCI probing in any way. They are only used for probing ISA and VLB cards. In addition, if the driver has been compiled with debugging enabled, the level of debugging output can be set by adding an `0xdeb[0-f]` parameter. The `0-f` allows setting the level of the debugging messages to any of 16 levels of verbosity.

Always IN2000 Host Adaptor (`in2000=')

Unlike other SCSI host boot arguments, the IN2000 driver uses ASCII string prefixes for most of its integer arguments. Here is a list of the supported arguments:

ioport:addr -- Where addr is IO address of a (usually ROM-less) card.

noreset -- No optional args. Prevents SCSI bus reset at boot time.

nosync:x -- x is a bitmask where the 1st 7 bits correspond with the 7 possible SCSI devices (bit 0 for device #0, etc). Set a bit to PREVENT sync negotiation on that device. The driver default is sync DISABLED on all devices.

period:ns -- ns is the minimum # of nanoseconds in a SCSI data transfer period. Default is 500; acceptable values are 250 to 1000.

disconnect:x -- x = 0 to never allow disconnects, 2 to always allow them. x = 1 does 'adaptive' disconnects, which is the default and generally the best choice.

debug:x If `DEBUGGING_ON' is defined, x is a bitmask that causes various types of debug output to printed - see the DB_xxx defines in in2000.h

proc:x -- If `PROC_INTERFACE' is defined, x is a bitmask that determines how the /proc interface works and what it does - see the PR_xxx defines in in2000.h

Some example usages are listed below:

```
in2000=ioport:0x220,noreset
in2000=period:250,disconnect:2,nosync:0x03
in2000=debug:0x1e
in2000=proc:3
```

AMD AM53C974 based hardware (`AM53C974=')

Unlike other drivers, this one does not use boot parameters to communicate i/o, IRQ or DMA channels. (Since the AM53C974 is a PCI device, there shouldn't be a need to do so.) Instead, the parameters are used to communicate the transfer modes and rates that are to be used between the host and the target device. This is best described with an example:

```
AM53C974=7,2,8,15
```

This would be interpreted as follows: `For communication between the controller with SCSI-ID 7 and the device with SCSI-ID 2, a transfer rate of 8MHz in synchronous mode with max. 15 bytes offset should be negotiated.' More details can be found in the file `linux/drivers/scsi/README.AM53C974`

BusLogic SCSI Hosts with v1.2 kernels (`buslogic=')

In older kernels, the buslogic driver accepts only one parameter, that being the I/O base. It expects that to be one of the following valid values: `0x130, 0x134, 0x230, 0x234, 0x330, 0x334`.

BusLogic SCSI Hosts with v2.x kernels (`BusLogic=')

With v2.x kernels, the BusLogic driver accepts many parameters. (Note the case in the above; upper case B and L!!!). There are simply too many to list here. A complete description is tucked away in the middle of the driver `linux/drivers/scsi/BusLogic.c` and searching for the string `BusLogic=` will put you right on it.

EATA SCSI Cards (`eata=')

As of late v2.0 kernels, the EATA drivers will accept a boot argument to specify the i/o base(s) to be probed. It is of the form:

```
eata=iobase1[,iobase2][,iobase3]...[,iobaseN]
```

The driver will probe the addresses in the order that they are listed.

Future Domain TMC-8xx, TMC-950 (`tmc8xx=')

The probe code for these SCSI hosts looks for an installed BIOS, and if none is present, the probe will not find your card. Or, if the signature string of your BIOS is not recognized then it will also not be found. In either case, you will then have to use a boot argument of the form:

```
tmc8xx=mem_base,irq
```

The `mem_base` value is the value of the memory mapped I/O region that the card uses. This will usually be one of the following values: `0xc8000, 0xca000, 0xcc000, 0xce000, 0xdc000, 0xde000`.

Future Domain TMC-16xx, TMC-3260, AHA-2920 (`fdomain=')

The driver detects these cards according to a list of known BIOS ROM signatures. For a full list of known BIOS revisions, please see `linux/drivers/scsi/fdomain.c` as it has a lot of information at the top of that file. If your BIOS is not known to the driver, you can use an override of the form:

```
fdomain=iobase,irq[,scsi_id]
```

IOMEGA Parallel Port / ZIP drive (`ppa=')

This driver is for the IOMEGA Parallel Port SCSI adapter which is embedded into the IOMEGA ZIP drives. It may also work with the original IOMEGA PPA3 device. The boot argument for this driver is of the form:

```
ppa=iobase,speed_high,speed_low,nybble
```

---

with all but iobase being optionally specified values. If you wish to alter any of the three optional parameters, you are advised to read `linux/drivers/scsi/README.ppa` for details of what they control.

NCR5380 based controllers (`ncr5380=')

Depending on your board, the 5380 can be either i/o mapped or memory mapped. (An address below 0x400 usually implies i/o mapping, but PCI and EISA hardware use i/o addresses above 0x3ff.) In either case, you specify the address, the IRQ value and the DMA channel value. An example for an i/o mapped card would be: `ncr5380=0x350,5,3`. If the card doesn't use interrupts, then an IRQ value of 255 (`0xff`) will disable interrupts. An IRQ value of 254 means to autoprobe. More details can be found in the file `linux/drivers/scsi/README.g_NCR5380`

NCR53c400 based controllers (`ncr53c400=')

The generic 53c400 support is done with the same driver as the generic 5380 support mentioned above. The boot argument is identical to the above with the exception that no DMA channel is used by the 53c400.

NCR53c406a based controllers (`ncr53c406a=')

This driver uses a boot argument of the form:

```
ncr53c406a=PORTBASE,IRQ,FASTPIO
```

where the IRQ and FASTPIO parameters are optional. An interrupt value of zero disables the use of interrupts. Using a value of one for the FASTPIO parameter enables the use of `insl` and `outsl` instructions instead of the single-byte `inb` and `outb` instructions. The driver can also use DMA as a compile-time option.

Pro Audio Spectrum (`pas16=')

The PAS16 uses a NCR5380 SCSI chip, and newer models support jumper-less configuration. The boot argument is of the form:

```
pas16=iobase,irq
```

The only difference is that you can specify an IRQ value of 255, which will tell the driver to work without using interrupts, albeit at a performance loss. The `iobase` is usually `0x388`.

Seagate ST-0x (`st0x=')

The probe code for these SCSI hosts looks for an installed BIOS, and if none is present, the probe will not find your card. Or, if the signature string of your BIOS is not recognized then it will also not be found. In either case, you will then have to use a boot argument of the form:

```
st0x=mem_base,irq
```

The `mem_base` value is the value of the memory mapped I/O region that the card uses. This will usually be one of the following values: `0xc8000, 0xca000, 0xcc000, 0xce000, 0xdc000, 0xde000`.

Trantor T128 (`t128=')

These cards are also based on the NCR5380 chip, and accept the following options:

```
t128=mem_base,irq
```

The valid values for `mem_base` are as follows: `0xcc000, 0xc8000, 0xdc000, 0xd8000`.

Ultrastor SCSI cards (`u14-34f=')

Note that there appears to be two independent drivers for this card, namely `CONFIG_SCSI_U14_34F` that uses `u14-34f.c` and `CONFIG_SCSI_ULTRASTOR` that uses `ultrastor.c`. It is the u14-34f one that (as of late v2.0 kernels) accepts a boot argument of the form:

```
u14-34f=iobase1[,iobase2][,iobase3]...[,iobaseN]
```

The driver will probe the addresses in the order that they are listed.

Western Digital WD7000 cards (`wd7000=')

The driver probe for the wd7000 looks for a known BIOS ROM string and knows about a few standard configuration settings. If it doesn't come up with the correct values for your card, or you have an unrecognized BIOS version, you can use a boot argument of the form:

```
wd7000=irq,dma,iobase
```

6.3 SCSI Host Adapters that don't Accept Boot Args

At present, the following SCSI cards do not make use of any boot-time parameters. In some cases, you can *hard-wire* values by directly editing the driver itself, if required.

```
Adaptec aha1740 (EISA probing),
NCR53c7xx,8xx (PCI, both drivers)
Qlogic Fast (0x230, 0x330)
Qlogic ISP (PCI)
```

7. Hard Disks

This section lists all the boot args associated with standard MFM/RLL, ST-506, XT, and IDE disk drive devices. Note that both the IDE and the generic ST-506 HD driver both accept the `hd=' option.

7.1 IDE Disk/CD-ROM Driver Parameters

The IDE driver accepts a number of parameters, which range from disk geometry specifications, to support for advanced or broken controller chips. The following is a summary of all the possible boot arguments. For full details, you *really* should consult the file `ide.txt` in the `linux/Documentation` directory, from which this summary was extracted.

```
"hdx="  is recognized for all "x" from "a" to "h", such as "hdc".
"idex=" is recognized for all "x" from "0" to "3", such as "ide1".

"hdx=noprobe"          : drive may be present, but do not probe for it
"hdx=none"             : drive is NOT present, ignore cmos and do not probe
"hdx=nowerr"           : ignore the WRERR_STAT bit on this drive
"hdx=cdrom"            : drive is present, and is a cdrom drive
"hdx=cyl,head,sect"    : disk drive is present, with specified geometry
"hdx=autotune"         : driver will attempt to tune interface speed
                                to the fastest PIO mode supported,
                                if possible for this drive only.
                                Not fully supported by all chipset types,
                                and quite likely to cause trouble with
                                older/odd IDE drives.

"idex=noprobe"         : do not attempt to access/use this interface
"idex=base"            : probe for an interface at the addr specified,
                                where "base" is usually 0x1f0 or 0x170
                                and "ctl" is assumed to be "base"+0x206
"idex=base,ctl"        : specify both base and ctl
"idex=base,ctl,irq"    : specify base, ctl, and irq number
"idex=autotune"        : driver will attempt to tune interface speed
                                to the fastest PIO mode supported,
                                for all drives on this interface.
                                Not fully supported by all chipset types,
                                and quite likely to cause trouble with
                                older/odd IDE drives.
"idex=noautotune"      : driver will NOT attempt to tune interface speed
                                This is the default for most chipsets,
                                except the cmd640.
```

```
"idex=serialize"        : do not overlap operations on idex and ide(x^1)
```

TOC

The following are valid ONLY on ide0, and the defaults for the base,ctl ports must not be altered.

```
"ide0=dtc2278"          : probe/support DTC2278 interface
"ide0=ht6560b"          : probe/support HT6560B interface
"ide0=cmd640_vlb"       : *REQUIRED* for VLB cards with the CMD640 chip
                          (not for PCI -- automatically detected)
"ide0=qd6580"           : probe/support qd6580 interface
"ide0=ali14xx"          : probe/support ali14xx chipsets (ALI M1439/M1445)
"ide0=umc8672"          : probe/support umc8672 chipsets
```

Everything else is rejected with a "BAD OPTION" message.

7.2 Standard ST-506 Disk Driver Options (`hd=')

The standard disk driver can accept geometry arguments for the disks similar to the IDE driver. Note however that it only expects three values (C/H/S) -- any more or any less and it will silently ignore you. Also, it only accepts `hd=' as an argument, i.e. `hda=', `hdb=' and so on are not valid here. The format is as follows:

```
hd=cyls,heads,sects
```

If there are two disks installed, the above is repeated with the geometry parameters of the second disk.

7.3 XT Disk Driver Options (`xd=')

If you are unfortunate enough to be using one of these old 8 bit cards that move data at a whopping 125kB/s then here is the scoop. The probe code for these cards looks for an installed BIOS, and if none is present, the probe will not find your card. Or, if the signature string of your BIOS is not recognized then it will also not be found. In either case, you will then have to use a boot argument of the form:

```
xd=type,irq,iobase,dma_chan
```

The `type` value specifies the particular manufacturer of the card, and are as follows: 0=generic; 1=DTC; 2,3,4=Western Digital, 5,6,7=Seagate; 8=OMTI. The only difference between multiple types from the same manufacturer is the BIOS string used for detection, which is not used if the type is specified.

The `xd_setup()` function does no checking on the values, and assumes that you entered all four values. Don't disappoint it. Here is an example usage for a WD1002 controller with the BIOS disabled/removed, using the `default' XT controller parameters:

```
xd=2,5,0x320,3
```

8. CD-ROMs (Non-SCSI/ATAPI/IDE)

This section lists all the possible boot args pertaining to CD-ROM devices. Note that this does not include SCSI or IDE/ATAPI CD-ROMs. See the appropriate section(s) for those types of CD-ROMs.

Note that most of these CD-ROMs have documentation files that you *should* read, and they are all in one handy place: `linux/Documentation/cdrom`.

8.1 The Aztech Interface (`aztcd=')

The syntax for this type of card is:

```
aztcd=iobase[,magic_number]
```

If you set the `magic_number` to `0x79` then the driver will try and run anyway in the event of an unknown firmware version. All other values are ignored.

8.2 The CDU-31A and CDU-33A Sony Interface (`cdu31a=')

This CD-ROM interface is found on some of the Pro Audio Spectrum sound cards, and other Sony supplied interface cards. The syntax is as follows:

```
cdu31a=iobase,[irq[,is_pas_card]]
```

Specifying an IRQ value of zero tells the driver that hardware interrupts aren't supported (as on some PAS cards). If your card supports interrupts, you should use them as it cuts down on the CPU usage of the driver.

The `is_pas_card' should be entered as `PAS' if using a Pro Audio Spectrum card, and otherwise it should not be specified at all.

8.3 The CDU-535 Sony Interface (`sonycd535=')

TOC

The syntax for this CD-ROM interface is:

```
sonycd535=iobase[,irq]
```

A zero can be used for the I/O base as a `placeholder' if one wishes to specify an IRQ value.

8.4 The GoldStar Interface (`gscd=')

The syntax for this CD-ROM interface is:

```
gscd=iobase
```

8.5 The ISP16 Interface (`isp16=')

The syntax for this CD-ROM interface is:

```
isp16=[port[,irq[,dma]]][[,]drive_type]
```

Using a zero for `irq` or `dma` means that they are not used. The allowable values for `drive_type` are `noisp16, Sanyo, Panasonic, Sony,` and `Mitsumi`. Using `noisp16` disables the driver altogether.

8.6 The Mitsumi Standard Interface (`mcd=')

The syntax for this CD-ROM interface is:

```
mcd=iobase,[irq[,wait_value]]
```

The `wait_value` is used as an internal timeout value for people who are having problems with their drive, and may or may not be implemented depending on a compile time `DEFINE`.

TOC

8.7 The Mitsumi XA/MultiSession Interface (`mcdx=')

At present this `experimental' driver has a setup function, but no parameters are implemented yet (as of 1.3.15). This is for the same hardware as above, but the driver has extended features.

8.8 The Optics Storage Interface (`optcd=')

The syntax for this type of card is:

```
optcd=iobase
```

8.9 The Phillips CM206 Interface (`cm206=')

The syntax for this type of card is:

```
cm206=[iobase][,irq]
```

The driver assumes numbers between 3 and 11 are IRQ values, and numbers between `0x300` and `0x370` are I/O ports, so you can specify one, or both numbers, in any order. It also accepts `cm206=auto' to enable autoprobing.

8.10 The Sanyo Interface (`sjcd=')

The syntax for this type of card is:

```
sjcd=iobase[,irq[,dma_channel]]
```

8.11 The SoundBlaster Pro Interface (`sbpcd=')

The syntax for this type of card is:

```
sbpcd=iobase,type
```

TOC

where `type` is one of the following (case sensitive) strings: `SoundBlaster', `LaserMate', or `SPEA'. I/O base is that of the CD-ROM interface, and *not* that of the sound portion of the card.

---

9. Serial and ISDN Drivers

9.1 The ICN ISDN driver (`icn=')

This ISDN driver expects a boot argument of the form:

```
icn=iobase,membase,icn_id1,icn_id2
```

where `iobase` is the i/o port address of the card, `membase` is the shared memory base address of the card, and the two `icn_id` are unique ASCII string identifiers.

9.2 The PCBIT ISDN driver (`pcbit=')

This boot argument takes integer pair arguments of the form:

```
pcbit=membase1,irq1[,membase2,irq2]
```

where `membaseN` is the shared memory base of the N'th card, and `irqN` is the interrupt setting of the N'th card. The default is IRQ 5 and membase `0xD0000`.

9.3 The Teles ISDN driver (`teles=')

This ISDN driver expects a boot argument of the form:

```
teles=iobase,irq,membase,protocol,teles_id
```

where `iobase` is the i/o port address of the card, `membase` is the shared memory base address of the card, `irq` is the interrupt channel the card uses, and `teles_id` is the unique ASCII string identifier.

9.4 The DigiBoard Driver (`digi=')

The DigiBoard driver accepts a string of six comma separated identifiers or integers. The 6 values in order are:

```
Enable/Disable this card
Type of card: PC/Xi(0), PC/Xe(1), PC/Xeve(2), PC/Xem(3)
Enable/Disable alternate pin arrangement
Number of ports on this card
I/O Port where card is configured (in HEX if using string identifiers)
```

```
Base of memory window (in HEX if using string identifiers)
```

An example of a correct boot prompt argument (in both identifier and integer form) is:

```
digi=E,PC/Xi,D,16,200,D0000
digi=1,0,0,16,512,851968
```

Note that the driver defaults to an i/o of `0x200` and a shared memory base of `0xD0000` in the absence of a `digi=` boot argument. There is no autoprobing performed. More details can be found in the file `linux/Documentation/digiboard.txt`.

9.5 The RISCom/8 Multiport Serial Driver (`riscom8=')

Up to four boards can be supported by supplying four unique i/o port values for each individual board installed. Other details can be found in the file `linux/Documentation/riscom8.txt`.

9.6 The Baycom Serial/Parallel Radio Modem (`baycom=')

The format of the boot argument for these devices is:

```
baycom=modem,io,irq,options[,modem,io,irq,options]
```

Using modem=1 means you have the ser12 device, modem=2 means you have the par96 device. Using options=0 means use hardware DCD, and options=1 means use software DCD. The `io` and `irq` are the i/o port base and interrupt settings as usual. There is more details in the file `README.baycom` which is currently in the `/linux/drivers/char/` directory.

---

10. Other Hardware Devices

Any other devices that didn't fit into any of the above categories got lumped together here.

10.1 Ethernet Devices (`ether=')

Different drivers make use of different parameters, but they all at least share having an IRQ, an I/O port base value, and a name. In its most generic form, it looks something like this:

```
ether=irq,iobase[,param_1[,param_2,...param_8]]],name
```

The first non-numeric argument is taken as the name. The `param_n` values (if applicable) usually have different meanings for each different card/driver. Typical `param_n` values are used to specify things like shared memory address, interface selection, DMA channel and the like.

The most common use of this parameter is to force probing for a second ethercard, as the default is to only probe for one. This can be accomplished with a simple:

```
ether=0,0,eth1
```

Note that the values of zero for the IRQ and I/O base in the above example tell the driver(s) to autoprobe.

IMPORTANT NOTE TO MODULE USERS: The above will *not* force a probe for a second card if you are using the driver(s) as run time loadable modules (instead of having them complied into the kernel). Most Linux distributions use a bare bones kernel combined with a large selection of modular drivers. The `ether=` only applies to drivers compiled directly into the kernel.

The Ethernet-HowTo has complete and extensive documentation on using multiple cards and on the card/driver specific implementation of the `param_n` values where used. Interested readers should refer to the section in that document on their particular card for more complete information. Ethernet-HowTo

10.2 The Floppy Disk Driver (`floppy=')

There are many floppy driver options, and they are all listed in `README.fd` in `linux/drivers/block`. There are too many options in that file to list here. Instead, only those options that may be required to get a Linux install to proceed on less than normal hardware are reprinted here.

`floppy=0,daring` Tells the floppy driver that your floppy controller should be used with caution (disables all daring operations).

`floppy=thinkpad` Tells the floppy driver that you have a Thinkpad. Thinkpads use an inverted convention for the disk change line.

`floppy=nodma` Tells the floppy driver not to use DMA for data transfers. This is needed on HP Omnibooks, which don't have a workable DMA channel for the floppy driver. This option is also useful if you frequently get "Unable to allocate DMA memory" messages. Use of `nodma' is not recommended if you have a FDC without a FIFO (8272A or 82072). 82072A and later are OK). The FDC model is reported at boot. You also need at least a 486 to use nodma.

`floppy=nofifo` Disables the FIFO entirely. This is needed if you get `Bus master arbitration error' messages from your Ethernet card (or from other devices) while accessing the floppy.

`floppy=broken_dcl` Don't use the disk change line, but assume that the disk was changed whenever the device node is reopened. Needed on some boxes where the disk change line is broken or unsupported. This should be regarded as a stopgap measure, indeed it makes floppy operation less efficient due to unneeded cache flushings, and slightly more unreliable. Please verify your cable connection and jumper settings if you have any DCL problems. However, some older drives, and also some Laptops are known not to have a DCL.

`floppy=debug` Print (additional) debugging messages.

`floppy=messages` Print informational messages for some operations (disk change notifications, warnings about over and underruns, and about autodetection).

10.3 The Sound Driver (`sound=')

The sound driver can also accept boot args to override the compiled in values. This is not recommended, as it is rather complex and the documentation for it in the kernel mysteriously vanished (a hint). You are better off to use sound as a module, or compile in your own values.

If you choose to use it regardless, then processing of the argument takes place in the file `dev_table.c` in `linux/drivers/sound`. It accepts a boot arg of the form:

```
sound=device1[,device2[,device3...[,device11]]]
```

---

where each `deviceN` value is of the following format `0xDTaaaId` and the bytes are used as follows:

D - second DMA channel (zero if not applicable)

T - device type: 1=FM, 2=SB, 3=PAS, 4=GUS, 5=MPU401, 6=SB16, 7=SB16-MIDI,... The listing of soundcard types up to 26 (don't forget to convert back to hex for command line use) are listed in the file `linux/include/linux/soundcard.h` and 27 to 999 (newer models) can be found in the file `linux/drivers/sound/dev_table.h`.

aaa - I/O address in hex.

I - interrupt line in hex (i.e 10=a, 11=b, ...)

d - First DMA channel.

As you can see it gets pretty messy, and you *really* are better off to use a modular driver or compile in your own personal values as recommended. Using a boot arg of `sound=0' will disable the sound driver entirely.

10.4 The Bus Mouse Driver (`bmouse=')

The busmouse driver only accepts one parameter, that being the hardware IRQ value to be used.

10.5 The MS Bus Mouse Driver (`msmouse=')

The MS mouse driver only accepts one parameter, that being the hardware IRQ value to be used.

10.6 The Printer Driver (`lp=')

With this boot argument you can tell the printer driver what ports to use and what ports *not* to use. The latter comes in handy if you don't want the printer driver to claim all available parallel ports, so that other drivers (e.g. PLIP, PPA) can use them instead.

The format of the argument is multiple i/o, IRQ pairs. For example, `lp=0x3bc,0,0x378,7` would use the port at `0x3bc` in IRQ-less (polling) mode, and use IRQ 7 for the port at `0x378`. The port at `0x278` (if any) would not be probed, since autoprobing only takes place in the absence of a `lp=` argument. To disable the printer driver entirely, one can use `lp=0`.

---

11. Copying, Translations, Closing, etc.

Hey, you made it to the end! (Phew...) Now just the legal stuff.

11.1 Copyright and Disclaimer

This document is Copyright (c) 1995-1999 by Paul Gortmaker. Copying and redistribution is allowed under the conditions as outlined in the Linux Documentation Project Copyright, available from where you obtained this document, OR as outlined in the GNU General Public License, version 2 (see linux/COPYING).

This document is *not* gospel. However, it is probably the most up to date info that you will be able to find. Nobody is responsible for what happens to your hardware but yourself. If your stuff goes up in smoke, or anything else bad happens, we take no responsibility. ie. THE AUTHOR IS NOT RESPONSIBLE FOR ANY DAMAGES INCURRED DUE TO ACTIONS TAKEN BASED ON THE INFORMATION INCLUDED IN THIS DOCUMENT.

A hint to people considering doing a translation. First, translate the SGML source (available via FTP from the HowTo main site) so that you can then generate other output formats. Be sure to keep a copy of the original English SGML source that you translated from! When an updated HowTo is released, get the new SGML source for that version, and then a simple `diff -u old.sgml new.sgml` will show you exactly what has changed so that you can easily incorporate those changes into your translated SMGL source without having to re-read or re-translate everything.

If you are intending to incorporate this document into a published work, please make contact (via e-mail) so that you can be supplied with the most up to date information available. In the past, out of date versions of the Linux HowTo documents have been published, which caused the developers undue grief from being plagued with questions that were already answered in the up to date versions.

11.2 Closing

If you have found any glaring typos, or outdated info in this document, please let me know. It is easy to overlook stuff, as the kernel (and the number of drivers) is huge compared to what it was when I started this.

Thanks,

Paul Gortmaker, `p_gortmaker@yahoo.com`

The Linux Bootdisk HOWTO

Tom Fawcett (fawcett@croftj.net)

3.4, May 1999

This document describes how to design and build your own boot/root diskettes for Linux. These disks could be used as rescue disks or to test new system components. If you haven't read the Linux FAQ and related documents, such as the Linux Installation HOWTO and the Linux Install Guide, you should not be trying to build boot diskettes. If you just want a rescue disk to have for emergencies, see Appendix Pre-made bootdisks.

1. Preface.

- 1.1 Version notes.
- 1.2 Feedback and credits.
- 1.3 Distribution policy.

2. Introduction.

3. Bootdisks and the boot process.

- 3.1 The boot process.
- 3.2 Disk types.

4. Building a root filesystem.

- 4.1 Overview.
- 4.2 Creating the filesystem.
- 4.3 Populating the filesystem.
- 4.4 Providing for PAM and NSS.
- 4.5 Modules.
- 4.6 Some final details.

- 4.7 Wrapping it up.

5. Choosing a kernel.

6. Putting them together: Making the diskette(s).

- 6.1 Transferring the kernel with LILO
- 6.2 Transferring the kernel without LILO.
- 6.3 Setting the ramdisk word.
- 6.4 Transferring the root filesystem.

7. Troubleshooting, or The Agony of Defeat.

8. Miscellaneous topics.

- 8.1 Reducing root filesystem size.
- 8.2 Non-ramdisk root filesystems.
- 8.3 Building a utility disk.

9. How the pros do it.

10. Frequently asked question (FAQ) list.

Appendix

11. Resources and pointers

- 11.1 Pre-made bootdisks.
- 11.2 Rescue packages.
- 11.3 Graham Chapman's shell scripts
- 11.4 LILO -- the Linux loader.
- 11.5 Linux FAQ and HOWTOs.
- 11.6 Ramdisk usage.
- 11.7 The Linux boot process.

12. LILO boot error codes.

13. Sample rootdisk directory listings.

14. Sample utility disk directory listing.

1. Preface.

Note: This document may be outdated. If the date on the title page is more than six months ago, please check the Linux Documentation Project homepage http://metalab.unc.edu/LDP/HOWTO/Bootdisk-HOWTO.html to see if a more recent version exists.

Although this document should be legible in its text form, it looks *much* better in Postscript (`.ps`) or HTML because of the typographical notation used. We encourage you to select one of these forms. The Info version, as of this writing, ends up so damaged as to be unusable.

1.1 Version notes.

Graham Chapman (`grahamc@zeta.org.au`) wrote the original Bootdisk-HOWTO and he supported it through version 3.1. Tom Fawcett (`fawcett@croftj.net`) added a lot of material for kernel 2.0, and he is the document's maintainer as of version 3.2. Much of Chapman's original content remains.

This document is intended for **Linux kernel 2.0 and later**. If you have an older kernel (1.2.xx or before), please consult previous versions of the Bootdisk-HOWTO archived on Graham Chapman's homepage.

This information is intended for Linux on the **Intel** platform. Much of this information may be applicable to Linux on other processors, but we have no first-hand experience or information about this. If anyone has experience with bootdisks on other platforms, please contact us.

1.2 Feedback and credits.

We welcome any feedback, good or bad, on the content of this document. We have done our best to ensure that the instructions and information herein are accurate and reliable. Please let us know if you find errors or omissions.

We thank the many people who assisted with corrections and suggestions. Their contributions have made it far better than we could ever have done alone.

Send comments, corrections and questions to the author at the email address above. I don't mind trying to answer questions, but please read section Troubleshooting first.

1.3 Distribution policy.

Copyright © 1995,1996,1997,1998,1999 by Tom Fawcett and Graham Chapman. This document may be distributed under the terms set forth in the Linux Documentation Project License at http://metalab.unc.edu/LDP/COPYRIGHT.html. Please contact the authors if you are unable to get the license.

This is free documentation. It is distributed in the hope that it will be useful, but **without any warra**
without even the implied warranty of **merchantability** or **fitness for a particular purpose**.

2. Introduction.

Linux boot disks are useful in a number of situations, such as:

- Testing a new kernel.
- Recovering from a disk failure -- anything from a lost boot sector to a disk head crash.
- Fixing a disabled system. A minor mistake as root can leave your system unusable, and you may have to boot from diskette to fix it.
- Upgrading critical system files, such as `libc.so`.

There are several ways of obtaining boot disks:

- Use one from a distribution such as Slackware. This will at least allow you to boot.
- Use a rescue package to set up disks designed to be used as rescue disks.
- Learn what is required for each of the types of disk to operate, then build your own.

Some people choose the last option so they can do it themselves. That way, if something breaks, they can work out what to do to fix it. Plus it's a great way to learn about how a Linux system works.

This document assumes some basic familiarity with Linux system administration concepts. For example, you should know about directories, filesystems and floppy diskettes. You should know how to use `mount` and `df`. You should know what /etc/passwd and fstab files are for and what they look like. You should know that most of the commands in this HOWTO should be run as root.

Constructing your own bootdisk from scratch can be complicated. If you haven't read the Linux FAQ and related documents, such as the Linux Installation HOWTO and the Linux Installation Guide, you should not be trying to build boot diskettes. If you just need a working bootdisk for emergencies, it is *much* easier to download a prefabricated one. See Appendix Pre-made bootdisks, below, for where to find these.

3. Bootdisks and the boot process.

A bootdisk is basically a miniature, self-contained Linux system on a floppy diskette. It must perform many of the same functions that a complete full-size Linux system performs. Before trying to build one you should understand the basic Linux boot process. We present the basics here, which are sufficient for understanding the rest of this document. Many details and alternative options have been omitted.

3.1 The boot process.

All PC systems start the boot process by executing code in ROM (specifically, the BIOS) to load the sector from sector 0, cylinder 0 of the boot drive. The boot drive is usually the first floppy drive (designated `A:` in DOS and /dev/fd0 in Linux). The BIOS then tries to execute this sector. On most bootable disks, sector 0, cylinder 0 contains either:

- code from a boot loader such as LILO, which locates the kernel, loads it and executes it to start the boot proper.
- the start of an operating system kernel, such as Linux.

If a Linux kernel has been raw-copied to a diskette, the first sector of the disk will be the first sector of the Linux kernel itself. This first sector will continue the boot process by loading the rest of the kernel from the boot device.

Once the kernel is completely loaded, it goes through some basic device initialization. It then tries to load and mount a **root filesystem** from some device. A root filesystem is simply a filesystem that is mounted as ``/''. The kernel has to be told where to look for the root filesystem; if it cannot find a loadable image there, it halts.

In some boot situations -- often when booting from a diskette -- the root filesystem is loaded into a **ramdisk**, which is RAM accessed by the system as if it were a disk. There are two reasons why the system loads to ramdisk. First, RAM is several orders of magnitude faster than a floppy disk, so system operation is fast; and second, the kernel can load a **compressed filesystem** from the floppy and uncompress it onto the ramdisk, allowing many more files to be squeezed onto the diskette.

Once the root filesystem is loaded and mounted, you see a message like:

```
VFS: Mounted root (ext2 filesystem) readonly.
```

At this point the system finds the `init` program on the root filesystem (in `/bin` or `/sbin`) and executes it. `init` reads its configuration file /etc/inittab, looks for a line designated `sysinit`, and executes the named script . The `sysinit` script is usually something like /etc/rc or /etc/init.d/boot. This script is a set of shell commands that set up basic system services, such as:

- Running `fsck` on all the disks,
- Loading necessary kernel modules,

TOC

- Starting swapping,
- Initializing the network,
- Mounting disks mentioned in `fstab`.

This script often invokes various other scripts to do modular initialization. For example, in the common SysVinit structure, the directory /etc/rc.d/ contains a complex structure of subdirectories whose files specify how to enable and shut down most system services. However, on a bootdisk the sysinit script is often very simple.

When the sysinit script finishes control returns to `init`, which then enters the *default runlevel*, specified in `inittab` with the `initdefault` keyword. The runlevel line usually specifies a program like `getty`, which is responsible for handling commununications through the console and ttys. It is the `getty` program which prints the familiar ``login:'' prompt. The `getty` program in turn invokes the `login` program to handle login validation and to set up user sessions.

3.2 Disk types.

Having reviewed the basic boot process, we can now define various kinds of disks involved. We classify disks into four types. The discussion here and throughout this document uses the term ``disk'' to refer to floppy diskettes unless otherwise specified, though most of the discussion could apply equally well to hard disks.

boot

A disk containing a kernel which can be booted. The disk can be used to boot the kernel, which then may load a root file system on another disk. The kernel on a bootdisk usually must be told where to find its root filesystem.

Often a bootdisk loads a root filesystem from another diskette, but it is possible for a bootdisk to be set up to load a hard disk's root filesystem instead. This is commonly done when testing a new kernel. (in fact, ```make zdisk`'' will create such a bootdisk automatically from the kernel source code).

root

A disk with a filesystem containing files required to run a Linux system. Such a disk does not necessarily contain either a kernel or a boot loader.

A root disk can be used to run the system independently of any other disks, once the kernel has been booted. Usually the root disk is automatically copied to a ramdisk. This makes root disk accesses much faster, and frees up the disk drive for a utility disk.

boot/root

A disk which contains both the kernel and a root filesystem. In other words, it contains everything necessary to boot and run a Linux system without a hard disk. The advantage of this type of disk is that is it compact -- everything required is on a single disk. However, the gradually increasing size of everything means that it is increasingly difficult to fit everything on a single diskette, even with compression.

utility

> A disk which contains a filesystem, but is not intended to be mounted as a root file system. It is an additional data disk. You would use this type of disk to carry additional utilities where you have too much to fit on your root disk.

In general, when we talk about ``building a bootdisk'' we mean creating both the boot (kernel) and root (files) portions. They may be either together (a single boot/root disk) or separate (boot + root disks). The most flexible approach for rescue diskettes is probably to use separate boot and root diskettes, and one or more utility diskettes to handle the overflow.

---

4. Building a root filesystem.

Creating the root filesystem involves selecting files necessary for the system to run. In this section we describe how to build a *compressed root filesystem*. A less common option is to build an uncompressed filesystem on a diskette that is directly mounted as root; this alternative is described in section Non-ramdisk Root Filesystem.

4.1 Overview.

A root filesystem must contain everything needed to support a full Linux system. To be able to do this, the disk must include the minimum requirements for a Linux system:

- The basic file system structure,
- Minimum set of directories: `/dev, /proc, /bin, /etc, /lib, /usr, /tmp`,
- Basic set of utilities: `sh, ls, cp, mv`, etc.,
- Minimum set of config files: `rc, inittab, fstab`, etc.,
- Devices: `/dev/hd*, /dev/tty*, /dev/fd0`, etc.,
- Runtime library to provide basic functions used by utilities.

Of course, any system only becomes useful when you can run something on it, and a root diskette usually only becomes useful when you can do something like:

- Check a file system on another drive, for example to check your root file system on your hard drive, you need to be able to boot Linux from another drive, as you can with a root diskette system. Then you can run `fsck` on your original root drive while it is not mounted.
- Restore all or part of your original root drive from backup using archive and compression utilities such as `cpio, tar, gzip` and `ftape`.

We will describe how to build a *compressed* filesystem, so called because it is compressed on disk and, when booted, is uncompressed onto a ramdisk. With a compressed filesystem you can fit many files (approximately six megabytes) onto a standard 1440K diskette. Because the filesystem is much larger than a diskette, it cannot be built on the diskette. We have to build it elsewhere, compress it, then copy it to the diskette.

4.2 Creating the filesystem.

In order to build such a root filesystem, you need a spare device that is large enough to hold all the files before compression. You will need a device capable of holding about four megabytes. There are several choices:

- Use a **ramdisk** (DEVICE = /dev/ram0). In this case, memory is used to simulate a disk drive. The ramdisk must be large enough to hold a filesystem of the appropriate size. If you use LILO, check your configuration file (/etc/lilo.conf) for a line like:

  ```
  RAMDISK_SIZE = nnn
  ```

  which determines the maximum RAM that can be allocated to a ramdisk. The default is 4096K, which should be sufficient. You should probably not try to use such a ramdisk on a machine with less than 8MB of RAM. Check to make sure you have a device like /dev/ram0, /dev/ram or /dev/ramdisk. If not, create /dev/ram0 with `mknod` (major number 1, minor 0).
- If you have an unused hard disk partition that is large enough (several megabytes), this is a good solution.
- Use a **loopback device**, which allows a disk file to be treated as a device. Using a loopback device you can create a three megabyte file on your hard disk and build the filesystem on it. Type `man losetup` for instructions on using loopback devices. If you don't have `losetup`, you can get it along with compatible versions of `mount` and `unmount` from the `util-linux` package in the directory ftp://ftp.win.tue.nl/pub/linux/utils/util-linux/.

TOC

If you do not have a loop device (/dev/loop0, /dev/loop1, etc.) on your system, you will have to create one with ``mk /dev/loop0 b 7 0''. One you've installed these special `mount` and `umount` binaries, create a temporary file o hard disk with enough capacity (eg, /tmp/fsfile). You can use a command like

```
dd if=/dev/zero of=/tmp/fsfile bs=1k count=<it/nnn/
```

to create an *nnn*-block file.

Use the file name in place of DEVICE below. When you issue a mount command you must include the option ``-o loop'' to tell mount to use a loopback device. For example:

```
mount -o loop -t ext2 /tmp/fsfile /mnt
```

will mount /tmp/fsfile (via a loopback device) at the mount point `/mnt`. A `df` will confirm this.

After you've chosen one of these options, prepare the DEVICE with:

```
dd if=/dev/zero of=DEVICE bs=1k count=3000
```

This command zeroes out the device. This step is important because the filesystem on the device will be compressed later, so all unused portions should be filled with zeroes to achieve maximum compression.

Next, create the filesystem. The Linux kernel recognizes two file system types for root disks to be automatically copied to ramdisk. These are minix and ext2, of which ext2 is the preferred file system. If using ext2, you may find it useful to use the `-i` option to specify more inodes than the default; `-i 2000` is suggested so that you don't run out of inodes. Alternatively, you can save on inodes by removing lots of unnecessary `/dev` files. `mke2fs` will by default create 360 inodes on a 1.44Mb diskette. I find that 120 inodes is ample on my current rescue root diskette, but if you include all the devices in the `/dev` directory then you will easily exceed 360. Using a compressed root filesystem allows a larger filesystem, and hence more inodes by default, but you may still need to either reduce the number of files or increase the number of inodes.

So the command you use will look like:

```
mke2fs -m 0 -i 2000 DEVICE
```

(If you're using a loopback device, the disk file you're using should be supplied in place of this DEVICE. In this case, `mke2fs` will ask if you really want to do this; say yes.)

The `mke2fs` command will automatically detect the space available and configure itself accordingly. The `-m 0` parameter prevents it from reserving space for root, and hence provides more usable space on the disk.

Next, mount the device:

```
mount -t ext2 DEVICE /mnt
```

(You must create a mount point `/mnt` if it does not already exist.) In the remaining sections, all destination directory names are assumed to be relative to `/mnt`.

4.3 Populating the filesystem.

Here is a reasonable minimum set of directories for your root filesystem:

- `/dev` -- Devices, required to perform I/O
- `/proc` -- Directory stub required by the proc filesystem
- `/etc` -- System configuration files
- `/sbin` -- Critical system binaries
- `/bin` -- Basic binaries considered part of the system
- `/lib` -- Shared libraries to provide run-time support

- `/mnt` -- A mount point for maintenance on other disks
- `/usr` -- Additional utilities and applications

(The directory structure presented here is for root diskette use only. Real Linux systems have a more complex and disciplined set of policies, called the Filesystem Hierarchy Standard, for determining where files should go.)

Three of these directories will be empty on the root filesystem, so they only need to be created with `mkdir`. The `/proc` directory is basically a stub under which the proc filesystem is placed. The directories `/mnt` and `/usr` are only mount points for use after the boot/root system is running. Hence again, these directories only need to be created.

The remaining four directories are described in the following sections.

/dev

A `/dev` directory containing a special file for all devices to be used by the system is mandatory for any Linux system. The directory itself is a normal directory, and can be created with `mkdir` in the normal way. The device special files, however, must be created in a special way, using the `mknod` command.

There is a shortcut, though -- copy your existing `/dev` directory contents, and delete the ones you don't want. The only requirement is that you copy the device special files using `-R` option. This will copy the directory without attempting to copy the contents of the files. *Be sure to use an upper case R*. If you use the lower case switch `-r`, you will probably end up copying the entire contents of all of your hard disks -- or at least as much of them as will fit on a diskette! Therefore, take care, and use the command:

```
cp -dpR /dev /mnt
```

assuming that the diskette is mounted at `/mnt`. The `dp` switches ensure that symbolic links are copied as links, rather than using the target file, and that the original file attributes are preserved, thus preserving ownership information.

If you want to do it the hard way, use `ls -l` to display the major and minor device numbers for the devices you want, and create them on the diskette using `mknod`.

However the devices are copied, it is worth checking that any special devices you need have been placed on the rescue diskette. For example, `ftape` uses tape devices, so you will need to copy all of these if you intend to access your floppy tape drive from the bootdisk.

Note that one inode is required for each device special file, and inodes can at times be a scarce resource, especially on diskette filesystems. It therefore makes sense to remove any device special files that you don't need from the diskette `/dev` directory. Many devices are obviously unnecessary on specific systems. For example, if you do not have SCSI disks you can safely remove all the device files starting with `sd`. Similarly, if you don't intend to use your serial port then all device files starting with `cua` can go.

Be sure to include the following files from this directory: `console, kmem, mem, null, ram, tty1`.

/etc

This directory must contain a number of configuration files. On most systems, these can be divided into three groups:

1. Required at all times, *e.g.* `rc, fstab, passwd`.
2. May be required, but no-one is too sure.
3. Junk that crept in.

Files which are not essential can be identified with the command:

```
ls -ltru
```

This lists files in reverse order of date last accessed, so if any files are not being accessed, they can be omitted from a root diskette.

On my root diskettes, I have the number of config files down to 15. This reduces my work to dealing with three sets of files:

TOC

1. The ones I must configure for a boot/root system:
   1. `rc.d/*` -- system startup and run level change scripts
   2. `fstab` -- list of file systems to be mounted
   3. `inittab` -- parameters for the `init` process, the first process started at boot time.
2. The ones I should tidy up for a boot/root system:
   1. `passwd` -- list of users, home directories, etc.
   2. `group` -- user groups.
   3. `shadow` -- passwords of users. You may not have this.
   4. `termcap` -- the terminal capability database.

   If security is important, `passwd` and `shadow` should be pruned to avoid copying user passwords off the system, and so that when you boot from diskette, unwanted logins are rejected. Be sure that `passwd` contains at least `root`. If you intend other users to login, be sure their home directories and shells exist. `termcap`, the terminal database, is typically several hundred kilobytes. The version on your boot/root diskette should be pruned down to contain only the terminal(s) you use, which is usually just the `linux-console` entry.
3. The rest. They work at the moment, so I leave them alone.

Out of this, I only really have to configure two files, and what they should contain is surprisingly small.

- `rc` should contain:

  ```
  #!/bin/sh
  /bin/mount -av
  /bin/hostname Kangaroo
  ```

  Be sure the directories are right. You don't really need to run `hostname` -- it just looks nicer if you do.
- `fstab` should contain at least:

  ```
  /dev/ram0       /               ext2    defaults
  /dev/fd0        /               ext2    defaults
  /proc           /proc           proc    defaults
  ```

  You can copy entries from your existing `fstab`, but you should not automatically mount any of your hard disk partitions; use the `noauto` keyword with them. Your hard disk may be damaged or dead when the bootdisk is used.

Your `inittab` should be changed so that its `sysinit` line runs `rc` or whatever basic boot script will be used. Also, if you want to ensure that users on serial ports cannot login, comment out all the entries for `getty` which include a `ttys` or `ttyS` device at the end of the line. Leave in the `tty` ports so that you can login at the console.

A minimal `inittab` file looks like this:

```
id:2:initdefault:
si::sysinit:/etc/rc
1:2345:respawn:/sbin/getty 9600 tty1
2:23:respawn:/sbin/getty 9600 tty2
```

The `inittab` file defines what the system will run in various states including startup, move to multi-user mode, etc. Be sure to check carefully the filenames mentioned in `inittab`; if `init` cannot find the program mentioned the bootdisk will hang, and you may not even get an error message.

Note that some programs cannot be moved elsewhere because other programs have hardcoded their locations. For example on my system, /etc/shutdown has hardcoded in it /etc/reboot. If I move `reboot` to /bin/reboot, and then issue a `shutdown` command, it will fail because it cannot find the `reboot` file.

For the rest, just copy all the text files in your `/etc` directory, plus all the executables in your `/etc` directory that you cannot be sure you do not need. As a guide, consult the sample listing in Section Sample rootdisk directory listings. Probably it will suffice to copy only those files, but systems differ a great deal, so you cannot be sure that the same set of files on your system is

equivalent to the files in the list. The only sure method is to start with `inittab` and work out what is required.

Most systems now use an /etc/rc.d/ directory containing shell scripts for different run levels. The minimum is a single `rc` script, but it may be simpler just to copy `inittab` and the /etc/rc.d directory from your existing system, and prune the shell scripts in the `rc.d` directory to remove processing not relevent to a diskette system environment.

/bin and /sbin

The `/bin` directory is a convenient place for extra utilities you need to perform basic operations, utilities such as `ls`, `mv`, `cat` and `dd`. See Appendix Sample rootdisk directory listings for an example list of files that go in a `/bin` and `/sbin` directories. It does not include any of the utilities required to restore from backup, such as `cpio`, `tar` and `gzip`. That is because I place these on a separate utility diskette, to save space on the boot/root diskette. Once the boot/root diskette is booted, it is copied to the ramdisk leaving the diskette drive free to mount another diskette, the utility diskette. I usually mount this as `/usr`.

Creation of a utility diskette is described below in the section Section Building a utility disk. It is probably desirable to maintain a copy of the same version of backup utilities used to write the backups so you don't waste time trying to install versions that cannot read your backup tapes.

Make sure you include the following programs: `init`, `getty` or equivalent, `login`, `mount`, some shell capable of running your rc scripts, a link from `sh` to the shell.

/lib

In `/lib` you place necessary shared libraries and loaders. If the necessary libraries are not found in your `/lib` directory then the system will be unable to boot. If you're lucky you may see an error message telling you why.

Nearly every program requires at least the `libc` library, `libc.so.`*N*, where *N* is the current version number. Check your `/lib` directory. `libc.so.N` is usually a symlink to a filename with a complete version number:

```
% ls -l /lib/libc*
-rwxr-xr-x   1 root     root      4016683 Apr 16 18:48 libc-2.1.1.so*
lrwxrwxrwx   1 root     root           13 Apr 10 12:25 libc.so.6 -> libc-2.1.1.so*
```

In this case, you want `libc-2.1.1.so`. To find other libraries you should go through all the binaries you plan to include and check their dependencies with the `ldd` command. For example:

```
% ldd /sbin/mke2fs
libext2fs.so.2 => /lib/libext2fs.so.2 (0x40014000)
libcom_err.so.2 => /lib/libcom_err.so.2 (0x40026000)
libuuid.so.1 => /lib/libuuid.so.1 (0x40028000)
libc.so.6 => /lib/libc.so.6 (0x4002c000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Each file on the right-hand side is required. The file may be a symbolic link.

Note that some libraries are *quite large* and will not fit easily on your root filesystem. For example, the `libc.so` listed above is about 4 meg. You will probably need to strip libraries when copying them to your root filesystem. See Section Reducing root filesystem size for instructions.

In `/lib` you must also include a loader for the libraries. The loader will be either `ld.so` (for a.out libraries) or `ld-linux.so` (for ELF libraries). Newer versions of `ldd` tell you exactly which loader is needed, as in the example above, but older versions may not. If you're unsure which you need, run the `file` command on the library. For example:

```
% file/lib/libc.so.4.7.2 /lib/libc.so.5.4.33 /lib/libc-2.1.1.so
/lib/libc.so.4.7.2: Linux/i386 demand-paged executable (QMAGIC), stripped
/lib/libc.so.5.4.33: ELF 32-bit LSB shared object, Intel 80386, version 1,
stripped
```

```
        /lib/libc-2.1.1.so: ELF 32-bit LSB shared object, Intel 80386, version 1,
stripped
```

TOC

The `QMAGIC` indicates that `4.7.2` is for a.out libraries, and `ELF` indicates that `5.4.33` and `2.1.1` are for ELF.

Copy the specific loader(s) you need to the root filesystem you're building. Libraries and loaders should be checked *carefully* against the included binaries. If the kernel cannot load a necessary library, the kernel will usually hang with no error message.

4.4 Providing for PAM and NSS.

Your system may require dynamically loaded libraries that are not visible to ldd.

PAM (Pluggable Authentication Modules).

If your system uses PAM (Pluggable Authentication Modules), you must make some provision for it on your bootdisk or you will not be able to login. PAM, briefly, is a sophisticated modular method for authenticating users and controlling their access to services. An easy way to determine if your system uses PAM is to check your hard disks's /etc directory for a file pam.conf or a pam.d directory; if either exists, you must provide some minimal PAM support. (Alternatively, run ldd on your login executable; if the output includes libpam.so, you need PAM.)

Fortunately, security is usually of no concern with bootdisks since anyone who has physical access to a machine can usually do anything they want anyway. Therefore, you can effectively disable PAM by creating a simple /etc/pam.conf file in your root filesystem that looks like this:

```
OTHER   auth       optional     /lib/security/pam_permit.so
OTHER   account    optional     /lib/security/pam_permit.so
OTHER   password   optional     /lib/security/pam_permit.so
OTHER   session    optional     /lib/security/pam_permit.so
```

Also copy the file `/lib/security/pam_permit.so` to your root filesystem. This library is only about 8K so it imposes minimal overhead.

Note that this configuration allows anyone complete access to the files and services on your machine. If you care about security on your bootdisk for some reason, you'll have to copy some or all of your hard disk's PAM setup to your root filesystem. Be sure to read the PAM documentation carefully, and copy any libraries needed in /lib/security onto your root filesystem.

You must also include /lib/libpam.so on your bootdisk. But you already know this since you ran ldd on /bin/login, which showed this dependency.

NSS (Name Service Switch).

If you are using glibc (aka libc6), you will have to make provisions for name services or you will not be able to log in. The file /etc/nsswitch.conf controls database lookups for various servies. If you don't plan to access services from the network (eg, DNS or NIS lookups), you need only prepare a simple nsswitch.conf file that looks like this:

```
    passwd:     files
    shadow:     files
    group:      files
    hosts:      files
    services:   files
    networks:   files
    protocols:  files
    rpc:        files
```

```
ethers:     files
netmasks:   files
bootparams: files
automount:  files
aliases:    files
netgroup:   files
publickey:  files
```

This specifies that every service be provided only by local files. You will also need to include `/lib/libnss_files.so.1`, which will be loaded dynamically to handle the file lookups.

If you plan to access the network from your bootdisk, you may want to create a more elaborate nsswitch.conf file. See the nsswitch man page for details. Keep in mind that you must include a file `/lib/libnss_`*service*`.so.1` for each *service* you specify.

4.5 Modules.

If you have a modular kernel, you must consider which modules you may want to load from your bootdisk after booting. You might want to include `ftape` and `zftape` modules if your backup tapes are on floppy tape, modules for SCSI devices if you have them, and possibly modules for PPP or SLIP support if you want to access the net in an emergency.

These modules may be placed in /lib/modules. You should also include `insmod`, `rmmod` and `lsmod`. Depending on whether you want to load modules automatically, you might also include `modprobe`, `depmod` and `swapout`. If you use `kerneld`, include it along with /etc/conf.modules.

However, the main advantage to using modules is that you can move non-critical modules to a utility disk and load them when needed, thus using less space on your root disk. If you may have to deal with many different devices, this approach is preferable to building one huge kernel with many drivers built in.

Note that in order to boot a compressed ext2 filesystem, you must have ramdisk and ext2 support built-in. They cannot be supplied as modules.

4.6 Some final details.

Some system programs, such as `login`, complain if the file /var/run/utmp and the directory /var/log do not exist. So:

```
mkdir -p /mnt/var/{log,run}
touch /mnt/var/run/utmp
```

Finally, after you have set up all the libraries you need, run `ldconfig` to remake /etc/ld.so.cache on the root filesystem. The cache tells the loader where to find the libraries. To remake `ld.so.cache`, issue the following commands:

```
chdir /mnt; chroot /mnt /sbin/ldconfig
```

The `chroot` is necessary because `ldconfig` always remakes the cache for the root filesystem.

4.7 Wrapping it up.

Once you have finished constructing the root filesystem, unmount it, copy it to a file and compress it:

```
umount /mnt
dd if=DEVICE bs=1k | gzip -v9 > rootfs.gz
```

When this finishes you will have a file `rootfs.gz` that is your compressed root filesystem. You should check its size to make sure it will fit on a diskette; if it doesn't you'll have to go back and remove some files. Section Reducing root filesystem size has

some hints for reducing the size of the root filesystem.

5. Choosing a kernel.

At this point you have a complete compressed root filesystem. The next step is to build or select a kernel. In most cases it would be possible to copy your current kernel and boot the diskette from that. However, there may be cases where you wish to build a separate one.

One reason is size. If you are building a single boot/root diskette, the kernel will be one of the largest files on the diskette so you will have to reduce the size of the kernel as much as possible. To reduce kernel size, build it with the minumum set of facilities necessary to support the desired system. This means leaving out everything you don't need. Networking is a good thing to leave out, as well as support for any disk drives and other devices which you don't need when running your boot/root system. As stated before, your kernel *must* have ramdisk and ext2 support built into it.

Having worked out a minimum set of facilities to include in a kernel, you then need to work out what to add back in. Probably the most common uses for a boot/root diskette system would be to examine and restore a corrupted root file system, and to do this you may need kernel support. For example, if your backups are all held on tape using Ftape to access your tape drive, then, if you lose your current root drive and drives containing Ftape, then you will not be able to restore from your backup tapes. You will have to reinstall Linux, download and reinstall ftape, and then try to read your backups.

The point here is that, whatever I/O support you have added to your kernel to support backups should also be added into your boot/root kernel.

The procedure for actually building the kernel is described in the documentation that comes with the kernel. It is quite easy to follow, so start by looking in /usr/src/linux. If you have trouble building a kernel, you should probably not attempt to build boot/root systems anyway. Remember to compress the kernel with ```make zImage`".

6. Putting them together: Making the diskette(s).

At this point you have a kernel and a compressed root filesystem. If you are making a boot/root disk, check their sizes to make sure they will both fit on one disk. If you are making a two disk boot+root set, check the root filesystem to make sure it will fit on a single diskette.

You should decide whether to use LILO to boot the bootdisk kernel. The alternative is to copy the kernel directly to the diskette and boot without LILO. The advantage of using LILO is that it enables you to supply some parameters to the kernel which may be necessary to initialize your hardware (Check the file /etc/lilo.conf on your system. If it exists and has a line like ```append=...`", you probably need this feature). The disadvantage of using LILO is that building the bootdisk is more complicated and takes slightly more space. You will have to set up a small separate filesystem, which we shall call the **kernel filesystem**, where you transfer the kernel and a few other files that LILO needs.

If you are going to use LILO, read on; if you are going to transfer the kernel directly, skip ahead to section Without using LILO.

6.1 Transferring the kernel with LILO .

The first thing you must do is create a small configuration file for LILO. It should look like this:

```
boot      =/dev/fd0
install   =/boot/boot.b
map       =/boot/map
read-write
backup    =/dev/null
compact
image     = KERNEL
label     = Bootdisk
root      =/dev/fd0
```

For an explanation of these parameters, see LILO's user documentation. You will probably also want to add an `append=...` line to this file from your hard disk's /etc/lilo.conf file.

Save this file as `bdlilo.conf`.

You now have to create a small filesystem, which we shall call a **kernel filesystem**, to distinguish it from the root filesystem.

First, figure out how large the filesystem should be. Take the size of your kernel in blocks (the size shown by ```ls -l KERNEL`" divided by 1024 and rounded up) and add 50. Fifty blocks is approximately the space needed for inodes plus other files. You can calculate this number exactly if you want to, or just use 50. If you're creating a two-disk set, you may as well overestimate the space since the first disk is only used for the

kernel anyway. Call this number `KERNEL_BLOCKS`.

Put a floppy diskette in the drive (for simplicity we'll assume /dev/fd0) and create an ext2 kernel filesystem on it:

```
mke2fs -i 8192 -m 0 /dev/fd0 KERNEL_BLOCKS
```

The ``-i 8192'' specifies that we want one inode per 8192 bytes. Next, mount the filesystem, remove the `lost+found` directory, and create `dev` and `boot` directories for LILO:

```
mount /dev/fd0 /mnt
rm -rf /mnt/lost+found
mkdir /mnt/{boot,dev}
```

Next, create devices /dev/null and /dev/fd0. Instead of looking up the device numbers, you can just copy them from your hard disk using `-R`:

```
cp -R /dev/{null,fd0} /mnt/dev
```

LILO needs a copy of its boot loader, `boot.b`, which you can take from your hard disk. It is usually kept in the `/boot` directory.

```
cp /boot/boot.b /mnt/boot
```

Finally, copy in the LILO configuration file you created in the last section, along with your kernel. Both can be put in the root directory:

```
cp bdlilo.conf KERNEL /mnt
```

Everything LILO needs is now on the kernel filesystem, so you are ready to run it. LILO's `-r` flag is used for installing the boot loader on some other root:

```
lilo -v -C bdlilo.conf -r /mnt
```

LILO should run without error, after which the kernel filesystem should look something like this:

```
total 361
  1 -rw-r--r--   1 root     root          176 Jan 10 07:22 bdlilo.conf
  1 drwxr-xr-x   2 root     root         1024 Jan 10 07:23 boot/
  1 drwxr-xr-x   2 root     root         1024 Jan 10 07:22 dev/
358 -rw-r--r--   1 root     root       362707 Jan 10 07:23 vmlinuz
boot:
total 8
  4 -rw-r--r--   1 root     root         3708 Jan 10 07:22 boot.b
  4 -rw-------   1 root     root         3584 Jan 10 07:23 map
dev:
total 0
```

```
 0 brw-r-----   1 root     root       2,   0 Jan 10 07:22 fd0
 0 crw-r--r--   1 root     root       1,   3 Jan 10 07:22 null
```


Do not worry if the file sizes are slightly different from yours.

Now leave the disk in the drive and go to section Setting the ramdisk word.

6.2 Transferring the kernel without LILO.

If you are *not* using LILO, transfer the kernel to the bootdisk with the `dd` command:

```
% dd if=KERNEL of=/dev/fd0 bs=1k
353+1 records in
353+1 records out
```

In this example, `dd` wrote 353 complete records + 1 partial record, so the kernel occupies the first 354 blocks of the diskette. Call this number `KERNEL_BLOCKS` and remember it for use in the next section.

Finally, set the root device to be the diskette itself, then set the root to be loaded read/write:

```
rdev /dev/fd0 /dev/fd0
rdev -R /dev/fd0 0
```

Be careful to use a capital `-R` in the second `rdev` command.

6.3 Setting the ramdisk word.

Inside the kernel image is the **ramdisk word** that specifies where the root filesystem is to be found, along with other options. The word can be accessed and set via the `rdev` command, and its contents are interpreted as follows:

```
bits  0-10:     Offset to start of ramdisk, in 1024 byte blocks
bits 11-13:     unused
bit     14:     Flag indicating that ramdisk is to be loaded
bit     15:     Flag indicating to prompt before loading rootfs
```

If bit 15 is set, on boot-up you will be prompted to place a new floppy diskette in the drive. This is necessary for a two-disk boot set.

There are two cases, depending on whether you are building a single boot/root diskette or a double ``boot+root'' diskette set.

1. If you are building a single disk, the compressed root filesystem will be placed right after the kernel, so the offset will be the first free block (which should be the same as `KERNEL_BLOCKS`). Bit 14 will be set to 1, and bit 15 will be zero.
2. If you are building a two-disk set, the root filesystem will begin at block zero of the second disk, so the offset will be zero. Bit 14 will be set to 1, and bit 15 will be 1.

After carefully calculating the value for the ramdisk word, set it with `rdev -r`. Be sure to use the *decimal* value. If you used LILO, the argument to `rdev` here should be the *mounted kernel path*, e.g. `/mnt/vmlinuz`; if you copied the kernel with `dd`, instead use the floppy device name (*e.g.*, `/dev/fd0`).

```
rdev -r KERNEL_OR_FLOPPY_DRIVE  VALUE
```

If you used LILO, unmount the diskette now.

6.4 Transferring the root filesystem.

The last step is to transfer the root filesystem.

- If the root filesystem will be placed on the *same* disk as the kernel, transfer it using `dd` with the `seek` option, which specifies how many blocks to skip:

```
dd if=rootfs.gz of=/dev/fd0 bs=1k seek=KERNEL_BLOCKS
```

- If the root filesystem will be placed on a *second* disk, remove the first diskette, put the second diskette in the drive, then transfer the root filesystem to it:

```
dd if=rootfs.gz of=/dev/fd0 bs=1k
```

Congratulations, you are done! *You should always test a bootdisk before putting it aside for an emergency!* If it fails to boot, read on.

---

7. Troubleshooting, or The Agony of Defeat.

When building bootdisks, the first few tries often will not boot. The general approach to building a root disk is to assemble components from your existing system, and try and get the diskette-based system to the point where it displays messages on the console. Once it starts talking to you, the battle is half over because you can see what it is complaining about, and you can fix individual problems until the system works smoothly. If the system just hangs with no explanation, finding the cause can be difficult. To get a system to boot to the stage where it will talk to you requires several components to be present and correctly configured. The recommended procedure for investigating the problem where the system will not talk to you is as follows:

- You may see a message like this:

  `Kernel panic: VFS: Unable to mount root fs on XX:YY`

  This is a common problem and it has only a few causes. First, check the device *XX:YY* against the list of device codes; is it the correct root device? If not, you probably didn't do an `rdev -R`, or you did it on the wrong image. If the device code is correct, then check carefully the device drivers compiled into your kernel. Make sure it has floppy disk, ramdisk and ext2 filesystem support built-in.
- Check that the root disk actually contains the directories you think it does. It is easy to copy at the wrong level so that you end up with something like /rootdisk/bin instead of `/bin` on your root diskette.
- Check that there is a /lib/libc.so with the same link that appears in your `/lib` directory on your hard disk.
- Check that any symbolic links in your `/dev` directory in your existing system also exist on your root diskette filesystem, where those links are to devices which you have included in your root diskette. In particular, `/dev/console` links are essential in many cases.
- Check that you have included `/dev/tty1, /dev/null, /dev/zero, /dev/mem, /dev/ram` and `/dev/kmem` files.
- Check your kernel configuration -- support for all resources required up to login point must be built in, not modules. So *ramdisk and ext2 support must be built-in*.
- Check that your kernel root device and ramdisk settings are correct.

Once these general aspects have been covered, here are some more specific files to check:

1. Make sure `init` is included as /sbin/init or /bin/init. Make sure it is executable.
2. Run `ldd init` to check init's libraries. Usually this is just `libc.so`, but check anyway. Make sure you included the necessary libraries and loaders.
3. Make sure you have the right loader for your libraries -- `ld.so` for a.out or `ld-linux.so` for ELF.
4. Check the /etc/inittab on your bootdisk filesystem for the calls to `getty` (or some `getty`-like

program, such as `agetty`, `mgetty` or `getty_ps`). Double-check these against your hard d `inittab`. Check the man pages of the program you use to make sure these make sense. `inittab` is possibly the trickiest part because its syntax and content depend on the init program used and the nature of the system. The only way to tackle it is to read the man pages for `init` and `inittab` and work out exactly what your existing system is doing when it boots. Check to make sure /etc/inittab has a system initialisation entry. This should contain a command to execute the system initialization script, which must exist.

5. As with `init`, run `ldd` on your `getty` to see what it needs, and make sure the necessary library files and loaders were included in your root filesystem.
6. Be sure you have included a shell program (e.g., `bash` or `ash`) capable of running all of your rc scripts.
7. If you have a /etc/ld.so.cache file on your rescue disk, remake it.

If `init` starts, but you get a message like:

```
Id xxx respawning too fast: disabled for 5 minutes
```

it is coming from `init`, usually indicating that `getty` or `login` is dying as soon as it starts up. Check the `getty` and `login` executables and the libraries they depend upon. Make sure the invocations in /etc/inittab are correct. If you get strange messages from `getty`, it may mean the calling form in /etc/inittab is wrong. The options of the *getty* programs are variable; even different versions of `agetty` are reported to have different incompatible calling forms.

If you get a login prompt, and you enter a valid login name but the system prompts you for another login name immediately, the problem may be with PAM or NSS. See Section PAM and NSS. The problem may also be that you use shadow passwords and didn't copy /etc/shadow to your bootdisk.

If you try to run some executable, such as `df`, which is on your rescue disk but you yields a message like: `df: not found`, check two things: (1) Make sure the directory containing the binary is in your PATH, and (2) make sure you have libraries (and loaders) the program needs.

---

8. Miscellaneous topics.

8.1 Reducing root filesystem size.

Sometimes a root filesystem is too large to fit on a diskette even after compression. Here are some ways to reduce the filesystem size, listed in decreasing order of effectiveness:

Increase the disk density

By default, floppy diskettes are formatted at 1440K, but higher density formats are available. fdformat will format disks for the following sizes: 1600, 1680, 1722, 1743, 1760, 1840, and 1920. Most 1440K drives will support 1722K, and this is what I always use for bootdisks. See the fdformat man page and /usr/src/linux/Documentation/devices.txt.

Replace your shell

Some of the popular shells for Linux, such as `bash` and `tcsh`, are large and require many libraries. Light-weight alternatives exist, such as `ash`, `lsh`, `kiss` and `smash`, which are much smaller and require few (or no) libraries. Most of these replacement shells are available from http://metalab.unc.edu/pub/Linux/system/shells/. Make sure any shell you use is capable of running commands in all the `rc` files you include on your bootdisk.

Strip libraries and binaries

Many libraries and binaries are typically unstripped (include debugging symbols). Running '`file`' on these files will tell you '`not stripped`' if so. When copying binaries to your root filesystem, it is good practice to use:

```
objcopy --strip-all FROM TO
```

When copying libraries, use:

```
objcopy --strip-debug FROM TO
```

Move non-critical files to a utility disk

If some of your binaries are not needed immediately to boot or login, you can move them to a utility disk. See section Building a utility disk for details. You may also consider moving modules to a utility disk as well.

8.2 Non-ramdisk root filesystems.

Section Building a root filesystem gave instructions for building a compressed root filesystem which is loaded to ramdisk when the system boots. This method has many advantages so it is commonly used.

However, some systems with little memory cannot afford the RAM needed for this, and they must u root filesystems mounted directly from the diskette.

Such filesystems are actually easier to build than compressed root filesystems because they can be built on a diskette rather than on some other device, and they do not have to be compressed. We will outline the procedure as it differs from the instructions above. If you choose to do this, keep in mind that you will have *much less space* available.

1. Calculate how much space you will have available for root files. If you are building a single boot/root disk, you must fit all blocks for the kernel plus all blocks for the root filesystem on the one disk.
2. Using `mke2fs`, create a root filesystem on a diskette of the appropriate size.
3. Populate the filesystem as described above.
4. When done, unmount the filesystem and transfer it to a disk file but *do not compress it*.
5. Transfer the kernel to a floppy diskette, as described above. When calculating the ramdisk word, **set bit 14 to zero**, to indicate that the root filesystem is not to be loaded to ramdisk. Run the `rdev`'s as described.
6. Transfer the root filesystem as before.

There are several shortcuts you can take. If you are building a two-disk set, you can build the complete root filesystem directly on the second disk and you need not transfer it to a hard disk file and then back. Also, if you are building a single boot/root disk and using LILO, you can build a *single* filesystem on the entire disk, containing the kernel, LILO files and root files, and simply run LILO as the last step.

8.3 Building a utility disk.

Building a utility disk is relatively easy -- simply create a filesystem on a formatted disk and copy files to it. To use it with a bootdisk, mount it manually after the system is booted.

In the instructions above, we mentioned that the utility disk could be mounted as `/usr`. In this case, binaries could be placed into a `/bin` directory on your utility disk, so that placing /usr/bin in your path will access them. Additional libraries needed by the binaries are placed in `/lib` on the utility disk.

There are several important points to keep in mind when designing a utility disk:

1. Do not place critical system binaries or libraries onto the utility disk, since it will not be mountable until after the system has booted.
2. You cannot access a floppy diskette and a floppy tape drive simultaneously. This means that if you have a floppy tape drive, you will not be able to access it while your utility disk is mounted.
3. Access to files on the utility disk will be slow.

Appendix Sample utility disk directory listing shows a sample of files on a utility disk. Here are some ideas for files you may find useful: programs for examining and manipulating disks (`format, fdisk`) and filesystems (`mke2fs, fsck, debugfs, isofs.o`), a lightweight text editor (`elvis, jove`), compression and archive utilities (`gzip, tar, cpio, afio`), tape utilities (`mt, tob, taper`), communications utilities (`ppp.o, slip.o, minicom`) and utilities for devices

(`setserial`, `mknod`).

9. How the pros do it.

You may notice that the bootdisks used by major distributions such as Slackware, RedHat or Debian seem more sophisticated than what is described in this document. Professional distribution bootdisks are based on the same principles outlined here, but employ various tricks because their bootdisks have additional requirements. First, they must be able to work with a wide variety of hardware, so they must be able to interact with the user and load various device drivers. Second, they must be prepared to work with many different installation options, with varying degrees of automation. Finally, distribution bootdisks usually combine installation and rescue capabilities.

Some bootdisks use a feature called **initrd** (**initial ramdisk**). This feature was introduced around 2.0.x and allows a kernel to boot in two phases. When the kernel first boots, it loads an initial ramdisk image from the boot disk. This initial ramdisk is a root filesystem containing a program that runs before the real root fs is loaded. This program usually inspects the environment and/or asks the user to select various boot options, such as the device from which to load the real rootdisk. It typically loads additional modules not built in to the kernel. When this initial program exits, the kernel loads the real root image and booting continues normally. For further information on `initrd`, see /usr/src/linux/Documentation/initrd.txt and ftp://elserv.ffm.fgan.de/pub/linux/loadlin-1.6/initrd-example.tgz

The following are summaries of how each distribution's installation disks seem to work, based on inspecting their filesystems and/or source code. We do not guarantee that this information is completely accurate, or that they have not changed since the versions noted.

Slackware (v.3.1) uses a straightforward LILO boot similar to what is described in section Transferring the kernel with LILO. The Slackware bootdisk prints a bootup message (``Welcome to the Slackware Linux bootkernel disk!") using LILO's `message` parameter. This instructs the user to enter a boot parameter line if necessary. After booting, a root filesystem is loaded from a second disk. The user invokes a `setup` script which starts the installation. Instead of using a modular kernel, Slackware provides many different kernels and depends upon the user to select the one matching his or her hardware requirements.

RedHat (v.4.0) also uses a LILO boot. It loads a compressed ramdisk on the first disk, which runs a custom `init` program. This program queries for drivers then loads additional files from a supplemental disk if necessary.

Debian (v.1.3) is probably the most sophisticated of the installation disk sets. It uses the SYSLINUX loader to arrange various load options, then uses an `initrd` image to guide the user through installation. It appears to use both a customized `init` and a customized shell.

10. Frequently asked question (FAQ) list.

Q. I boot from my boot/root disks and nothing happens. What do I do?

See section Troubleshooting, above.

Q. How does the Slackware/Debian/RedHat bootdisk work?

See section What the pros do, above.

Q. How can I make a boot disk with a XYZ driver?

The easiest way is to obtain a Slackware kernel from your nearest Slackware mirror site. Slackware kernels are generic kernels which atttempt to include drivers for as many devices as possible, so if you have a SCSI or IDE controller, chances are that a driver for it is included in the Slackware kernel.

Go to the `a1` directory and select either IDE or SCSI kernel depending on the type of controller you have. Check the xxxxkern.cfg file for the selected kernel to see the drivers which have been included in that kernel. If the device you want is in that list, then the corresponding kernel should boot your computer. Download the xxxxkern.tgz file and copy it to your boot diskette as described above in the section on making boot disks.

You must then check the root device in the kernel, using the rdev command:

```
rdev zImage
```

`rdev` will then display the current root device in the kernel. If this is not the same as the root device you want, then use `rdev` to change it. For example, the kernel I tried was set to /dev/sda2, but my root SCSI partition is /dev/sda8. To use a root diskette, you would have to use the command:

```
rdev zImage /dev/fd0
```

If you want to know how to set up a Slackware root disk as well, that's outside the scope of this HOWTO, so I suggest you check the Linux Install Guide or get the Slackware distribution. See the section in this HOWTO titled ``References''.

Q. How do I update my boot diskette with a new kernel?

Just copy the kernel to your boot diskette using the dd command for a boot diskette without a filesystem, or the cp command for a boot/root disk. Refer to the section in this HOWTO titled ``Boot'' for details on creating a boot disk. The description applies equally to updating a kernel on a boot disk.

Q. How do I update my root diskette with new files?

The easiest way is to copy the filesystem from the rootdisk back to the DEVICE you used (from section Creating the filesystem, above). Then mount the filesystem and make the changes. You have to remember where your root filesystem started and how many blocks it occupied:

```
dd if=/dev/fd0 bs=1k skip=ROOTBEGIN count=BLOCKS | gunzip > DEVICE
```

```
mount -t ext2 DEVICE /mnt
```

After making the changes, proceed as before (in Section Wrapping it up) and transfer the root filesystem back to the disk. You should not have to re-transfer the kernel or re-compute the ramdisk word if you do not change the starting position of the new root filesystem.

Q. How do I remove LILO so that I can use DOS to boot again?

This is not really a Bootdisk topic, but it is asked often. Within Linux, you can run:

```
/sbin/lilo -u
```

You can also use the `dd` command to copy the backup saved by LILO to the boot sector. Refer to the LILO documentation if you wish to do this.

Within DOS and Windows you can use the DOS command:

```
FDISK /MBR
```

MBR stands for Master Boot Record, and it replaces the boot sector with a clean DOS one, without affecting the partition table. Some purists disagree with this, but even the author of LILO, Werner Almesberger, suggests it. It is easy, and it works.

Q. How can I boot if I've lost my kernel *and* my boot disk?

If you don't have a boot disk standing by, probably the easiest method is to obtain a Slackware kernel for your disk controller type (IDE or SCSI) as described above for ``How do I make a boot disk with a XXX driver?''. You can then boot your computer using this kernel, then repair whatever damage there is.

The kernel you get may not have the root device set to the disk type and partition you want. For example, Slackware's generic SCSI kernel has the root device set to /dev/sda2, whereas my root Linux partition happens to be /dev/sda8. In this case the root device in the kernel will have to be changed.

You can still change the root device and ramdisk settings in the kernel even if all you have is a kernel, and some other operating system, such as DOS.

`rdev` changes kernel settings by changing the values at fixed offsets in the kernel file, so you can do the same if you have a hex editor available on whatever systems you do still have running -- for example, Norton Utilities Disk Editor under DOS. You then need to check and if necessary change the values in the kernel at the following offsets:

```
HEX     DEC  DESCRIPTION
0x01F8  504  Low byte of RAMDISK word
0x01F9  505  High byte of RAMDISK word
0x01FC  508  Root minor device number - see below
0X01FD  509  Root major device number - see below
```

The interpretation of the ramdisk word was described in Section Setting the ramdisk word, above.

The major and minor device numbers must be set to the device you want to mount your root filesystem on. Some useful values to select from are:

```
DEVICE          MAJOR MINOR
/dev/fd0            2     0   1st floppy drive
/dev/hda1           3     1   partition 1 on 1st IDE drive
/dev/sda1           8     1   partition 1 on 1st SCSI drive
/dev/sda8           8     8   partition 8 on 1st SCSI drive
```

Once you have set these values then you can write the file to a diskette using either Norton Utilities Disk Editor, or a program called `rawrite.exe`. This program is included in all distributions. It is a DOS program which writes a file to the ``raw'' disk, starting at the boot sector, instead of writing it to the file system. If you use Norton Utilities you must write the file to a physical disk starting at the beginning of the disk.

Q. How can I make extra copies of boot/root diskettes?

Because magnetic media may deteriorate over time, you should keep several copies of your rescue disk, in case the original is unreadable.

The easiest way of making copies of any diskettes, including bootable and utility diskettes, is to use the `dd` command to copy the contents of the original diskette to a file on your hard drive, and then use the same command to copy the file back to a new diskette. Note that you do not need to, and should not, mount the diskettes, because `dd` uses the raw device interface.

To copy the original, enter the command:

```
dd if=DEVICENAME of=FILENAME
where   DEVICENAME is the device name of the diskette drive
and     FILENAME is the name of the (hard-disk) output file
```

Omitting the `count` parameter causes dd to copy the whole diskette (2880 blocks if high-density).

To copy the resulting file back to a new diskette, insert the new diskette and enter the reverse command:

```
dd if=FILENAME of=DEVICENAME
```

Note that the above discussion assumes that you have only one diskette drive. If you have two of the same type, you can copy diskettes using a command like:

```
dd if=/dev/fd0 of=/dev/fd1
```

Q. How can I boot without typing in "ahaxxxx=nn,nn,nn" every time?

Where a disk device cannot be autodetected it is necessary to supply the kernel with a command device parameter string, such as:

```
aha152x=0x340,11,3,1
```

This parameter string can be supplied in several ways using LILO:

- By entering it on the command line every time the system is booted via LILO. This is boring, though.
- By using the LILO ```lock`'' keyword to make it store the command line as the default command line, so that LILO will use the same options every time it boots.
- By using the `append=` statement in the LILO config file. Note that the parameter string must be

enclosed in quotes.

For example, a sample command line using the above parameter string would be:

```
zImage  aha152x=0x340,11,3,1 root=/dev/sda1 lock
```

This would pass the device parameter string through, and also ask the kernel to set the root device to /dev/sda1 and save the whole command line and reuse it for all future boots.

A sample APPEND statement is:

```
APPEND = "aha152x=0x340,11,3,1"
```

Note that the parameter string must NOT be enclosed in quotes on the command line, but it MUST be enclosed in quotes in the APPEND statement.

Note also that for the parameter string to be acted on, the kernel must contain the driver for that disk type. If it does not, then there is nothing listening for the parameter string, and you will have to rebuild the kernel to include the required driver. For details on rebuilding the kernel, cd to /usr/src/linux and read the README, and read the Linux FAQ and Installation HOWTO. Alternatively you could obtain a generic kernel for the disk type and install that.

Readers are strongly urged to read the LILO documentation before experimenting with LILO installation. Incautious use of the `BOOT` statement can damage partitions.

Q. At boot time, I get error "`A: cannot execute B`". Why?

There are several cases of program names being hardcoded in various utilities. These cases do not occur everywhere, but they may explain why an executable apparently cannot be found on your system even though you can see that it is there. You can find out if a given program has the name of another hardcoded by using the `strings` command and piping the output through `grep`.

Known examples of hardcoding are:

- Shutdown in some versions has /etc/reboot hardcoded, so `reboot` must be placed in the `/etc` directory.
- `init` has caused problems for at least one person, with the kernel being unable to find `init`.

To fix these problems, either move the programs to the correct directory, or change configuration files (e.g. `inittab`) to point to the correct directory. If in doubt, put programs in the same directories as they are on your hard disk, and use the same `inittab` and /etc/rc.d files as they appear on your hard disk.

Q. My kernel has ramdisk support, but initializes ramdisks of 0K

Where this occurs, a kernel message like this will appear as the kernel is booting:

```
Ramdisk driver initialized : 16 ramdisks of 0K size
```

This is probably because the size has been set to 0 by kernel parameters at boot time. This could possibly be because of an overlooked LILO configuration file parameter:

```
ramdisk= 0
```

This was included in sample LILO configuration files in some older distributions, and was put there to override any previous kernel setting. If you have such a line, remove it.

Note that if you attempt to use a ramdisk which has been set to 0K the behaviour can be unpredictable, and can result in kernel panics.

---

11. Resources and pointers.

When retrieving a package, always get the latest version unless you have good reasons for not doing so.

11.1 Pre-made bootdisks.

These are sources for distribution bootdisks. *Please use one of the mirror sites to reduce the load on these machines.*

- Slackware bootdisks, rootdisks and Slackware mirror sites
- RedHat bootdisks and Red Hat mirror sites
- Debian bootdisks and Debian mirror sites

In addition to the distribution bootdisks, the following rescue disk images are available. Unless otherwise specified, these are available in the directory http://metalab.unc.edu/pub/Linux/system/recovery/!INDEX.html

- `tomsrtbt`, by Tom Oehser, is a single-disk boot/root disk based on kernel 2.0, with a large set of features and support programs. It supports IDE, SCSI, tape, network adaptors, PCMCIA and more. About 100 utility programs and tools are included for fixing and restoring disks. The package also includes scripts for disassembling and reconstructing the images so that new material can be added if necessary.
- `rescue02`, by John Comyns, is a rescue disk based on kernel 1.3.84, with support for IDE and Adaptec 1542 and NCR53C7,8xx. It uses ELF binaries but it has enough commands so that it can be used on any system. There are modules that can be loaded after booting for all other SCSI cards. It probably won't work on systems with 4 mb of ram since it uses a 3 mb ram disk.
- `resque_disk-2.0.22`, by Sergei Viznyuk, is a full-featured boot/root disk based on kernel 2.0.22 with built-in support for IDE, many difference SCSI controllers, and ELF/AOUT. Also includes many modules and useful utilities for repairing and restoring a hard disk.
- `cramdisk` images, based on the 2.0.23 kernel, available for 4 meg and 8 meg machines. They include math emulation and networking (PPP and dialin script, NE2000, 3C509), or support for the parallel port ZIP drive. These diskette images will boot on a 386 with 4MB RAM. MSDOS support is included so you can download from the net to a DOS partition.

  http://metalab.unc.edu/pub/Linux/system/recovery/images

11.2 Rescue packages.

Several packages for creating rescue disks are available on metalab.unc.edu. With these packages you specify a set of files for inclusion and the software automates (to varying degrees) the creation of a bootdisk. See http://metalab.unc.edu/pub/Linux/system/recovery/!INDEX.html for more information.

Check the file dates carefully -- some of these packages have not been updated in several years and not support the creation of a compressed root filesystem loaded into ramdisk. To the best of our knowledge, Yard is the only package that will.

11.3 Graham Chapman's shell scripts

Graham Chapman has written a set of scripts that may be useful as examples of how to create bootdisks. In previous versions of this HOWTO the scripts appeared in an appendix, but they have been deleted from the documented and placed on a web page:

http://www.zeta.org.au/~grahamc/linux.html

You may find it convenient to use these scripts, but if so, read the instructions carefully -- for example, if you specify the wrong swap device, you will find your root filesystem has been throroughly and permanently erased. Be sure you have it correctly configured before you use it!

11.4 LILO -- the Linux loader.

Written by Werner Almesberger. Excellent boot loader, and the documentation includes information on the boot sector contents and the early stages of the boot process.

Ftp from ftp://tsx-11.mit.edu/pub/linux/packages/lilo/. It is also available on Metalab and mirrors.

11.5 Linux FAQ and HOWTOs.

These are available from many sources. Look at the usenet newsgroups `news.answers` and `comp.os.linux.announce`.

The FAQ is available from http://metalab.unc.edu/pub/Linux/docs/faqs/linux-faq and the HOWTOs from http://metalab.unc.edu/pub/Linux/docs/HOWTO.

Most documentation for Linux may be found at The Linux Documentation Project homepage.

If desperate, send mail to `mail-server@rtfm.mit.edu` with the word ``help'' in the message, then follow the mailed instructions.

11.6 Ramdisk usage.

An excellent description of the how the new ramdisk code works may be found with the documentation supplied with the Linux kernel. See /usr/src/linux/Documentation/ramdisk.txt. It is written by Paul Gortmaker and includes a section on creating a compressed ramdisk.

11.7 The Linux boot process.

TOC

For more detail on the Linux boot process, here are some pointers:

- The Linux System Administrators' Guide has a section on booting, See http://metalab.unc.edu/LDP/LDP/sag/c1582.html
- The LILO ``Technical overview'' http://metalab.unc.edu/pub/Linux/system/boot/lilo/lilo-t-21.ps.gz has the definitive technical, low-level description of the boot process, up to where the kernel is started.
- The source code is the ultimate guide. Below are some kernel files related to the boot process. If you have the Linux kernel source code, you can find these under /usr/src/linux on your machine; alternatively, Shigio Yamaguchi (shigio@wafu.netgate.net) has a very nice hypertext kernel browser at http://wafu.netgate.net/linux/. Here are some relevant files:

  arch/i386/boot/bootsect.S,setup.S

  Contain assembly code for the bootsector.

  arch/i386/boot/compressed/misc.c

  Contains code for uncompressing the kernel.

  arch/i386/kernel/

  Directory containing kernel initialization code. `setup.c` contains the ramdisk word.

  drivers/block/rd.c

  Contains the ramdisk driver. The procedures `rd_load` and `rd_load_image` load blocks from a device into a ramdisk. The procedure `identify_ramdisk_image` determines what kind of filesystem is found and whether it is compressed.

---

12. LILO boot error codes.

Questions about these errors are asked so often on Usenet that we include them here as a public service. This summary is excerpted from Werner Almsberger's LILO User Documentation, available at http://metalab.unc.edu/pub/Linux/system/boot/lilo/lilo-u-21.ps.gz.

When LILO loads itself, it displays the word ```LILO`''. Each letter is printed before or after performing some specific action. If LILO fails at some point, the letters printed so far can be used to identify the problem.

(nothing)

No part of LILO has been loaded. LILO either isn't installed or the partition on which its boot sector is located isn't active.

L

The first stage boot loader has been loaded and started, but it can't load the second stage boot loader. The two-digit error codes indicate the type of problem. (See also section ``Disk error codes''.) This condition usually indicates a media failure or a geometry mismatch (e.g. bad disk parameters)

LI

The first stage boot loader was able to load the second stage boot loader, but has failed to execute it. This can either be caused by a geometry mismatch or by moving /boot/boot.b without running the map installer.

LIL

The second stage boot loader has been started, but it can't load the descriptor table from the map file. This is typically caused by a media failure or by a geometry mismatch.

LIL?

The second stage boot loader has been loaded at an incorrect address. This is typically caused by a subtle geometry mismatch or by moving /boot/boot.b without running the map installer.

LIL-

The descriptor table is corrupt. This can either be caused by a geometry mismatch or by moving /boot/map without running the map installer.

LILO

All parts of LILO have been successfully loaded.

If the BIOS signals an error when LILO is trying to load a boot image, the respective error code is displayed. These codes range from `0x00` through `0xbb`. See the LILO User Guide for an explanation of these.

13. Sample rootdisk directory listings.

Here are the contents of a sample root filesystem and a utility diskette.

```
Root directory:
drwx--x--x   2 root     root         1024 Nov  1 15:39 bin
drwx--x--x   2 root     root         4096 Nov  1 15:39 dev
drwx--x--x   3 root     root         1024 Nov  1 15:39 etc
drwx--x--x   4 root     root         1024 Nov  1 15:39 lib
drwx--x--x   5 root     root         1024 Nov  1 15:39 mnt
drwx--x--x   2 root     root         1024 Nov  1 15:39 proc
drwx--x--x   2 root     root         1024 Nov  1 15:39 root
drwx--x--x   2 root     root         1024 Nov  1 15:39 sbin
drwx--x--x   2 root     root         1024 Nov  1 15:39 tmp
drwx--x--x   7 root     root         1024 Nov  1 15:39 usr
drwx--x--x   5 root     root         1024 Nov  1 15:39 var

/bin:
-rwx--x--x   1 root     root        62660 Nov  1 15:39 ash
-rwx--x--x   1 root     root         9032 Nov  1 15:39 cat
-rwx--x--x   1 root     root        10276 Nov  1 15:39 chmod
-rwx--x--x   1 root     root         9592 Nov  1 15:39 chown
-rwx--x--x   1 root     root        23124 Nov  1 15:39 cp
-rwx--x--x   1 root     root        23028 Nov  1 15:39 date
-rwx--x--x   1 root     root        14052 Nov  1 15:39 dd
-rwx--x--x   1 root     root        14144 Nov  1 15:39 df
-rwx--x--x   1 root     root        69444 Nov  1 15:39 egrep
-rwx--x--x   1 root     root          395 Nov  1 15:39 false
-rwx--x--x   1 root     root        69444 Nov  1 15:39 fgrep
-rwx--x--x   1 root     root        69444 Nov  1 15:39 grep
-rwx--x--x   3 root     root        45436 Nov  1 15:39 gunzip
-rwx--x--x   3 root     root        45436 Nov  1 15:39 gzip
-rwx--x--x   1 root     root         8008 Nov  1 15:39 hostname
-rwx--x--x   1 root     root        12736 Nov  1 15:39 ln
-rws--x--x   1 root     root        15284 Nov  1 15:39 login
-rwx--x--x   1 root     root        29308 Nov  1 15:39 ls
-rwx--x--x   1 root     root         8268 Nov  1 15:39 mkdir
-rwx--x--x   1 root     root         8920 Nov  1 15:39 mknod
-rwx--x--x   1 root     root        24836 Nov  1 15:39 more
-rws--x--x   1 root     root        37640 Nov  1 15:39 mount
-rwx--x--x   1 root     root        12240 Nov  1 15:39 mt
-rwx--x--x   1 root     root        12932 Nov  1 15:39 mv
-r-x--x--x   1 root     root        12324 Nov  1 15:39 ps
-rwx--x--x   1 root     root         5388 Nov  1 15:39 pwd
-rwx--x--x   1 root     root        10092 Nov  1 15:39 rm
lrwxrwxrwx   1 root     root            3 Nov  1 15:39 sh -> ash
-rwx--x--x   1 root     root        25296 Nov  1 15:39 stty
-rws--x--x   1 root     root        12648 Nov  1 15:39 su
-rwx--x--x   1 root     root         4444 Nov  1 15:39 sync
-rwx--x--x   1 root     root       110668 Nov  1 15:39 tar
-rwx--x--x   1 root     root        19712 Nov  1 15:39 touch
-rwx--x--x   1 root     root          395 Nov  1 15:39 true
-rws--x--x   1 root     root        19084 Nov  1 15:39 umount
-rwx--x--x   1 root     root         5368 Nov  1 15:39 uname
```

TOC

```
-rwx--x--x   3 root     root        45436 Nov  1 15:39 zcat

/dev:
lrwxrwxrwx   1 root     root            6 Nov  1 15:39 cdrom -> cdu31a
brw-rw-r--   1 root     root      15,   0 May  5  1998 cdu31a
crw-------   1 root     root       4,   0 Nov  1 15:29 console
crw-rw-rw-   1 root     uucp       5,  64 Sep  9 19:46 cua0
crw-rw-rw-   1 root     uucp       5,  65 May  5  1998 cua1
crw-rw-rw-   1 root     uucp       5,  66 May  5  1998 cua2
crw-rw-rw-   1 root     uucp       5,  67 May  5  1998 cua3
brw-rw----   1 root     floppy     2,   0 Aug  8 13:54 fd0
brw-rw----   1 root     floppy     2,  36 Aug  8 13:54 fd0CompaQ
brw-rw----   1 root     floppy     2,  84 Aug  8 13:55 fd0D1040
brw-rw----   1 root     floppy     2,  88 Aug  8 13:55 fd0D1120
brw-rw----   1 root     floppy     2,  12 Aug  8 13:54 fd0D360
brw-rw----   1 root     floppy     2,  16 Aug  8 13:54 fd0D720
brw-rw----   1 root     floppy     2, 120 Aug  8 13:55 fd0D800
brw-rw----   1 root     floppy     2,  32 Aug  8 13:54 fd0E2880
brw-rw----   1 root     floppy     2, 104 Aug  8 13:55 fd0E3200
brw-rw----   1 root     floppy     2, 108 Aug  8 13:55 fd0E3520
brw-rw----   1 root     floppy     2, 112 Aug  8 13:55 fd0E3840
brw-rw----   1 root     floppy     2,  28 Aug  8 13:54 fd0H1440
brw-rw----   1 root     floppy     2, 124 Aug  8 13:55 fd0H1600
brw-rw----   1 root     floppy     2,  44 Aug  8 13:55 fd0H1680
brw-rw----   1 root     floppy     2,  60 Aug  8 13:55 fd0H1722
brw-rw----   1 root     floppy     2,  76 Aug  8 13:55 fd0H1743
brw-rw----   1 root     floppy     2,  96 Aug  8 13:55 fd0H1760
brw-rw----   1 root     floppy     2, 116 Aug  8 13:55 fd0H1840
brw-rw----   1 root     floppy     2, 100 Aug  8 13:55 fd0H1920
lrwxrwxrwx   1 root     root            7 Nov  1 15:39 fd0H360 -> fd0D360
lrwxrwxrwx   1 root     root            7 Nov  1 15:39 fd0H720 -> fd0D720
brw-rw----   1 root     floppy     2,  52 Aug  8 13:55 fd0H820
brw-rw----   1 root     floppy     2,  68 Aug  8 13:55 fd0H830
brw-rw----   1 root     floppy     2,   4 Aug  8 13:54 fd0d360
brw-rw----   1 root     floppy     2,   8 Aug  8 13:54 fd0h1200
brw-rw----   1 root     floppy     2,  40 Aug  8 13:54 fd0h1440
brw-rw----   1 root     floppy     2,  56 Aug  8 13:55 fd0h1476
brw-rw----   1 root     floppy     2,  72 Aug  8 13:55 fd0h1494
brw-rw----   1 root     floppy     2,  92 Aug  8 13:55 fd0h1600
brw-rw----   1 root     floppy     2,  20 Aug  8 13:54 fd0h360
brw-rw----   1 root     floppy     2,  48 Aug  8 13:55 fd0h410
brw-rw----   1 root     floppy     2,  64 Aug  8 13:55 fd0h420
brw-rw----   1 root     floppy     2,  24 Aug  8 13:54 fd0h720
brw-rw----   1 root     floppy     2,  80 Aug  8 13:55 fd0h880
brw-rw----   1 root     disk       3,   0 May  5  1998 hda
brw-rw----   1 root     disk       3,   1 May  5  1998 hda1
brw-rw----   1 root     disk       3,   2 May  5  1998 hda2
brw-rw----   1 root     disk       3,   3 May  5  1998 hda3
brw-rw----   1 root     disk       3,   4 May  5  1998 hda4
brw-rw----   1 root     disk       3,   5 May  5  1998 hda5
brw-rw----   1 root     disk       3,   6 May  5  1998 hda6
brw-rw----   1 root     disk       3,  64 May  5  1998 hdb
brw-rw----   1 root     disk       3,  65 May  5  1998 hdb1
brw-rw----   1 root     disk       3,  66 May  5  1998 hdb2
brw-rw----   1 root     disk       3,  67 May  5  1998 hdb3
brw-rw----   1 root     disk       3,  68 May  5  1998 hdb4
brw-rw----   1 root     disk       3,  69 May  5  1998 hdb5
brw-rw----   1 root     disk       3,  70 May  5  1998 hdb6
```

TOC

```
crw-r-----   1 root     kmem       1,   2 May  5  1998 kmem
crw-r-----   1 root     kmem       1,   1 May  5  1998 mem
lrwxrwxrwx   1 root     root          12 Nov  1 15:39 modem -> ../dev/ttyS1
lrwxrwxrwx   1 root     root          12 Nov  1 15:39 mouse -> ../dev/psaux
crw-rw-rw-   1 root     root       1,   3 May  5  1998 null
crwxrwxrwx   1 root     root      10,   1 Oct  5 20:22 psaux
brw-r-----   1 root     disk       1,   1 May  5  1998 ram
brw-rw----   1 root     disk       1,   0 May  5  1998 ram0
brw-rw----   1 root     disk       1,   1 May  5  1998 ram1
brw-rw----   1 root     disk       1,   2 May  5  1998 ram2
brw-rw----   1 root     disk       1,   3 May  5  1998 ram3
brw-rw----   1 root     disk       1,   4 May  5  1998 ram4
brw-rw----   1 root     disk       1,   5 May  5  1998 ram5
brw-rw----   1 root     disk       1,   6 May  5  1998 ram6
brw-rw----   1 root     disk       1,   7 May  5  1998 ram7
brw-rw----   1 root     disk       1,   8 May  5  1998 ram8
brw-rw----   1 root     disk       1,   9 May  5  1998 ram9
lrwxrwxrwx   1 root     root           4 Nov  1 15:39 ramdisk -> ram0
***  I have only included devices for the IDE partitions I use.
***  If you use SCSI, then use the /dev/sdXX devices instead.
crw-------   1 root     root       4,   0 May  5  1998 tty0
crw--w----   1 root     tty        4,   1 Nov  1 15:39 tty1
crw-------   1 root     root       4,   2 Nov  1 15:29 tty2
crw-------   1 root     root       4,   3 Nov  1 15:29 tty3
crw-------   1 root     root       4,   4 Nov  1 15:29 tty4
crw-------   1 root     root       4,   5 Nov  1 15:29 tty5
crw-------   1 root     root       4,   6 Nov  1 15:29 tty6
crw-------   1 root     root       4,   7 May  5  1998 tty7
crw-------   1 root     tty        4,   8 May  5  1998 tty8
crw-------   1 root     tty        4,   9 May  8 12:57 tty9
crw-rw-rw-   1 root     root       4,  65 Nov  1 12:17 ttyS1
crw-rw-rw-   1 root     root       1,   5 May  5  1998 zero

/etc:
-rw-------   1 root     root         164 Nov  1 15:39 conf.modules
-rw-------   1 root     root         668 Nov  1 15:39 fstab
-rw-------   1 root     root          71 Nov  1 15:39 gettydefs
-rw-------   1 root     root         389 Nov  1 15:39 group
-rw-------   1 root     root         413 Nov  1 15:39 inittab
-rw-------   1 root     root          65 Nov  1 15:39 issue
-rw-r--r--   1 root     root         746 Nov  1 15:39 ld.so.cache
***  ld.so.cache is created by ldconfig and caches library locations.
***  Many things break at boot time if ld.so.cache is missing.
***  You can either remake it after creating the bootdisk, or
***  include ldconfig on the bootdisk and run it from an rc.x script
***  to update the cache.
-rw-------   1 root     root          32 Nov  1 15:39 motd
-rw-------   1 root     root         949 Nov  1 15:39 nsswitch.conf
drwx--x--x   2 root     root        1024 Nov  1 15:39 pam.d
-rw-------   1 root     root         139 Nov  1 15:39 passwd
-rw-------   1 root     root         516 Nov  1 15:39 profile
-rwx--x--x   1 root     root         387 Nov  1 15:39 rc
-rw-------   1 root     root          55 Nov  1 15:39 shells
-rw-------   1 root     root         774 Nov  1 15:39 termcap
-rw-------   1 root     root          78 Nov  1 15:39 ttytype
lrwxrwxrwx   1 root     root          15 Nov  1 15:39 utmp -> ../var/run/utmp
lrwxrwxrwx   1 root     root          15 Nov  1 15:39 wtmp -> ../var/log/wtmp
```

```
/etc/pam.d:
-rw-------   1 root     root          356 Nov  1 15:39 other

/lib:
*** I have an ELF system with glibc, so I need the ld-2.so loader.
-rwxr-xr-x   1 root     root        45415 Nov  1 15:39 ld-2.0.7.so
lrwxrwxrwx   1 root     root           11 Nov  1 15:39 ld-linux.so.2 -> ld-2.0.7.so
-rwxr-xr-x   1 root     root       731548 Nov  1 15:39 libc-2.0.7.so
lrwxrwxrwx   1 root     root           13 Nov  1 15:39 libc.so.6 -> libc-2.0.7.so
lrwxrwxrwx   1 root     root           17 Nov  1 15:39 libcom_err.so.2 ->
libcom_err.so.2.0
-rwxr-xr-x   1 root     root         6209 Nov  1 15:39 libcom_err.so.2.0
-rwxr-xr-x   1 root     root       153881 Nov  1 15:39 libcrypt-2.0.7.so
lrwxrwxrwx   1 root     root           17 Nov  1 15:39 libcrypt.so.1 ->
libcrypt-2.0.7.so
-rwxr-xr-x   1 root     root        12962 Nov  1 15:39 libdl-2.0.7.so
lrwxrwxrwx   1 root     root           14 Nov  1 15:39 libdl.so.2 -> libdl-2.0.7.so
lrwxrwxrwx   1 root     root           16 Nov  1 15:39 libext2fs.so.2 ->
libext2fs.so.2.4
-rwxr-xr-x   1 root     root        81382 Nov  1 15:39 libext2fs.so.2.4
-rwxr-xr-x   1 root     root        25222 Nov  1 15:39 libnsl-2.0.7.so
lrwxrwxrwx   1 root     root           15 Nov  1 15:39 libnsl.so.1 -> libnsl-2.0.7.so
-rwx--x--x   1 root     root       178336 Nov  1 15:39 libnss_files-2.0.7.so
lrwxrwxrwx   1 root     root           21 Nov  1 15:39 libnss_files.so.1 ->
libnss_files-2.0.7.so
lrwxrwxrwx   1 root     root           14 Nov  1 15:39 libpam.so.0 -> libpam.so.0.64
-rwxr-xr-x   1 root     root        26906 Nov  1 15:39 libpam.so.0.64
lrwxrwxrwx   1 root     root           19 Nov  1 15:39 libpam_misc.so.0 ->
libpam_misc.so.0.64
-rwxr-xr-x   1 root     root         7086 Nov  1 15:39 libpam_misc.so.0.64
-r-xr-xr-x   1 root     root        35615 Nov  1 15:39 libproc.so.1.2.6
lrwxrwxrwx   1 root     root           15 Nov  1 15:39 libpwdb.so.0 ->
libpwdb.so.0.54
-rw-r--r--   1 root     root       121899 Nov  1 15:39 libpwdb.so.0.54
lrwxrwxrwx   1 root     root           19 Nov  1 15:39 libtermcap.so.2 ->
libtermcap.so.2.0.8
-rwxr-xr-x   1 root     root        12041 Nov  1 15:39 libtermcap.so.2.0.8
-rwxr-xr-x   1 root     root        12874 Nov  1 15:39 libutil-2.0.7.so
lrwxrwxrwx   1 root     root           16 Nov  1 15:39 libutil.so.1 ->
libutil-2.0.7.so
lrwxrwxrwx   1 root     root           14 Nov  1 15:39 libuuid.so.1 -> libuuid.so.1.1
-rwxr-xr-x   1 root     root         8039 Nov  1 15:39 libuuid.so.1.1
drwx--x--x   3 root     root         1024 Nov  1 15:39 modules
drwx--x--x   2 root     root         1024 Nov  1 15:39 security

/lib/modules:
drwx--x--x   4 root     root         1024 Nov  1 15:39 2.0.35

/lib/modules/2.0.35:
drwx--x--x   2 root     root         1024 Nov  1 15:39 block
drwx--x--x   2 root     root         1024 Nov  1 15:39 cdrom

/lib/modules/2.0.35/block:
-rw-------   1 root     root         7156 Nov  1 15:39 loop.o

/lib/modules/2.0.35/cdrom:
-rw-------   1 root     root        24108 Nov  1 15:39 cdu31a.o
```

TOC

```
/lib/security:
-rwx--x--x   1 root     root         8771 Nov  1 15:39 pam_permit.so

***  Directory stubs for mounting
/mnt:
drwx--x--x   2 root     root         1024 Nov  1 15:39 SparQ
drwx--x--x   2 root     root         1024 Nov  1 15:39 cdrom
drwx--x--x   2 root     root         1024 Nov  1 15:39 floppy

/proc:

/root:
-rw-------   1 root     root          176 Nov  1 15:39 .bashrc
-rw-------   1 root     root          182 Nov  1 15:39 .cshrc
-rw-------   1 root     root           47 Nov  1 15:39 .glintrc
-rwx--x--x   1 root     root          455 Nov  1 15:39 .profile
-rw-------   1 root     root         4014 Nov  1 15:39 .tcshrc

/sbin:
-rwx--x--x   1 root     root        23976 Nov  1 15:39 depmod
-rwx--x--x   2 root     root       274600 Nov  1 15:39 e2fsck
-rwx--x--x   1 root     root        41268 Nov  1 15:39 fdisk
-rwx--x--x   1 root     root         9396 Nov  1 15:39 fsck
-rwx--x--x   2 root     root       274600 Nov  1 15:39 fsck.ext2
-rwx--x--x   1 root     root        29556 Nov  1 15:39 getty
-rwx--x--x   1 root     root         6620 Nov  1 15:39 halt
-rwx--x--x   1 root     root        23116 Nov  1 15:39 init
-rwx--x--x   1 root     root        25612 Nov  1 15:39 insmod
-rwx--x--x   1 root     root        10368 Nov  1 15:39 kerneld
-rwx--x--x   1 root     root       110400 Nov  1 15:39 ldconfig
-rwx--x--x   1 root     root         6108 Nov  1 15:39 lsmod
-rwx--x--x   2 root     root        17400 Nov  1 15:39 mke2fs
-rwx--x--x   1 root     root         4072 Nov  1 15:39 mkfs
-rwx--x--x   2 root     root        17400 Nov  1 15:39 mkfs.ext2
-rwx--x--x   1 root     root         5664 Nov  1 15:39 mkswap
-rwx--x--x   1 root     root        22032 Nov  1 15:39 modprobe
lrwxrwxrwx   1 root     root            4 Nov  1 15:39 reboot -> halt
-rwx--x--x   1 root     root         7492 Nov  1 15:39 rmmod
-rwx--x--x   1 root     root        12932 Nov  1 15:39 shutdown
lrwxrwxrwx   1 root     root            6 Nov  1 15:39 swapoff -> swapon
-rwx--x--x   1 root     root         5124 Nov  1 15:39 swapon
lrwxrwxrwx   1 root     root            4 Nov  1 15:39 telinit -> init
-rwx--x--x   1 root     root         6944 Nov  1 15:39 update

/tmp:

/usr:
drwx--x--x   2 root     root         1024 Nov  1 15:39 bin
drwx--x--x   2 root     root         1024 Nov  1 15:39 lib
drwx--x--x   3 root     root         1024 Nov  1 15:39 man
drwx--x--x   2 root     root         1024 Nov  1 15:39 sbin
drwx--x--x   3 root     root         1024 Nov  1 15:39 share
lrwxrwxrwx   1 root     root           10 Nov  1 15:39 tmp -> ../var/tmp

/usr/bin:
-rwx--x--x   1 root     root        37164 Nov  1 15:39 afio
-rwx--x--x   1 root     root         5044 Nov  1 15:39 chroot
-rwx--x--x   1 root     root        10656 Nov  1 15:39 cut
```

```
-rwx--x--x   1 root     root        63652 Nov  1 15:39 diff
-rwx--x--x   1 root     root        12972 Nov  1 15:39 du
-rwx--x--x   1 root     root        56552 Nov  1 15:39 find
-r-x--x--x   1 root     root         6280 Nov  1 15:39 free
-rwx--x--x   1 root     root         7680 Nov  1 15:39 head
-rwx--x--x   1 root     root         8504 Nov  1 15:39 id
-r-sr-xr-x   1 root     bin          4200 Nov  1 15:39 passwd
-rwx--x--x   1 root     root        14856 Nov  1 15:39 tail
-rwx--x--x   1 root     root        19008 Nov  1 15:39 tr
-rwx--x--x   1 root     root         7160 Nov  1 15:39 wc
-rwx--x--x   1 root     root         4412 Nov  1 15:39 whoami

/usr/lib:
lrwxrwxrwx   1 root     root           17 Nov  1 15:39 libncurses.so.4 ->
libncurses.so.4.2
-rw-r--r--   1 root     root       260474 Nov  1 15:39 libncurses.so.4.2

/usr/sbin:
-r-x--x--x   1 root     root        13684 Nov  1 15:39 fuser
-rwx--x--x   1 root     root         3876 Nov  1 15:39 mklost+found

/usr/share:
drwx--x--x   4 root     root         1024 Nov  1 15:39 terminfo

/usr/share/terminfo:
drwx--x--x   2 root     root         1024 Nov  1 15:39 l
drwx--x--x   2 root     root         1024 Nov  1 15:39 v

/usr/share/terminfo/l:
-rw-------   1 root     root         1552 Nov  1 15:39 linux
-rw-------   1 root     root         1516 Nov  1 15:39 linux-m
-rw-------   1 root     root         1583 Nov  1 15:39 linux-nic

/usr/share/terminfo/v:
-rw-------   2 root     root         1143 Nov  1 15:39 vt100
-rw-------   2 root     root         1143 Nov  1 15:39 vt100-am

/var:
drwx--x--x   2 root     root         1024 Nov  1 15:39 log
drwx--x--x   2 root     root         1024 Nov  1 15:39 run
drwx--x--x   2 root     root         1024 Nov  1 15:39 tmp

/var/log:
-rw-------   1 root     root            0 Nov  1 15:39 wtmp

/var/run:
-rw-------   1 root     root            0 Nov  1 15:39 utmp

/var/tmp:
```

14. Sample utility disk directory listing.

```
total 579
-rwxr-xr-x   1 root     root        42333 Jul 28 19:05 cpio*
-rwxr-xr-x   1 root     root        32844 Aug 28 19:50 debugfs*
-rwxr-xr-x   1 root     root       103560 Jul 29 21:31 elvis*
-rwxr-xr-x   1 root     root        29536 Jul 28 19:04 fdisk*
-rw-r--r--   1 root     root       128254 Jul 28 19:03 ftape.o
-rwxr-xr-x   1 root     root        17564 Jul 25 03:21 ftmt*
-rwxr-xr-x   1 root     root        64161 Jul 29 20:47 grep*
-rwxr-xr-x   1 root     root        45309 Jul 29 20:48 gzip*
-rwxr-xr-x   1 root     root        23560 Jul 28 19:04 insmod*
-rwxr-xr-x   1 root     root          118 Jul 28 19:04 lsmod*
lrwxrwxrwx   1 root     root            5 Jul 28 19:04 mt -> mt-st*
-rwxr-xr-x   1 root     root         9573 Jul 28 19:03 mt-st*
lrwxrwxrwx   1 root     root            6 Jul 28 19:05 rmmod -> insmod*
-rwxr-xr-x   1 root     root       104085 Jul 28 19:05 tar*
lrwxrwxrwx   1 root     root            5 Jul 29 21:35 vi -> elvis*
```

The Linux Busmouse HOWTO

Chris Bagwell, cbagwell@sprynet.com

v1.91, 15 June 1998

This document describes how to install, configure and use a busmouse under Linux. It lists the supported mice and attempts to answer the most frequently asked questions that come up in newsgroups and mailing lists.

1. Introduction.

- 1.1 Disclaimer.
- 1.2 Feedback.
- 1.3 Acknowledgements.

2. Determining your mouse type.

- 2.1 Mouse interfaces.
- 2.2 Inport mice.
- 2.3 Logitech mice.
- 2.4 PS/2 mice.
- 2.5 ATI combo video/mice.
- 2.6 Mouse protocols.

3. Getting your mouse working.

- 3.1
- 3.2 Inport and Logitech mice.
- 3.3 ATI-XL mice.
- 3.4 PS/2 mice.
- 3.5 Configuring the kernel.
- 3.6 The mouse devices.

TOC

4. Using your mouse.

- 4.1 gpm.
- 4.2 XFree86.
- 4.3 XFree86 and gpm.

5. Still can't get your mouse going?

---

1. Introduction.

This document is a guide to getting your busmouse working with Linux. With the more advanced distributions available today setting up your busmouse is generally easy but when you do run into problems this document will help give you a better understanding of how to manually setup your busmouse hardware and software.

Busmouse support has been in the kernel for as long as I can remember, and hasn't changed much in a long time, so this document should be relevant to any version of Linux you're likely to have.

1.1 Disclaimer.

The information in this document is correct to the best of my knowledge, but there's a always a chance I've made some mistakes, so don't follow everything too blindly, especially if it seems wrong. Nothing here should have a detrimental effect on your computer, but just in case I take no responsibility for any damages incurred from the use of the information contained herein.

`Microsoft(R) is a Trademark of Microsoft Corporation.`

1.2 Feedback.

If you find any mistakes in this document, have any comments about its contents or an update or addition, send them to me at the address listed at the top of this howto.

1.3 Acknowledgements.

This howto has been, in the spirit of Linux, a community effort. Thanks goes out to Mike Battersby, `mib@deakin.edu.au` as he started this FAQ. Any errors are most likely added by me.

Many thanks go to Johan Myreen for the sections on the PS/2 mice, Robert T. Harris for help on the ATI-XL sections and Reuben Sumner for miscellaneous info and constructive criticism.

Thanks also to the multitudes of people who have sent me mouse information, fixes or words of encouragement.

2. Determining your mouse type.

There are two separate but important characteristics you will need to know about your mouse before you go on: what mouse interface it uses and what mouse protocol it uses. The interface is the hardware aspect of the mouse, taking into account things like which i/o ports it uses and how to check if it is installed. This is the part which the kernel is concerned with, so that it knows how to read data from the mouse. The protocol is the software aspect of the mouse. Applications need to know the protocol to interpret the raw mouse data they receive from the kernel.

2.1 Mouse interfaces.

The Linux kernel currently supports four different kinds of bus mouse interface : Inport (Microsoft), Logitech, PS/2 and ATI-XL. There is no sure-fire way of determining your mouse interface --- mouse developers generally do their own thing when it comes to standards. The following sections may help, otherwise you'll just have to make it up.

2.2 Inport mice.

Inport mice includes most of the old style Microsoft mice which are shaped like a bar of dove soap. U.S. users who have purchased Gateway computers should note that the mice that come with them are not Inport mice but PS/2 mice (see below). Inport mice generally connect to an interface card which plugs into the bus on your motherboard. If the plug which connects your mouse cord to the interface card is round, has 9 pins, and a notch in one side you likely have an Inport mouse.

As far as I can tell, apart from the ATI-XL, all ATI mice (such as those on the Graphics Ultra cards) are plain Inport mice.

2.3 Logitech mice.

Logitech mice in general appear almost exactly the same as Inport mice. They too connect to an interface card via a 9 pin mini-din connector. Hopefully, it will have come in a Logitech box or have ``Logitech'' printed on the connector card so that you can tell it actually is a Logitech mouse.

There are also some truly ancient Microsoft mice (ones with ball bearings on the bottom as well as the mouse ball and a DB9 connector) which also use the Logitech protocol.

TOC

2.4 PS/2 mice.

PS/2 mice aren't really bus mice at all. The PS/2 mouse interface is not on an expansion card, the mouse is connected to the PS/2 Auxiliary Device port on the keyboard controller. A PS/2 mouse port uses a 6-pin mini DIN connector, similar to the keyboard connector. Many laptops also use this kind of interface to their trackballs --- except for the connector, of course.

2.5 ATI combo video/mice.

ATI-XL mice are a variant of Inport mice, with some slight differences. They come on the ATI-XL combined video adaptor/mouse card. Unless you know you have an ATI-XL card (and thus an ATI-XL mouse), you probably don't have one of these. It is possible for ATI-XL mice to use either the ATI-XL or Inport kernel drivers, although the ATI-XL driver should give better results.

There is also an older ATI video adaptor/mouse card called either ATI VGA1024 or ATI VGA Wonder. These cards are setup the same as the ATI-XL but use the Logitech mouse protocol. For these mice the hardware is setup the same as for the ATI-XL but you setup the software (i.e. IRQ) the same as the Logitech mice.

2.6 Mouse protocols.

The PC world is full of different and conflicting mouse protocols. Fortunately, the choice for bus mice is considerable smaller than that for serial mice. Most Inport, Logitech and ATI-XL mice use the ``BusMouse'' protocol, although there are some ancient Logitech mice which use the ``MouseSystems'' protocol, and some even older Microsoft mice which use the "Logitech" protocol. PS/2 mice use the ``PS/2'' protocol.

---

3. Getting your mouse working.

Once you have figured out your mouse interface and protocol types, you're ready to proceed.

3.1 Setting the mouse interrupt.

Now, you'll need to know which interrupt number your mouse is using, and make sure it doesn't conflict with any other peripherals you have installed. That last part deserves to be repeated! **Make sure that it does not conflict with any other peripherals you have installed!**

You should make sure that your mouse is not trying to use the same interrupt as any of your other devices --- it is not possible for the mouse to share an interrupt under Linux, even though it may work fine under other operating systems. Check the documentation for all your peripherals to see which interrupt they use.

If you have Plug-n-Play cards, bear in mind that other operating systems may be initializing the card to an IRQ that is not in conflict with any other device but Linux doesn't do the same checking. It is up to you to make sure there are no IRQ conflicts between all of your equipment.

In most cases IRQ4 is used for the first serial port (`/dev/ttyS0`), IRQ3 for the second (`/dev/ttyS1`) (these are assuming you actually have such devices --- if you don't you can happily use their IRQ's), IRQ5 for some SCSI adaptors, and IRQ12 for some network cards. Having a card use IRQ12 is a big problem for machines with PS/2 ports as you are generally forced to use IRQ12 only for the PS/2 port.

For ATI-XL, Inport and Logitech mice the kernel default is to use IRQ5, so if you are stuck with a pre-compiled kernel (eg, CD-ROM users) you will have to use that. If you are using an Inport or Logitech mice with a newer kernel you may be able to pass a command line option to the kernel to tell it what interrupt to use without recompiling.

3.2 Inport and Logitech mice.

If you open up your computer's case and look at the card which your mouse plugs into, you should notice a block of jumpers on the card (hopefully labeled ``INTERRUPT'') with positions for interrupt (otherwise known as IRQ) numbers 2,3,4 and 5. To change the interrupt simply move the jumper from its current position onto the correct pair of pins.

```
***     MAKE SURE YOUR COMPUTER IS TURNED OFF    ***
***     BEFORE CHANGING THE JUMPERS AROUND!!     ***
```

TOC

3.3 ATI-XL mice.

ATI-XL and a few other ATI busmice have a software selectable IRQ - you should have received with your mouse a MS-DOS program (`VSETUP.EXE`) to set the IRQ. In order to do so you must (temporarily) boot MS-DOS and run this program. Note that the VSETUP program takes an optional parameter ``/70" to increase the vertical refresh rate (which results in less flicker). The VSETUP program also allows you to select either the primary or secondary mouse address - you should set this to the primary address or the kernel will not be able to detect your mouse.

Once VSETUP has been run you must perform a hard reset for the new configuration to take effect.

3.4 PS/2 mice.

The PS/2 mouse always uses IRQ12 -- there is no way of changing this (except with a soldering gun.) In the rare case that some other device is using IRQ12, you'll have to rejumper that peripheral to use another IRQ number.

3.5 Configuring the kernel.

In order for your busmouse to operate correctly you will need to configure your kernel to compile in busmouse support. If you are using a pre-compiled kernel then it often comes with support for all three busmouse included. This may still not be enough. The kernel could be trying to use the wrong interrupt or the detection can get confused and treat your mouse as the wrong type. When in doubt, try recompiling your kernel with only support for your mouse type and set it to use the correct interrupt.

Compiling the kernel.

Change to your kernel directory (here assumed to be (`/usr/src/linux`) and do a

```
make config
```

If you are unsure as to your mouse type, the first time you recompile the kernel you may wish to enable all of the busmouse options in the hope that the kernel will autodetect your mouse properly. People have mixed success with this: it doesn't always work, but on the other hand it might save you any further compiles.

Answer ``y" or "m" to the question pertaining to your type of busmouse interface and ``n" to all the other busmouse questions. Use the "m" option if you have your system setup to support loading kernel modules if you do not or do not know what that means then it will be safe to always answer "y" to have the support directly compiled into your kernel.

As an example, if you have an Inport mouse you should answer ``y" to

```
Microsoft busmouse support
```

and ``n" to all other busmouse questions. Answer the non-mouse related questions as you usually would.

TOC

To compile the kernel with PS/2 mouse support answer ``y'' to the question.

```
PS/2 mouse (aka "auxiliary device") support
```

The PS/2 mouse driver actually supports two kinds of devices: the standard PS/2 Auxiliary Device controller and a special PS/2 mouse interface chip from Chips & Technologies which is used in the Texas Instruments Travelmate and Gateway Nomad laptops. To compile in support for the trackballs on these computers, answer ``y'' to the

```
C&T 82C710 mouse port support (as on TI Travelmate)
```

question. Note that you will still have to answer ``y'' to the question about the standard PS/2 driver to even get a chance to answer this question, since the 82C710 driver is actually an add-on to the standard PS/2 mouse driver.

When configured both for a standard PS/2 mouse device and the 82C710 device, the driver first tries to locate a 82C710 chip at boot time. Failing this, the standard driver is used instead, so using a kernel configured for both types of interface on a machine with a standard PS/2 mouse port should work too. However, there has been one report of a falsely detected 82C710 chip, so to be on the safe side do not configure in support for the 82C710 if you don't need it.

You will now need to tell the kernel what interrupt your mouse uses. You can skip this step if your using a PS/2 mouse as it always uses IRQ 12.

If you have a Logitech, Inport mouse, or an ATI mouse that uses the Logitech protocol, edit the file `/usr/src/linux/include/linux/busmouse.h` and change the line which says

```
#define MOUSE_IRQ 5
```

to reflect the interrupt number for your mouse (see the section Setting the mouse interrupt for details on finding your interrupt number).

If you have an ATI-XL mouse, edit the file `/usr/src/linux/drivers/char/atixlmouse.c` and change the line which says

```
#define ATIXL_MOUSE_IRQ 5
```

to reflect your mouse's interrupt number.

Due to the vagaries of the PC architecture, if you have set your mouse to use interrupt 2, you must set the #define to use interrupt 9.

Examples

For a mouse on interrupt 3, you should change the line to read

```
#define MOUSE_IRQ 3
```

For a mouse on interrupt 2, you should change the line to read

```
#define MOUSE_IRQ 9
```

Next, compile your kernel as per the instructions which come with it, and boot from the new kernel. You should now have the busmouse support correctly compiled in.

selection.

It appears that in older kernels you had to compile in support to use the program selection (a program that allows you to cut and past from virtual consoles the same way you do under X). This option does not appear in modern kernels and the program selection has generally been replaced with the program gpm (see the section gpm for more details).

If you are working with an old kernel then you may wish to set this option to ``y'' regardless of your mouse type so that you may use the selection program.

Changing interrupts with newer kernels.

The steps to compile into the kernel what interrupt it uses works with any version of the kernel to date. Newer kernels (starting somewhere in the 2.x.x's) allow you to pass arguments to the kernel during load time using something like LILO or LOADLIN to specify the interrupt number for Logitech and Microsoft Inport mice. This can be a real time saver as you do not need to recompiler your kernel (or know how to). If you've configured your kernel to load the mouse drivers as modules then you will need to pass this information when loading the module.

You can add the following options to your boot line in LILO to change interrupt:

```
bmouse=3  (Logitech Busmice)
msmouse=3 (for Microsoft Inport mice)
```

Substitute the 3 with your mouse's actual interrupt. An example of using this with lilo is:

```
LILO:linux msmouse=3
```

You can consult your LILO or LOADLIN docs to see how to add this type information to their configuration files so that you do not need to type it.

When using modules you can manually set the interrupt by using insmod as follows:

```
insmod msbusmouse.o mouse_irq=3    (use for Inport mice)
insmod busmouse.o mouse_irq=3      (use for Logitech mice)
```

If your system uses kerneld to auto load modules, you can edit your /etc/conf.modules or /etc/modules.conf file, which ever your system uses, and add one of the following lines.

```
options msbusmouse mouse_irq=3
options busmouse mouse_irq=3
```

3.6 The mouse devices.

Mice under Linux are accessed via the devices in the `/dev` directory. The following table gives a list of interface types and which device you should use.

```
INTERFACE        DEVICE        MAJOR    MINOR
-----------------------------------------------
Logitech        /dev/logibm      10       0
PS/2            /dev/psaux       10       1
Inport          /dev/inportbm    10       2
ATI-XL          /dev/atibm       10       3
```

Note: If you are using your ATI-XL mouse with the Inport driver, you should use the `inportbm` device, not the `/dev/atibm` device.

The major and minor entries are the device numbers for that particular device.

If you find that you do not have these devices, you should create them first. To do so, execute the following as root.

```
mknod /dev/logimm    c 10 0
mknod /dev/psaux     c 10 1
mknod /dev/inportbm  c 10 2
mknod /dev/atibm     c 10 3
```

Note: Some time in the (progressively less) recent history of Linux the names for the busmouse devices have changed. The following device names have been superceded by those above and should be removed: `bmousems, bmouseps2, bmouseatixl, and bmouselogitech.`.

Many people like to create a symbolic link from their mouse device to `/dev/mouse` so that they don't have to remember which device they need to be using. If you have one of the current Linux distributions you will almost certainly find that you have such a link. If you have such a link, or create one, you should make sure that it is pointing to the correct device for your mouse.

---

4. Using your mouse.

This section deals with the general use of your mouse with various applications.

4.1 gpm.

Gpm is a program which allows you to do mouse based 'cut- and-paste' between Linux virtual consoles, much like you can under X, and is a good way of testing your mouse out. The current version of gpm is gpm-1.13.tar.gz and can be found at your friendly Linux FTP site (such as sunsite.unc.edu), and contains instructions for getting it compiled. Some Linux distributions, such as Red Hat, come with a precompiled gpm binary.

When invoking gpm, use the -t switch to indicate which protocol your mouse is using and the -m option to indicate which mouse device you are using. Three protocols useful for most busmice are logi, bm, and ps2. The default for mouse device is to use `/dev/mouse`, so you can omit the -m option if you have the appropriate symbolic link. An example for a Microsoft Inport mouse is:

```
gpm -t bm
```

or if you use the PS/2 protocol:

```
gpm -t ps2
```

You should then be able move your mouse and see a block move around the screen and also be able to cut and paste text between virtual consoles using the mouse buttons. Read the documentation with gpm, or do a ``man gpm'' for more information on how to operate it.

4.2 XFree86.

To use your busmouse under XFree86, you will need to set your mouse protocol type in your Xconfig file. If you have a BusMouse protocol mouse, your Xconfig should contain (including the quotes)

```
Section "Pointer"
  Protocol "Busmouse"
  Device   "/dev/mouse"

  # Any other options such as Emulate3Buttons
EndSection
```

For PS/2 mice change the protocol line to:

```
  Protocol        "PS/2"
```

TOC

If you have a two button mouse, it should also contain the line

```
Emulate3Buttons
```

which will allow you to emulate the use of the middle mouse button by pressing both mouse buttons simultaneously. All other mouse related lines, such as ``BaudRate'' and ``SampleRate'' should be commented out, as these have no effect on bus mice.

4.3 XFree86 and gpm.

For a long period of the kernel developement, it was not possible to share busmice between processes. Because of this it was hard to run both XFree86 and gpm at the same time. If you try to run X with gpm running and you get errors like the following then you know you are using one of these older kernels.

```
Fatal server error:
Cannot open mouse (Device or resource busy)
```

There are two meathods of getting gpm working with XFree86 with these kernels. The first is to kill any copy of gpm you have running before you start up XFree86. The second is to use gpm's "repeater" option (it takes mouse data and repeats the information to multiple applications).

I would recommend upgrading your kernel if possible so that you can share busmice between processes. For this document, I will only explain the simplest meathod of using XFree86 and gpm together with older kernels. Please see gpm's documentation if you would like to use the repeater meathod.

Gpm allows you to terminate running copies of itself by executing:

```
gpm -k
```

This should be done before starting up X11. Take whatever script you use to start up your X session, such as `startx`, and add the above command to the top of the script so that gpm is shut down automatically. You may wish to also put a command that restarts gpm at the bottom of the script so that it restarts upon exiting your X session.

---

5. Still can't get your mouse going?

So you've read through this howto a dozen times, done everything exactly as you think you should have, and your mouse still doesn't work? The best advice I can give you is this: experiment. Sure, it's a pain in the posterior, but in the end the only way to find out what is going to work with your mouse is to try all of the alternatives until you have success.

As always, if there is something you don't understand, try reading the manual page first and see if that helps. If you have a specific question, or a problem you think I might be able to help with, feel free to contact me at the address listed at the top of this howto, and I'll see if I can help you out or point you to someone who can.

The `comp.os.linux.setup` newsgroup or `comp.os.linux.hardware` is the appropriate forum for discussion and/or questions regarding setup --- please don't post questions to other groups, and especially don't crosspost questions to two or more of the Linux groups, they are more than cluttered enough as it is! When posting, you will get a much better response (and much fewer flames) if you use appropriate Subject: and Keywords: lines. For example:

```
Subject: BUSMICE - Gateway 2000 mouse wont work.
Keywords: mouse busmouse gateway
```

CD-Writing HOWTO

Winfried Trümper <winni@xpilot.org>

v2.8.2, 7 August 1999

This document explains how to write CD-ROMs under Linux.

1. Introduction

- 1.1 Copyright, license and terms of usage
- 1.2 Suggested readings
- 1.3 Terminology ... lasers at maximum ... fire!
- 1.4 Supported CD-Writers
- 1.5 Supported "features"
- 1.6 Mailinglists
- 1.7 Availability

2. Setup the Linux-system for writing CD-ROMs

- 2.1 Set up the Linux-kernel
- 2.2 Hardware and devicefiles
- 2.3 Get the user-software for burning CD-Rs

3. Burning CD-Rs

- 3.1 Writing CD-ROMs (pure data)
- 3.2 Writing audio CDs
- 3.3 Mixed mode CD-ROMs

TOC

4. Dear Winfried,...

- 4.1 How sensitive is the burning process?
- 4.2 Has fragmentation a bad impact on the throughput?
- 4.3 Is it possible to store the CD-image on an UMSDOS-filesystem?
- 4.4 Isn't there some way to get around the ISO-9660 limitations?
- 4.5 How to read the tracks from audio CDs?
- 4.6 How to probe for SCSI-devices after boot?
- 4.7 Is it possible to make a 1:1 copy of a data CD?
- 4.8 Can Linux read Joliet CD-ROMs?
- 4.9 How do I read/mount CD-ROMs with the CD-writer?
- 4.10 How to put even more data on the CD-R?
- 4.11 How to make bootable CD-ROMs?
- 4.12 How to make CD-ROMs writable somehow?
- 4.13 Is it possible to use several writers at once?
- 4.14 Which media is the best?
- 4.15 What about Solaris, *BSD, AIX, HP-UX, etc.?
- 4.16 Where to store the local configuration permanently?
- 4.17 How can the CD-info be retrieved?
- 4.18 What about re-writing
- 4.19 How to create a multi-session CD?

5. Troubleshooting

- 5.1 It doesn't work: under Linux
- 5.2 Error-message: No read access for 'dev=0,6,0'.
- 5.3 It doesn't work: under DOS and friends
- 5.4 SCSI errors during the burning phase

6. Credits

1. Introduction

Many people use Linux to burn CD-ROMs, because it is reliable and easy. No bluescreens while burning and no headaches about getting the right combination of hard- and software. It just works once properly set up. The CD-writing HOWTO explains the setup, how to put data on the media and gives some interesting applications kindly submitted by the readers.

1.1 Copyright, license and terms of usage

Copyright Winfried Trümper 1996,1997,1998,1999. All rights reserved.

Redistribution and use, with or without modification, are permitted provided that the name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

The author disclaims all warranties with regard to this document, including all implied warranties of merchantability and fitness for a certain purpose; in no event shall the author be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use of this document.

Short: read and use at your own risk. No money-back guarantee.

1.2 Suggested readings

The CD-R FAQ is a general FAQ about compact-disk recordables (CD-R), CD-writers and the required software.

As most CD-writers can be used to read CD-ROMs, too, you may want to read the *Linux CD-ROM HOWTO*, the *Linux SCSI HOWTO* and the *Linux Kernel HOWTO*.

1.3 Terminology ... lasers at maximum ... fire!

CD-ROM stands for *Compact Disc Read Only Memory*, a storage medium utilizing an optical laser to sense microscopic pits on a silver shimmering disk. The silver shimmering comes from an aluminized layer which is the carrier. The pits represent the bits of the information (in some way) and are so petite that some billions of them fit on the disc. Thus a CD is a mass-storage medium.

The term *CD-R* is a short form of *CD-ROM recordable* and refers to a CD that doesn't have those "microscopic pits" on its surface. Thus its empty. Instead of the silver aluminium layer a CD-R has a special film (colored) into which "microscopic pits" can be burned in. This is done by giving the laser which normally only senses the pits a little bit more power so he burns the pits. This action can only be taken **once** on a CD-R, although you can leave out some areas for later writing, creating a so called *multi-session CD*.

TOC

The *CD-ROM rewritable* (short: *CD-RW*) was developed to come around the limitation of CD-R media. Here the laser can not even burn pits into the media, but it can also melt the media back into its original state. This is possible, because the laser does not really burn holes into the media - you do not see a puff of smoke while burning and no voice from above tells you "target destroyed". What the laser really does is to trigger a phase change, thus the name "phase change devices" for CD-RW-writers. An example for a very similar technique used in daily life is the cleaning car (called Zamboni) in an ice-hockey stadion: during the pauses of the game, the car drives over the ice and fills the scratches in the ice (-> bits containing some sort of information about the game) with melted splinter.

This HOWTO deals with the task of writing CD-Rs and CD-RWs. Welcome on board, captain.

1.4 Supported CD-Writers

You can assume that all newer IDE/ATAPI- and SCSI-writers work under Linux. Newer drives are mostly MMC-compliant and thus supported. However, some people want to get a warm and fuzzy feeling by reading the exakt model of their writer in some sort of compatibilty list and that is the reason why I didn't threw the following out of the HOWTO:

Here is a comprehensive summary of drives reported to work with cdrecord:

```
Acer:           CDRW 6206A
BTC:            BCE 621E (IDE)
Compro:         CW-7502, CW-7502B
Creative:       RW 4224E, MK 4211
Dysan:          CRW-1622
Elite:          Elite b444.41
Grundig:        CDR 100 IPW
Guillemot:      Maxi CD-R 4X/8X
HP:             SureStore 4020i, SureStore 6020i,
                C4324, C4325
                CD-Writer+ 7100, CD-Writer+ 7200i, CD-Writer+ 7500e,
                CD-Writer+ 8100i, CD-Writer+ 8110i,
Hi-Val:         CDD 2242, CDD-3610,
JVC:            XR-W2001, XR-W2010, XR-W2042, R-2626
Kodak:          PCD 200, PCD 225, PCD 260, PCD 600
Matsushita:     CW-7502
Memorex:        CRW-620, CRW-1622, CRW-2224
Microboards:    PlayWrite 2000, PlayWrite 4000RW, PlayWrite 4001RW
MicroNet:       MasterCD Plus 4x4, MasterCD Plus 4x6
Mitsubishi:     CDRW-226
Mitsumi:        CR-2401-TS, CR-2600 TE,  CR-2801 TE,
                CR-4801 TE, CR-4802 TE
Nomai:          680.RW
OTI:            CDRW 965
Olympus:        CDS 615E, CDS 620E
Optima:         DisKovery 650 CD-R
```

TOC

```
OTI:              CDRW 965
Panasonic:        CW-7502, CW-7582
Philips:          CDD-521/10, CDD-522
                  CDD-2000, CDD-2600, CDD-3600, CDD-3610
                  Omniwriter 26, Omniwriter 26A
Plasmon:          CDR 480, CDR 4220, RF-4100, RF-4102, CDR 4400
Plextor:          CDR PX-24 CS, PX-412 C, PX-R412 C
                  PX-R810Ti, PleXwriter 412C
Procom:           PCDR 4
Ricoh:            RO-1420C+, MP 1420C, MP 6200S, MP 6201S, MP 7040A
Sanyo:            CRD-R24S
Smart and
Friendly:         CD-RW226, CD-R1002, CD-R1002/PRO, CD-R1004,
                  CD-R2004, CD-R2006 PLUS, CD-R2006 PRO,
                  CD-R4000, CD-R4006, CD-R4012
Sony:             CDU 920S, CDU 924, CDU 926S, CDU-928E,
                  CDU 948S, CDRX 100E, CDRX 120E
Taiyo Yuden:      EW-50
TEAC:             CD-R50S, CD-R55S, CDR-55S, CDR-56S-400
Traxdata:         CRW 2260, CDR 4120, CDR 4120 Pro, CDRW 4260,
Turtle Beach:     2040R
WPI (Wearnes):    CDRW-622, CDR-632P
YAMAHA:           CDR-100, CDR 102, CDR-200, CDR-200t, CDR-200tx
                  CDR-400, CDR-400c, CDR-400t, CDR-400tx, CDR-400Atx
                  CRW-2260, CRW-2260t,
                  CRW-4250tx, CRW-4260 t, CRW-4260 tx, CRW-4261,
                  CRW-4416 S
```

Table 1: CD-writers supported under Linux

The detailed list of models which have been reported to work or not to work under various Unix-like operating systems is available online from http://www.guug.de:8080/cgi-bin/winni/lsc.pl.

If your hardware isn't supported you can still use Linux to create an image of the CD. You may wish to do so because most DOS-software cannot deal with RockRidge-Extensions. In a second step, you use DOS- or Macintosh-software to write the image to the CD-R.

1.5 Supported "features"

There are two classes of utilities for writing CD-Rs: the hardware-drivers and the data-formatters. The hardware drivers support the following features:

Supported Feature	cdwrite-2.1	cdrecord-1.6	cdrdao
IDE/ATAPI	no	yes	yes
Parallel Port	no	yes	yes
CD-RW	no	yes	yes

TOC

Audio CD	yes	yes	yes
Data CD-ROM	yes	yes	partial
Multisession	partial	yes	no
TAO (Track at once)	yes	yes	yes
DAO (Disk at once)	no	no	yes

Table 2:

`cdwrite` is an old software only referenced for completeness. Please use `cdrecord` instead, as it supports a wider range of hardware and has significantly more features. The main purpose of cdrdao is to create audio CDs without two seconds of silence between the tracks (writing in disk-at-once mode).

The data-formatters organise the data on the media ("put a filesystem on it").

Feature	mkisofs	mkhybrid
ISO 9660	yes	yes
RockRidge	yes	yes
El Torito	yes	yes
HFS	no	yes
Joliet	yes	yes
Multisession	yes	yes

Table 3:

The most obvious difference of the ISO 9660 filesystem compared to the Extended-2 filesystem is, that you can't modify files once they are written to the media. The limitations of the ISO-9660-filesystem are:

- only 8 levels of sub-directories allowed (counted from the top-level directory of the CD) (use RockRidge Extensions to enlarge this number)
- maximum length for filenames: 32 characters
- 650 MB capacity

RockRidge is an extension to allow longer filenames and a deeper directory hierarchy for the ISO-9660 filesystem. These extensions are not available when reading the CD-ROM under DOS or the Windows-family of operating systems. When reading a CD-ROM with RockRidge extensions under Linux, all the known properties of files like owner, group, permissions, symbolic links appear.

El Torito can be used to produce bootable CD-ROMs. The BIOS of your PC must support this feature, too. Roughly speaking, the first 1.44 Mbyte of the CD-ROM contain a floppy-disk image supplied by you. This image is treated like a floppy by the BIOS and booted from.

HFS lets a Macintosh read the CD-ROM as if it were an HFS volume (the native filesystem for MacOS).

Joliet brings long filenames (among other things) to newer variants of Windows (95, 98, NT). The author knows of no tool to have long filenames under plain DOS or Windows 3.11.

Section 2.8 lists the availability of the mentioned software.

TOC

1.6 Mailinglists

If you want to join the development team (with the intention to actively *help* them), send e-mail to cdwrite-request@other.debian.org and put the word `subscribe` in body of the message.

1.7 Availability

The newest version of this document is always available from http://www.guug.de/~winni/linux/.

---

2. Setup the Linux-system for writing CD-ROMs

If you succeed setting up your hardware and the rest of your Linux-system, the command `cdrecord -scanbus` shows you a list of devices available for writing. The goal of this chapter is to guide you in setting up your Linux-system, so that you finally end up seeing something like:

```
shell> cdrecord -scanbus
Cdrecord release 1.7a1 Copyright (C) 1995-1998 Jörg Schilling
scsibus0:
         0) 'Quantum ' 'XP34300         ' 'F76D' Disk
         1) 'SEAGATE ' 'ST11200N        ' '8334' Disk
         2) *
         3) 'TOSHIBA ' 'MK537FB/        ' '6258' Disk
         4) 'WANGTEK ' '5150ES SCSI-36  ' 'ESB6' Removable Tape
         5) 'EXABYTE ' 'EXB-8500-85QUE  ' '0428' Removable Tape
         6) 'TOSHIBA ' 'XM-3401TASUNSLCD' '3593' Removable CD-ROM
         7) *
scsibus1:
       100) 'Quantum ' 'XP31070W        ' 'L912' Disk
       101) *
       102) *
       103) 'TEAC    ' 'CD-R55S         ' '1.0H' Removable CD-ROM
       104) 'MATSHITA' 'CD-R   CW-7502  ' '4.02' Removable CD-ROM
       105) *
       106) 'YAMAHA  ' 'CDR400t         ' '1.0d' Removable CD-ROM
       107) *
```

Listing 1: Detecting devices on your SCSI-bus

The example was provided by Jörg Schilling and shows a total of four CD-writers. Please note that -scanbus also reports other devices, e.g. regular CD-ROMs and harddisk drives. The last column gives the SCSI description of the device, from which you cannot distinguish ordinary CD-ROM drives from those with burning capability. But the product identification often has hints about the feature in form of a R, -R or -RW.

Don't worry about the keyword "scsibus". If you don't have one, you just read the following about how to load the Linux kernel-module to emulate a SCSI hostadaptor. Same is goes for owners of a writer working over the parallel port. As ATAPI means "speak the SCSI protocol over the IDE-bus", it is more or less straightforward to make IDE/ATAPI-devices appear as if they were real SCSI-devices. But once you access your writer as a SCSI-device with the CD-writing software, you must also access your writer as a SCSI-device, when you use it as a regular CD-ROM. Usally mounting of filesystems or playing audio CDs will go through /dev/scd0 then.

2.1 Set up the Linux-kernel

TOC

The Linux-kernel can be equipped with drivers for various features. You can compile the drivers into the kernel-image statically or you can compile them as a module for on-demand loading. The last method is preferred for drivers not essential for bringing your Linux-system into life, because your kernel will be smaller and faster then. However, some drivers are essential for the system to come up and you can't compile them as a module. Example: if your system lives on a IDE-harddisk, you must have the driver for IDE-harddisks in the kernel - not as a module.

There are three different types of CD-writers: SCSI, IDE/ATAPI and external writers for the parallel port. Table 4 shows how to configure the Linux-kernel for those hardware types. The first column of the table is the section of the kernel configuration menu, where you can find the setting. The second column is the description of the feature (taken from the kernel configuration menu, too). The third column gives the name of the resulting module. The columns named SCSI, IDE and PP contain the necessary options for the associated hardware (PP = parallel port).

Sect.	Description	Module	SCSI	IDE	PP
BLOCK	Enhanced IDE/MFM/RLL...			Y	
BLOCK	IDE/ATAPI CDROM	ide-cd		M	
BLOCK	SCSI hostadaptor emulation	ide-scsi		M	
BLOCK	Loopback device	loop	M	M	M
PARIDE	Parallel port IDE device	paride			Y/M
PARIDE	Parallel port ATAPI CD-ROMs				M
PARIDE	Parallel port generic ATAPI				M
PARIDE	(select a low-level driver)				Y
SCSI	SCSI support	scsi_mod	Y/M	Y/M	
SCSI	SCSI CD-ROM support	sr_mod	Y/M	Y/M	
SCSI	Enable vendor-specific		Y	Y	
SCSI	SCSI generic support	sg	Y/M	Y/M	
SCSI	(select a low-level driver)		Y		
FS	ISO 9660 CDROM filesystem	iso9660	Y	Y	Y
FS	Microsoft Joliet cdrom...	joliet	M	M	M

Table 4:

Y stands for yes and means you should put the beast into the kernel. M stands for module and means you should or must compile this feature as a module. Y/M gives you the option to decide between both (order indicates choices with less potential problems). Empty settings don't need to be modified and not changing them increases the chance that the resulting kernel will work for you (if it did before...).

The loopback device is an optional feature, which helps you to test the image before writing it to the media. If you want to be able to read CD-ROMs, you need support for the ISO 9660 filesystem. This driver automatically includes RockRidge Extensions. If you want to have MS Joliet cdrom extensions, compile them as a module and load them. In any case, you need a low-level driver for your hardware. Low-level refers to the driver, which interacts directly with the hardware. For SCSI and the parallel port, there are a lot of low-level

drivers available.

Especially in environments, where SCSI- and ATAPI-devices are mixed, you better build most things as modules. To make your life with modules easier, you can add the following lines to /etc/conf.modules:

```
alias   scd0 sr_mod                  # load sr_mod upon access of scd0
alias   scsi_hostadapter ide-scsi    # SCSI hostadaptor emulation
options ide-cd ignore=hdb            # if /dev/hdb is your CD-writer
```

Some settings for using IDE/ATAPI-writers

The aliases provide alternate names for the same module and are not essential. The options provide a way to make options for module loading permanent. To learn about different options for different modules read on.

Description	command
Show active drivers	cat /proc/devices
List active modules	cat /proc/modules
List available modules	modprobe -l
Load a precompiled module	modprobe
Remove a module	modprobe -r

Table 4:

If you need a module not available on your system, you must recompile and reinstall the kernel. Please read the Kernel-HOWTO for more information upon this. The most recent information about handling of kernel-modules can be read in the Module-HOWTO. If you have success in building or finding the required modules, either add their names to /etc/modules or let an automatic facility (kmod, kerneld) load them. Consult the documentation of your Linux-distribution if you are unsure.

If you are lucky, then just install a package called "Linux Kernel source", which was shipped with your distribution. Afterwards issue the following commands:

```
cd /usr/src/linux
make menuconfig  # follow instructions and menu
make dep
make zImage      # or "make bzImage"
# insert a floppy-disk before proceding
dd if=arch/i386/boot/zImage of=/dev/fd0
make modules
make modules_install
```

Table 4:

This will produce a bootable floppy disk. If you are really clueless about this whole stuff, you better get help from a friend before you completely mess up your system. Remember the kernel is the heart of the system.

TOC

Special hints up to kernel version 2.2.9

Owners of ATAPI-CD-writers please head up: if your kernel detects your writer as an ATAPI-device during system startup, you have no chance of getting it to work as a CD-writer under Linux. Once the IDE/ATAPI-driver grabs the device, the SCSI-driver can no longer do so. Please recompile a kernel with the CD-ROM-stuff as a module like it is shown in the table above.

With 2.2 and above, you can control the greediness of the ide-cd driver with the parameter ignore=. The driver won't grab the specified device, so it will be available when the SCSI-hostadapotr emulation is activated. Example: use "modprobe ide-cd ignore=hdb", if your ATAPI-writer is hdb and you want the ide-scsi-driver to treat hdb as a scsi-device (usally sr0). There is no way to specify this on the kernel-commandline (like in 2.0).

Up to kernel-version 2.2.9, don't enable CONFIG_SCSI_MULTI_LUN ('probe for multiple luns') and ide-scsi support at the same time, there is an ide-scsi bug that prevents this.

Some users reported conflicts with pre-compiled binaries and the 2.2-release. This is a problem of the Linux-kernel. Solutions:

- It may go away if you recompile cdrecord so that it adopts to the actual values in linux/include/scsi/sg.h. These values must be the values of the runing kernel due to the dumb sg user interface.
- It may be caused by a loabdable sg driver which cannot get enough memory that is suitable for DMA (Kernel releases up to 2.2.5).

Special hints with kernel version 2.2.10 and later

It was reported to me, that 2.2.10 provides you with both, the option from the 2.0-series, and the option from the series before 2.2.9. Personally the author prefers the more positive strategie of telling the kernel which device should be managed by the ide-scsi driver like it is done in 2.0.37.

Special hints for the 2.0-series

Kernels prior to 2.0.31 have several bugs and can't read the last file on a CD-ROM. Please upgrade to 2.0.37. For this version, the misleading feature name "SCSI emulation" has been renamed into "SCSI hostadaptor emulation", which is more accurate.

With 2.0.36 and above, you can select which driver (IDE or SCSI) to use for which ATAPI-device. To be more precise, you can add the feature "SCSI emulation" to your kernel and activate the emulation for specific devices on the kernel commandline (before/while booting). For example, if your CD-writer has the devicefile /dev/hdb, then add the statement `hdb=ide-scsi`. Compared to the 2.2 series, this is a more positive way of thinking.

```
image=/boot/zImage-2.0.37
  label=Linux
  read-only
  append="hdb=ide-scsi"
```

Listing 2: Example configuration for lilo (/etc/lilo.conf)

```
linux "Linux 2.0.37" {
        image=/boot/zImage-2.0.37
```

```
        cmdline= root=/dev/hda5 readonly hdb=ide-scsi
}
```

Listing 3: Example configuration for chos (/etc/chos.conf)

Please note that this is different from 2.2, where you can only pass the options while loading the modules.

2.2 Hardware and devicefiles

After you have installed all necessary drivers, boot the new kernel. It should have all the features you need for the following sections.

[Users of devfs need not to care about the following steps of manually setting up device files. If you use devfs, everything will automacigally appear on the right place.]

Goto the `/dev`-directory and check for *loopback devices*. Its not critical if you don't have those devices, but its convenient if you do (see 3.5). If you already have them, the `ls`-command should show `loop0-loop7`:

```
shell> cd /dev
shell> ls loop*
loop0 loop1 loop2 loop3 loop4 loop5 loop6 loop7
```

Listing: devicefiles for loopback

If you don't have those device-files, then create them by using the `/dev/MAKEDEV`-script:

```
shell> cd /dev/
shell> modprobe loop   # load the module called "loop"
shell> ./MAKEDEV loop
```

Listing: commands to create loopback devices

The last command only succeeds if you have the `loop`-module in your kernel. What you read so far about handling of loop-devicefiles can be applied to all other devicefiles (sg*, pg*, hd*, ...) and is not explicitely mentioned in the following.

In case your Linux-distribution misses the script /dev/MAKEDEV, then create the devices manually with this loop:

```
for i in 0 1 2 3 4 5 6 7
do
        mknod /dev/loop$i c 7 $i
done
```

IDE/ATAPI CD-writers

ATAPI means "SCSI-commands over the IDE-bus". In other words, your IDE/ATAPI-drives speak the SCSI-protocol (with some limitations) and thus can act as if they were SCSI-drives.

Under Linux you can make your IDE-writer appear as a SCSI-devices using the kernel-features "SCSI (hostadaptor) emulation". So activate the associated kernel module ide-scsi like described above and proceed

as if you had a SCSI-writer.

SCSI CD-Writers

Please make sure that your writer is recognized by the BIOS of your computer. It makes no sense to proceed if your computer does not accept the hardware (the fact that it doesn't spit it out should not be interpreted as a sign of confirmation; a message on the screen is required).

If you plan to connect your SCSI-drive to your parallel port (not to confuse with the IDE-drives for the parallel port), you need a special active cable and a special kernel-driver. Read read http://www.torque.net/parport/parscsi.html to learn more about this option.

Generic SCSI devices

The devicefiles /dev/sd* and /dev/sr* for accessing SCSI harddisks and SCSI CD-ROM drives are limited to block-oriented data transfer. This makes them so fast and idealy suited for data storage. Driving the laser of a CD-writer is a more complicated task and requires more than block-oriented data transfer. To keep the sd* and sr* devices clean and fast, a new type of SCSI-devices was introduced, the so called *generic SCSI-devices*. As everthing can be done to SCSI-hardware through the generic devices they are not fixed to a certain purpose - therefore the name *generic*.

As true for other devices, you should find them in the /dev directory (by convention):

```
shell> cd /dev
shell> ls sg*
sg0  sg1  sg2  sg3  sg4  sg5  sg6  sg7
```

If you don't have those device-files then create them by using the `/dev/MAKEDEV`-script:

```
shell> cd /dev/
shell> ./MAKEDEV sg
```

If your Linux-distribution misses the script /dev/MAKEDEV, then create the devices manually with this loop:

```
for i in 0 1 2 3 4 5 6 7
do
        mknod /dev/sg$i c 21 $i
done
```

CD-Writers for the parallel port

I have no clue about this, sorry. Please read http://www.torque.net/parport/paride.html or your local file /usr/src/linux/Documentation/paride.txt.

2.3 Get the user-software for burning CD-Rs

TOC

A more detailed survey of tools related to produce CD-ROMs is available from http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdb.html.

Command line utilities

One of the following packages are required to generate images of CD-Rs (only required for data CD-ROMs):

ftp://tsx-11.mit.edu/pub/linux/packages/mkisofs/ (mkisofs)

ftp://ftp.ge.ucl.ac.uk/pub/mkhfs (mkhybrid)

To write images to the CD-R, you need one of the following software packages:

ftp://ftp.fokus.gmd.de/pub/unix/cdrecord/ (cdrecord)

http://www.ping.de/sites/daneb/cdrdao.html (cdrdao)

Don't trust the man-page of (old) `mkisofs` which states you need version 1.5 of `cdwrite`. Just use cdrecord and you are fine. Please note that use the mkisofs shipped with cdrecord has a few extensions (referenced in the documentation of cdrecord).

Graphical user interfaces (optional)

Front-ends are really front-ends under Linux. That means, you still have to install the command-line utilities, but you access them in a better looking manner.

X-CD-Roast is a program-package dedicated to easy CD creation under Linux. It combines command line tools like `cdrecord` and `mkisofs` into a nice graphical user interface.

http://www.fh-muenchen.de/home/ze/rz/services/projects/xcdroast/e_overview.html

BurnIT is a JAVA front-end to cdrecord, mksiofs and cdda2wav-0.95 making it a complete package for burning CD's on the Unix platform. Its available from

http://sunsite.auc.dk/BurnIT/

XDaodio

This is a graphical frontend for cdrdao. Its main purpose is to make 1:1-copies of audio CDs.

http://www.lrz-muenchen.de/~MarkusTschan/

---

3. Burning CD-Rs

> *"If to smoke you turn I shall not cease to fiddle while you burn."* (Roman emperor Nero about burning his own classic-CDs, AD64. He misunderstood it completely.)

Writing CD-ROMs consists of two steps under Linux:

- packaging the desired data (files, music or both) into files with special formats
- writing the data from the files to the CD-R with the utility `cdrecord`

This chapter describes the steps for data- and audio-CDs in greater detail.

3.1 Writing CD-ROMs (pure data)

t Note that collecting the software usually takes up longer than one expects. Consider that missing files cannot be added once the CD is written and fixated.

Also keep in mind that a certain amount of the free space of a CD is used for storing the information of the ISO-9660-filesystem (usually a few MB). 620 MB data will always fit on a 650 MB CD-R.

Creating an image of the later CD-ROM

Before any storage medium (e.g. floppy disk, harddisk or CD) can be used, it must get a filesystem (DOS speak: get formatted). The filesystem is responsible for organizing and incorporating the files that should be stored on the medium.

The usal utilities for creating filesystems on harddisk partitions write an empty filesystem onto them, which is then mounted and filled with files by the user as he needs it. A writable CD is only writable once so if we would write an empty filesystem to it, it would get formated - but remain completely empty forever. This is also true for rewritable media, as you cannot change arbritary sectors, but you must erase their whole content.

So what we need is a tool that creates the filesystem while copying the files to the CD. This tool is called `mkisofs`. A sample usage looks as follows:

```
mkisofs  -r   -o cd_image   private_collection/
              `---------'   `-----------------'
                   |               |
           write output to   take directory as input
```

The option '-r' sets the permissions of all files to be public readable on the CD and enables Rock Ridge extensions. That is what one usually wants and use of this option is recommended until you know what you're doing (hint: without '-r' the mount-point gets the permissions of `private_collection`!).

`mkisofs` will try to map all filenames to the 8.3-format used by DOS to ensure highest possible compatibility. In case of naming conflicts (different files have the same 8.3-name), numbers are used in the filenames and information about the chosen filename is printed via STDERR (usually the screen). Don't panic:

TOC

Under Linux you will never see these 8.3 filenames because Linux makes use of the Rock Ridge extension which contain the original file-information (permissions, filename, etc.).

Now you may wonder why the output of `mkisofs` is not directly sent to the writer-device. This has two reasons:

- `mkisofs` knows nothing about driving CD-writers.
- You may want to test the image before burning it.
- On slow machines it would not be reliable (see section 4.).

The method to write a CD-R in one go will be described below.

One also could think of creating an extra partition and writing the image to that partition instead to a file. I vote against such a strategy because if you write to the wrong partition (due to a typo), you can lose your complete Linux-system (read: that happened to me...). Furthermore, its a waste of disk-space because the CD-image is temporary data that can be deleted after writing the CD. However, using raw partitions saves you the time for deleting files of 650 MB size. (If you have enough disk-space, an extra partition may save you the time of deleting the image.)

Test the CD-image

Linux has the ability to mount files as if they were disk-partitions. This feature is useful to check that the directory layout and file access permissions of the CD-image matches your wishes. Although media is very cheap today, the writing process is still time-consuming, and you may at least want to save your time by doing a quick test.

To mount the file `cd_image` created above on the directory `/cdrom`, give the command

```
mount -t iso9660 -o ro,loop=/dev/loop0 cd_image /cdrom
```

Now you can inspect the files under `/cdrom` - they appear exactly as they were on a real CD. To umount the CD-image, just say `umount /cdrom`. (Warning: On Linux-kernels prior to 2.0.31 the last file on `/cdrom` may not be fully readable. Please use more recent kernels like 2.0.36. The option -pad for cdrecord applies to audio-CDs only and the optional option -pad for mkisofs requires a patch, which is as much work to apply than to upgrade to a bug-free Linux-kernel.)

Note:

> some ancient versions of `mount` are not able to deal with loopback-devices. If you have such an old version of `mount` it is a hint to upgrade your Linux-system.
>
> Several people already suggested to put information about how to get the newest mount-utilities into this HOWTO. I always refuse this. If your Linux-Distribution ships with an ancient `mount`: report it as a bug. If your Linux-Distribution is not easily upgradable: report it as a bug.
>
> If I had included all the information that is necessary to work around bugs in bad designed Linux-Distributions, this HOWTO would be a lot bigger and harder to read.

Write the CD-image to a CD

TOC

Not much more left to do. If you didn't already tried, its a good time for the command

```
cdrecord -scanbus
```

This will tell you to which SCSI-device your CD-writer is attached to. All other methods of guessing the information printed so nicely by cdrecord have been removed from the HOWTO (especially the somewhat dangerous naming scheme for generic SCSI devices).

Before showing you the last command, let me warn you that CD-writers want to be fed with a constant stream of data because they have only small data-buffers. So the process of writing the CD-image to the CD must not be interupted or a corrupt CD will be the result. Its easy to interrupt the data-stream by deleting a very large file. Example: if you delete a previous image of 650 Mbyte size, the kernel must update information about 650.000 blocks on the harddisk (assuming you have a blocksize of 1 Kbyte for your filesystem). That takes some time and is very likely to slow down disk-activity long enough for the data stream to pause a few seconds. However, reading mail, browsing the web or even compiling a kernel generally do not affect the writing process on modern machines.

Please note that no writer can re-position its laser and can't continue at the original spot on the CD when it gets disturbed. Therefore any strong vibrations or other mechanical shocks will probably destroy the CD you are writing.

When you are mentally prepared, dress up in a black robe, multiply the SCSI-id of the CD-writer with its SCSI-revision and light as many candles, speak two verses of the ASR-FAQ (newsgroup alt.sysadmin.recovery) and finally type:

```
shell> SCSI_BUS=0   # taken from listing 1 "scsibus0:"
shell> SCSI_ID=6    # taken from listing 1 "TOSHIBA XM-3401"
shell> SCSI_LUN=0
shell> cdrecord -v speed=2 dev=$SCSI_BUS,$SCSI_ID,$SCSI_LUN \
                   -data  cd_image

# same as above, but shorter:
shell> cdrecord -v speed=2 dev=0,6,0  -data  cd_image
```

For better readability, the coordinates of the writer are stored in three environment variables with speaking names: SCSI_BUS, SCSI_ID, SCSI_LUN. The option -data is non-mandatory but is given to make the commandline comparable to the one used for writing audio-CDs.

If you use cdrecord to overwrite a CD-RW, you must add the option "blank=..." to erase the old content. Please read the man-page to learn more about the various methods to blank the CD-RW.

In times where everybody except me owns a 400 Mhz machine, people feed the output of mkisofs directly into cdrecord:

```
shell> IMG_SIZE=`mkisofs -R -q -print-size private_collection/  2>&1 \
| sed -e "s/.* = //"`
shell> echo $IMG_SIZE
shell> [ "0$IMG_SIZE" -ne 0 ] && mkisofs -r  private_collection/  \
```

TOC

```
|cdrecord  speed=2  dev=0,6,0
            tsize=${IMG_SIZE}s  -data  -
#       don't forget the s --^         ^-- read data from STDIN
```

The first command is an empty run to determine the size of the image (you need the mkisofs from the cdrecord-distribution for this to work). Maybe your writer does not need to know the size of the image to be written, so you can leave this out. The printed sized must be passed as a tsize-parameter to cdrecord (its stored in the enviroment variable IMG_SIZE). The second command is a sequence of mkisofs and cdrecord, coupled via a pipe.

3.2 Writing audio CDs

To write audio-CDs is very similar compared to the steps described above. The two main differences are, that the audio-CD consists of audio-tracks, which are organized in separate images. So if you want to have ten tracks on your CD, you have to make ten images. The other differnce is, that the format of the images is not ISO-9660 (or whatever filesystem you prefer), but it is "16 bit stereo samples in PCM coding at 44100 samples/second (44.1 kHz)".

One utility to convert your soundfiles into the required format is sox. Its usage is straightforward:

```
shell> sox  killing-my-software.wav  killing-my-software.cdr
```

This command would convert the song killing-my-software from the WAV-format into the CDR-audio-format. See the man-page for sox for more details about formats and filename-extensions sox recognizes. Because the conversion takes up much diskspace, it was made a built-in feature of cdrecord for the formats WAV and AU. So as long as your soundfiles have the extensions .wav or .au (and the samplerate "stereo, 16 bit, 44.1 kHz"), you can use them as images without manual conversion.

Cdrecord writes the images as audio-tracks, if the option -audio is specified. The other options are identical to those used for writing data-CDs (unless you have very special requirements). Three examples all doing the same, but reading the tracks from different soundfile-formats:

```
shell> cdrecord -v speed=2 dev=0,6,0  -audio  track1.cdr track2.cdr...
shell> cdrecord -v speed=2 dev=0,6,0  -audio  track1.wav track2.wav...
shell> cdrecord -v speed=2 dev=0,6,0  -audio  track1.au  track2.au...
```

A special case are the MPEG3-files, which can be converted to the required format with the command "mpg123 -s track1.mp3 > track1.cdr". (Warning: this creates a file with a byte order, which needs to be reversed by using the option -swab to cdrecord.) The other direction can be done with "8hz-mp3" for WAV-files (extract the track with cdda2wav from the audio CD and encode it into MP3 with 8hz-mp3). To create a CD-R from a whole bunch of MP3-files, you can use the following command sequence:

```
for I in *.mp3
do
    mpg123 -s $I | cdrecord -audio -pad -swab -nofix -
done
cdrecord -fix
```

Depending on the speed of your machine, you may want to slow down writing to "speed=1"

TOC

(cdrecord-Option). If you use "speed=4", your machine must be able to play the MP3-file at quadruple sp mpg123 consumes much CPU-time! If you are in doubt, try an empty run with -dummy (keeps the laser switched off). Currently you will produce an audio-CD, which has a 2 seconds pause between two audio tracks.

DAO

If you want to get rid of the pauses between the audio tracks, you have to use disk-at-once (DAO) recording in opposite to the (individual) track-at-once (TAO) recording described above. Support for DAO is currently most advanced in cdrdao. Please see it's homepage for details.

Using the parameter read-cd, you are even able to produce 1:1 copies of audio CDs.

3.3 Mixed mode CD-ROMs

Not much left to say. Just indicate the type of the (subsequent) images with the options -data and -audio. Example:

```
cdrecord  -v dev=0,6,0  -data  cd_image  -audio  track*.cdr
```

---

4. Dear Winfried,...

In other words, this is the section usally known as "frequently asked questions with answers". If you have a problem with your partner, kids or dog, just send it in, as long as it is related to writing CD-Rs or is otherwise entertaining.

4.1 How sensitive is the burning process?

Test it. Use the option -dummy to do an empty run of cdrecord. Do everything you would do otherwise and watch if the burning process survives.

If you feed cdrecord directly from mkisofs, then disk intensive processes such as updating the *locate*-database lower the maximum flow-rate and may corrupt the CD. You better check such processes are not started via `cron`, `at` or `anacron` while you burn CD-Rs.

4.2 Has fragmentation a bad impact on the throughput?

Fragmentation is usually so low that its impact isn't noticed. However, you can easily construct pathological cases of fragmentation, which lower the throughput of your harddisks under 100 kbyte/second. So don't do that. :-)

Yes, files on a harddisk get fragmented over the years. The faster, the fuller the filesystem is. Always leave 10% or 20% free space, and you should run fine with respect to writing CD-Rs.

If you're uncertain then look at the messages printed while booting, the percentage of fragmentation is reported while checking the filesystems. You can check for this value with the very dangerous command

```
shell> e2fsck -n  /dev/sda5        # '-n' is important!
[stuff deleted - ignore any errors]
/dev/sda5: 73/12288 files (12.3% non-contiguous)
```

In this example the fragmentation seems to be very high - but there are only 73 very small files on this filesystem (used as `/tmp`) so the value is _not_ alarming.

There is an experimental utility called e2defrag to defragment extended-2 filesystems. The current version does not work reliable enough yet, to use it even for private environments. If you really want to defragment your filesystem, make a backup copy (better: two copies), practise restoring the data, then create a new filesystem (will destroy the old) and restore the data. In a few words, this is currently the safest technique.

4.3 Is it possible to store the CD-image on an UMSDOS-filesystem?

Yes. The only filesystem that isn't reliable and fast enough for writing CD-ROMs from is the *network filesystem* (*NFS*).

TOC

I used UMSDOS myself to share the disk-space between Linux and DOS/Win on a PC (486/66) dedicated to writing CD-ROMs.

4.4 Isn't there some way to get around the ISO-9660 limitations?

Yes. You can put any filesystem you like on the CD. But other operating systems than Linux won't be able to deal with this CD.

Here goes the recipe:

- Create an empty file of 650MB size.

```
dd if=/dev/zero of="empty_file" bs=1024k count=650
```

- Create an extended-2 filesystem on this file

```
shell> /sbin/mke2fs  -b 2048  empty_file
empty_file is not a block special device.
Proceed anyway? (y,n) y
```

- Mount this empty file through the loopback-devices

```
mount -t ext2 -o loop=/dev/loop1 empty_file /mnt
```

- Copy files to /mnt and umount it afterwards.
- Use `cdrecord` on empty_file (which is no longer empty) as if it were an ISO-9660-image.

If you want to make an entry in `/etc/fstab` for such a CD, disable the checking of it, e.g.:

```
/dev/cdrom  /cdrom  ext2  defaults,ro  0 0
```

The first 0 means "don't include in dumps", the second (=important) one means "don't check for errors on startup" (fsck will fail to check the CD for errors).

4.5 How to read the tracks from audio CDs?

There are several software packages available. The newest one is "cdpranoia" and can be downloaded from

http://www.mit.edu/afs/sipb/user/xiphmont/cdparanoia/index.html

Or you want to try the combination of "cdda2wav" and "sox", available from sunsite and its mirrors:

ftp://sunsite.unc.edu/pub/Linux/apps/sound/cdrom/cdda2wav0.71.src.tar.gz

ftp://sunsite.unc.edu/pub/Linux/apps/sound/convert/sox-11gamma-cb3.tar.gz

`cdda2wav` enables you to get a specific interval (or a whole track) from your audio CD and converts it into a .wav-file. `sox` converts the wav-files back into the (audio-CD) cdda-format so it can be written to the CD-R using `cdrecord`. You don't necessarily need `sox` if you use a recent version of `cdrecord`, because it has built-in support for .au and .wav files.

4.6 How to probe for SCSI-devices after boot?

TOC

The file `drivers/scsi/scsi.c` contains the information

```
/*
 * Usage: echo "scsi add-single-device 0 1 2 3" >/proc/scsi/scsi
 * with  "0 1 2 3" replaced by your "Host Channel Id Lun".
 * Consider this feature BETA.
 *     CAUTION: This is not for hotplugging your peripherals. As
 *     SCSI was not designed for this you could damage your
 *     hardware !
 * However perhaps it is legal to switch on an
 * already connected device. It is perhaps not
 * guaranteed this device doesn't corrupt an ongoing data transfer.
 */
```

Please note that this should only be used if your add SCSI-devices at the end of the chain. Inserting new SCSI-devices into an existing chain disturbs the naming of devices (directory /dev) and may destroy the complete content of your harddisk.

4.7 Is it possible to make a 1:1 copy of a data CD?

Yes. But you should be aware of the fact that any errors while reading the original (due to dust or scratches) will result in a defective copy.

First case: you have a CD-writer and a separate CD-ROM drive. By issuing the command

```
cdrecord -v dev=0,6,0 speed=2 -isosize /dev/scd0
```

you read the data stream from the CD-ROM drive attached as `/dev/scd0` and write it directly to the CD-writer.

Second case: you don't have a separate CD-ROM drive. In this case you have to use the CD-writer to read out the CD-ROM first:

```
dd if=/dev/scd0 of=cdimage
```

If you are running a recent version of cdrecord, then use the tool "readcd" (found under misc/). Both commands are equivalent to the result of `mkisofs`, so you should procede as described in chapter 3. Please note that this method will fail on audio CDs! You have to use cdrdao or cdda2wav on audio CDs.

4.8 Can Linux read Joliet CD-ROMs?

Yes. Newer Kernels (2.0.36 and the upcoming 2.2) have built-in support for the joliet format. Remember you have to use both options in your /etc/fstab: the keywords iso9660 and joliet (later is really an extension). For more details, see http://www-plateau.cs.berkeley.edu/people/chaffee/joliet.html.

4.9 How do I read/mount CD-ROMs with the CD-writer?

TOC

Just as you do with regular CD-ROM drives. No tricks at all. Note that you have to use the scd-devices (SCSI CD-ROM) to mount CD-ROMs for reading, even if you have an ATAPI CD-ROM (remember you configured your ATAPI-devices to act like SCSI). Example-entry for /etc/fstab:

```
/dev/scd0  /cdrom  iso9660  ro,user,noauto  0  0
```

4.10 How to put even more data on the CD-R?

Use `bzip2` instead of any other compressor like `gzip` or `pkzip`. It will save you up to 30% of disk-space for larger (>100kb) files. You can download it from

http://www.muraroa.demon.co.uk/

Instead of writing a true audio-CD, you can optionally convert your wav-audio-files into mp3-audio-files and store them on a ISO-9660 filesystem as regular files. Usually MPEG III give you a compression of 1:10. Of course, most CD-players are not able to read files... this is the drawback. On the other hand, why not running the music for your next party from harddisk? 18 Gbyte are enough for 3000-4000 titles. :-)

A software MPEG III-encoder is available from

http://www.stud.ifi.uio.no/~larsi/other/8hz-mp3-cheng.tar.gz

A MPEG III-player is available from

http://homepages.uni-tuebingen.de/student/michael.hipp/mpg123/

For recorded speech, you may want to try to reduce its size using `shorten` or "GSM lossy speech compression":

ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/

http://kbs.cs.tu-berlin.de/~jutta/toast.html

4.11 How to make bootable CD-ROMs?

You must have an 1.44 MB bootable floppy-disk. Create an exact image of this floppy-disk by issuing the command

```
dd if=/dev/fd0 of=boot.img bs=18k
```

Place this floppy-image into the directory holding the collection of your files (or into a subdirectory of it, just as you like). Tell mkisofs about this file with the option '-b' and also use '-c'. For details read the file README.eltorito in the mkisofs-distribution.

An interesting application for a custom bootable CD is as a virus-safe DOS- or Windows-system. It saves you the money for the harddisks (if you have a network and use samba to put the user-data on a fileserver). However, this is purely theoretical as nobody reported an actual recipe to me.

Some details about the bootable RedHat CD-ROM is available from

http://www.channel1.com/users/rodsmith/rhjol-technical.html.

4.12 How to make CD-ROMs writable somehow?

There is an *overlay-filesystem* available for Linux, which is mounted over the CD-ROM and intercepts all writing operations. New and modified files are stored elsewhere, but for the user it looks like the CD-ROM is modified. For more information, see http://home.att.net/~artnaseef/ovlfs/ovlfs.html.

If that is not enough for your needs: wait for the UDF-filesystem to be supported by Linux or help developing it (see http://trylinux.com/projects/udf/.

4.13 Is it possible to use several writers at once?

Yes. However, it has been only tested with two writers yet. You need either a recent version of the Linux kernel (2.2.10 as of writing) or a a kernel patch for more buffers in the SCSI-generic driver (ftp://ftp.fokus.gmd.de/pub/unix/cdrecord/alpha; works up to 2.2.5).

4.14 Which media is the best?

The german computer magazine "c't" had a list of tips regarding the blank CD-Rs in their november 1996 issue:

- "No-name" discs are generally not of highest quality and should better not be used.
- If a recordable CD is defective, this is likely to apply to the whole batch (if you bought more than one at a time); maybe you are lucky and can at least use the first 500MB of such CD-Rs ...
- Don't touch the CD-Rs at their shimmering side before writing.

4.15 What about Solaris, *BSD, AIX, HP-UX, etc.?

Only chapter 2 is Linux-specific. You can apply chapter 3 and 4 even if you run another family of operating systems than Linux. Please see the files README.NetBSD, README.aix, README.hpux, README.next, README.solaris, README.sunos, README.vms or README.xxxBSD from the cdrecord-distribution.

4.16 Where to store the local configuration permanently?

You have two options. Either you use the built-in configuration file for cdrecord, or you use a shell-wrapper like the one shown below. This shell-script reads a configuration file, which lists the options and parameters for cdrecord line by line. The names are exactly the same as on the commandline, but without the leading -. Comments are allowed. Example:

```
# be verbose
v
# set the speed of the writer
speed=2
# the device-coordinates in the form BUS,ID,LUN
```

```
dev=0,6,0
```

The configuration files for the wrapper belong into /etc/cdrecord and must be referenced on the commandline. Example: if you want to refer to the configuration /etc/cdrecord/mywriter.cfg, then you can issue the command "cdrecord.sh mywriter.cfg -audio track1...". Everthing after mywrite.cfg is passed to cdrecord.

```
#! /bin/bash

CFGDIR="/etc/cdrecord"

CFG="$1"
shift
ARGS_LEFT="$@"

if [ ! -f "$CFGDIR/$CFG" ]
then
    echo "Configuration file $CFGDIR/$CFG not found. Exiting."
    exit 1
fi

while read LINE
do
    case $LINE in
        \#*|"") continue;;
    esac
    old_IFS="$IFS"
    IFS="$IFS="
    set -- $LINE
    IFS="$old_IFS"
    O_NAME="$1"
    O_VALUE=""
    while shift
    do
        case $1 in
            "") continue;;
        esac
        O_VALUE="$1"
    done

    if [ -z "$O_VALUE" ]
    then
        O_CDRECORD="$O_CDRECORD -$O_NAME "
        continue
    fi
    O_CDRECORD="$O_CDRECORD $O_NAME=$O_VALUE "

done < "$CFGDIR/$CFG"

set -x    #DEBUG
```

```
exec cdrecord $O_CDRECORD $ARGS_LEFT
echo "Execution of cdrecord failed."
```

4.17 How can the CD-info be retrieved?

Somewhere behind the first 32 k on the CD, a block with information about the CD is located. You can extract the information with the following shell-script:

```
#! /bin/bash

RD=/dev/cdrom
for i in 32768,7   32776,32 32808,32 32958,128 33086,128 33214,128 \
         33342,128 33470,32 33581,16 33598,16  33615,16  33632,16
do
    old_IFS="$IFS"
    IFS=","
    set -- $i
    IFS="$old_IFS"
    OFFSET=$1
    LENGTH=$2
    echo "*`dd if=$RD bs=1 skip=$OFFSET count=$LENGTH   2> /dev/null`#"
done
```

4.18 What about re-writing

When overwriting CD-RW media, specify the parameter blank=fast to cdrecord. That's all. See the man-page of cdrecord for details about this parameter.

4.19 How to create a multi-session CD?

First of all, the image for a multi-session CD must be formatted using the ISO-9660-filesystem using RockRidge-Extensions. And you must use the option -multi for cdrecord as long as you want to add further sessions. So at least for the first session, you must use -multi.

The images for the second and subsequent sessions are a little bit more complicated to generate. Mkisofs must know where the free space on the CD-R begins. That information can be gathered by using the option -msinfo on cdrecord (see example below).

```
shell> NEXT_TRACK=`cdrecord -msinfo dev=0,6,0`
shell> echo $NEXT_TRACK
shell> mkisofs -R -o cd_image2 -C $NEXT_TRACK -M /dev/scd5
         private_collection/ \
```

For more information, please read the file README.multi, which is distributed with cdrecord.

5. Troubleshooting

Always remember, that you can still use corrupt CD-ROMs as coasters. :-)

5.1 It doesn't work: under Linux

Please check first, that the writer works under the software it is shipped with (=under another operating system). Concretely:

- Does the controller recognize the writer as a SCSI device?
- Does the driver software recognize the writer?
- Is it possible to make a CD using the accompanied software?

If "it doesn't even work" with the accompanied software you have a hardware conflict or defective hardware. If it works and you use loadlin to boot Linux, then that is your problem. Loadlin makes a warm-boot with most of the hardware already initialized and that can confuse the Linux-kernel.

5.2 Error-message: No read access for 'dev=0,6,0'.

Under Linux, some versions of the C-library are incompatible (buggy), so that an application linked against one version will not work with another. An example for an error triggered by pre-compiled binaries is the following:

```
[root@Blue /dev]# cdrecord -eject dev=0,6,0
cdrecord: No such file or directory. No read access for 'dev=0,6,0'.
```

5.3 It doesn't work: under DOS and friends

Try to use Linux. Installation and configuration of SCSI-drivers for DOS is the hell. Linux is too complicated? Ha!

5.4 SCSI errors during the burning phase

Most likely those errors are caused by

- missing dis-/reconnect feature on the SCSI bus
- insufficiently cooled hardware
- defective hardware (should be detected by 5.1.)

Under various circumstances SCSI devices dis- and reconnect themselves (electronically) from the SCSI bus. If this feature is not available (check controller and kernel parameters) some writers run into trouble

during burning or fixating the CD-R.

Especially the NCR 53c7,8xx SCSI driver has the feature disabled by default, so you might want to check it first:

```
NCR53c7,8xx SCSI support                    [N/y/m/?] y
   always negotiate synchronous transfers [N/y/?] (NEW) n
   allow FAST-SCSI [10MHz]                [N/y/?] (NEW) y
   allow DISCONNECT                       [N/y/?] (NEW) y
```

6. Credits

Many thanks go to the readers of this HOWTO, who contributed actively to its contents. As I don't have access to a cd-writer myself, reports about real-live setups and experience have always been of great value to me.

Doug Alcorn <doug@lathi.net>

helped to improve the handling of newer kernels

Kalle Andersson <kalle@sslug.dk>

Howto write audio-cds directly from mp3.

Alan Brown <alan@manawatu.net.nz>

Rick Cochran <rick@msc.cornell.edu>

hint about dis-/reconnect disabled by default in the ncr driver

Robert Doolittle <bob.doolittle@sun.com>

good arguments for dropping cdwrite from the HOWTO

Markus Dickebohm <m.dickebohm@uni-koeln.de>

Jos van Geffen <jos@tnj.phys.tue.nl>

noted the problem in 4.9.

Bernhard Gubanka <beg@ipp-garching.mpg.de>

noticed the need of a recent version of mount to utilize the loopback device

Stephen Harris <sweh@mpn.com>

contributed hint about writing audio-CDs

Janne Himanka <shem@oyt.oulu.fi>

pointer to kernel patch to read Joliet CD-ROMs

Stephan Noy <stnoy@mi.uni-koeln.de>

information and experience about writing audio-CDs

Don H. Olive <don@andromeda.campbellsvil.edu>

URL of the mkhybrid tool

Jesper Pedersen <jews@imada.ou.dk>

Pierre Pfister <pp@uplift.fr>

helped to develop the recipe on 1:1 copies.

Daniel A. Quist <dquist@cs.nmt.edu>

TOC

information about IDE-CD-R and newer kernel versions

Martti.Rahkila@hut.fi

Reported problem with pre-initialized writers when booting via loadlin.

Dale Scheetz <dwarf@polaris.net>

Joerg Schilling <schilling@fokus.gmd.de>

information about cdrecord

Martin Schulze <joey@Infodrom.North.DE>

gave information about the cdwrite-mailinglist

Gerald C Snyder <gcsnyd@loop.com>

tested writing of an ext2 CD-ROM (see 4.4)

Art Stone <stone@math.ubc.ca>

had the idea to put non-ISO-9660 filesystems on a CD

The Sheepy One <kero@escape.com>

suggested using defective CD-ROMs as coasters for drinks

Erwin Zoer <ezoer@wxs.nl>

Futhermore, I would like to thank the following people for reporting spelling mistakes: Bartosz Maruszewski <B.Maruszewski@zsmeie.torun.pl>, Ian Stirling <ian@opus131.com>, Brian H. Toby.

End of the Linux CD-Writing HOWTO. (You can stop reading here.)

The Linux CD-ROM HOWTO

Jeff Tranter, tranter@pobox.com

v1.14, 24 March 1999

This document describes how to install, configure, and use CD-ROM drives under Linux. It lists the supported hardware and answers a number of frequently asked questions. The intent is to bring new users up to speed quickly and reduce the amount of traffic in the Usenet news groups and mailing lists.

1. Introduction

- 1.1 Acknowledgments
- 1.2 New Versions Of This Document
- 1.3 Feedback
- 1.4 Distribution Policy

2. CD-ROM Technology

3. Supported Hardware

- 3.1 ATAPI CD-ROM Drives
- 3.2 SCSI CD-ROM Drives
- 3.3 Proprietary CD-ROM Drives
- 3.4 Parallel Port Drives
- 3.5 Alternate Drivers

4. Installation

- 4.1 Installing the Hardware
- 4.2 Configuring and Building the Kernel
- 4.3 Creating Device Files and Setting Boot Time Parameters

- 4.4 Booting the Linux Kernel
- 4.5 Mounting, Unmounting, and Ejecting Devices
- 4.6 Troubleshooting

5. Applications

- 5.1 Audio CD Players
- 5.2 PhotoCD
- 5.3 Mkisofs
- 5.4 ISO-9660 Utilities

6. Answers to Frequently Asked Questions

- 6.1 How can a non-root user mount and unmount discs?
- 6.2 Why do I get device is busy when unmounting a CD-ROM?
- 6.3 How do I export a CD-ROM to other hosts over NFS?
- 6.4 Can I boot Linux from a CD-ROM?
- 6.5 How can I read digital data from audio CDs?
- 6.6 Why doesn't the find command work properly?
- 6.7 Does Linux support any recordable CD-ROM drives?
- 6.8 Why do I get mount: Read-only file system when mounting a CD-ROM?
- 6.9 Why does the disc tray open when I shut down the system?
- 6.10 I have a "special" CD that can't be mounted
- 6.11 Do multi-platter CD-ROM drives work with Linux?
- 6.12 I get "/cdrom: Permission denied" errors
- 6.13 How do I interpret IDE CD kernel error messages?
- 6.14 How can I tell what speed CD-ROM I have?
- 6.15 My CD-ROM stopped working after Linux was installed
- 6.16 There are "hidden" files on a CD which I can't read
- 6.17 Where is the CD-ROM API documented?
- 6.18 Why don't I see long filenames on this Windows CD-ROM?
- 6.19 Is Microsoft's Joliet filesystem supported?
- 6.20 Does Linux support "enhanced" CDs?
- 6.21 Does Linux support DVD-ROM?

- 6.22 Does Linux support CD-RW?

7. References

1. Introduction

This is the Linux CD-ROM HOWTO. It is intended as a quick reference covering everything you need to know to install and configure CD-ROM hardware under Linux. Frequently asked questions related to CD-ROM are answered, and references are given to other sources of information related to CD-ROM applications and technology.

1.1 Acknowledgments

Much of this information came from the documentation and source files provided with the Linux kernel, the Internet alt.cdrom newsgroup FAQ, and input from Linux users.

Thanks to the SGML Tools package, this HOWTO is available in several formats, all generated from a common source file.

1.2 New Versions Of This Document

New versions of this document will be periodically posted to the comp.os.linux.answers newsgroup. They will also be uploaded to various anonymous ftp sites that archive such information including ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/.

Hypertext versions of this and other Linux HOWTOs are available on many world-wide web sites, including http://metalab.unc.edu/LDP/HOWTO/. Most Linux CD-ROM distributions include the HOWTOs, often under the /usr/doc directory, and you can also buy printed copies from several vendors. Sometimes the HOWTOs available from CD-ROM vendors, ftp sites, and printed format are out of date. If the date on this HOWTO is more than six months in the past, then a newer copy is probably available on the Internet.

Please note that, due to the dynamic nature of the Internet, all web and ftp links listed in this document are subject to change.

Translations of this document are available in several languages:

Chinese: http://www.linux.org.tw/CLDP/CDROM-HOWTO.html

French: http://www.freenix.org/unix/linux/HOWTO/

Japanese: http://jf.linux.or.jp/JF/JF.html/

Polish: http://www.jtz.org.pl/Html/CDROM-HOWTO.pl.html

Spanish: ftp://ftp.insflug.org/es

Swedish: http://www.swe-doc.linux.nu/

Most translations of this and other Linux HOWTOs can also be found at http://metalab.unc.edu/pub/Linux/docs/HOWTO/translations/ and ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/translations/. If you make a translation of this document into another language, let me know and I'll include a reference to it here.

1.3 Feedback

I rely on you, the reader, to make this HOWTO useful. If you have any suggestions, corrections, or comments, please send them to me, tranter@pobox.com, and I will try to incorporate them in the next revision.

I am also willing to answer general questions on CD-ROM under Linux, as best I can. Before doing so, please read all of the information in this HOWTO, and then send me detailed information about the problem. Please do not ask me about using CD-ROM drives under operating systems other than Linux.

If you publish this document on a CD-ROM or in hardcopy form, a complimentary copy would be appreciated; mail me for my postal address. Also consider making a donation to the Linux Documentation Project to help support free documentation for Linux. Contact the Linux HOWTO coordinator, Tim Bynum (linux-howto@metalab.unc.edu), for more information.

1.4 Distribution Policy

Copyright (c) 1995-1999 by Jeff Tranter. This document may be distributed under the terms set forth in the LDP license at http://metalab.unc.edu/LDP/COPYRIGHT.html.

---

2. CD-ROM Technology

```
"CD-ROM is read-only memory, and audio compact disc system is
available as package-media of digital data for those purpose. For
playing audio CD, please insert Head-phone jack."
--- from a CD-ROM instruction manual
```

Don't Panic! The world of CD-ROM technology is not as confusing as your instruction manual.

CD-ROM stands for *Compact Disc Read-Only Memory*, a mass storage medium utilizing an optical laser to read microscopic pits on the aluminized layer of a polycarbonate disc. The same format is used for audio Compact Discs. Because of its high storage capacity, reliability, and low cost, CD-ROM has become an increasingly popular storage media.

The storage capacity of a CD-ROM disc is approximately 650 megabytes, equivalent to over 500 high density 3.5" floppy disks or roughly 250,000 typed pages.

First generation drives (known as *single speed*), provided a transfer rate of approximately 150 kilobytes per second. Hardware manufacturers then introduced double speed (300 kB/sec), quad speed (600 kB/sec), and higher. Current drives operate at up to 40 times speed, although the maximum rate is only achievable over certain portions of the disc surface.

Most CD-ROM drives use either the Small Computer Systems Interface (SCSI), ATAPI enhanced IDE interface, or a vendor proprietary interface. They also typically support playing audio CDs via an external headphone jack or line level output. Some CDs also allow reading the frames of data from audio CDs in digital form.

CD-ROMs are usually formatted with an ISO-9660 (formerly called *High Sierra*) file system. This format restricts filenames to the MS-DOS style (8+3 characters). The *Rock Ridge Extensions* use undefined fields in the ISO-9660 standard to support longer filenames and additional Unix style information (e.g. file ownership, symbolic links, etc.). Microsoft has defined a proprietary ISO file system extension called Joliet which supports long filenames using the 16-bit UNICODE character encoding.

PhotoCD is a standard developed by Kodak for storing photographic images as digital data on a CD-ROM. With appropriate software, you can view the images on a computer, manipulate them, or send them to a printer. Information can be added to a PhotoCD at a later date; this is known as *multi-session* capability.

CD recorders (CD-R) allow writing onto a special "gold" CD which can then be read by any CD-ROM drive. Data can only be written once, although using multi-session new data can be appended to a disc. These drives are now becoming increasingly affordable.

CD-RW (read/write) drives support multiple writing of data to CDs. They use special discs which cannot

TOC

be read using standard CD-ROM drives, although the CD-RW drives themselves can read standard d

DVD-ROM expands the storage of a CD to as much as 17 gigabytes. They are commonly used as a medium for distributing full length motion pictures encoded using the MPEG-2 format. The MPEG video decoding is usually performed using specialized decoder hardware.

---

3. Supported Hardware

This section lists the CD-ROM drivers and interfaces that are currently supported under Linux. The information here is based on the latest stable Linux kernel, which at time of writing was version 2.2.4.

This information is only valid for Linux on the Intel platform. Much of it is applicable to Linux on other processor architectures as well.

3.1 ATAPI CD-ROM Drives

ATAPI (ATA Packet Interface) is a protocol for controlling mass storage devices. It builds on the ATA (AT Attachment) interface, the official ANSI standard name for the IDE interface developed for hard disk drives. ATAPI is commonly used for hard disks, CD-ROM drives, tape drives, and other devices. Currently the most popular type of interface, it offers most of the functionality of SCSI, without the need for an expensive controller or cables.

The Linux kernel has a device driver that should work with any ATAPI compliant CD-ROM drive. Vendors shipping compatible drives include Aztech, Mitsumi, NEC, Sony, Creative Labs, and Vertos. If you have recently purchased a CD-ROM drive, especially if it is quad speed or faster, it is almost guaranteed to be IDE/ATAPI.

Linux also has an IDE SCSI emulation kernel driver that makes an IDE/ATAPI device appear in software to be a SCSI device, allowing the use of a SCSI device driver instead of the native ATAPI driver. This is useful if you have an ATAPI device for which no native driver has been written (for example, an ATAPI PD-CD or CDR drive); you can then use this emulation together with an appropriate SCSI device driver.

3.2 SCSI CD-ROM Drives

SCSI (Small Computer Systems Interface) is a popular format for CD-ROM drives. Its chief advantages are a reasonably fast transfer rate, multi-device capability, and support on a variety of computer platforms. Some disadvantages of SCSI are the need for a relatively expensive controller card and cables.

Any SCSI CD-ROM drive with a block size of 512 or 2048 bytes should work under Linux; this includes the vast majority of CD-ROM drives on the market.

You will also need a supported SCSI controller card; see the SCSI HOWTO for more information on interface hardware.

Note that some CD-ROMs include a proprietary controller with a modified interface that is not fully SCSI compatible (e.g. it may not support adding other SCSI devices on the bus). These will most likely *not* work under Linux.

3.3 Proprietary CD-ROM Drives

TOC

Several CD-ROM drives using proprietary interfaces are available; the interface is often provided on a sound card. Simple interface cards equivalent to that provided on the sound card are also available. These drives generally tend to be lower in cost and smaller than SCSI drives. Their disadvantages are the lack of standardization and expandability.

Note that proprietary interfaces are sometimes erroneously referred to as IDE interfaces, because like IDE hard disks, they use a simple interface based on the PC/AT bus. To add to the confusion, some vendors, most notably Creative Labs, have shipped many different types of CD-ROM drives and have offered proprietary, SCSI, and ATAPI interfaces on their sound cards.

The table below lists the proprietary CD-ROM drives that are known to be supported under Linux. Drivers for additional devices may be available in the latest development kernels or as kernel patches. The latter can most often be found at ftp://metalab.unc.edu/pub/Linux/kernel/patches/cdrom/. Also check the documentation files included with the kernel distribution, usually installed in /usr/src/linux/Documentation/cdrom, for the latest information.

Proprietary CD-ROM Drives

Vendor	Model	Kernel Driver	Notes
Panasonic	CR-521	sbpcd	Note 1
Panasonic	CR-522	sbpcd	Note 1
Panasonic	CR-523	sbpcd	Note 1
Panasonic	CR-562	sbpcd	Note 1
Panasonic	CR-563	sbpcd	Note 1
Creative Labs	CD-200	sbpcd	
IBM	External ISA	sbpcd	Note 2
Longshine	LCS-7260	sbpcd	
Teac	CD-55A	sbpcd	
Sony	CDU-31A	cdu31a	
Sony	CDU-33A	cdu31a	
Sony	CDU-535	sonycd535	Note 3
Sony	CDU-531	sonycd535	
Aztech	CDA268-01A	aztcd	Note 4
Orchid	CDS-3110	aztcd	
Okano/Wearnes	CDD110	aztcd	
Conrad	TXC	aztcd	
CyCDROM	CR520ie	aztcd	
CyCDROM	CR940ie	aztcd	
GoldStar	R420	gscd	Note 5
Philips/LMS	CM206	cm206	Note 6
Mitsumi	CRMC LU005S	mcd/mcdx	Note 7, 8
Mitsumi	FX001	mcd/mcdx	Note 7, 8

```
Optics Storage  Dolphin 8000AT  optcd
Lasermate       CR328A          optcd
Sanyo           H94A            sjcd
various         various         isp16           Note 9
```

Notes:

1. These drives may be sold under the names Creative Labs, Panasonic, Matsushita, or Kotobuki.
2. This drive is the same as a Panasonic CR-562.
3. May also be sold under the Procomm name.
4. This driver is for the CDA268-01A only. Other models, including the CDA268-03I and CDA269-031SE are not proprietary and should use the IDECD (ATAPI) kernel driver.
5. May also be sold as part of a Reveal Multimedia Kit.
6. The Philips CM205 is not supported by this driver, but there is a separate alpha release driver available from ftp://metalab.unc.edu in /pub/Linux/kernel/patches/cdrom/lmscd0.4.tar.gz
7. May also be sold under the Radio Shack name.
8. There are two drivers available. "mcd" is the original one, and "mcdx" is a newer driver with more features (but possibly less stable).
9. This driver works with CD-ROM drives that are attached to the interface on an ISP16, MAD16 or Mozart sound card.

If a drive listed here is not supported by your kernel, you probably need to upgrade to a newer version.

If your drive is not one of the models listed here, particularly if it was bought recently and is quad speed or faster, it probably uses the IDE/ATAPI interface listed in a previous section. *The single most common error among Linux CD-ROM users is to assume that any drive connected to a SoundBlaster card should use the SBPCD driver*. Creative Labs and most other vendors are no longer selling proprietary interface drives, they are following the standard ATAPI/IDE interface.

3.4 Parallel Port Drives

There are external storage devices, including CD-ROM drives, that attach to the parallel port of personal computers. In many cases the devices internally use an IDE interface in conjunction with an adaptor which interfaces the internal IDE bus to the PC parallel port.

Linux has a parallel port IDE driver which supports most parallel port devices. At the time of writing it supported devices from the following vendors (as well as most no-name and clone drives compatible with these): ATEN, Avatar, DataStor, Fidelity International Technology, Freecom, Hewlett-Packard, Imation, KT Technology, KingByte Information Corp., Maxell, MicroSolutions, OnSpec, Shuttle Technology, SyQuest, and ValuStore.

Additional information can be found at http://www.torque.net/parport/.

3.5 Alternate Drivers

TOC

There is an alternate kernel driver available for Panasonic/Matsushita CR-56x drives written by Zoltan Vorosbaranyi. It can be found at ftp://ftp.honlap.net/pub/linux/pcd/pcd-0.30.tar.gz.

---

4. Installation

Installation of a CD-ROM under Linux consists of these steps:

1. Installing the hardware.
2. Configuring and building the Linux kernel.
3. Creating device files and setting boot time parameters
4. Booting the Linux kernel.
5. Mounting the media.

The next sections will cover each of these steps in detail.

4.1 Installing the Hardware

Follow the manufacturer's instructions for installing the hardware or have your dealer perform the installation. The details will vary depending on whether the drive is internal or external and on the type of interface used. There are no special installation requirements for Linux. You may need to set jumpers on the drive and/or interface card for correct operation; some of the kernel drivers include documentation files that include this information.

As explained in the file ide-cd, ATAPI CD-ROMS should be jumpered as "single" or "master", and not "slave" when only one IDE device is attached to an interface (although this restriction is no longer enforced with recent kernels).

4.2 Configuring and Building the Kernel

When initially installing Linux from CD-ROM you will likely be using a boot and/or root disk provided as part of a Linux distribution. If possible, you should choose a boot disk with the kernel driver for your CD-ROM device type. If you cannot find a boot disk with the necessary CD-ROM driver, you have several options:

1. Boot directly from CD-ROM (if supported by your computer and Linux CD)
2. Install over a network
3. Boot DOS, and install the Linux files onto your hard disk
4. Boot DOS, and create a set of floppies to install Linux
5. Find someone who can build you a boot disk with the needed CD-ROM driver

The Linux Installation HOWTO has more information on installing Linux. If you purchased Linux on CD-ROM, it likely also came with some installation instructions (that little booklet inside the jewel case, and/or files on the CD).

Once Linux has initially been installed, most users will want to compile their own kernel, usually for one of these reasons:

- to support a CD-ROM drive or other hardware
- to upgrade to a newer kernel release
- to free up memory resources by minimizing the size of the kernel

The Linux Kernel HOWTO should be consulted for the details of building a kernel. I will just mention here some issues that are specific to CD-ROM drives.

Obviously, you need to compile in support for your CD-ROM drive when you do a "make config".

If you have an ATAPI CD-ROM drive, you need to answer `yes` to the questions:

```
Enhanced IDE/MFM/RLL disk/cdrom/tape support (CONFIG_BLK_DEV_IDE) [Y/n/?]
Include IDE/ATAPI CDROM support (CONFIG_BLK_DEV_IDECD) [Y/n/?]
```

For SCSI CD-ROM drives, enable these options:

```
SCSI support (CONFIG_SCSI) [Y/n/m/?]
SCSI CDROM support (CONFIG_BLK_DEV_SR) [Y/n/m/?]
```

Also enable support for your SCSI host adapter when prompted, e.g.

```
Adaptec AHA152X/2825 support (CONFIG_SCSI_AHA152X) [Y/n/m/?]
```

For proprietary interface CD-ROM drives, enable the appropriate driver. You can use the table listed previously to determine the driver to use for your model.

Virtually all CD-ROMs use the ISO-9660 file system, so you must also enable:

```
ISO 9660 CDROM filesystem support (CONFIG_ISO9660_FS) [Y/n/m/?]
```

Although not needed for CD-ROM operation, if you have a sound card that is supported under Linux you might want to enable and configure the kernel sound driver at this time as well. The Sound HOWTO can be a useful reference here.

You should then follow the usual procedure for building the kernel and installing it. Don't boot with the new kernel until you create the device files and set up any boot time parameters as described in the next section.

The ISO-9660 filesystem and almost all of the CD-ROM drivers can be built as loadable kernel modules. This scheme allows the kernel drivers to be loaded and unloaded without rebooting the kernel, freeing up memory. I recommend you get your CD-ROM installation running using compiled-in drivers first. How to use modules is described in the modules documentation and the Kernel HOWTO.

If a drive type listed here is not supported by your kernel, you likely need to upgrade to a newer version.

It is possible that you need to use a driver that is distributed separately from the kernel source code. This usually involves patching the kernel. Again, the Kernel HOWTO explains how to do this.

Note that there is a menu-based kernel configuration program invoked by "make menuconfig" and an X11-based graphical configuration invoked as "make xconfig". All three configuration methods offer on-line help.

4.3 Creating Device Files and Setting Boot Time Parameters

The kernel uses device files to identify which device driver to use. If you are running a standard Linux distribution you may have created the necessary device files during installation. Under Slackware Linux, for example, there is a menu-based `setup` tool that includes CD-ROM setup, and most systems have a /dev/MAKEDEV script. If you don't use these methods, you can use the more manual procedure listed in this section. Even if you use either of these methods, it is recommended that you at least verify the device files against the information in this section.

You create the device file by running the shell commands indicated for your drive type. This should be done as user `root`. Note that some Linux distributions may use slightly different CD-ROM device names from those listed here.

It is recommended that you also create a symbolic link to the CD-ROM device to make it easier to remember. For example, for an IDE CD-ROM drive that is the second device on the secondary interface, the link would be created using

```
# ln -s /dev/hdd /dev/cdrom
```

If you want to play audio CDs, you will need to set the protection on the device file (the real file, not the symbolic link to it) to allow all users to read, e.g.

```
# chmod 664 /dev/hdd
# ls -l /dev/hdd
brw-rw-r--   1 root     disk      22,  64 Feb  4  1995 /dev/hdd
```

When booting Linux, the device drivers attempt to determine whether the appropriate devices are present, typically by probing specific addresses. Many of the drivers auto-probe at several addresses, but because of differences in configuration, possible

device conflicts, and hardware limitations, the drivers sometimes need help identifying the addresses and other parameters. l drivers support an option on the kernel command line to pass this information to the device driver. This can be done interact or more commonly, configured into your boot loader. With LILO, for example, you would add an `append` command such as the following to your /etc/lilo.conf file:

```
append = "sbpcd=0x230,SoundBlaster"
```

See the LILO documentation for more information.

In the next section I discuss issues specific to individual device drivers, including device files, boot parameters, and the capabilities of the different drivers. You probably only need to read the section relevant to your drive type. The documentation files are usually found in the directory /usr/src/linux/Documentation/cdrom.

Sbpcd Driver

```
       Principal author: Eberhard Moenkeberg (emoenke@gwdg.de)
  Multi-session support: yes (but not all drives)
 Multiple drive support: yes
Loadable module support: yes
   Reading audio frames: yes (CR-562, CR-563, CD-200 only)
           Auto-probing: yes
            Device file: /dev/sbpcd, major 25
     Configuration file: sbpcd.h
   Kernel config option: Matsushita/Panasonic CDROM support?
     Documentation file: sbpcd
```

This driver accepts a kernel command line of the form:

```
sbpcd=<io-address>,<interface-type>
```

where the first parameter is the base address of the device (e.g. 0x230), and <interface-type> is one of "SoundBlaster", "LaserMate", or "SPEA". See the file sbpcd.h for hints on what interface type to use. Using `sbpcd=0` disables auto-probing, disabling the driver.

The device file can be created using:

```
# mknod /dev/sbpcd b 25 0
```

Up to four drives per controller are supported. The next three drives on the first controller would use minor device numbers 1 through 3. If you have more than one controller, create devices with major numbers 26, 27, and 28, up to a maximum of 4 controllers (this is 16 CD-ROM drives in total; hopefully enough for most users :-).

See the file sbpcd for more information on this driver.

If you recently bought a CD-ROM drive, don't assume that if it connects to a SoundBlaster card it should use this kernel driver. Most CD-ROM drives being sold by Creative Labs are now EIDE/ATAPI drives.

Sonycdu535 Driver

```
       Principal author: Ken Pizzini (ken@halcyon.com)
  Multi-session support: no
 Multiple drive support: no
Loadable module support: yes
   Reading audio frames: no
           Auto-probing: no
            Device file: /dev/sonycd535, major 24
     Configuration file: sonycd535.h
```

```
   Kernel config option: Sony CDU535 CDROM support?
     Documentation file: sonycd535
```

This driver accepts a kernel command line of the form:

```
sonycd535=<io-address>
```

where <io-address> is the base address of the controller (e.g. 0x320). Alternatively you can set the address in the file sonycd535.h and compile it in.

The device file can be created using:

```
# mknod /dev/sonycd535 b 24 0
```

Some Linux distributions use /dev/sonycd for this device. Older versions of the driver used major device number 21; make sure your device file is correct.

This driver was previously distributed as a patch but is now part of the standard kernel. See the file sonycd535 for more information on this driver.

Cdu31a Driver

```
       Principal author: Corey Minyard (minyard@-rch.cirr.com)
  Multi-session support: yes
 Multiple drive support: no
Loadable module support: yes
   Reading audio frames: yes
           Auto-probing: no
            Device file: /dev/cdu31a, major 15
     Configuration file: cdu31a.h
   Kernel config option: Sony CDU31A/CDU33A CDROM support?
     Documentation file: cdu31a
```

This driver accepts a kernel command line of the form:

```
cdu31a=<io-address>,<interrupt>,PAS
```

The first number is the I/O base address of the card (e.g. 0x340). The second is the interrupt number to use (0 means to use polled i/o). The optional third parameter should be "PAS" if the drive is connected to a Pro-Audio Spectrum 16 sound card, otherwise left blank.

If the driver is loaded as a module, it uses a slightly different format. When loading the driver using the modprobe or insmod command, the parameters take the form:

```
cdu31a_port=<io-address> cdu31a_irq=<interrupt>
```

The base io-address is required while the interrupt number is optional.

The device file can be created using:

```
# mknod /dev/cdu31a b 15 0
```

See the file cdu31a for more information on this driver.

Also see the web page put together by Jeffrey Oxenreider (zureal@infinet.com) that covers a lot of common problems with these drives. It can be found at http://lemures.shinma.symix.com/~zureal/cdu31a.html.

Aztcd Driver

```
       Principal author: Werner Zimmermann (zimmerma@rz.fht-esslingen.de)
  Multi-session support: yes
 Multiple drive support: no
Loadable module support: yes
   Reading audio frames: no
           Auto-probing: no
            Device file: /dev/aztcd0, major 29
     Configuration file: aztcd.h
   Kernel config option: Aztech/Orchid/Okano/Wearnes (non IDE) CDROM support?
     Documentation file: aztcd
```

This driver accepts a kernel command line of the form:

```
aztcd=<io-address>
```

where the parameter is the I/O base address of the card (e.g. 0x340).

The device file can be created using:

```
# mknod /dev/aztcd0 b 29 0
```

Note that this driver is for the CDA268-01A only. Other models, including the CDA268-03I and CDA269-031SE are not proprietary and should use the IDECD (ATAPI) kernel driver.

See the file aztcd for more information on this driver.

Gscd Driver

```
       Principal author: Oliver Raupach (raupach@nwfs1.rz.fh-hannover.de)
  Multi-session support: no
 Multiple drive support: no
Loadable module support: yes
   Reading audio frames: no
           Auto-probing: no
            Device file: /dev/gscd0, major 16
     Configuration file: gscd.h
   Kernel config option: Goldstar R420 CDROM support?
     Documentation file: gscd
```

This driver accepts a kernel command line of the form:

```
gscd=<io-address>
```

specifying the I/O base address of the card (e.g. 0x340).

The device file can be created using:

```
# mknod /dev/gscd0 b 16 0
```

See the file gscd and the world-wide web site http://linux.rz.fh-hannover.de/~raupach/ for more information on this driver.

TOC

Mcd Driver

```
       Principal author: Martin  (martin@bdsi.com)
    Multi-session support: no
 Multiple drive support: no
Loadable module support: yes
   Reading audio frames: no
           Auto-probing: no
            Device file: /dev/mcd, major 23
     Configuration file: mcd.h
   Kernel config option: Standard Mitsumi CDROM support?
     Documentation file: mcd
```

This is the older driver for Mitsumi drivers that has been available for some time. You might want to try the newer `mcdx` driver, which has some new features but is possibly less stable.

This driver accepts a kernel command line of the form:

```
mcd=<io-address>,<irq>
```

specifying the I/O base address of the card (e.g. 0x340) and the IRQ request number used.

The device file can be created using:

```
# mknod /dev/mcd b 23 0
```

See the file mcd for more information on this driver.

Mcdx Driver

```
       Principal author: Heiko Schlittermann
  Multi-session support: yes
 Multiple drive support: yes
Loadable module support: yes
   Reading audio frames: no (not supported by hardware)
           Auto-probing: no
            Device file: /dev/mcdx0, major 20
     Configuration file: mcdx.h
   Kernel config option: Experimental Mitsumi support?
     Documentation file: mcdx
```

This is a newer driver for Mitsumi drivers. The older and possibly more stable `mcd` driver is still available.

This driver accepts a kernel command line of the form:

```
mcdx=<io-address>,<irq>
```

specifying the I/O base address of the card (e.g. 0x340) and the IRQ request number used.

The device file can be created using:

```
# mknod /dev/mcdx0 b 20 0
```

If you recently bought a Mitsumi CD-ROM drive, don't assume that it should use this kernel driver. New Mitsumi drives are now EIDE/ATAPI compatible and should use the idecd kernel driver instead.

See the file mcdx for more information on this driver.

Cm206 Driver

```
       Principal author: David A. van Leeuwen (david@tm.tno.)
  Multi-session support: yes
 Multiple drive support: no
Loadable module support: yes
   Reading audio frames: no
           Auto-probing: yes
            Device file: /dev/cm206cd, major 32
     Configuration file: cm206.h
   Kernel config option: Philips/LMS CM206 CDROM support?
     Documentation file: cm206
```

The driver accepts a kernel command line of the form:

```
    cm206=<io-address>,<interrupt>
```

where the first number is the I/O base address of the card (e.g. 0x340). The second is the interrupt channel.

The device file can be created using:

```
    # mknod /dev/cm206cd b 32 0
```

See the file cm206 for more information on this driver.

Optcd Driver

```
       Principal author: Leo Spiekman (spiekman@dutette.et.tudelft.nl)
  Multi-session support: yes
 Multiple drive support: no
Loadable module support: yes
   Reading audio frames: no
           Auto-probing: no
            Device file: /dev/optcd0, major 17
     Configuration file: optcd.h
   Kernel config option: Experimental Optics Storage ... CDROM support?
     Documentation file: optcd
```

The driver accepts a kernel command line of the form

```
    optcd=<io-address>
```

to specify the I/O base address of the card (e.g. 0x340).

The device file can be created using:

```
    # mknod /dev/optcd0 b 17 0
```

See the file optcd for more information on this driver.

Sjcd Driver

```
       Principal author: Vadim V. Model (vadim@rbrf.msk.su)
  Multi-session support: no
 Multiple drive support: no
```

```
Loadable module support: yes
   Reading audio frames: no
           Auto-probing: no
            Device file: /dev/sjcd, major 18
     Configuration file: sjcd.h
   Kernel config option: Experimental Sanyo H94A CDROM support?
     Documentation file: sjcd
```

The driver accepts a kernel command line of the form:

```
    sjcd=<io-address>,<interrupt>,<dma>
```

indicating the base address, interrupt, and DMA channel to be used (e.g. sjcd=0x340,10,5).

The device file can be created using:

```
    # mknod /dev/sjcd b 18 0
```

See the file sjcd for more information on this driver.

Paride Driver

```
       Principal author: Grant R. Guenther (grant@torque.net)
  Multi-session support: no
 Multiple drive support: yes
Loadable module support: yes
   Reading audio frames: no
           Auto-probing: yes
            Device file: /dev/pcd0, major 46
     Configuration file: bpcd.h
   Kernel config option: Parallel port IDE device support?
     Documentation file: paride.txt
```

This is the driver for various types of storage devices that attach to a parallel port. Normally the driver will auto-detect the parallel port device. The documentation describes parameters that can be used to specify the device parameters if auto-detection does not work.

The device file for the first drive can be created using:

```
    # mknod /dev/pcd0 b 46 0
```

See the file /usr/src/linux/Documentation/paride.xt for more information on this driver. Note that this replaces the bpcd driver that was present in older kernels.

SCSI Driver

```
       Principal author: David Giller
  Multi-session support: yes (depending on drive)
 Multiple drive support: yes
Loadable module support: yes
   Reading audio frames: no
           Auto-probing: yes
            Device file: /dev/scd0, major 11
     Configuration file: cdrom.h
   Kernel config option: SCSI CDROM support?
     Documentation file: scsi.txt
```

There are kernel command line option specific to each type of SCSI controller. See the SCSI HOWTO for more information.

Multiple drives are supported (up to the limit of the maximum number of devices on the SCSI bus). Create device files with major number 11 and minor numbers starting at zero:

```
# mknod /dev/scd0 b 11 0
# mknod /dev/scd1 b 11 1
```

While the kernel driver itself does not support reading digital audio frames, some SCSI drives have the capability and will work with the cdda2wav program (which uses the generic SCSI kernel interface).

Also see the discussion of the IDE SCSI emulation driver earlier in this document.

IDECD Driver

```
       Principal author: Scott Snyder (snyder@fnald0.fnal.gov)
  Multi-session support: yes
 Multiple drive support: yes
Loadable module support: no
   Reading audio frames: yes (on supported drives)
           Auto-probing: yes
            Device file: /dev/hd{a,b,c,d},  major 22
     Configuration file: cdrom.h
   Kernel config option: Include support for IDE/ATAPI CDROMs?
     Documentation file: ide-cd
```

This is the driver for ATAPI CD-ROMS. The driver accepts a kernel command line of the form

```
hdx=cyls,heads,sects,wpcom,irq
  or
hdx=cdrom
```

where `hdx` can be any of {hda,hdb,hdc,hdd}, or simply hd, for the "next" drive in sequence. Only the first three parameters are required (cyls,heads,sects). For example `hdc=1050,32,64 hdd=cdrom`.

Getting the IDE driver to recognize your CD-ROM drive can be tricky, especially if you have more than 2 devices or more than one IDE controller. Usually all that is required is to pass the right command line options from LILO. The file /usr/src/linux/Documentation/cdrom/ide-cd explains how to do this. Read it carefully.

Recent Linux kernels have better support for multiple IDE devices. If you have problems with an older kernel, upgrading may help.

Some IDE controllers have hardware problems which the kernel driver can work around. You may need to pass additional parameters to the driver to enable this. See the documentation for details.

4.4 Booting the Linux Kernel

You can now reboot with the new kernel. Watch for a message such as the following indicating that the CD-ROM has been found by the device driver (the message will vary depending on the drive type):

```
hdd: NEC CD-ROM DRIVE:282, ATAPI CDROM drive
```

If the bootup messages scroll by too quickly to read, you should be able to retrieve them using `dmesg` or `tail /var/log/messages`.

If the drive is not found, then a problem has occurred, See the section on troubleshooting.

4.5 Mounting, Unmounting, and Ejecting Devices

To mount a CD-ROM, insert a disc in the drive, and run the `mount` command as `root` (this assumes you created a symbolic link to your device file as recommended above and that an empty directory /mnt/cdrom exists):

```
# mount -t iso9660 -r /dev/cdrom /mnt/cdrom
```

The CD can now be accessed under the directory /mnt/cdrom.

There are other options to the mount command that you may wish to use; see the `mount(8)` man page for details.

You can add an entry to /etc/fstab to automatically mount a CD-ROM when Linux boots or to specify parameters to use when it is mounted; see the `fstab(5)` man page.

Note that to play audio CDs you should *not* try to mount them.

To unmount a CD-ROM, use the `umount` command as `root`:

```
# umount /mnt/cdrom
```

The disc can only be unmounted if no processes are currently accessing the drive (including having their default directory set to the mounted drive). You can then eject the disc. Most drives have an eject button; there is also a standalone eject program that allows ejecting CD-ROMs under software control.

Note that you should not eject a disc while it is mounted (this may or may not be possible depending on the type of drive). Some CD-ROM drivers can automatically eject a CD-ROM when it is unmounted and insert the CD tray when a disc is mounted (you can turn this feature off when compiling the kernel or by using a software command).

Its possible that after playing an audio CD you may not be able to mount a CD-ROM. You need to send a CD audio "stop" command (using a CD player program) before trying the mount. This problem only appears to occur with the SBPCD driver.

Recent kernels support a kernel-based automounter which provides transparent mounting of removable media including CD-ROM. You can find the tools needed to use it at ftp://ftp.kernel.org/pub/linux/daemons/autofs/.

4.6 Troubleshooting

If you still encounter problems after following the instructions in the HOWTO, here are some things to check. The checks are listed in increasing order of complexity. If a check fails, solve the problem before moving to the next stage.

Step 1: Make sure you are really running the kernel you compiled

You can check the date stamp on the kernel to see if you are running the one that you compiled with CD-ROM support. You can do this with the `uname` command:

```
% uname -a
Linux fizzbin 2.2.4 #1 Tue Mar 232 11:23:21 EST 1999 i586 unknown
```

or by displaying the file /proc/version:

```
% cat /proc/version
Linux version 2.2.4 (root@fizzbin) (gcc version 2.7.2.3) #1 Tue Mar 23 11:23:21 EST
1999
```

If the date stamp doesn't match when you compiled the kernel, then you are running an old kernel. Did you remember to reboot? If you use LILO, did you re-install it (typically by running /sbin/lilo)? If booting from floppy, did you create a new boot floppy and use it when booting?

Step 2: Make sure the proper kernel drivers are compiled in

You can see what drivers are compiled in by looking at /proc/devices:

```
% cat /proc/devices
Character devices:
 1 mem
 2 pty
 3 ttyp
 4 ttyS
 5 cua
 7 vcs

Block devices:
 3 ide0
22 ide1
```

First look for your CD-ROM device driver. These are all block devices, in this case we can see that the `idecd` driver with major number 22 was present.

Also make sure that ISO-9660 filesystem support was compiled in, by looking at /proc/filesystems:

```
% cat /proc/filesystems
        ext2
nodev   proc
nodev   devpts
        vfat
        iso9660
```

You can also see what i/o port addresses are being used by a driver with the file /proc/ioports:

```
howto % cat /proc/ioports
 ...
0230-0233 : sbpcd
 ...
```

If any of the drivers you thought you compiled in are not displayed, then something went wrong with the kernel configuration or build. Start the installation process again, beginning with configuration and building of the kernel.

Step 3: Did the kernel detect your drive during booting?

Make sure that the CD-ROM device was detected when the kernel booted. You should have seen a message on bootup. If the messages scrolled off the screen, you can usually recall them using the `dmesg` command:

```
% dmesg
```

or

```
% tail /var/log/messages
```

If your drive was not found then something is wrong. Make sure it is powered on and all cables are connected. If your drive has hardware jumpers for addressing, check that they are set correctly (e.g. drive 0 if you have only one drive). ATAPI CD-ROMS must be jumpered as "single" or "master", and not "slave" when only one IDE device is attached to an interface. If the drive works under DOS then you can be reasonably confident that the hardware is working.

Many kernel drivers using auto-probing, but some do not, and in any case the probing is not always reliable. Use the kernel command line option listed for your kernel driver type. You may want to try several different values if you are not sure of the i/o address or other parameters. LILO can be (and usually is) configured to allow you to enter the parameters manually when booting.

TOC

Another possibility is that you used the wrong kernel driver for your CD-ROM driver. Some documentation may refer to proprietary interfaces as IDE, leading some to mistakenly believe they are ATAPI drives.

Another possibility is that your drive (or interface card) is one of the "compatible" type that requires initialization by the DOS driver. Try booting DOS and loading the vendor supplied DOS device driver. Then soft boot Linux using `Control-Alt-Delete`.

If your drive is not listed in this document, it is possible that there are no drivers for it available under Linux. You can check with some of the references listed at the end of this document for assistance.

Step 4: Can you read data from the drive?

Try reading from the CD-ROM drive. Typing the following command should cause the drive activity light (if present) to come on and no errors should be reported. Use whatever device file is appropriate for your drive and make sure a CD-ROM is inserted; use Control-C to exit.

```
# dd if=/dev/cdrom of=/dev/null bs=2048
^C
124+0 records in
124+0 records out
```

If this works, then the kernel is communicating with the drive and you can move on to step 5.

If not, then a possible cause is the device file. Make sure than the device file in the /dev directory has the correct major and minor numbers as listed previously for your drive type. Check that the permissions on the device file allow reading and writing.

A remote possibility is a hardware problem. Try testing the drive under another operating system, if possible, to determine if this could be the case.

Step 5: Can you mount the drive?

If you can read from the drive but cannot mount it, first verify that you compiled in ISO-9660 file system support by reading /proc/filesystems, as described previously.

Make sure you are mounting the drive with the "-t iso9660" and "-r" options and that a known good ISO-9660 CD-ROM (not Audio CD) is inserted in the drive. You normally must mount drives as user `root`.

Make sure that the mount point exists and is an empty directory.

If you are automatically mounting the CD-ROM on bootup, make sure that you have correct entries in the /etc/fstab file.

If you are running the syslog daemon, there may be error messages from the kernel that you are not seeing. Try using the "dmesg" command:

```
% dmesg
SBPCD: sbpcd_open: no disk in drive
```

There may also be errors logged to files in /var/log, depending on how your system is configured.

Debugging Audio Problems

If the drive works with CD-ROMs, but not for playing audio CDs, here are some possible solutions.

You need an application program to play audio CDs. Some applications may be broken or may not be compatible with your drive. Try other applications and/or try recompiling them yourself. A good place to look for software is ftp://metalab.unc.edu/pub/Linux/apps/sound/cdrom/.

A few of the CD-ROM drivers do not support playing Audio CDs. Check the documentation file or source code to see if that is the case.

Check if the audio can be played through the headphone jack. If so, then the problem is likely related to your sound card. Us mixer program to set the input device and volume levels. Make sure you have installed an audio cable from the CD-ROM d the sound card. Make sure that the kernel sound card driver is installed and working (see the Sound HOWTO).

When All Else Fails

If you still have problems, here are some final suggestions for things to try:

- carefully re-read this HOWTO document
- read the references listed at the end of this document, especially the relevant kernel source files
- post a question to one of the `comp.os.linux` or other usenet newsgroups
- send a question to the Linux mailing list
- try using the latest Linux kernel
- contact your computer dealer
- contact the CD-ROM manufacturer
- send mail to the maintainer of the relevant kernel driver (look in the file /usr/src/linux/MAINTAINERS)
- send mail to me
- fire up emacs and type `Esc-x doctor` :-)

---

5. Applications

This section briefly lists a few of the many applications related to CD-ROM that are available under Linux. Check the Linux Software Map for the latest versions and archive sites.

5.1 Audio CD Players

Several programs are available for playing audio CDs, either through a headphone jack or an attached sound card.

Workman

a graphical player running under X11 and supporting a CD database and many other features

WorkBone

an interactive text-mode player

xcdplayer

a simple X11 based player

cdplayer

a very simple command line based player

Xmcd

an X11/Motif based player

xmitsumi

another X11 based player for Mitsumi drives

xplaycd

another X11 based player, bundled with sound mixer and VU meter programs

cdtool

command line tools for playing audio CDs

Some of these programs are coded to use a specific device file for the CD-ROM (e.g. /dev/cdrom). You may be able to pass the correct device name as a parameter, or you can create a symbolic link in the /dev directory. If sending the CD output to a sound card, you may wish to use a mixer program to set volume settings or select the CD-ROM input for recording.

TOC

5.2 PhotoCD

PhotoCDs use an ISO-9660 file system containing image files in a proprietary format. Not all CD-ROM drives support reading PhotoCDs.

The `hpcdtoppm` program by Hadmut Danisch converts PhotoCD files to the portable pixmap format. It can be obtained from ftp://ftp.gwdg.de/pub/linux/hpcdtoppm or as part of the PBM (portable bit map) utilities, available on many archive sites (look for "pbm" or "netpbm").

The `photocd` program by Gerd Knorr (kraxel@cs.tu-berlin.de) can convert PhotoCD images into Targa or Windows and OS/2 bitmap files.

The same author has written the program `xpcd`, an X11-based program for handling PhotoCD images. You can select the images with a mouse, preview the image in a small window, and load the image with any of the five possible resolutions. You can also mark a part of the Image and load only the selected part. Look for these packages at ftp://ftp.cs.tu-berlin.de/pub/linux/Local/misc/.

The ImageMagick image file manipulation program also supports PhotoCD files. It is available from ftp://ftp.x.org/contrib/applications/ImageMagick/.

5.3 Mkisofs

Eric Youngdale's `mkisofs` package allows creating an ISO-9660 file system on a hard disk partition. This can then be used to assist in creating and testing CD-ROM file systems before mastering discs.

The tools for actually writing data to writable CD-ROM drives tend to be vendor specific. They also require writing the data with no interruptions, so a multitasking operating system like Linux is not particularly well suited.

5.4 ISO-9660 Utilities

These are some utilities for verifying the format of ISO-9660 formatted discs; you may find them useful for testing suspect CDs. The package can be found at ftp://ftp.cdrom.com/pub/unixfreeware/archive/. They were written by Bill Siegmund and Rich Morin.

---

6. Answers to Frequently Asked Questions

6.1 How can a non-root user mount and unmount discs?

Most `mount` commands support the *user* option. If you make an entry such as the following in /etc/fstab:

```
/dev/sbpcd  /mnt/cdrom   iso9660     user,noauto,ro
```

then an ordinary user will be allowed to mount and unmount the drive using these commands:

```
% mount /mnt/cdrom
% umount /mnt/cdrom
```

By default the disc will be mounted with some options that help enforce security (e.g. programs cannot executed, device files are ignored). If this is too restrictive you can use additional options (e.g. the "exec" option will enable execution or programs). See the mount(8) man page for details.

Another method is to get the `usermount` package which allows non-root users to mount and unmount removable devices such as floppies and CD-ROMs, but restricts access to other devices (such as hard disk partitions). It is available on major archive sites.

The archive site `ftp.cdrom.com` has the source file `mount.c` which allows mounting an unmounting of CD-ROMs (only) by normal users. It runs as a setuid executable.

6.2 Why do I get device is busy when unmounting a CD-ROM?

The disc cannot be unmounted if any processes are accessing the drive, including having their default directory set to the mounted filesystem. If you cannot identify the processes using the disc, you can use the `fuser` command, as shown in the following example.

```
% umount /cdrom
umount: /dev/hdd: device is busy
% fuser -v /cdrom
                      USER       PID ACCESS COMMAND
/mnt/cdrom            tranter    133 ..c..  bash
```

On some systems you may need to be root when running the `fuser` command in order to see the processes of other users.

TOC

6.3 How do I export a CD-ROM to other hosts over NFS?

You need to add an entry to the /etc/exports file. Users on other machines will then be able to mount the device. See the `exports(5)` man page for details.

6.4 Can I boot Linux from a CD-ROM?

When initially installing Linux the most common method is to use a boot floppy. Some distributions allow booting a Linux kernel on CD directly from DOS.

With the right CD-ROM, ROM BIOS, and ATAPI CD-ROM drive it is possible to boot directly from CD. The latest version of `mkisofs` supports creating such disks using the El Torito standard for bootable CDs.

6.5 How can I read digital data from audio CDs?

Heiko Eissfeldt (heiko@colossus.escape.de) and Olaf Kindel have written a utility that reads audio data and saves it as `.wav` format sound files. The package is called `cdda2wav.tar.gz` and can be found on `metalab.unc.edu`.

Another utility to extract digital audio is `cdparanoia` available from http://www.mit.edu/afs/sipb/user/xiphmont/cdparanoia.

Because CD-ROM drives are changing very quickly, it is difficult to list which models support reading digital data. You best bet is to get the latest `cdda2wav` or `cdparanoia` package and read the documentation.

For more information on this subject, see the web site http://www.tardis.ed.ac.uk/~psyche/cdda/ and the alt.cd-rom FAQ listed in the references section.

6.6 Why doesn't the find command work properly?

On ISO-9660 formatted discs without the Rock Ridge Extensions, you need to add the `-noleaf` option to the `find` command. See the find(1) man page for details.

(In my experience virtually all recent Linux CDs use the Rock Ridge extensions, so this problem should occur very rarely.)

6.7 Does Linux support any recordable CD-ROM drives?

TOC

The X-CD-Roast package for Linux is a graphical front-end for using CD writers. The package can be found at metalab.unc.edu in /pub/Linux/utils/disk-management/xcdroast-0.95.tar.gz

Also see the Linux CD-Writing HOWTO document, found at ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/CD-Writing-HOWTO or http://metalab.unc.edu/LDP/HOWTO/CD-Writing-HOWTO.html.

6.8 Why do I get mount: Read-only file system when mounting a CD-ROM?

CD-ROM is a read-only media. With some early kernels you could mount a CD-ROM for read/write; attempts to write data to the CD would simple be ignored. As of kernel version 1.1.33 this was corrected so that CD-ROMs must be mounted read only (e.g. using the -r option to mount).

6.9 Why does the disc tray open when I shut down the system?

The sbpcd driver supports automatically ejecting the CD when it is unmounted. In some older kernel versions this was the default behaviour. If you shut down the system, a mounted CD will be unmounted, causing it to eject.

This feature is for convenience when changing discs. If the tray is open when you mount or read a CD, it will also automatically be closed.

I found that this caused problems with a few programs (e.g. cdplay and workbone). As of the 1.1.60 kernel you can control this feature under software control. A sample program is included in the sbpcd documentation file (or use the eject program). You can also control the default behaviour by editing the kernel source file sbpcd.h.

6.10 I have a "special" CD that can't be mounted

The "special" CD is likely an XA disc (like all Photo CDs or "one-offs" created using CD-R drives). Most of the Linux kernel CD-ROM drivers do not support XA discs, although you may be able to find a patch to add support on one of the archive sites.

The sbpcd driver *does* support XA. If you are using this driver you can determine if the disc is XA using the following procedure: go into the file sbpcd.c and enable the display of the "Table of Contents" (DBG_TOC). Build and install the new kernel and boot from it. During each mount the TOC info will be written (either to the console or to a log file). If the first displayed value in the TOC header line is "20",

TOC

then it is an XA disc. That byte is "00" with normal disks. If the TOC display shows different tracks, is also a sign that it is an XA disc.

(thanks to Eberhard Moenkeberg for the above information)

Other possibilities for unreadable CDs are:

1. The disc doesn't use an ISO-9660 file system (e.g. some use SunOS or HFS)
2. It is an audio CD
3. The CD is damaged or defective
4. You put it in the drive upside down :-)

6.11 Do multi-platter CD-ROM drives work with Linux?

Several users have reported success with SCSI multi-disc CD-ROM changers. You probably need to enable the "Probe all LUNs on each SCSI device" kernel configuration option (this is not necessary if your CD changer is already known to the SCSI-driver. Have a look at /usr/src/linux/drivers/scsi.c).

At least one user also had to increase a SCSI timeout value in the kernel driver. A symptom of this is an error message like "wrong fs type, bad option, bad superblock on /dev/sr5, or too many mounted file systems" when trying to mount a CD for the first time, but a second mount immediately afterwards succeeds. To increase the timeout increase the value of IOCTL_TIMEOUT at the beginning of /usr/src/linux/drivers/scsi/sr_ioctl.c and recompile the kernel. A value of 10000 instead of the default 3000 is reported to work with a NEC Multispin 4Xc.

It might be necessary to create additional block special files, so that all LUNs can be accessed. A device file is needed for every LUN. So for a 7 disc changer `/dev/sr0` to `/dev/sr6` are needed (more if you have additional SCSI CD-ROM drives). To create the block special file execute `mknod /dev/sr? b 11 ?` as root with ? being the required number.

The Nakamichi MBR-7 7 disc changer, NEC Multispin 4Xc and Pioneer 12 disc changer have been reported to work.

EIDE/ATAPI multi-disc changers are also available. The kernel has support for some drives using the CDROM_SELECT_DISC ioctl function. The IDE-CD kernel driver documentation file includes source code for a program to select changer slots, or you can use various utilities such as the `eject` program described earlier.

6.12 I get "/cdrom: Permission denied" errors

Some CDs have root directory file permissions that only allow user `root` to read them. This is an error on the part of the CD-ROM vendor and is a real inconvenience. A more common occurrence is for certain files or directories not to be world readable. Some people have patched their kernels to work around the problem.

Also see the related question on hidden files later in this document.

6.13 How do I interpret IDE CD kernel error messages?

What does it mean when I get a kernel message from the IDE CD-ROM driver like "hdxx: code: xx key: x asc: xx ascq: x"?

This is an status/error message from the IDE CD-ROM drive. By default the IDECD driver prints out the raw information instead of wasting kernel space with error messages. You can change the default to display the actual error messages by going into /usr/src/linux/drivers/block/ide-cd.c, changing the value of VERBOSE_IDE_CD_ERRORS to 1, and recompiling the kernel.

6.14 How can I tell what speed CD-ROM I have?

Here's one way. This command measures how long is takes to read 1500K of data from CD:

```
% time -p dd if=/dev/cdrom of=/dev/null bs=1k count=1500
1500+0 records in
1500+0 records out
real 5.24
user 0.03
sys 5.07
```

The transfer rate of single speed drives is 150 kilobytes per second, which should take about 10 seconds. At double speed it would take five seconds, quad speed would take 2.5, etc.

The "real" time above is probably the best number to look at -- in this case it indicates a double speed drive. You can increase the amount of data transferred to get a more accurate value (in case you were wondering, the data does not get cached). You should probably run the command a few times and take the average.

I've also written a small C program that measures and reports CD-ROM data transfer rate; I can send it to you on request.

6.15 My CD-ROM stopped working after Linux was installed

The usual symptom is that the boot disk used to initially install Linux recognized your CD-ROM drive, but after Linux was installed on the hard drive or floppy and rebooted it no longer recognizes the CD-ROM.

The most common reason for this problem is that with some Linux distributions the kernel that is installed on your hard drive (or floppy) is not necessarily the same one that was on your boot disk. You

selected a boot disk that matched your CD-ROM hardware, while the kernel you installed is a "gene kernel that is lacking CD-ROM support. You can verify this by following the troubleshooting guidel discussed previously in this document (e.g. start by checking /proc/devices).

The solution is to recompile the kernel, ensuring that the drivers for your CD-ROM drive and any others that are needed (e.g. SCSI controller, ISO-9660 file system) are included. See the Kernel HOWTO if you don't know how to do this.

If you passed any command line options to the boot disk (e.g. "hdc=cdrom") you need to add these to your boot program configuration file (typically /etc/lilo.conf).

6.16 There are "hidden" files on a CD which I can't read

Some CDs have files with the "hidden" bit set on them. Normally these files are not visible. If you mount the CD with the "unhide" option then the files should be accessible (this doesn't seem to be documented anywhere).

6.17 Where is the CD-ROM API documented?

If you want to write your own application, such as an audio CD player program, you will need to understand the application programming interface (API) provided by Linux.

Originally the CD-ROM kernel drivers used their own ioctl() functions to support features specific to each drive. Header files such as /usr/include/linux/sbpcd.h describe these. Because many of the drivers were based on other drivers, the interfaces, while not identical, have a lot in common.

More recently there has been an initiative headed by David van Leeuwen (david@tm.tno.nl) to standardize the API for CD-ROM drives, putting common code in one place and ensuring that all drivers exhibit the same behaviour. This is documented in the file /usr/src/linux/Documentation/cdrom/cdrom-standard.tex. Several kernel drivers support this. As of the 2.0 kernel all CD-ROM drivers conform to this API.

My book, *Linux Multimedia Guide*, goes into quite a bit of detail on how to program CD-ROM drives, especially for audio functions. See the end of the References section.

6.18 Why don't I see long filenames on this Windows CD-ROM?

If you have a CD-ROM which has long filenames under Windows but not under Linux, it may be formatted using Microsoft's proprietary Joliet filesystem. See the next question for a solution.

6.19 Is Microsoft's Joliet filesystem supported?

TOC

Microsoft has created an extension to the ISO CD-ROM format called Joliet. It allows for long filenames encoded using the 16-bit UNICODE format.

Starting with version 2.0.34 the Linux kernel has support for the Microsoft Joliet file system extensions. You need to enable support for it in the kernel.

If you want to display filenames with native language characters from Joliet CD-ROMs correctly on the screen, you need to enable support in the kernel for the appropriate NLS ISO8859 character sets.

6.20 Does Linux support "enhanced" CDs?

Some audio CDs are "enhanced" with additional data. Typically you will find that these CDs have the usual audio CD tracks as well as a data track which can be mounted as an ISO-9660 file system.

An enhanced CD I examined had Microsoft Windows and Apple Macintosh applications on it (which won't run directly under Linux of course, although I had partial success running the Windows application under the WINE Windows emulator). There were some GIF images which could be viewed using a standard viewer such as `xv` and some animations in Apple QuickTime format which could be viewed using the `xanim` program. It was a multisession disk which some very old CD-ROM drives do not support reading. On the disc was a `readme.txt` file containing an Enhanced CD FAQ.

6.21 Does Linux support DVD-ROM?

SCSI and ATAPI compliant DVD-ROM drives should work under Linux for reading discs formatted with an ISO-9660 file system. In other words they will work as a (possibly large) CD-ROM drive.

Many DVD-ROM discs use the UDF file system. At the time of writing this was still in development. Kernel patches were available from http://trylinux.com/projects/udf/.

However, I am not aware of any support for playing MPEG-2 encoded DVD videos either in software or in conjunction with DVD decoder hardware. Apparently the standard document for the encoding format can only be obtained at a high cost and under a non-disclosure agreement. Another issue is that MPEG decoding typically uses proprietary hardware for which the vendor may not be willing to release programming information. These factors may preclude any freely available DVD video software being developed for Linux.

6.22 Does Linux support CD-RW?

SCSI and ATAPI compliant CD-RW drives should work under Linux for discs formatted with an ISO-9660 file system. This includes the ability to write to the disc.

Many CD-RW discs use the UDF file system. At the time of writing this was still in development. Kernel patches were available from http://trylinux.com/projects/udf/.

7. References

I have already mentioned the documentation files, typically installed in /usr/src/linux/Documentation/cdrom. These can be a gold mine of useful information.

The following Usenet FAQs are posted periodically to news.answers and archived at Internet FTP sites such as ftp://rtfm.mit.edu/:

- alt.cd-rom FAQ
- comp.periphs.scsi FAQ
- Enhanced IDE/Fast-ATA/ATA-2 FAQ

Several other Linux HOWTOs have useful information relevant to CD-ROM:

- SCSI HOWTO
- Hardware Compatibility HOWTO
- Sound HOWTO
- Kernel HOWTO
- Distribution HOWTO
- CD Writing HOWTO

At least a dozen companies sell Linux distributions on CD-ROM; most of them are listed in the Distribution HOWTO.

The following Usenet news groups cover CD-ROM related topics:

- comp.publish.cdrom.hardware
- comp.publish.cdrom.multimedia
- comp.publish.cdrom.software
- comp.sys.ibm.pc.hardware.cd-rom
- alt.cd-rom
- alt.cd-rom.reviews

The `comp.os.linux` newsgroups are also good sources of Linux specific information.

There is a large archive of CD-ROM information and software at ftp://ftp.cdrom.com/pub/cdrom/.

A FAQ document on IDE and ATA devices can be found at ftp://rtfm.mit.edu/pub/usenet/news.answers/pc-hardware-faq/enhanced-IDE/ and at http://www.seagate.com/techsuppt/faq/faqlist.html.

Western Digital, the company that started the IDE protocol, has information available on the IDE

protocol available on their FTP site at ftp://fission.dt.wdc.com/pub/standards/atapi.

A web site dedicated to multimedia can be found at http://viswiz.gmd.de/MultimediaInfo/. Creative Labs has a web site at http://www.creaf.com/.

The *Linux Software Map* (LSM) is an invaluable reference for locating Linux software. The LSM can be found on various anonymous FTP sites, including ftp://metalab.unc.edu/pub/Linux/docs/LSM/ (formerly known as sunsite). There are also various web sites that maintain databases of Linux applications. One such site is http://www.freshmeat.net.

The Linux mailing list has a number of "channels" dedicated to different topics. To find out how to join, send a mail message with the word "help" as the message body to majordomo@vger.rutgers.edu

The Linux Documentation Project has produced several books on Linux, including *Linux Installation and Getting Started*. These are freely available by anonymous FTP from major Linux archive sites or can be purchased in hardcopy format.

Finally, a shameless plug: If you want to learn a lot more about multimedia under Linux (especially CD-ROM and sound card applications and programming), check out my book *Linux Multimedia Guide*, ISBN 1-56592-219-0, published by O'Reilly and Associates. As well as the original English version, French and Japanese translations are now in print. For details, call 800-998-9938 in North America or check the web page http://www.ora.com/catalog/multilinux/noframes.html or my home page http://www.pobox.com/~tranter/.

Linux Chinese HOWTO English Version

Chih-Wei Huang cwhuang@phys.ntu.edu.tw
Cd Chen cdchen@linux.ntcic.edu.tw

v1.04, 2 June 1998

This document demonstrates how to implement Chinese available on Linux system, including those common problems encountered on Linux/UNIX system while using Chinese, the ways to obtain, and shows how to install and setup a variety of different Chinese softwares. Finally, we would like to take a bit little glance at the work for making a complete Chinese environment with Linux system.

1. Foreword

- 1.1 Introduction
- 1.2 Awareness
- 1.3 Newest Version
- 1.4 Copyright and Announcement
- 1.5 Goal
- 1.6 Acknowledgement
- 1.7 Words from The Translator--Frank J.S. Chen

2. Difficulties of Using Chinese on Linux System

3. Where to Get Chinese Softwares?

- 3.1 Websites for Chinese softwares
- 3.2 Using Archie

TOC

4. Display and Input Chinese

- 4.1 xcin+crxvt
- 4.2 yact
- 4.3 bcs16
- 4.4 chdrv
- 4.5 cxterm
- 4.6 XA (Xcin Anywhere)
- 4.7 New Added Inputs
- 4.8 Problems coming with input

5. Chinese X Window System

- 5.1 CXWin
- 5.2 Installation of Chinese fonts
- 5.3 TaBE & B5LE

6. Printing for Chinese

- 6.1 cnprint
- 6.2 ps2cps
- 6.3 bg2ps
- 6.4 gb2ps

7. Chinese Printing Softwares

- 7.1 LaTeX + CJK
- 7.2 ChiTeX
- 7.3 Dtop
- 7.4 ChinesePower
- 7.5 EasyFlow

TOC

8. Useful Chinese Softwares

- 8.1 cjoe - Joe's Own Chinese Editor
- 8.2 celvis
- 8.3 cvim
- 8.4 he
- 8.5 hztty
- 8.6 ktty
- 8.7 Cemacs and CChelp For Emacs
- 8.8 Mule
- 8.9 hc
- 8.10 ctin
- 8.11 Some Other Toys

9. Related Chinese Problems with Others

- 9.1 Netscape Navigator
- 9.2 sendmail
- 9.3 procmail
- 9.4 telnet
- 9.5 less
- 9.6 pine
- 9.7 samba
- 9.8 lynx
- 9.9 minicom
- 9.10 fingerd
- 9.11 tintin++

10. Chinese Linux

- 10.1 Chinese Linux Documentation's Projects
- 10.2 Chinese Linux Project
- 10.3 Chinese websites of Linux
- 10.4 Discussed groups of Linux

TOC

11. FAQs

- 11.1 Why Can't I Enter Chinese?
- 11.2 I have got the NTU TTF fonts, but how can I decompress it?
- 11.3 Netscape's title and tag cannot see Chinese?
- 11.4 My Netscape cannot see Chinese?
- 11.5 How to enter Chinese in Netscape?
- 11.6 What's the difference between ChiTeX and CJK? Can I install them at the same time?

12. Appendix - Chinese Softwares of FTP.IFCSS.ORG

---

1. Foreword

1.1 Introduction

NOTE: I write the document to help novices. However, if it didn't, that's *your* problem, not *mime*! Don't complain that I WASTE your time, because nobody has to answer your questions, and nobody has to write documents to help you! You have been warned.

This is a document, as a part of Chinese Linux Documentation Projects, introduces the progress in fulfilling Chinese on Linux. The special point different from the other CLDP documents is that this HOWTO document, in turn, will be the first one and also *the only* one written in Chinese first and then translated into English later.^_^

I have been made my best effort to ensure the exactness of this document, though; however, I couldn't give a guarantee for myself that all the operations and configurations can execute exactly and smoothly on your system. If any problems or losses have happened to your own system because of the executable examples issued in this document, which you have been followed, the authors, us, *won't* be in charge of any responsibility.

The basic purpose carried with this document is to enormously reduce the questions repeated again and again, just like "Why can't I input/see Chinese?", "Why can't I set xxx up?" and so forth, occurred on Internet. In fact, it is impossible as I knew it...

1.2 Awareness

The shell program used in this document will be sh/bash. If other system(such as csh/tcsh) is your choice, you need to pay attention to the differences amongst them. On the configuration of environment variables, for instance, usage for sh/bash will be,

```
# export TERM=vt100
```

whereas for csh/tcsh ,

```
# setenv TERM vt100
```

1.3 Newest Version

Simultaneously, I provide the following various formats of different versions due to the mechanism of SGML document formats.

- Plain Text

- HTML
- PostScript
- SGML

You can download these formats from the ftp site ftp://ftp.phys.ntu.edu.tw/pub/CLDP/howto-translations/.

Also, you may read directly the newest version of this document on line.

- http://www.phys.ntu.edu.tw/~cwhuang/pub/os/linux/CLDP/Chinese-HOWTO.html
- HomePage for Chinese Linux Documentation Projects

English version could be obtained from:

- http://sunsite.unc.edu/LDP/HOWTO/Chinese-HOWTO.html
- http://www.phys.ntu.edu.tw/~cwhuang/documents/linux-howto/Chinese-HOWTO.html

Japanese translation is also available at(thanks to `isaji@mxu.meshnet.or.jp`):

- http://jf.gee.kyoto-u.ac.jp/JF/JF-ftp/other-formats/Chinese/Chinese-HOWTO.html
- ftp://ftp.kuis.kyoto-u.ac.jp/Linux/JF/other-formats/Chinese/

1.4 Copyright and Announcement

The copyright of this Chinese HOWTO document belongs to Chih-Wei Huang < cwhuang@phys.ntu.edu.tw>. You can make any copies, distributions and spread all or parts of this document. Acyually, you are encouraged very much to fulfill that statement real so that more and more netters could get some helps from it under the use with non-business. (DO NOT remove my name out of it, please.) For business, contact the authors first. The trademarks or packages mentioned in this document belong to the companies or the personal, respectively.

1.5 Goal

I wish this document can offer a complete and self-sufficient explanation in the aspect of using Chinese on Linux system. What matters concerned with implementations of Chinese on Linux will put it all together here! However, limited to the intelligence and private ability, the ultimate goal of my willingness is almost a "mission impossible"; therefore, if there are losses or drawbacks in it, please let me know it.

1.6 Acknowledgement

I want to thank to Hong Zhang < zhangh@earthlink.net> for being the original founder of this Chinese HOWTO document and for his papers of being the foundation of this new version. Also, I need to thank to Cd Chen < cdchen@linux.ntcic.edu.tw> for completing the section Display and Input Chinese.

The translator, Frank_J.S._Chen < frank63@ms5.hinet.net>, also deserved the acknowledgments for translating this document into English.

Here, there are many people ought to be acknowledged much better for the reasons of devoting their efforts to make Linux more reliable in Chinese. Because of their hard works, we can nowadays enjoy ourselves on a comfortable Chinese environment of Linux.

1.7 Words from The Translator--Frank J.S. Chen

This work begins on 1/8/98 and ends on 1/11/98, which I call the version of 0.1, as I always did for the initial states of anything. I appreciate the hard work of Chinese version for its delicate statements and not leaving out any detail in the field of Chinese practices. The more I translated, the more I learned. After examining all the sentences I've made, there are definitely many of them won't like real American-English, or the so called Chinese-styled English. Well, if you are skilled in this field of translation or are interested in the circle of computers science, anything about this English article, just like grammar, writing, internationalization of locale with Linux, translation, programming with C/C++, and so forth, will be sincerely grateful and excessively welcome, too.(1/11/98, modified on 6/2/98) This is the secondary modification of it, focusing on the falses of typing, incorrect grammar, and correcting the meanings of ambiguous words(maybe a little research on sematics, just for this document:-)). I have no ideal about the translating quality of this paper, if you feel something strange or ill while reading, please let me know about it in my real heart. Pointing out the bad parts will be marvelous in spreading out this howto. As far as I know, English version of it had been translated into several foreign languages including Japanese, Swedish, etc. If you want to do another one, just go ahead, but we suggest you to inform anyone of us about that.(6/2/98)

---

2. Difficulties of Using Chinese on Linux System

This section makes an attempt to do a general description for the possible obstacles in using Chinese on Linux; then you could find the key points out much easier as you meet with these problems. As a matter of fact, the shortcomings described here not only appear on Linux but also the other system. Even more, we can say that the whole computers environments are concerned. If this section is not suited for your tastes or you are eager to act directly, then you can jump onto the section Display and Input Chinese!

A Chinese word is composite of two bytes in computers, as we all know. The most popular encoding methods includes BIG5 codes available in the area of Taiwan and GB codes available in the mainland China. The first byte of each word is almost bigger than numeric values 128, which is what we called the non-ASCII codes.(The ASCII codes means codes smaller than 128.)

Yes! Then so what? Here are the points! Because of different kinds of reasons, in the early days, many programs didn't consider the possibility of non-ASCII codes as a part of entering data.

These kinds of programs always assume that the data prepared to manipulation are all limited in the range of ASCII codes, and the most worst is that when they meet with non-ASCII codes, an assumption of their non-existence and a truncation of the 8th bit is the most frequent method they took. This is the so called 8-bit clean problem.

Your program, for example, always take it for granted that your inputs are all the 7-bit-width ASCII codes. When you enter Chinese words, it will erase the 8th bit so that the inputs under circumstances of Chinese will become disturbed codes all the way.

Communication programs on Internet are usually could only transmit 7-bit data. A notorious substance is the earlier `sendmail` program. `sendmail` could only send and receive 7-bit mails, causing that the strategies of many odd encoding methods, Encoding which made the receivers an excessive disturbance, are recognized as sending out Chinese mails(like uuencode, base64, QP and so on). (Frequently, I thought by myself that if the founders of emails could have put much foresight on it, then we could have little problems nowadays perhaps.)

This problem seems to be more complicated on Internet. Even you and your receivers all have the machines installed with `sendmail` program which might manipulate with Chinese mails, the receiver might get disturbed mails in any way. This is because this mail before its arrival at the target may travel over several hosts settled on Internet, if one of the hosts' `sendmail` cuts the 8th bit off, then things go down. For the programs with the architecture of client/server, the problem may be on the end of client, or on the end of server; otherwise both of them are.

Applications which are incapable of identifying the Chinese encoding are also a major problem, apart from being unable to deal with non-ASCII codes' data. That is, most programs(even if they can deal with 8-bit data accurately) all take a Chinese word as two individual bytes. This won't cause problems under some conditions, but it will show an unfortunate disaster under some spots.

The most obvious matter is that, for instance, even if you can input Chinese words properly, but when you hit the backspace key a time trying to delete a complete word, the whole word will be split into two parts meaning that only one byte(column) can backspace on monitor and the redundant half one then become a disturbed code. More over than that, you might change new line at the second byte of a Chinese word in some text editors and then disturbed codes occurred. Besides, these text editors might think that a long Chinese sentence as a long English sentence without changing to a new line, making the picture of screen ugly and chaotic.

There are more worse matters, too ! Some Chinese words contain special codes which correspond to some particular meaning for some applications and might make these programs producing severe faults while meeting with that codes or just collapse.

Here below will try to propose some resolved methods but segmental, incomplete and also unsatisfactory. Only when all softwares can fit with Chinese, then the problems could really resolve perhaps.

However, more and more programs have noticed the significance of internationalization, for example, most hosts' `sendmail` programs now can cope with 8-bit mails exactly --- Not only transmitting Chinese mails need 8-bit, but also many multi-media mails need 8-bit. Lots of softwares already don't need to modify at all or just open some special options for the purpose of using Chinese. Simultaneously, there are more and more persons devoting to the birth of Chinese softwares. Let's us wait and expect for it.

---

3. Where to Get Chinese Softwares?

3.1 Websites for Chinese softwares

Most of the Chinese softwares issued in this document can get from these sites below:

- ftp://linux.cis.nctu.edu.tw/
- ftp://linux.csie.nctu.edu.tw/pub/chinese/
- ftp://chinese.linux.org.tw/pub/Chinese/
- ftp://NCTUCCCA.edu.tw/Chinese/ifcss/software/
- http://cnapps.ifcss.org/
- http://irpslibrary.ucsd.edu/software/chinese_software/index.html

3.2 Using Archie

There are usually some ordinary questions such as "Where can I find xxx...?" appeared on Internet. Actually, there was an excellent utility named archie could help you to search softwares you needed. Make a link to http://archie.edu.tw/archie.html, and then enter the name of softwares that you want. If you have a desire to use `xcin`, for instance, just input xcin can you get all these softwares concerned with xcin displayed on screen. Then, choose the nearest site to download those softwares. More illustrations in detail can be found on the archie's homepage for self-reference.

You can use text mode, too. `telnet archie.edu.tw` and login as `archie`, then type `prog FILENAME` and press ENTER key, for instance,

```
Archie > prog xcin-2.1d.tar.gz
# Search type: sub.
# Your queue position: 1
# Estimated time for completion: 5 seconds.
working... =

Host linux.cis.nctu.edu.tw    (140.113.166.117)
Last updated 08:47 27 Nov 1997

    Location: /packages/chinese/xcin
      FILE    -rw-r--r-- 1106789 bytes  08:58 25 Jul 1997  xcin-2.1d.tar.gz
```

4. Display and Input Chinese

For environments of Chinese system under DOS or of Windows 95 for Chinese that you have been always using to get through Linux hosts, there is no doubt that you don't need to take a good look at this section. However, when your local host is Linux-based system, configurations for Chinese system are definitely necessary.

4.1 xcin+crxvt

It is truly suggested that you can take the combinations of xcin and crxvt as a pathway to solve the problems of Chinese I/O if the X Window System is acquainted with you.

Xcin, with a contraction of X Window Chinese INput, is a Chinese Input System executed under the X mode. Because xcin is utilizing the architecture of client/server, all you have to do is just to start one xcin window so that you can manipulate with many crxvt virtual terminals under the same window, which can exhaust much little resources. Also, it offers some wise programs of input, like Wang-Hsing Input or Natural Input, which can select the matching phases by themselves. In the early age, in order to convert fonts and inputs table to fit with xcin, you need the ETen Chinese System mounted first before the installation of xcin. After the man, Tung-Han Hsieh, < thhsieh@twclx.phys.ntu.edu.tw> became responsible for the voluntary maintenance of xcin, this terrible problem had been resolved!

The newest version is 2.3.02 now. But this one is just a version of beta. For the more stable one, I advise you to fetch the xcin-2.1d for a suggestion in my mind.

Getting the software

Download the sources of xcin from this ftp site below:

ftp://linux.cis.nctu.edu.tw/packages/chinese/xcin/xcin-2.1d.tar.gz
ftp://linux.cis.nctu.edu.tw/packages/chinese/xcin/xcin-2.3.02.tar.gz

Installation of xcin

You need the xcin-2.3.02.tar.gz file at hand first to setup xcin of version 2.3.02 and untar and unzip it under any certain directory.

```
# tar xzvf xcin-2.3.02.tar.gz
....
# cd xcin-2.3.02
# ./configure (Follow the instructions on screen to modify the options you picked up
in turn.)
# make
# make install
```

So, you have finished the setup of xcin of version 2.3.02 from now on.

You can also get the same sort of xcin, named xcin-2.3.02.i386.rpm, maintained by Cd Chen for RedHat Linux System.

ftp://linux.ntcic.edu.tw/personal/cdchen/Chinese-RedHat-Packages/XCIN/RPMS/xcin-2.3.02-1.i386.rpm

Install it by

```
# rpm -Uvv xcin-2.3.02-3.i386.rpm
```

4.2 yact

yact is the system of displaying and inputing Chinese running under the terminal mode. The most different point from chdrv is that yact uses your computers' displaying card through svgalib library. Without the information of your video card contained in the svgalib, you may not probably make yact work on your Linux.

The most admirable point for yact is that the fonts are scanned with 24x24 on the monitor and are more beautiful than other terminal Chinese input system. And it is more smooth than the others in dealing with scrolling pages. The newest version of yact is yact-p4 now.

You can get yact here below,

ftp://linux.cis.nctu.edu.tw/packages/chinese/yact/yactp4.tar.gz

The ways to setup yact are simple, too. After getting the sources of yact, decompress it and examine whether the Makefile file is correct or not, then type make all install directly on shell prompt to complete the installation of bits files. Read the README file for more detailed information.

Next is the step of fonts' setup. yact uses the HBF fonts, but you can't find the fonts in the packages of yact, which means you need to take extra actions to make it available. Having ETen Chinese System mounted is an original recommendation from the founder for its fonts. Copy the HBF fonts' description file `et24.hbf` under the fonts directory of sources of yact to the directory `/usr/local/lib/yact` and rename it as `hzfont.hbf`. Then copy STDFONT.24, SPCFSUPP.24, SPCFONT.24 and ASCFONT.24 fonts files to the directory `/usr/local/lib/yact` and rename ASCFONT.24 as `12x24`.

The free HBF fonts are available, too. Change names of these 256 ASCII fonts as `12x24` and names of the HBF fonts' description file as `hzfont.hbf`. Put both of description files and fonts files into `/usr/local/lib/yact`, then everything will be ok.

4.3 bcs16

Because yact needs the svgalib 1024x768 mode to work, many netters without ET4000 series cards cannot get well supports from it, causing their sickness of this excellent Chinese system. The bcs16 is modified from yact according to this drawbacks. It needs only 640x480 and can work very well on most video cards.

Get bcs16 from this site:

ftp://linux.cis.nctu.edu.tw/packages/chinese/bcs/bcs007a.tgz

Installation

- Decompress and compile it to setup

```
# tar zxvf bcs007a.tgz
# cd bcs16
# make
# cp bcs16 /usr/local/bin
```

- bcs16 take yact's input table as default and this is the reason that you must setup yact first before bcs16 can work. However, version of v0.05 or laters have been emerged with the support of individual Boshiamy input, so if you don't need other inputs provided from yact, you don't need to install yact first. Just fetch Boshiamy input table, `liubig5.tab`, of DOS version and put it into the directory `/usr/local/lib/yact` .
- Put files under directory `data` into `/usr/local/lib/yact` and all the legal fonts files, namely `spcfont.15`, `spcfsupp.15`, and`stdfont.15` , are need to put together in the same place, too.
- If you want to see GB codes or Japanese or Korean, get fonts `cclib.16.gz cclib.16a.gz`, `jis.16.gz`, `ksc.16.gz` from the site below and decompress them into `/usr/local/lib/yact`.

  ftp://nctuccca.edu.tw/Chinese/ifcss/software/fonts/
- If you have the files of creating fonts or sets of Sea-Fonts, you can copy them to the directory `/usr/local/lib/yact/usrfont.15m` too.
- Execute `bcs16`. Pressing `Alt-H` can get more advanced explanations of instructions.

The founder of bcs16 is cnoize.bbs@bbs.cis.nctu.edu.tw.

4.4 chdrv

chdrv is a Chinese emulator program displaying and entering Chinese through console. Because chdrv access the tty device directly, it must be activated by the root. Now, chdrv is maintained by Yu-Chung Wang < wycc@iis.sinica.edu.tw> and the newest one is chdrv-1.0.10.

Getting chdrv

You can get it from

ftp://linux.cis.nctu.edu.tw/packages/chinese/chdrv/

Get the source package, chdrv-1.0.10.tar.gz, binary package, chdrvbin-1.0.10.tar.gz, and fonts package, chdrvfont.tar.gz, respectively.

Installation of chdrv

Unzip and untar the compressed binary file,

```
# tar xvzf chdrvbin-1.0.10.tar.gz
# mv chdrvfont.tar.gz chdrv-1.0.10/
# cd chdrv-1.0.10
```

Read the illustrations for installation in file `INSTALL.1.0` to modify file `install`. If you want your shadow password to works, you need to change the settings in `chinese.conf`. Erase the following comments off,

```
LOGINPROGRAM /bin/telnet
```

Freeze this line into remarks,

```
LOGINPROGRAM /bin/login
```

Now, you can execute the installation script.

```
# ./installbin
```

4.5 cxterm

cxterm is a Chinese virtual terminal running under X Window System, and is the oldest Chinese displaying and inputing virtual terminal, which is provided various codes for Chinese, including BIG5 codes, GB codes, HZ codes and so on. Because each cxterm opened needs to load Chinese data into the main memory, system resources are exhausted quite largely for it.

Getting cxterm

The newest one is version of 5.0.p3 named `cxterm5.0.p3.tar.gz`, which contains both of cxterm codes and Chinese fonts. You can get it from here below,

ftp://ftp.ifcss.org/pub/software/x-win/cxterm/

Or the rpm at `ftp.redhat.com`: `cxterm-color-5.0p3-1.i386.rpm`, `cxterm-color-big5-5.0p3-1.i386.rpm`, `cxterm-color-gb-5.0p3-1.i386.rpm`.

ftp://ftp.redhat.com/pub/contrib/hurricane/i386/

Installation of cxterm

Decompress the packed,

```
# tar -xvzf cxterm5.0.p3.tar.gz
```

TOC

The upper instruction will produce a new directory `cxterm-5.0`, and then invoke instructions as follows:

```
# cd cxterm-5.0
# ./config.sh
```

You need login as root to execute ``./config.sh'' if you wish all users on your computer can use cxterm well, then you can finish the steps of setup following the descriptions below:

```
  0. Read COPYRIGHT Notice
  1. Compile, Install, and Configure "CXTERM 5.0" in One Step

  2. Compile cxterm (not to install)
  3. Install cxterm (after successful compilation in 2)
  4. Install additional Chinese font(s) for your X window
  5. Configure your account for using cxterm (after installation in 3)

  x. Exit
Please choose (0/1/2/3/4/5/x) :
```

If you wish all things going through automatically, please choose `1`, and then enter the name of directory where to store the cxterm. If asking me, I will suggest this place, `/usr/local/chinese`¡D There are two kinds of Chinese fonts coming with the packed package, choosing `1` and `3` will make the procedure of installation all automatic. In addition, you can select `4` to install some extra fonts, too. After achieving this setup, you need to put `cxterm` and `CXterm` into the path of searching directory.

```
# export PATH=$PATH:/usr/local/chinese/bin
```

`CXterm` is just a shell script to load the resources of X and cxterm in. For using GB codes, invoke this following command,

```
# CXterm -gb
```

For BIG5 codes, invoke this following command,

```
# CXterm -big5
```

Color patch of CXterm

On the website `ftp://ftp.ifcss.org/pub/software/x-win/cxterm`, there is a color patch for cxterm as well. Using this patch can make cxterm showing colors of ANSI. Assuming that you put the original files of cxterm under the `/tmp/cxterm-5.0`, then

```
# cp cxterm-5.0.p3-color.patch.gz /tmp
# gzip -d cxterm-5.0.p3-color.patch.gz
# patch < cxterm-5.0.p3-color.patch
# cd cxterm-5.0
# ./config.sh
```

4.6 XA (Xcin Anywhere)

XA is a small tool of an abbreviation for Xcin Anywhere, which can let you enter Chinese words with xcin under any common X-based softwares. If making XA coordinating with CXWin(or XA+CV), you can get an environment of accessing Chinese for softwares not supported with Chinese. Thus, the xterm will become cxterm-like naturally. Cool, doesn't it? XA is now extremely unstable as though, and you have to take charge of all risks if you want to explore it. As most things do, before working with XA, you must make xcin installed first. You can pick XA up into your pocket from here below:

ftp://linux.cis.nctu.edu.tw/packages/chinese/xcin/XA/

Decompress the packed. Run `./configure` to produce `mk` and `config.h` on account for the compilation by using `./mk`. If it works, key in `./mk test xterm` next to see if the xcin can be called out and enter Chinese under xterm to test. If there is no

problem, copy `wrap.so` to `/usr/local/lib/` following the syntax below:

```
# LD_PRELOAD=/usr/local/lib/wrap.so netscape &
```

Then, just follow the general method of using xcin.

Founder of XA is weijr.bbs@bbs.ntu.edu.tw.

4.7 New Added Inputs

At present, there are two common formats of input table, namely `tit` and `cin`, which both are plain text of formats. (That's means you can edit them directly from text editors.) However, most Chinese Systems almost provide utilities for the purpose of exchanging pure text of formats into special binary of formats in order to speed up searching method. Before you setup some certain input, you must own it's `tit` , `cin`, or formats after transformation first .

I will take the Bo-Shia-My input as an example to show how to add it under each kind of Chinese system. The input tables mentioned here can be found at `ftp://ftp.cis.nctu.edu.tw/UNIX/Chinese/Boshiamy/`.

Other `tit` files are also available at `ftp://ftp.ifcss.org/pub/software/x-win/cxterm/dict/`.

xcin

Make use of utility `cin2tab` provided from xcin to transform the `cin` table into `tab` one.

```
# cin2tab boshiamy.cin
```

It will produce the two files: `boshiamy.tab` and `boshiamy.tab.rev`. Put them into the directory of xcin and activate xcin next:

```
# xcin -in9 boshiamy.tab
```

To use Bo-Shia-My input, press the combination keys `CTRL-ALT-9`.

yact & bcs16

yact takes the use of cit of version 2, and the same as cxterm. You can use the tool `tit2cit` accompanied with yact to make transformation available between `boshiamy.tit` and `cit`.

Move `boshiamy.cit` into `/usr/local/lib/yact` and establish a symbol link for it:

```
# ln -s boshiamy.cit 9
```

Like xcin does, Hit `CTRL-ALT-9` for calling Bo-Shia-My input.

chdrv

At first, put `boshiamy.tbl` into `/usr/local/lib/chinese`. Secondly, modify `/etc/chinese.conf`, adding the follow section `INPUT` into it.

```
BEGIN INPUT
PHONETIC /usr/local/lib/chinese/phone.def
MULTI /usr/local/lib/chinese/boshiamy.tbl
END INPUT
```

Finally, use utility `chconfig` of chdrv to make the contents of `/etc/chinese.conf` effective.

cxterm

Change `boshiamy.tit` into `cit` or `citnf` with the utility `tit2cit` of cxterm, then modify `.Xdefaults` to set a combination key for acting Bo-Shia-My input. Please refer to the technical document coming with cxterm for more information about installation and implementation in detail.

4.8 Problems coming with input

After you accomplished the establishments of Chinese System, you have already been able to display Chinese on your Linux through monitors. However, as using a Chinese editor, you will find that the Linux system can only display Chinese but cannot accept the input of Chinese. You have to modify two spots, by yourself, to make Linux system becoming acceptable with output and input of Chinese if you want to improve these problems At beginning, you need to add the locale setting to the shell profile file (Referring to the locale mini-HOWTO when concerning with locale). Additionally, adding sets about inputs for the `.inputrc` file under your home directory is necessary too. (Please create it yourself if `.inputrc` didn't ever exist.) Here are the configurations concerning with shell profile file and `.inputrc` file which are distributing on public domain(just as a reference, though):

Bash Shell: Appending the following contents into `/etc/profile`, please.

```
stty cs8 -istrip
stty pass8
export LANG=C
export LC_CTYPE=iso-8859-1
```

Tcsh Shell: Appending the following contents into `/etc/csh.login` or in `/etc/csh.cshrc`, please.

```
stty cs8 -istrip
stty pass8
setenv LANG C
setenv LC_CTYPE iso-8859-1
```

`$HOME/.inputrc` file for setup lists as follows:

```
set convert-meta off
set output-meta on
```

Eventually, prepare a text file contained with Chinese words by yourself, and use tool `grep` to search it. If it can find something exact, this means that your Linux system can work with Chinese words already.

5. Chinese X Window System

X Window System is a software with powerful environment of graphical user interface under UNIX System. XFree86 is a modified version from MIT X Window System and also freely distributed.

5.1 CXWin

The Chinese X, abbreviated with CXWin, is really a patch of XFree86, making it can show Chinese words under X Window System. Founder of it is srlee. CXWin only support BIG5 code at present, and can let you be possessed of Chinese pop-up menus, of Chinese titles of windows, and of showing Chinese under a variety of window managers and applications.

How to get it?

You can get CXWin 3.3 from this ftp site below:

ftp://linux.cis.nctu.edu.tw/packages/X/Xserver/CXwin/3.3/

If you use XFree86 3.2, you can get the Linux ELF format of version 3.2.

ftp://linux.cis.nctu.edu.tw/packages/chinese/CXwin/binaries/

All you need is to get the corresponding X server. For example, if your video card is matched with `XF86_SVGA`, just take `XF86_SVGA.gz` home. `XF86_SVGA.gz` should work properly on most video cards.

If you system is compatible with rpm, you can get the newest CXWin 3.3.1 of rpm.

ftp://linux.cis.nctu.edu.tw/packages/X/Xserver/CXwin/3.3.1/

Patches

You can get the patches at the same place to compile to binary files by yourself if you've got the sources of XFree86.

Installation

Make sure that you have these shared libraries if your choice is CXWin 3.3 (You can use `ldconfig -p` for an observation):

```
libm.so.5 => /lib/libm.so.5.0.6
libdl.so.1 => /lib/libdl.so.1.7.14
libc.so.5 => /lib/libc.so.5.3.12
```

If your choice is CXWin 3.2, then you need the libc of version 5.2.18 or newer.

First, unzip the compressed files of gzip.

```
# gzip -d XF86_SVGA.gz
```

Duplicate the original X server as a copy as the root identification.

```
# cd /usr/X11R6/bin
# mv XF86_SVGA XF86_SVGA_BACKUP
```

Put the decompressed CXWin into the exact place and create the decent symbolic link:

```
# mv /tmp/XF86_SVGA /usr/X11R6/bin
# chown root.bin XF86_SVGA
```

```
# chmod 4755 XF86_SVGA
# ln -sf XF86_SVGA X
```

If your choice is rpm, then everything is more simple and more compact.

```
# rpm -Uvv XFree86-SVGA-3.3.1-1c.i386.rpm
```

CXWin needs the following *four BIG5 fonts* to work normally: `taipei15`, `taipei16`, `taipeik20` and `taipeik24`. Install these Chinese fonts in accordance with the method of next section.

5.2 Installation of Chinese fonts

Where to get these fonts?

Maybe you have already installed some Chinese fonts; for GB fonts, you can find them out through the next way:

```
# xlsfonts | grep gb
-cclib-song-medium-r-normal--0-0-72-72-c-0-gb2312.1980-1
-cclib-song-medium-r-normal--16-160-72-72-c-160-gb2312.1980-1
-isas-fangsong ti-medium-r-normal--0-0-72-72-c-0-gb2312.1980-0
-isas-fangsong ti-medium-r-normal--16-160-72-72-c-160-gb2312.1980-0
-isas-song ti-medium-r-normal--0-0-72-72-c-0-gb2312.1980-0
-isas-song ti-medium-r-normal--16-160-72-72-c-160-gb2312.1980-0
-isas-song ti-medium-r-normal--24-240-72-72-c-240-gb2312.1980-0
hanzigb13fs
hanzigb13st
hanzigb16fs
hanzigb16st
hanzigb24st
```

The last five fonts are the previous five ones' aliases. To check the installed BIG5 fonts, using

```
# xlsfonts | grep big5
-kc-kai-medium-r-normal-fantizi-0-0-75-75-c-0-big5.eten.3.10-1
-kc-kai-medium-r-normal-fantizi-20-200-75-75-c-200-big5.eten.3.10-1
-kc-kai-medium-r-normal-fantizi-24-240-75-75-c-240-big5.eten.3.10-1
-kc-ming-medium-r-normal-fantizi-0-0-75-75-c-0-big5.eten.3.10-1
-kc-ming-medium-r-normal-fantizi-15-150-75-75-c-160-big5.eten.3.10-1
-kc-ming-medium-r-normal-fantizi-24-240-75-75-c-240-big5.eten.3.10-1
-uw-ming-medium-r-normal-fantizi-0-0-75-75-c-0-big5.eten.3.10-1
-uw-ming-medium-r-normal-fantizi-16-160-75-75-c-160-big5.eten.3.10-1
-uw-ming-medium-r-normal-fantizi-20-200-75-75-c-200-big5.eten.3.10-1
-uw-songti-medium-r-normal-fantizi-0-0-75-75-c-0-big5.eten.3.10-1
-uw-songti-medium-r-normal-fantizi-20-200-75-75-c-200-big5.eten.3.10-1
-uw-songti-medium-r-normal-fantizi-24-240-75-75-c-240-big5.eten.3.10-1
```

You can take all of it from this ftp site `ftp://ftp.ifcss.org/pub/software/fonts/` if you haven't installed any of them or if you want more of them. The `gb/bdf/` amongst them is a GB font while the `big5/bdf` is a BIG5 font.

Installation of X fonts

Assuming that you have already got a BIG5 font, `eb5-24k2.bdf.gz`, and a GB font, `gb24st.bdf.gz`. What you have to do first is that changing them into the format of `.pcf`.

```
# gzip -cd eb5-24k2.bdf.gz | bdftopcf -t > eb5-24k2.pcf
# gzip -cd gb24st.bdf.gz | bdftopcf -t > gb24st.pcf
```

```
# compress *.pcf
```

Then install them under `/usr/X11R6/lib/X11/fonts/misc`

```
# mv *.pcf.Z /usr/X11R6/lib/X11/fonts/misc/
# cd /usr/X11R6/lib/X11/fonts/misc
# mkfontdir .
```

Now it should contain the following two lines in `fonts.dir`:

```
eb5-24k2.pcf.Z -kc-kai-medium-r-normal-fantizi-24-240-75-75-c-240-big5.eten.3.10-1
gb24st.pcf.Z -isas-song ti-medium-r-normal--24-240-72-72-c-240-gb2312.1980-0
```

You can create other aliases for these fonts for the advantages of saving typing time when using Chinese softwares. Add the two lines in `fonts.alias`,

```
gb24st    "-isas-song ti-medium-r-normal--24-240-72-72-c-240-gb2312.1980-0"
taipeik24 "-kc-kai-medium-r-normal-fantizi-24-240-75-75-c-240-big5.eten.3.10-1"
```

Next, tell X Server to reload these fonts,

```
# xset fp rehash
```

If everything is fine, you can use the new fonts now:

```
# cxterm -GB -fn 12x24 -fh gb24st &
```

You can put these fonts into another directory, too, for instance, `/usr/X11R6/lib/X11/fonts/chinese`. If this works, you have to add this line `/etc/XF86Config` to the Section "Files",

```
    FontPath    "/usr/X11R6/lib/X11/fonts/chinese/"
```

So, when you start X next time, you can use these fonts without any problems.

Transformations from other fonts to X fonts

There is a TTF2BDF program, written by lwj, can let you transform the TTF fonts of Windows into BDF fonts. You can get it from here below:

ftp://linux.cis.nctu.edu.tw/packages/chinese/TTF2BDF.EXE

This is a Win32 program, thus you must execute it on Windows 95 or NT.

Next step is that change BDF into PCF, then you can use it under X. (You can use BDF directly under X; but that BDF carried with bigger sizes means PCF will be a better choice though.)

```
# bdftopcf -t < ming.bdf > ming.pcf
```

You may change HBF into BDF and PCF fonts, too. Here is the ftp site for the utility of transformation.

ftp://ftp.ifcss.org/pub/software/fonts/utils/hbf.tar.gz

```
# hbftobdf ccfs24.hbf > ccfs24.bdf
# bdftopcf -t < ccfs24.bdf > ccfs24.pcf
```

Other nice font conversion programs are available at:

ftp://crl.nmsu.edu/CLR/multiling/General/ttf2bdf-1.8-ELF.tar.gz
ftp://crl.nmsu.edu/CLR/multiling/General/xmbdfed-3.0.tar.gz

5.3 TaBE & B5LE

TaBE and B5LE(Big5 Locale Environment) are projects both based on locale for solving the Chinese ability under X. If we can finish them, then the problems coming with Chinese I/O under X can be resolved thoroughly.

The TaBE author, shawn, has been taken for the military service in Taiwan, and the homepage of TaBE had been removed either(so awfully?), so this project has been dead from that time.

Author of B5LE is Thinker < Thinker.bbs@bbs.yzu.edu.tw>. However, the major platform is FreeBSD , though. If someone install B5LE successfully on Linux platform, please let me know about it. The concerned homepage is listed as follows:

http://ftp-cnpa.yzit.edu.tw/~thinker/B5LE/

---

6. Printing for Chinese

This section will teach you how to print Chinese documents, but without the demonstrations of how to set up your printer; that you have to make it all ready by yourself. Tools introduced in this section are all for transformations to PostScript format; hence, you need to set your printer up in order to print PostScript files. If your printer doesn't support PostScript directly, you can install `ghostscript` instead. Please refer to Printing HOWTO.

6.1 cnprint

`cnprint` is a printing tool for the purpose of changing Chinese text files into PostScript format, of which commands are the same as the standard ones. It supports GB, HZ, and BIG5 codes simultaneously.

Where to get it?

Download it from `ftp://ftp.ifcss.org/pub/software/unix/print`, of which name is `cnprint280.tar.gz`.

Setting up cnprint

`cnprint280.tar.gz` contains only five files.

```
# ls
cnprint.1          cnprint.cmd        cnprint280.README
cnprint.c          cnprint.help
```

Compile it through this way below:

```
# gcc cnprint.c -o cnprint
# mv cnprint /usr/local/bin
# mv cnprint.1 /usr/local/man/man1
```

Installing HBF fonts

HBF fonts includes both description files and fonts files. A `.hbf` is the header file describing fonts, with a texture of plain text, recording the file names of its fonts files. Both of two files must be installed yourself. You can find them at `ftp://ftp.ifcss.org/pub/software/fonts/`.

If you want to use `ccfs24.hbf`, for example, which is a sort of simplified Sung-imitated style font, you must obtain these three following files: `ccfs24.hbf`, `cclib.n24` and `ccsym.24`. In the ifcss fonts directory, `00index.txt` lists all HBF fonts' filenames. Establish a specific directory to store HBF fonts, just like `/usr/local/lib/chinese/HBF/`, for instance. Put all HBF description files and fonts files you've got together under this directory, then export the complete directory pathname for HBF out of environmental variables.

```
# export HBFPATH="/usr/local/lib/chinese/HBF/"
```

TOC

This file `cnprint.cmd` contains some default values for `cnprint`, that you have to fix them up to point t HBF fonts you have installed, and then export it to `$HBFPATH`.

```
# cp cnprint.cmd $HBFPATH
```

Now, you can use ```cnprint -w FILENAME`'' to change Chinese text files into PostScript files well. For more information in detail, take a look at `man cnprint` yourself.

Code Transformation

The `cnprint 2.80` has been added a great deal of *splendid abilities* such as transformations amongst various code formats, for example, BIG5 <=> GB <=> HZ. For BIG5 <=> GB, you need another tabulation for installing, though, which can be found from this following site:

ftp://ftp.ifcss.org/pub/software/unix/convert/hc.tab

Put it under the fonts directory of HBF, and modify `cnprint.cmd`, adding this

```
DEFAULT_GBB5_TABLEFILE: /usr/local/lib/chinese/HBF/hc.tab
```

Not only, for BIG5 into GB codes, can the transformation of `cnprint 2.80` against **words to words** be done, but also the work against **phases to phases**. However, `cnprint` itself didn't offer the dictionary for transformations, that means that you have to create a dictionary yourself if you need one. Please refer to `man cnprint` for more details.

6.2 ps2cps

The purpose of this small program is to make PostScript files, which could not pork Chinese out to output devices, available for Chinese output. For example, when Netscape is printing files, it will transport files into PostScript formats first; however, the outputs of PostScript won't load Chinese fonts so that the original parts of Chinese words will become disturbed codes to output. This program can read PostScript in, and replace the disturbed codes with corresponding words; afterwards pork the results out to the standard output devices, which the output data can send to the printer that can print PostScript documents.

Pick ps2cps up from this site:

ftp://linux.cis.nctu.edu.tw/packages/chinese/misc/ps2cps-0.1.tgz

Untar and unzip this file, and modify Makefile according to your demands:

```
BINPATH    : Installing pathname for binary ps2cps
PS2CPSPATH : Resources files' pathname for PS2CPS
PS2CPSRC   : Filename of resources files for PS2CPS
```

Next step is `make all install`.

You have to install HBF Chinese fonts first according to the previous section, and then modify your `ps2cpsrc` file:

```
HBF_PATH:      Define the directory pathname of HBF fonts
HBF_NAME:      Define filenames of HBF fonts(NOT including pathname)
CH_WORD_SHIFT: Define shift of Chinese fonts
```

The last item is used for adjustment of Chinese fonts' locations. Some Chinese fonts and the original English may probably not locate on the same horizontal level line, so you can set this variable, of which numeric val ranging from -1.0 to +1.0, activated to adjust it up or down.

So, you can use that examples made from the founder to do some tests:

```
# ps2cps thhsieh.ps > c-thhsieh.ps
```

Use `ghostview` to see whether you can see Chinese words or not.

However, this program is still under beta, there are so many problems waitting for resolving. If you have any questions or recommendations, you can send messages to the author directly: Tung-Han Hsieh < thhsieh@twclx.phys.ntu.edu.tw>.

6.3 bg2ps

This is another program which can transform BIG5-based Chinese files into PostScript available for printing(the same as cnprint). But because it uses the TrueType fonts, the output is more beautiful than others. In addition, it has a script which can transform PostScript out of Netscape into Chinese. The author is Chen-Shan Chin < cschin@u.washington.edu>.

Get this software from this web site:

http://weber.u.washington.edu/~cschin/bg2ps/

Installing bg2ps

Decompress and compile it:

```
# mkdir bg2ps
# cd bg2ps
# tar xzvf bg5ps.tgz
# gcc -O2 ttf2psm.c -o ttf2psm
```

Install TrueType fonts next. You can install it under the same directory of bg2ps or can assign another directory for use. Refer to Installing TTF fonts for more explanations. Then create a `.bg5ps.conf` under the directory, and you can pick directly up the sample file coming with this program to modify. The most important is to assign the directory where you installed the TrueType fonts to `chineseFontPath`, and rename the content of `fontName` to the fonts name you used.

Test the sample coming with it, and view the output as `ghostview` or `gv`.

```
# ./bg5ps -if test.big5 -of test.ps
# ghostview test.ps
```

Installing nps2cps

The nps2cps has no extra configuration files. You need to modify yourself `chineseFontPath` and `fontName` in `nps2cps` script.

Test nps2cps:

```
# nps2cps < netscape.ps > test.ps
# ghostview test.ps
```

6.4 gb2ps

The gb2ps program is another tool for printing GB and HZ codes.

How to get gb2ps?

Package:

```
gb2ps.2.02.tar.gz
```

ftp://ftp.ifcss.org/pub/software/unix/print

Fonts:

```
csong24.ccf     ckai24.ccf
cfang24.ccf     chei24.ccf
```

ftp://ftp.ifcss.org/pub/software/fonts/gb/misc/

Put the fonts under certain self-established directory, just like `/usr/local/lib/chinese/CFONT`.

Installing gb2ps

Change the settings in `Makefile` before compiling gb2ps.

```
CFONT=/usr/local/lib/chinese/CFONT/
COVERPAGE=/usr/local/lib/chinese/lib/cover.ps
```

Then type these commands and hit return key,

```
# make
# cp gb2ps /usr/local/bin
```

---

7. Chinese Printing Softwares

7.1 LaTeX + CJK

TeX/LaTeX is a set of printing softwares, of which excellent and elegant output quality have been admiring and adopting out of the academic circles for several years. The CJK is a LaTeX2e macro package, which can let you use CJK (Chinese/Japanese /Korean) literal codes in TeX documents.

You need install TeX/LaTeX first on your Linux system; many Linux distributions have been included teTeX/LaTeX already. If haven't, you could install it by yourself. Please link to teTeX HOWTO for more information.

Getting this software

Obtain CJK 4.1.3 from this site:

ftp://nctuccca.edu.tw/Chinese/ifcss/software/tex/CJK-4.1.3.src.tar.gz

Bring the demanded TTF fonts home:

ftp://nctuccca.edu.tw/Chinese/ifcss/software/fonts/big5/ms-win/

Installation

You must have an clear ideal about the teTeX/LaTeX installing directory, `$TEXMF`, for example, `/usr/lib/texmf`. And I assume that you will use the `ntu_kai.ttf` font; if doesn't, make a change to match it.

- Put the downloaded fonts, after decompression, into the `$TEXMF/fonts/truetype/chinese`.
- Unzip and untar CJK-4.1.3.src.tar.gz, and mv the sub-directory `texinput` to `$TEXMF/tex/latex`, and rename it to `CJK`, and create the fonts directory.

```
# cd 4_1.3/; mv ./texinput $TEXMF/tex/latex/CJK
# mkdir -p $TEXMF/fonts/tfm/chinese/ntukai
# mkdir $TEXMF/ttf2pk
# mkdir $TEXMF/hbf2gf
```

- To come immediately, it will probably be the most difficult step. You need to invoke `patch` command to append `*.diff` under `4_1.3/doc/teTeX` upon these following files:

```
/usr/bin/MakeTeX*
$TEXMF/web2c/texmf.cnf
$TEXMF/fontname/special.map
```

  For instance:

```
# cd /usr/bin
# patch -s < 4_1.3/doc/teTeX/MakeTeXPK.diff
```

  According to the enormous TeX versions, your action with `patch` may fail perhaps. If so, you need to do it manually by yourself to patch these parts of failure(the portions of failure will record in `.rej`). If you don't understand the theory of `patch` well, you had better get someone experienced to help you.

- Compiling and installing `bg5conv`

```
# cd 4_1.3/utils/Bg5conv; gcc -o bg5conv bg5conv.c
# chmod 755 bg5latex
# cp bg5conv bg5latex /usr/local/bin/
# gzip bg5conv.1; cp bg5conv.1.gz /usr/local/man/man1/
```

  Because there are some codes in BIG5 TeX documents may contain `{`, `}`, & and so forth, which are the specific tokens to TeX , the `bg5conv` can change this codes into certain format that can let TeX manipulate.

- Compile and install `ttf2pk`, which can change TrueType fonts into TeX's pk fonts.

```
# cd 4_1.3/utils/ttf2pk/src
# make all OS=unix
# cp ttf2pk /usr/local/bin/
# cd ..
# cp config/ttf2pk.cfg  $TEXMF/ttf2pk/
# gzip ttf2pk.1; cp ttf2pk.1.gz /usr/local/man/man1/
# cp c00kai.fd $TEXMF/tex/latex/CJK/Bg5/
# cp MakeTTFPK /usr/local/bin
```

- If you want to print documents vertically, you have to edit a `c00kair.fd` file under `$TEXMF/tex/latex/CJK/Bg5/`:

```
\def\fileversion{4.1.0}
\def\filedate{1996/11/20}
\ProvidesFile{c00kair.fd}[\filedate\space\fileversion]

% traditional Chinese characters in Big 5 encoding scheme.

% font shape: kai
% ntu_kai.ttf is Kai3 Shu1 (&quotmodel book")

\DeclareFontFamily{C00}{kair}{}

\DeclareFontShape{C00}{kair}{m}{n}{<-> CJK * ntukar}{}
\DeclareFontShape{C00}{kair}{bx}{n}{<-> CJKb * ntukar}{\CJKbold}

\endinput
```

- Execute `texconfig`:

```
# texconfig rehash
# texconfig hyphen
```

Tests

- Horizontal printing tests:

```
# cd 4_1.3/examples
# bg5latex Big5.tex          (See if there is a Big5.dvi appeared ?)
# xdvi Big5.dvi              (Can you see Chinese? Of course, in X mode)
# dvips Big5.dvi -o Big5.ps  (Change to PostScript format)
# ghostview Big5.ps          (View it as ghostview)
# lpr Big5.ps                (Send it to printers, see if there are Chinese printed.)
```

- Vertical printing tests: You can use the `Big5vert.tex` file under `examples` to repeat those previous procedures.

Adding New Fonts

For example, if you want to change to the Ming style font `ntu_mm.ttf`, then

- put your TrueType fonts under `$TEXMF/fonts/truetype/chinese`.
- And then add the two lines in `$TEXMF/ttf2pk/ttf2pk.cfg`:

```
ntumm:  -e Big5 $TEXMF/fonts/truetype/chinese/ntu_mm.ttf
ntummr: -r 1 -e Big5 $TEXMF/fonts/truetype/chinese/ntu_mm.ttf
```

- Create the `c00ming.fd` file:

```
# cd 4_1.3/utils/ttf2pk
# cp c00ming.fd $TEXMF/tex/latex/CJK/Bg5/
```

- Execute the `texconfig` again.

- Change kai to ming in file `4_1.3/examples/Big5.tex`, and then repeat this steps, `bg5latex`, `xdvi`, `dvips` and so on see if the results correct or not.
- If you want to print files vertically, redo this section's installation and create the `c00mingr.fd` file listed as follows, and then change kair to mingr in `Big5vert.tex` file, and repeat tests processes.

```
\def\fileversion{4.1.0}
\def\filedate{1996/11/20}
\ProvidesFile{c00kair.fd}[\filedate\space\fileversion]

\DeclareFontFamily{C00}{mingr}{}

\DeclareFontShape{C00}{mingr}{m}{n}{<-> CJK * ntummr}{}
\DeclareFontShape{C00}{mingr}{bx}{n}{<-> CJKb * ntummr}{\CJKbold}

\endinput
```

Creating CJK Documents

The dominant difference between Chinese CJK TeX documents and general LaTeX ones is that:

- There should exist the \usepackage{CJK} command in the preamble (\documentclass to \begin{document}¡^section, that means you must have CJK.sty loaded.
- Chinese words must be under the circumstances of CJK or CJK*.
- If you want to change fonts, you can use \CJKfamily command, for example, the command \CJKfamily{fs} will change the following fonts as Song-imitated fonts(Certainly, the `fs` fonts' name must be defined in `c00fs.fd`).

This is a demonstration for CJK document:

```
\documentclass[12pt]{article}
\usepackage{CJK}
\begin{document}
  \begin{CJK*}{Bg5}{kai}

    \section{first section}
    \section{second section}
    Paragraphs, sections, pictures, tables, references and so forth...
    ...

  \end{CJK*}
\end{document}
```

7.2 ChiTeX

Developed by professor Chen Hung-Yih < yih@math.ncu.edu.tw>. Operations with the ChiTeX are easy enough just like the English TeX, apart from some special instructions.

Obatining it

From here, you can find its vestige:

ftp://dongpo.math.ncu.edu.tw/tex-archive/local/chitex/chitex/Linux/

Installing

You should know what's the version of TeX installed on your Linux.. On older system, the installed one usually is NTeX while on newer one, it will be always teTeX . How to tell which TeX you've installed, teTeX or NTeX, on your Linux System. Just run the TeX. If the monitor appears

```
This is TeX, Version 3.14159
```

, then it is teTeX. Otherwise,

```
This is TeX, Version 3.1415N
```

It is NTeX. Download the exact ChiTeX corresponding to your version of TeX.

It is simple to install ChiTeX. Just put `chitex60.tgz`(teTeX) or `chitexN.tgz`(NTeX) and `fonts1.tgz`, `fonts2.tgz` under `/usr/local`. And decompress `chitex60.tgz` to execute setup program `csetup`.

```
# tar zxvf chitex60.tgz
# cd chitex60
# ./csetup
```

If you are using Red Hat 5.0, you may install the rpm package I made:

ftp://ftp.phys.ntu.edu.tw/pub/os/linux/rpm/chitex-6.0.7-1.i386.rpm

Installing fonts

You can install Chinese TrueType fonts for ChiTeX.

- Put TrueType fonts under `$TEXMF/fonts/chinese/ttf`.
- Modify `$TEXMF/tex/chinese/chitex.fdf`, and add a new line:

  ```
  \choosechfont{fontname}{filename}
  ```

  In it, the `filename` is the fonts' names erased the `.ttf` part, and `\fontname` is the macro that you can use for this font in your document. For example, if you want to use a font named `avntmv.ttf`, put a line `\choosechfont{ming}{avntmv}`, and use the macro `\ming` to use the font `avntmv.ttf`.
- In `chitex.fdf`, there were several fonts defined; thus, if you want to install these fonts, you don't have to change `chitex.fdf` any more.

  ```
  \kai    ---->  ntukai.ttf
  \li     ---->  ntuli.ttf
  \mr     ---->  ntumr.ttf
  \fs     ---->  ntufs.ttf
  ```

Tests

Now, you can take the samples enclosed with ChiTeX to test:

```
# chilatex math2.tex        (Compiling)
# xdvi math2                (pre-viewing)
# dvips math2               (Changing to PostScript file)
# ghostview math2.ps        (Using ghostview to view the output)
```

You can also write a small document to test the new installed fonts:

```
\documentclass[12pt]{article}
\begin{document}
\ming
   This is a test(You should type these words in Chinese).
\end{document}
```

For more information, please refer to the homepage of professor Chen.

http://www.math.ncu.edu.tw/yih/intro.htm

7.3 Dtop

The Dtop is a Chinese printing software for the UNIX platform, developed by the Behavior Design Corporation. In early November, 1997, it released the version of v1.4, which was composite of five various platforms, simultaneously. The Linux beta version provided the users with a trial of free and endless expiration, which has created another nice Chinese environment for Linux fans. However, the formal version is late for publishing until now due to the tiny scope of business market. It seems that this had been ceased to develop.

How to Get it?

The beta version of Dtop for Linux can download from each school's ftp site:

- ftp://nctuccca.edu.tw/Chinese/ifcss/software/linux/X11R6/dtop1.4/pub/
- ftp://ftp.ntu.edu.tw/Chinese/ifcss/software/linux/X11R6/dtop1.4/pub/
- ftp://ftp.ccu.edu.tw/pub3/chinese/linux/X11R6/dtop1.4/pub/
- ftp://ftp.ncu.edu.tw/OS/linux/X/ifcss/X11R6/dtop1.4/pub/
- ftp://ftp.ifcss.org/pub/software/linux/X11R6/dtop1.4/pub/

There are three sub-directories saying respectively,

dtop.linux

Storage for binary files and data files of Dtop, which can divide into three major files. After decompressing all files, the disk space demanded is about 40MB.

dtop.readme

Storage for related documents of Dtop, which are stored as the file format defined by Dtop. Users can read this papers through this beta version directly.

dtop.manual

Storage of references for X version of Dtop, which are stored as the file format defined by Dtop. The X version can access through Linux, IBM AIX, HP-UX, Sun 4.1.x, and Solaris directly.

Installing

The hardware required is something like this: CPU 486 DX-33 or later, RAM 16MB or larger, 50MB disk space or so. If there is no enough memory on your system, you will get nuts someday.

Decompress all the files under `dtop.linux` to a self-selected directory, for example, `/usr/local/dtop`.

Before implementing it, you must first set environment variable `$DTOPHOME`, which is the directory that preparing to install Dtop. The `$DTOPTMP` is a temporary directory for Dtop, which is usually `/tmp`. The `XAPPLRESDIR` is the directory for resources files of Dtop, which is set at `$DTOPHOME/user` .

```
# export DTOPHOME=/usr/local/dtop
# export DTOPTMP=/tmp
# export XAPPLRESDIR=DTOPHOME/user
```

You need to set the Chinese fonts directory used by Dtop.

```
# xset fp+ $DTOPHOME/pcf.chn
```

Finally, you have to configure the Keysym matching table caused by the matching between Keysym and Keycode under X different from workstations. The Keysyms of `BackSpace` and `Delete`, for example, are all matching to the same Keycode under X, regarding the behavior of `BackSpace` as the `Delete`'s in reality . To solve this problem, the Dtop creates a Keysym file, locating under `$DTOPHOME/user` of which filename is `dtop_keysym.linux`. After starting X, invoke the command,

```
# xmodmap dtop_keysym.linux
```

Another solution is that invoke commands directly on shell:

```
# xmodmap -e "keycode 22 = BackSpace"
```

```
# xmodmap -e "keycode 107 = Delete"
```

When everything is ready, you can execute Dtop.

```
# $DTOPHOME/bin/dtop14
```

You may read all the on-line references enclosed in Dtop for more detailed introductions.

7.4 ChinesePower

The ChinesePower is a Far-East Editor running under X, which is easy and classified as WYSIWYG. It supports BIG5, GB, Japanese and Korean mixed inputs, and the printing and displaying PostScript format. It can generate GIF graphics of seven colors for documents, which is adequate for Chinese homepages. ChinesePower uses HBF or TTF fonts.

You can get it from,

ftp://ftp.ifcss.org/pub/software/x-win/editor/chpower-2.0.tar.gz

To compile Chinese Power, you will need HBF fonts and Motif libraries. At first, modify `Makefile` to fit your system, then

```
# make
```

After making step, it will generate binary executable files. Then export environment variables:

```
# export HBFPATH=/usr/local/lib/chinese/HBF/
# export TTFPATH=/usr/local/lib/chinese/TTF/
# export HZINPUTDIR=/usr/local/lib/chinese/dict/
# export CHPOWERPATH=path_of_chinesepower
```

7.5 EasyFlow

This is developed by wycc < wycc@iis.sinica.edu.tw> , which can make simple printing for Chinese documents. Relative information is locating at

http://formosa.iis.sinica.edu.tw/~wycc/

---

8. Useful Chinese Softwares

8.1 cjoe - Joe's Own Chinese Editor

JOE is a free and professional ASCII codes' editor operated on UNIX platform, which is just like other text editors on IBM PC. To use Chinese in JOE, you have to modify `.joerc` under your home directory. Refer to the homepage for details:

http://www.neto.net/~bv1al/linux/cjoe.html

CJOE is a Chinese version of JOE. Get it from here below:

ftp://ftp.ifcss.org/pub/software/unix/editor/cjoe-2.8.tgz

It is very simple to compile and install. Unless you want to change locations of binary or man page files, or just follows the steps below:

```
# make
# make install
```

8.2 celvis

Celvis is a vi/ex-like editor on UNIX, which is almost supporting all instructions that vi/ex has. You can edit articles including Chinese and English by using Celvis. Simultaneously, it also supports GB2312-80 and BIG5 codes.

You can get it from here below,

ftp://ftp.ifcss.org/pub/software/unix/editor/celvis-1.3.tar.gz

Decompress it,

```
# tar zxvf celvis-1.3.tar.gz
# cd celvis
```

You need to change `tmp.c`, erasing 93-95 lines.

```
#if OS9
                    if we don't have write permission...
#endif
```

Then you can continue to compile it.

```
# cp Makefile.s5 Makefile
# make install
```

Because its `Makefile` is not for Linux particularly, you may see many warning message while compiling and linking; however, don't mention it, just keep it away. After it is done, the `celvis` will be installed under

/usr/local/bin.

8.3 cvim

cvim is a Chinese patch program out of vim-4.2, including some features like vi but has no track in celvis-1.3, such as line number, circling lines and large files' editions. You can take `vim-4.2-Chinese-patch` and `vim-4.2.tar.gz:` from here,

ftp://ftp.csie.nctu.edu.tw/pub/Unix/Chinese/cvim

Untar and unzip following the steps below:

```
# tar -xvzf vim-4.2.tar.gz
# cd vim-4.2/src
# patch < ../../vim-4.2-Chinese-patch
```

Change `vim-4.2/src/feature.h` to fit your requires. It is simple to compile and install, that is,

```
# make
# make install
```

8.4 he

he was a famous editor on DOS, and is a diversion of Linux version. But this is a shareware program, limited with a hundred lines' edition.

Obtain it from here,

ftp://linux.cis.nctu.edu.tw/packages/he_linux/he_linux.tar.Z

Login as root,

```
# cd /
# decompress he_linux.tar.Z
# tar xvf he_linux.tar
```

Refer to `/usr/lib/he/notes2.2`, `/usr/lib/he/chap15` and `/usr/lib/he/appendix` for more simple illustrations.

8.5 hztty

Hztty can make transformations among various Chinese codes. Decompress `hztty-2.0.tar.gz` first.

```
# tar -xvzf hztty-2.0.tar.gz
# cd hztty-2.0
# make linux
```

After compilation, move the binary file `hztty` to the directory `bin`, and move man pages to the directory `man`.

```
# mv hztty /usr/local/bin
```

```
# chmod 555 /usr/local/bin/hztty
# cp hztty.1 /usr/local/man/man1
# chmod 444 /usr/local/man/man1/hztty.1
```

Please refer to the on-line manual of hztty to use it.

8.6 ktty

This is another tool, like hztty, using for reading Chinese on kterm or pxvt. Get it at this site:

ftp://ftp.ifcss.org/pub/software/unix/convert/ktty1.3.tar.gz

However, it cannot be compiled on Linux, that you need use the ``tty.c" in `hztty-2.0.tar.gz` and add the two lines to it.

```
linux:
      ${MAKE} ${MFLAGS} DEFINES='-DPOSIX -DSYSV -DUSE_SYSV_UTMP' ${PROG}
```

Then, you can make it work.

```
# make linux
```

8.7 Cemacs and CChelp For Emacs

Cemacs, using the GNU Emacs to show and edit Chinese files, has to run Emacs under Chinese virtual terminals while CCHELP is a system of providing Chinese assistant messages. After installed CCHELP, you can slip mouse to any Chinese word and click it without loosing, then there coming out with the messages about that word , including its pronunciation, English explanation and so forth. It supports both GB and BIG5 codes.

You can get them from this site:

ftp://ftp.math.psu.edu/pub/simpson/chinese/

Install cemacs and cchelp in accordance with the README file.

8.8 Mule

The MULE is an abbreviation of MULtilingual Enhancement to GNU Emacs. In simplicity, it adds some materials to GNU Emacs to make it dealing with multi-languages(encoding systems). It encodes the encoding system of multi bytes again in its inner mechanism; hence, a piece of paper can simultaneously use Chinese(BIG5 and GB), Japanese, Korean, English, Thai, and so on.

Obtaining Mule-2.3 and patch for Linux

You can get `mule-2.3.tar.gz` and patch `mule-2.3+lx.diff.gz` for Linux:

ftp://sunsite.unc.edu/pub/Linux/distributions/je/sources/mule/

If your system is Red Hat Linux, you can take `mule-2.3-1.i386.rpm`, `mule-common-2.3-1.i386.rpm` and `mule-elispsrc-2.3-1.i386.rpm`:

ftp://sunsite.unc.edu/pub/Linux/distributions/redhat/contrib/contrib-2.0.x/RPMS/

TOC

Compiling and Installation

Decompress packed files and add the patch in.

```
# tar -xvzf mule-2.3.tar.gz
# patch < mule-2.3+lx.diff
# cd mule-2.3/
# ./configure "i386-*-linux" --with-x11 --with-x-toolkit --with-gcc
```

If you want to rectify the out-looking of mule to fit your taste, read `INSTALL` file, please. And run ``./configure --help''. Then, correct ``src/Makefile'', changing ``-lcurses'' to ``-lncurses''. Then,

```
# make
# make install
```

Default installing directory is `/usr/local`.

Using Chinese in Mule-2.3

If you have already installed fonts, you can use mule to enter and show Chinese. Most fonts are 16 or 24 points, so you can:

```
# mule -fn 8x16 &
```

or

```
# mule -fn 12x24 &
```

According to `M-x load-library RETURN chinese RETURN`. ``Ctrl-]'' to shift inputs.

8.9 hc

This is a program of conversion for BIG5 and GB codes. Get it at this site:

ftp://ftp.ifcss.org/pub/software/unix/convert/hc-30.tar.gz

Decompress and install it:

```
# tar zxvf hc-30.tar.gz
# cd hc3
# make
# mv hc /usr/local/bin
# mv hc.tab /usr/local/lib/chinese
# mv hc.1 /usr/local/man/man1
```

For converting GB into BIG5 , using

```
hc -m g2b -t /usr/local/lib/chinese/hc.tab < INPUT_FILE > OUTPUT_FILE
```

For exchanging BIG5 into GB, using

```
hc -m b2g -t /usr/local/lib/chinese/hc.tab < INPUT_FILE > OUTPUT_FIL
```

You can write a shell script to simplify that instructions.

8.10 ctin

The ctin a news reader of all complete Chinese messages. Get it from here,

ftp://linux.cis.nctu.edu.tw/packages/chinese/ctin/ctin-970625.src.bin.tgz

Set environment variables `$NNTPSERVER` first to export to the news server that you want to link before executing ctin.

```
# export NNTPSERVER=netnews.ntu.edu.tw
```

Then run `tin`.

8.11 Some Other Toys

You can find several small tools capable of showing Chinese, such as `cuptime`, `cw`, `cless`, `cwrite`, `cytalk`, and so forth.

ftp://ftp.vit.edu.tw/unix/linux/chinese/

---

9. Related Chinese Problems with Others

9.1 Netscape Navigator

Netscape 2.0 or later support both BIG5 and GB codes, which have already discussed on several homepages. To display Chinese in Netscape, there are three steps:

1. Install Chinese fonts of X. Refer to the section Installing Chinese Fonts .
2. Run `netscape`, and configure `Document Encoding` as `Traditional Chinese (Big5)` or `Simplified Chinese`.
   - In Netscape 2.x or 3.x, this will configure at `Options/Document Encoding/....`
   - In Netscape Communicator 4.x, configure it at `View/Encoding/....`
3. Select the Chinese fonts you want.
   - In Netscape 2.x, 3.x, configure it at `Options/General Preferences/Fonts`.
   - In Netscape Communicator 4.x, configure it at `Edit/Preferences.../Appearance/Fonts`.

Maybe you will feel so strange that why you can only use `Fixed(HKU) Size 16.0`, which is causing from the settings in `/usr/X11R6/lib/X11/app-defaults/Netscape`. Assuming that you have the following BIG5 fonts:

```
# xlsfonts | grep big5
-hku-fixed-medium-r-normal--0-0-72-72-c-0-big5.hku-0
-hku-fixed-medium-r-normal--16-160-72-72-c-160-big5.hku-0
-kc-kai-medium-r-normal-fantizi-0-0-75-75-c-0-big5.eten.3.10-1
-kc-kai-medium-r-normal-fantizi-20-200-75-75-c-200-big5.eten.3.10-1
-kc-kai-medium-r-normal-fantizi-24-240-75-75-c-240-big5.eten.3.10-1
-kc-ming-medium-r-normal-fantizi-0-0-75-75-c-0-big5.eten.3.10-1
-kc-ming-medium-r-normal-fantizi-15-150-75-75-c-160-big5.eten.3.10-1
-kc-ming-medium-r-normal-fantizi-24-240-75-75-c-240-big5.eten.3.10-1
```

Add one line to `/usr/X11R6/lib/X11/app-defaults/Netscape`

```
*documentFonts.charset*big5.eten.3.10-1:              x-x-big5
```

Then, Netscape can use these fonts well.

Another solution is to name aliases for your fonts's names: Adding aliases in the `fonts.alias` file(create it yourself if there is no such a file) under your fonts directory.

```
-big5-taipeik-medium-r-normal--16-160-72-72-c-160-big5.hku-1 taipei16
-big5-taipeik-medium-r-normal--20-200-72-72-c-200-big5.hku-1 taipeik20
-big5-taipeik-medium-r-normal--24-240-72-72-c-240-big5.hku-1 taipeik24
```

The fonts filenames, like `taipeik20`, are that you need to give aliases to them. After modification, do

```
# xset fp rehash
```

to reload fonts's names to make them available.

You may find that the sizes of Chinese and English are out of proportion in homepages of Chinese-English languages(such as too small English words), which is the reason that what you've selected for English fonts and Chinese fonts is unbalanced at the magnitude. Go back to the configuration for fonts, and select Western(iso-8859-1). Then, in

your opinion, change the size of English fonts to make it perfect in proportions.

9.2 sendmail

Because defaults of `sendmail 8.8.x` will encode all 8-bit mails first before sending them out, which may or may not reduce influences to those receiver. (Refer to Next Sub-Section for solutions). Most hosts can directly send 8-bit mails now, so you had better re-compile `sendmail` and close the encoding ability.

- Get the newest version from this site(or mirror site):

  ftp://ftp.sendmail.org/ucb/src/sendmail/

- Decompress it and modify `Makefile`:

```
# tar zxvf sendmail-8.8.8.tar.gz
# cd src/Makefiles
# chmod u+w Makefile.Linux
# vi Makefile.Linux
```

  Add `-DMIME8TO7=0` after `CFLAGS+=`.

- Compile and install `sendmail`:

```
# cd ..
# ./makesendmail all install
```

- Install man pages to the exact directory manually(It's ok if not installing).
- Erase the old `sendmail` process:

```
# kill -9 `head -1 /var/run/sendmail.pid`
```

- Restart `sendmail`:

```
# /usr/sbin/sendmail -bd -q1h
```

9.3 procmail

`procmail` is an electronic mail filter, which can manipulate mails first before storing them into your mailbox, for example, classification for mails and so on. Though, I want to teach you how to make use of `procmail` to uncode the encoded ones into pure text files and how to convert the various encoded mails into common encoding you used here.

Create `.procmailrc` file under your home directory, listed as follows:

```
# To avoid accidents, store mails before using procmail.
# You can filter mails to ensure which mail doesn't need to duplicate
# by, for example, MAILER-DAEMON.
:0 c
* !^From.*MAILER-DAEMON
mail/procmail-backup

# Uncode mails encoded by quoted-printable or base64.
:0
* ^Content-Type: *text/plain
{
    :0 fbw
```

```
    * ^Content-Transfer-Encoding: *quoted-printable
    | mimencode -u -q

        :0 Afhw
        | formail -I "Content-Transfer-Encoding: 8bit" \
                  -I "X-Mimed-Autoconverted: quoted-printable to 8bit by procmail"

    :0 fbw
    * ^Content-Transfer-Encoding: *base64
    | mimencode -u -b

        :0 Afhw
        | formail -I "Content-Transfer-Encoding: 8bit" \
                  -I "X-Mimed-Autoconverted: base64 to 8bit by procmail"
}

# Here will judge whether a mail is GB codes or not; if it is, convert it
# into BIG5 codes. If you want to convert BIG5 codes into GB codes, you
# need to exchange big5 with gb2312 and replace 'hc -m b2g' with 'hc -m g2b'.

:0
* ^Content-Type:.*text/plain;.*charset=gb2312
{
    :0 fw
    | hc -m b2g -t /usr/local/lib/chinese/hc.tab

        :0 Afhw
        | formail -I "Content-Type: text/plain; charset=big5" \
                  -I "X-Charset-Autoconverted: gb2312 to big5 by procmail"
}

# Store mails into mailbox.
:0:
${ORGMAIL}
```

In this case, integrating with `mimencode`, `formail` to decode mails, and use hc to convert GB codes mails into BIG5 codes mails. Thus, you have to install these tools in order to use procmail.

Eventually, rectify your `.forward` file.

```
"|IFS=' ' && exec /usr/bin/procmail -f- ~/.procmailrc ||exit 75 #login"
```

Notes: change the `login` as your login name.

9.4 telnet

If you can access Chinese on your own machine, but cannot access Chinese while telnet to other machines, then the problems may cause from `telnet` itself. When telnet is being under connection, you can press `Ctrl-]` to jump out temporarily, and enter **set binary** at the prompt of `telnet>` to assure the fully transmission of bits. You can also use `telnet -8` to make connections (but it seems won't work for some sites...?). Another more simple method is to use directly a 8-bit `telnet` program. Download the binary file from here:

ftp://linux.cis.nctu.edu.tw/packages/chinese/misc/

TOC

9.5 less

`less` itself can display Chinese directly; just set the following environment variable can Chinese display:

```
# export LESSCHARSET=latin1
```

9.6 pine

Pine of version 3.95 later can support reading and entering Chinese mails. As you execute pine, select `SETUP/Config` in `feature list`, choose this(hit `X`):

```
[X]  enable-8bit-esmtp-negotiation
[X]  enable-8bit-nntp-posting
```

Then, hit `C` on `character-set`, and change its value to big5 or gb2312. Finally, hit `E` to escape and store this change.

You can also install the cpine capable of showing Chinese messages:

ftp://ftp.vit.edu.tw/unix/linux/chinese/cpine-3.91.bin.tgz

9.7 samba

- At first, take the newest version of samba-1.9.17p5 and big5 patch home.

  ftp://linux.cis.nctu.edu.tw/packages/network/samba/samba-1.9.17p5.tar.gz
  ftp://linux.cis.nctu.edu.tw/packages/chinese/samba-big5-patch/samba-1.9.17p2-BIG5-patch-0.gz
- Decompress samba-1.9.17p5.tar.gz, and add the patch upon:

  ```
  # tar zxvf samba-1.9.17p5.tar.gz
  # cd samba-1.9.17p5/source
  # zcat samba-1.9.17p2-BIG5-patch-0.gz | patch -s
  ```
- According to the normal procedures, you can compile and install it. Refer to the Samba HomePage.

9.8 lynx

Lynx v2.7 can view Chinese homepages directly. Execute lynx, and press `O)ption` / `C)haracter set`, and then choose `Taipei(Big5)` by arrow keys, and finally press '>' to store this set.

9.9 minicom

Use the switch `-l` to start `minicom`, then you can see Chinese in minicom.

9.10 fingerd

If you type Chinese in your `.plan` but can't be saw by others, this is probably caused by the inability of `fingerd` to send out 8-bit Chinese words. Get `cfingerd` from this site:

ftp://nctuccca.edu.tw/OS/Linux/packages/network/finger/cfingerd-1.3.2.tar.gz

Decompress it and execute `./Configure`, then implement `make all`.

Then, modify `/etc/inetd.conf`, changing the original `finger` to:

```
finger  stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.cfingerd
```

Let `inetd` read the configuration file again:

```
# kill -HUP `cat /var/run/inetd.pid`
```

9.11 tintin++

The TinTin++ is not a Chinese software but just a useful tool when playing Chinese mud.You can download it from this place:

ftp://ftp.princeton.edu/pub/tintin++

Here are some Chinese mud sites:

```
FengYun                 fengyun.com       5555
Xi You Ji               129.105.79.24     6905
Xia ke Xing             207.76.64.2       5555
```

10. Chinese Linux

If you have known some projects else about Chinese Linux, please tell me about it¡D

10.1 Chinese Linux Documentation's Projects

CLDP is now abandoned itself to translating Linux HOWTO documents, locating at http://www.linux.org.tw/CLDP/. CLDP has a mailing list ldp-trans@linux.org.tw. Its purposes are:

- Combine everyone's power to implement Linux HOWTO translations available.
- Discuss SGML Tools of dealing with Chinese.
- Share and maintain experience of translation for Linux HOWTO.
- To propagate the translation projects of Linux HOWTO.
- Others concerned with Chinese Linux.

If you want to join us, please send a mail contented(*not titled*) with `subscribe` to ldp-trans-request@linux.org.tw For more advanced explanations of this mailing list, please send a mail contented with `info` to the upper address. If you want to know the usage of mailing list, send a mail contented with `help` to the same address. If there is any suggestion or suspicion to the mailing list, please write to ldp-trans-approval@linux.org.tw¡D

If you are willing to contribute your own power, please join with this mailing list.

10.2 Chinese Linux Project

This is established for a Chinese environment of Linux, originated by cdchen. This is still under the period of setting, waiting for everyone's all efforts.

Homepage of this project is at http://chinese.linux.org.tw/ which is responsible for formal announcements; and it provides discussions of problems, suggestions and criticisms, delivering ourselves of achievements and so forth.

10.3 Chinese websites of Linux

Here are some websites for Chinese Linux information. If you have others not listed here, tell me about that, please.

BIG5

- http://www.linux.org.tw/
- http://chinese.linux.org.tw/

- http://linux.cis.nctu.edu.tw/
- http://linux.ntcic.edu.tw/
- http://freebsd.ee.ntu.edu.tw/bbs/6/index.html
- http://cc.shu.edu.tw/~rick/wwwguide/c_linux_hopenet.html
- http://henry.cis.nctu.edu.tw/~linuxwww/
- http://marr.dorm9.nccu.edu.tw/~marr/Comp/PC-Unix/index.html
- http://www.phys.ntu.edu.tw/~cwhuang/pub/os/linux/

GB

- http://csun01.ihep.ac.cn/
- http://www.clinux.ml.org/

10.4 Discussed groups of Linux

If you have any problem, you can post you question here, using Chinese is ok, too. However, make sure that this question is still unknown or hasn't answered in related documents or HOWTO, or *repeat some FAQs are absolutely not popular with these places.*

- news://tw.bbs.comp.linux
- telnet://henry.cis.nctu.edu.tw

---

11. FAQs

11.1 Why Can't I Enter Chinese?

Answer: This question is quite ambiguous, though. I cannot tell which condition that you have met? If you can not enter Chinese on console's shell, refer to the section Chinese Input Problems for detailed.

Or that you cannot enter Chinese in an editor, it is possible that your editor doesn't support Chinese. Please refer to the section Useful Chinese Softwares to install Chinese-supported editors.

If this is matched at the period of telnet, then refer to the section `telnet` for more explanations.

11.2 I have got the NTU TTF fonts, but how can I decompress it?

Answer: The NTU TFF fonts is compressed with arj format and divided it into several files for the convenience of putting it into floppy disks, causing that decompression is a little trifling. If you want `ntu_kai.ttf` font, for example, you must take `ntu_kai.arj`, `ntu_kai.a01`, `ntu_kai.a02`, and `ntu_kai.a03` home. Assuming that you have an arj tool of **DOS** version, put these four files altogether under the same directory and invoke this commands to decompress:

```
C:\> arj x -va -y ntu_kai
```

If you want to decompress them on Linux, you must use `unarj` to unarj them one by one, and then use `cat` to concatenate them together:

```
# unarj e ntu_kai.arj ; mv ntu_kai.ttf ntu_kai.ttf0
# unarj e ntu_kai.a01 ; mv ntu_kai.ttf ntu_kai.ttf1
# unarj e ntu_kai.a02 ; mv ntu_kai.ttf ntu_kai.ttf2
# unarj e ntu_kai.a03 ; mv ntu_kai.ttf ntu_kai.ttf3
# cat ntu_kai.ttf* > ntu_kai.ttf
```

11.3 Netscape's title and tag cannot see Chinese?

Answer: To install CXWin will solve this problem. Refer to CXWin for detailed.

11.4 My Netscape cannot see Chinese?

Answer: If you could not see any Chinese on all homepages, then you might probably not install Chinese fonts, or haven't make Netscape settled. Refer to the section Netscape.

But if you can see Chinese on most parts of Chinese homepages, only on some websites could see disturbed codes, then this may not be your private problems. A little websites take Frontpage to generate homepages, and their language encoding is `x-x-big5` not general `big5`.

This wil make all browsers except IE cannot identify this encoding, which just a bad trick adopted by Microsoft to fulfill its ambitions to occupy the market of browsers. To solve this problem, send a mail to the administrator of that site, telling them there are other browsers except IE all over the world. If they want to manage it perpetually, they should correct that unusual encoding to normal one.

Of course, if you have better solutions, please let me know of it.

11.5 How to enter Chinese in Netscape?

Answer: Install xcin+XA. Refer to this section XA.

11.6 What's the difference between ChiTeX and CJK? Can I install them at the same time?

Answer:

- The system of Chinese reading and searching of CJK which is independent and efficnet is based on the NFSS of LaTeX.
- CJK can use Chinese, Japanese, and Korean; and ChiTeX is just for BIG5.
- CJK can only use on LaTeX while ChiTeX can use on plain TeX and LaTeX .
- Instructions of ChiTeX are simple and useful.
- ChiTeX is approaching to real Chinese LaTeX.
- Difference in functionality...

ChiTeX and CJK should be compatible while installing them at the same system, but if the `ttf2pk` has the same name for both ChiTeX and CJK, then it could cause certain problems. A solution to this is that define the `$PATH` variable in each shell script, respectively, to point to the correct locations of `ttf2pk`. (*Thanks to professor Chen Hung-Yih.*.)

---

12. Appendix - Chinese Softwares of FTP.IFCSS.ORG

Warnning¡GMost pre-compiled Linux executable files are out-of-date and cannot use normally.

```
****
UNIX
****

========
UNIX:BBS
========

software: Phoenix BBS
version : 4.0, 1995.08.31
function: o UNIX platform multi-user BBS system,
          o discussion boards, private email
          o multichannel chat, one-to-one chat
          o Internet Email, News gateway,
          o 0Announce : Gopher-like information query interface.
          o fine tuned to allow more than 256 users on-line
          o configurable menu, screen display
          o Chinese message
URL     : ftp://ftp.ifcss.org/pub/software/unix/bbs/PhoenixBBS-4.0.tar.gz
author  : Ji-Tzay Yang, Ming-Feng Chen, Tzung-Yu Wen

============
UNIX:C-UTILS
============

software: addpy
version : 1.0
function: To annotate Pinyin to rare Hanzi and a portion of
          randomly selected common Hanzi.  Both GB (simplified) and
          Big5 (traditional) versions.  Based on statistics derived
          from huge Chinese corpus and well prepared hazni-freq-pinyin
          tables.  Source code and raw data provided.
URL     : ftp://ifcss.org/pub/software/unix/c-utils/addpy.tar.gz
filename: README.addpy, b5addpy.l, gbaddpy.l, b5addpy.dat,
          gbaddpy.dat, makefile
author  : Guo Jin

software: rm4mat
version : n/a
function: remove formating codes for printing in some GB files
URL     : ftp://ftp.ifcss.org/pub/software/unix/c-utils/rm4mat.c
author  : Chenghong Wang

software: mail_hxwz
version : 1.1
```

```
function: It extracts HXWZ from your mail (Suppose you are subscribing
          it).  It is a Bourne shell program which works on most Unix
          systems.  It can rerun itself every Friday automatically.
URL     : ftp://ftp.ifcss.org/pub/software/unix/c-utils/mail_hxwz
author  : Yaoen Zhang

software: auto_get_hxwz
Version : 1.2
Function: It gets the current issue of HXWZ in GB or postscipt format.
          It also process and print these files, and do the clean up.
          It save your time and server machine time.
URL     : ftp://ftp.ifcss.org/pub/software/unix/c-utils/auto_get_hxwz
author  : Yaoen Zhang

software: Chinese less
version : 290, 1995.5.25
function: Browse BIG5 and GB text files
URL     : ftp://ftp.ifcss.org/pub/software/unix/c-utils/cless-290.tar.gz
author  : Paul W. Shew
comment : requires a chinese terminal, like cxterm.

software: cscreen
version : 3.2b
function: This is a modified screen to minic ETen in cxterm.
          After you run cscreen in cxterm, it will intercept
          every key movement and examine if the current
          screen postion has a Chinese char. If it has,
          auto-key movements will be made accordingly.
URL     : ftp://ftp.ifcss.org/pub/software/unix/c-utils/cscreen-3.2b.tar.gz
author  : Shih-Kun Huang

software: ytalk
version : 3.0.2c6
function: ytalk-3.02c provides better compatibility for DEC's
          workstations, two-byte refresh for terminal like cxterm, and
          capable of passing character code 254, internally used as erase
          function in ytalk-3.0; Compatible with ytalk-3.0, cytalk-3.0,
          cytalk-3.0.2, ytalk-3.02c0/1/2/3/4/5, that is, you can still use
          Delete, or Backspace key.  If both ends use ytalk-3.02c4, the code
          254 is transparent, you can use more Chinese words to talk to
          others.
URL     : ftp://ftp.ifcss.org/pub/software/unix/c-utils/ytalk-3.0.2c6.tar.gz
author  : Sze-Yao Ni

software: lunar
version : 2.1
function: conversion program between Solar and Chinese lunar calendars;
          calculation of birthday in "4-column" astrology as well
URL     : ftp://ftp.ifcss.org/pub/software/unix/c-utils/lunar-2.1.tar.gz
author  : Ricky Yeung and Fung Fung Lee
comment : can output GB code, bitmap, or just ASCII

software: DateStar - Chinese Calendar Producer
```

```
version : 1.1
function: Displays Chinese and western calendar in
          ASCII code, BIG-5 code (Hongkong, Taiwan),
              GuoBiao code (PRC Standard), and HZ code (Network)
          Prints on two most popular laser printers
              PostScript laser printers, and
              HP LaserJet (PCL) printers
          Generates four different format
              Yearly calendar, Monthly calendar,
              One-page Weekly calendar, and Two-page Weekly calendar
          Shows calendar with 24 Solar Terms (JieQi),
              the Heavenly Stems and Earthly Branches
              (TianGan DiZhi Eight Characters)
          Supports user defined annotations
          Applicable from year 1841 through to 2060
URL     : ftp://ftp.ifcss.org/pub/software/unix/c-utils/datestar-1.1.tar.gz
author  : Youzhen Cheng
comment : UNIX version works on SUN Workstation with SUN OS 4.1.x
          see /software/dos/c-utils for DOS version

software: pull
version : 2.9, 1996.4.7
function: 1. Extracts the original file(s) from uuencoded/compressed/split
             file(s).
             a) uudecode a file and display the name of its decoded file
             b) decompress .zz (= .gz/.Z/.zip) file
             c) uudecode + decompress
             d) uudecode + cat (concatenate) + gunzip
          2. Packs file. Reverse the above `extracting' process.
             compresses, (splits), then encodes file(s)
URL     : ftp://ftp.ifcss.org/pub/software/unix/utils/pull.c.gz
author  : Yao Li

software: utf-utils
version : 15 Oct 1993.
function: some utility programs for manipulating Unicode/ISO-10646 text
          in the FSS-UTF encoding.
URL     : ftp://ftp.ifcss.org/pub/software/unix/c-utils/utf-utils.tar.gz
author  : Ross Paterson

software: gbfmt
version : 1.0
function: GB formatting with variable line width, GB<->HZ
URL     : ftp://ftp.ifcss.org/pub/software/unix/c-utils/gbformat.tar.gz
author  : Dongxiao Yue (http://www.cs.umn.edu/~dyue/wiihist/gbfmt.html)

===========
UNIX:EDITOR
===========

software: celvis
version : 1.3
function: vi editor with Chinese handling capabilities.
```

```
          Most run on some Chinese terminal, e.g. cxterm, or IBM-PC with
          some kind of Chinese DOS.  Work exactly like Unix vi editor,
          except that it side-scrolls long lines instead of wrapping.
URL     : ftp://ftp.ifcss.org/pub/software/unix/editor/celvis.tar.gz
          ftp://ftp.cs.purdue.edu/pub/ygz
author  : Man-Chi Pong, Yongguang Zhang
comment : comments, bug-reports, modifications to: c-elvis@cs.purdue.edu

software: Chinese JOE (Joe's Own Editor)
version : 2.8c
function: Chinese BIG5 localized text screen editor.  The key sequences are
          remeniscent of WordStar and TURBO-PASCAL. Makes full use of
          termcap/terminfo, is designed to work well over slow networks and
          low baud rate modems, and has the best features of vi. Most
          useful for editing unformatted text, such as USENET news articles
          and for editing block-structured languages such as C and PASCAL.
URL     : ftp://ftp.ifcss.org/pub/software/unix/editor/joe2.8_c.tar.gz
author  : <u8222034@cc.nctu.edu.tw>

============
UNIX:CONVERT
============


software: cn2jp
version : 1.3.2, 1996.4.11
function: code conversion routines for Chinese and Japanese
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/cn2jp1.3.2.tar.gz
author  : Seke Wei

software: ktty
version : 1.3, 1996.4.11
function: This is a kanji terminal translator among Chinese and Japanese.
          It allows online translation of codes so that you can view a
          specific code real time using your Chinese or Japanese terminal.
          It runs on Unix and is derived from the 'hztty' package
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/ktty1.3.tar.gz
author  : Seke Wei

software: code1
version : 1.3, 1996.4.11
function: This is a multilingual file browser for Chinese and Japanese.
          It runs on Unix and allows browsing of files in various codes
          using a Chinese or Japanese terminal.
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/code1.3.tar.gz
author  : Seke Wei

software: HUG program
version : 1.0, 1995.5.20
function: Converting between HZ, Uudecode/uuencode, GB codes
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/shug.osf.zip
          ftp://ftp.ifcss.org/pub/software/unix/convert/shug.sun4.zip
author  : Yinrong Huang
```

```
software: hztty
version : 2.0 Jan 29, 1994
Function: This program turns a tty session from one encoding to another.
          For example, running hztty on cxterm can allow you to read/write
          Chinese in HZ format, which was not supported by cxterm.
          If you have many applications in different encodings but your
          favor terminal program only supports one, hztty can make life easy.
          For example, hztty can your GB cxterm into a HZ terminal, a
          Unicode (16bit, or UTF8, or UTF7) terminal, or a Big5 terminal.
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/hztty-2.0.tar.gz
author  : Yongguang Zhang

software: EHZ
version : 2.0
function: Conversion among GB/BIG5/CNS and EHZ-BIG5/GB/CNS.
          Patch to hztty to support EHZ-BIG5/GB/CNS.
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/EHZ-2.0.tar.gz
author  : Ricky Yeung
comment : Also contains the EHZ spec and Fung Fung Lee's EHZ-BIG5-CNS spec.

software: b5cns
version : prototype
function: functions to convert Big5 <-> CNS
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/b5cns.tar.gz
author  : Ross Paterson

software: c2t
version : n/a
function: converts GB or BIG5 coded chinese to _pinyin_
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/c2t.tar.gz
authors : Tommi Kaikkonen and Katya Ta

software: BeTTY/CCF/B5Encode package
version : 1.534, 1995.03.22
function: a chinese code conversion package for codes widely used
          in Taiwan and the GB code widely used in Mainland, plus
          a 7-bit Big5 encoding method (B5Encode3/B5E3, an extension
          to HZ encoding for GB),
          including off-line converters (CCF/Chinese Code Filters and
          B5E/B5Encode) and an on-line converter (BeTTY) which simulates
          your native chinese terminal to become aware of the coding
          systems widely used in Taiwan and GB, HZ encoding.
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/BeTTY-1.534.tar.gz
          ftp://hermes.ee.nthu.edu.tw/shin/betty/BeTTY-1.534.tar.gz
author  : Jing-Shin Chang

software: BeTTY-ws_2fl.p1
version : 1.0 (patch to BeTTY-1.534) Oct. 1995.
function: 1. makes BeTTY respect the window(tty) size.
          2. a second Chinese code converting filter can be added on.
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/BeTTY-ws2fl.p1.tar.gz
author  : Wei Dong
```

```
software: c2gif
version : 0.01, 1995.10.21
function: convert a BIG5 text file to a GIF file
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/c2gif001.tar.gz
          http://www.math.ncu.edu.tw/~luors/c2gif/
author  : Luoh Ren-Shan

software: gb2jis
version : 1.5, 1995.11.19
function: convert GB (or HZ) to JIS with two-letter pinyin
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/gb2jis.tar.gz
          ftp://ftp.ifcss.org/pub/software/fonts/gb/bdf/guobiao16.bdf.gz
author  : Koichi Yasuoka

software: jis2gb
version : 1.5, 1996.1.10
function: convert JIS to GB (or HZ)
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/jis2gb.tar.gz
          ftp://ftp.ifcss.org/pub/software/fonts/gb/bdf/guobiao16.bdf.gz
author  : Koichi Yasuoka

software: HZ
version : 2.0
function: convert among GB, HZ and zW
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/HZ-2.0.tar.gz
author  : Fung F. Lee

software: HZ+ specification and conversion utilities
version : 0.77
function: HZ+ is a convenient 7-bit representation of mixed Big5, GB,
          and ASCII text for use in Internet e-mail, news, etc.
          Source code for Big5 <-> HZ+ and GB <-> HZ+ conversion
          utilities is here.  DOS executables can be found in
          another archive, /software/dos/convert/hzp.zip.  New in
          this Unix version is a simple HZ+ terminal program for cxterm
          which allows the user to transparently read HZ+ mail and news.
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/hzp.tar.gz
author  : Stephen G. Simpson

software: hc
version : 3.0
function: convert between GB and BIG5
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/hc-30.tar.gz
author  : Fung F. Lee and Ricky Yeung

software: Support Table for Hanzi Convert (hc)
version : 1994/05/01
function: Convert table supports the program Hanzi Convert
          (Author  : Fung F. Lee and Ricky Yeung) GB<->Big5.
          Include Russian, number, Japanese, graphing symbols and
          "incorrect" codes. Text file, comments in it.
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/sym-supp.tab
```

```
          ftp://ftp.ifcss.org/pub/software/unix/convert/in-corr.tab
author  : Chi-Ming Tsai

software: pbmbig5
version : 0.01, 1995.11.2
function: convert big5 coded Chinese text file to pbm graphic file
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/pbmbig5-0.01.tar.gz
          ftp://ftp.ifcss.org/pub/software/fonts/big/hbf/kck24.hbf
          ftp://ftp.ifcss.org/pub/software/fonts/big/hbf/kcchin24.f02
author  : Wei-Jou Chen

software: UTF utilities
version : 31 May 1994.
function: various utilities for the UTF encoding of Unicode/ISO-10646,
          including conversion from ISO-2022 and (partially) back again.
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/utf.tar.gz
author  : Ross Paterson

software: utf7
version : prototype, use at your own risk
function: functions to convert between UTF-7 and other codes
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/utf7.tar.gz
author  : Ross Paterson

software: ISO-2022-CN encoder and decoder
version : beta 960408
function: Convert between CN-GB and CN-CNS codes
URL     : ftp://ftp.ifcss.org/pub/software/unix/convert/iso-cn.tgz
author  : handa@etl.go.jp
comment : BIG5 code is not yet supported.

===============
UNIX:NETWORKING
===============

software: gopher2.014c
version : 2.0.14
function: A Chinese localized gopher client capable of 8-bit BIG5 Chinese
          string search on IBM AIX, SUN OS, and any other machines.
          Compatable with any BIG5 Chinese system such as ET and
          cxterm. executable binary for ibm and sun included.
URL     : ftp://ftp.ifcss.org/pub/software/unix/networking/gopher2.014c.tar.gz
          ftp://ftp.ifcss.org/pub/software/unix/networking/gopher.ibm.tar.gz
          ftp://ftp.ifcss.org/pub/software/unix/networking/gopher.sun.tar.gz
author  : Hoo-Tung Cheuk (NCTU CIS, Taiwan)

software: Chinese Tin
version : 1.2PL2a
function: tin 1.2PL2 newsreader with English/Big5 message toggle.
URL     : ftp://ftp.ifcss.org/pub/software/unix/networking/ctin122a.tar.gz
author  : Shih-Kun Huang

software: NcFTP with chinese message compatible
```

```
version : 2.3.0c, 1996.1.17
function: Chinese (BIG5) patch to NcFTP.
          Now it can display any chinese message from ftp server instead
          of "\xxx".  ANSI color compatible in Line Mode.
URL     : ftp://ftp.ifcss.org/pub/software/unix/networking/ncftp-2.3.0.chinese.tgz
author  : NCEMRSoft (orig), Aiken Sam (chinese patch)

==========
UNIX:PRINT
==========

software: C2PS
version : 1.30 Aug 1 1995
function: Translate Big5 coded Chinese document into Level 2 PostScript.
          This is the version for Sparcstation. Using Chinese TrueType fonts,
          you can create the most beautiful document with C2PS.
          This is a DEMO version. You can freely copied and use it.
URL     : ftp://ftp.ifcss.org/pub/software/unix/print/c2ps130sos.tar.gz
          ftp://ftp.ifcss.org/pub/software/fonts/big5/ms-win/
author  : Hsueh-I Lu

software: cnprint
version : 2.60  JAN-25-95
function: print GB/Hz/BIG5/JIS/KSC/UTF8 etc or convert to PostScript
          (conforms to EPSF-3.0). Fast.  Multicolumn. Vertical printing.
          Small disk space requirement. "Intelligent" treatment of
          punctuations.  Flexible change of fonts, char size,
          width/height, char and line spaces, paper orientation and
          margins, etc.  Support of European chars.  Special modes for
          printing HXWZ.  See readme for more
URL     : ftp://ftp.ifcss.org/pub/software/unix/print/cnprint260.tar.gz
          ftp://ftp.ifcss.org/pub/software/fonts/{gb,big5,misc,unicode}/hbf/
author  : Yidao Cai
comment : v2.60 is also for VMS, use v2.61 for DOS

software: GBscript
version : 1.11
function: Convert GB/ASCII mixed text to PostScript output.
          High print speed (4ppm on LaserWriter NTX).
          Support Adobe-2.1.
          Small PS file size (400K for one HXWZ issue).
URL     : ftp://ftp.ifcss.org/pub/software/unix/print/gbscript-1.11.tar.gz
author  : Yan Zhou

software: gb2ps
version : 2.02
function: convert GB/HZ to postscript, supports simple page formatting
          (change chinese fonts and font size, cover page, page
          number, etc). Five chinese fonts are provided in this
          release, they are Song, Kai, Fang Song, Hei and FanTi
          The HZ ENCODING is also supported.
URL     : ftp://ftp.ifcss.org/pub/software/unix/print/gb2ps.2.02.tar.gz
          ftp://ftp.ifcss.org/pub/software/fonts/gb/misc/fan24.ccf.gz
```

```
          ftp://ftp.ifcss.org/pub/software/fonts/gb/misc/fang24.ccf.gz
          ftp://ftp.ifcss.org/pub/software/fonts/gb/misc/hei24.ccf.gz
          ftp://ftp.ifcss.org/pub/software/fonts/gb/misc/kai24.ccf.gz
          ftp://ftp.ifcss.org/pub/software/fonts/gb/misc/song24.ccf.gz
author  : Wei SUN

software: news2ps
version : n/a
function: news2ps converts BIG5 to Postscript
URL     : ftp://ftp.ifcss.org/pub/software/unix/print/news2ps.c
          ftp://ftp.ifcss.org/pub/software/fonts/big5/misc/chinese.16.new
author  : The Society of HKU Postgraduate on Chinese Affairs
comment : rename chinese.16.new to chinese.16

software: hz2ps
version : 3.1
function: Convert hanzi (GB/BIG5) text to PostScript.
          Use HBF font files.
URL     : ftp://ftp.ifcss.org/pub/software/unix/print/hz2ps-3.1.tar.gz
          ftp://ftp.ifcss.org/pub/software/fonts/{gb,big5}/hbf/
author  : Fung F. Lee

===========
UNIX:VIEWER
===========

software: ChiRK
version : 1.2a
function: GB/HZ/BIG5 text viewer on terminals (or emulations) capable
          of displaying Tektronics 401x graphics, such as GraphOn,DEC
          VT240/330, Xterm, Tektool on Sun, EM4105 on PC,
          VersaTerm-Pro on Mac, etc.
          displays up to 17x40 Chinese characters per screen. works directly
          under UNIX mail and news programs. Comes with four fonts.
URL     : ftp://ftp.ifcss.org/pub/software/unix/viewer/ChiRK-1.2a.tar.gz
          ftp://ftp.ifcss.org/pub/software/fonts/gb/hbf/cclib.v
          ftp://ftp.ifcss.org/pub/software/fonts/gb/hbf/cclib.16
          ftp://ftp.ifcss.org/pub/software/fonts/gb/hbf/cclibf.16.gz
          ftp://ftp.ifcss.org/pub/software/fonts/big5/misc/chinese.16.new.gz
author  : Bo Yang
comment : rename chinese.16.new to chinese.16

software: Cbanner
version : 1.10, 950821
function: To show Chinese word's banner
URL     : ftp://ftp.ifcss.org/pub/software/unix/banner/cbanner1.10.tar.gz
author  : Sheen Cherng-Dar, rewritten by Jonen Liu
comment : requires ETen Big5 Chinese System's fonts.

software: gb2text
version : n/a
function: convert GB to text
URL     : ftp://ftp.ifcss.org/pub/software/unix/viewer/gb2text.c
```

```
author  : Ding Yijun

software: hzbanner
version : 1.1, Feb 15, 1995
function: Display Song style GuoBiao in large ASCII characters,
          supports GB2312-80 (^[$A), GB2312-80 + GB8565-88 (^[$(E),
          Chinese-EUC (8-bit Guobiao) CNS Plane 1 & 2, BIG5 and HZ-encoding
          ( ?·ã??)
URL     : ftp://ftp.ifcss.org/pub/software/unix/viewer/hzbanner11.tar.gz
          ftp://ftp.ifcss.org/pub/software/fonts/gb/bdf/guobiao16.bdf.gz
          ftp://ftp.ifcss.org/pub/software/fonts/cns/bdf/cns1hku16.bdf.gz
          ftp://ftp.ifcss.org/pub/software/fonts/cns/bdf/cns2hku16.bdf.gz
author  : Koichi Yasuoka

software: hzview
version : 3.1
function: Display hanzi (GB/BIG5) text on dumb terminal.
          Use HBF font files.
URL     : ftp://ftp.ifcss.org/pub/software/unix/viewer/hzview-3.1.tar.gz
          ftp://ftp.ifcss.org/pub/software/fonts/{gb,big5}/hbf/
author  : Fung F. Lee

Software: cnview
Version : 3.1 (UNIX version. DOS version available under /software/dos/viewer)
Function: View GB/Hz/Big5 encoded Chinese text file on * HP-UNIX (X-window) *
URL     : ftp://ftp.ifcss.org/pub/software/unix/viewer/cnview.tar.gz
          ftp://ftp.ifcss.org/pub/software/fonts/{big5,gb}/hbf/
Author  : Jifang Lin

software: readgb
version : n/a
function: convert GB to text
URL     : ftp://ftp.ifcss.org/pub/software/unix/viewer/readgb.c
author  : Yuzhao Lu
comment : modified from readnews.c

software: readnews
version : n/a
function: readnews converts BIG5 to ascii dot-matrix picture
URL     : ftp://ftp.ifcss.org/pub/software/unix/viewer/readnews.c
          ftp://ftp.ifcss.org/pub/software/fonts/big5/misc/chinese.16.new.gz
author  : The Society of HKU Postgraduate on Chinese Affairs
comment : rename chinese.16.new to chinese.16

==========
UNIX:INPUT
==========

software: CCTeach
version : 1.0
function: Chinese Character input method Teacher.
          Help new user to learn CC input and some utility programs
              of "cxterm NewFace" for associate dictionary tool,
```

```
              hotkey tool, WuBi phrase encoder, and converter with
              ".tit" <==> ".titnf".
          Based on GB and Big5 (ETen and HongKong).
          Support all input method by external dictionary file.
          Need cxterm in unix, CC DOS or ZW DOS in PC.
URL     : ftp://ftp.ifcss.org/pub/software/unix/input/CCTeach1.0.tar.gz
author  : Xiaokun Zhu

=========
UNIX:MISC
=========

software: ICCS 1.3
version : 1.3, June 26, 1994
function: Internet Chinese Chess Server
URL     : ifcss.org:/software/unix/misc/iccs-1.3.tar.gz
author  : Xi Chen
comment : file off-lined due to legal status July 3, 1996. Please contact
          the author Xi Chen at xichen@abel.math.harvard.edu for further info.

========
UNIX:WWW
========

software: cdelegate
version : 1.4a, 1996.5.1
function: This is a Chinese patch on DeleGate, a gateway for the WWW services.
          provides code translation between Chinese/Japanese for WWW browsers
URL     : ftp://ftp.ifcss.org/pub/software/unix/www/cdelegate1.4.tar.gz
author  : Seke Wei

software: Chinese Lynx
version : 2.5FMc, 1996.7.19
function: Chinese BIG5/GB patch to lynx, a WWW client for vt100 terminals.
          Volunteers needed to continue on the patch.
URL     : ftp://ftp.ifcss.org/pub/software/unix/www/clynx25.zip
author  : Nelson Chin

software: Internet MahJong Server (server + client applet)
version : 0.2beta
function: provides a server and a graphic client for playing MahJong on
          the Internet.
URL     : ftp://ftp.ifcss.org/pub/software/unix/www/MJ_dist.tar.gz
author  : Zuwei Thomas Feng
*****
LINUX
*****

software: D Series (Chinese Tools, ELF binary)
version : 1.00,  March 25, 1995
function: Binary distribution of various useful Chinese tools:
          Viewers -- cxterm, crxvt; Input server -- xcin; GB,Big5,HZ,B5E3
          code converters -- ccf, hc, hz2gb, gb2hz, zw2hz, b5decode,
```

```
          b5encode; Print tool -- lunar, ttf2ps; Pseudo tty -- hztty,
          betty; Various handy scripts, man pages, dictionaries, HBF fonts,
          xfonts included; chdrv, celvis, elm, sendmail
URL     : ftp://ftp.ifcss.org/pub/software/linux/CLinux/CTool/d1
          ftp://ftp.ifcss.org/pub/software/linux/CLinux/CTool/d2
          ftp://ftp.ifcss.org/pub/software/linux/CLinux/CTool/d3
          ftp://ftp.ifcss.org/pub/software/linux/CLinux/CTool/d4
author  : Eric Lin
comment : requires XFree86 3.1+, ELF libraries

software: C Series (Chinese packages for Slackware)
version : N/A
function: The Chinese packages collected by Wei-Jou Chen can be installed by
          Slackware's setup tools. Basic idea are that we have right to install
          and remove Chinese softwares easily and beginners can play them
          without much trouble.
URL     : ftp://ftp.ifcss.org/pub/software/linux/CLinux/c1/
          ftp://ftp.ifcss.org/pub/software/linux/CLinux/c2/
          ftp://ftp.ifcss.org/pub/software/linux/CLinux/c3/
          ftp://ftp.ifcss.org/pub/software/linux/CLinux/c4/
          ftp://ftp.ifcss.org/pub/software/linux/CLinux/c5/
compiler: CHEN, Wei-Jou

software: MU Series (Mule packages for Slackware)
version : N/A
function: The Multilingual Emacs 2.0 packages for XFree86 2.X and 3.1
          collected by Shawn Hsiao can be installed by Slackware's setup
          tools
URL     : ftp://ftp.ifcss.org/pub/software/linux/CLinux/mu1/
          ftp://ftp.ifcss.org/pub/software/linux/CLinux/mu2/
          ftp://ftp.ifcss.org/pub/software/linux/CLinux/mu3/
          ftp://ftp.ifcss.org/pub/software/linux/CLinux/mu4/
          ftp://ftp.ifcss.org/pub/software/linux/CLinux/mu5/
          ftp://ftp.ifcss.org/pub/software/linux/CLinux/mu6/
compiler: Shawn Hsiao

=============
LINUX:C-UTILS
=============

software: GNU fileutils-3.9
version : 3.9, 1 August 1994
function: Chinese version of the GNU file utility programs for Linux.
URL     :
ftp://ftp.ifcss.org/pub/software/linux/c-utils/fileutils-3.9-bin-chinese.tar.gz
author  : Patrick D'Cruze

software: C2PS
version : 1.30 Aug 1 1995
function: Translate Big5 coded Chinese document into Level 2 PostScript.
          This is the version for Linux. Using Chinese TrueType fonts, you
          can create the most beautiful Chinese documents with C2PS.
          This is a DEMO version. You can freely copied and use it.
```

```
URL     : ftp://ftp.ifcss.org/pub/software/linux/chinese_utils/c2ps130lnx.tar.
          ftp://ftp.ifcss.org/pub/software/fonts/big5/ms-win/
author  : Hsueh-I Lu

============
LINUX:EDITOR
============

software: ?????? for Linux (promotion version)
version : v2.163
function: PE2-like text editor, special designed for Chinese
URL     : ftp://ftp.ifcss.org/pub/software/
agent   : LU, Heman

=========
LINUX:TTY
=========

software: chdrv
version : 1.0.7, 1995.12.20
function: Chinese Terminal Simulator. Does not require X-Windows.
URL     : ftp://ftp.ifcss.org/pub/software/linux/tty/chdrv-1.0.7.tar.gz
          ftp://ftp.ifcss.org/pub/software/linux/tty/chdrvbin-1.0.7.tar.gz
          ftp://ftp.ifcss.org/pub/software/linux/tty/chdrvfont.tar.gz
author  : WANG, Yu-Chung

===========
LINUX:X11R6
===========

software: Behavior DTop (for Linux)
version : 1.4, Beta
function: A full-featured Chinese DeskTop Publishing Software Package
          characterized by object-oriented design for manipulating
          various document objects, including text, tables, graphics,
          equations, images, in an integrated way. Two outline fonts
          are provided in the Beta Version. PostScript output.
          Good as an English DeskTop Publishing Software Package too.
          (See the README's & formated DTop manual files for a long and
          complete list of functions and characteristics).
URL     : ftp://ftp.ifcss.org/pub/software/linux/X11R6/dtop1.4/
author  : DTop Development Group
```

Linux Commercial HOWTO

Mr. Poet <poet@linuxports.com>

v5.38, 7 March 1999

This document contains a listing of commercial software and applications which are offered for Linux. It is maintained by Mr. Poet <poet@linuxports.com>.

1. About this document

2. Copyright Information

3. Disclaimer

4. Related Information

5. Linux for the Commercial Market?

- 5.1 Turnkey systems
- 5.2 Large end-user customers
- 5.3 Specialized markets
- 5.4 Biography

6. Website Development

- 6.1 ASWedit, HTML editor
- 6.2 Empress DataWEB
- 6.3 EZ-EDIT
- 6.4 LinkScan
- 6.5 TalentSoft Web+ (WebPlus)
- 6.6 VirtuFlex 1.1

- 6.7 Visual prolog
- 6.8 Web Crossing
- 6.9 ThreadTrack and WebTailor from Webthreads.

7. Databases

- 7.1 c-tree Plus
- 7.2 Empress
- 7.3 Essentia
- 7.4 FairCom Server
- 7.5 INFORMIX-SE
- 7.6 Just Logic/SQL
- 7.7 KE Texpress
- 7.8 Qddb
- 7.9 Raima Database Manager++
- 7.10 Empress Embedded RDBMS
- 7.11 SOLID Server
- 7.12 Velocis Database Server
- 7.13 Yard SQL

8. Data Visualisation and CAD

- 8.1 IDL (Interactive Data Language)
- 8.2 Megahedron
- 8.3 Tecplot 7.0
- 8.4 VariCAD
- 8.5 VARKON
- 8.6 XVScan

9. Development Tools

- 9.1 ACUCOBOL-GT
- 9.2 Amzi! Prolog & Logic Server
- 9.3 Basmark QuickBASIC

- 9.4 Critical Mass CM3
- 9.5 Dynace
- 9.6 Absoft Fortran 77
- 9.7 Finesse
- 9.8 ISE Eiffel
- 9.9 EiffelBench
- 9.10 C-Forge IDE
- 9.11 IdeaFix
- 9.12 j-tree
- 9.13 KAI C++
- 9.14 Khoros Pro 2.1
- 9.15 MetaCard
- 9.16 ObjectManual Release 3.0
- 9.17 Critical Mass Reactor
- 9.18 Resource Standard Metrics
- 9.19 r-tree
- 9.20 sdoc (Source Documenter)
- 9.21 SEDIT, S/REXX
- 9.22 SNiFF+
- 9.23 ST/X (Smalltalk/X)
- 9.24 tdb (Tcl Debugger)
- 9.25 tprof (Tcl Profiler)
- 9.26 View Designer/X (VDX)
- 9.27 XBasic
- 9.28 XMove 4.0 for Linux

10. Emulation

- 10.1 Emulus
- 10.2 Executor 2
- 10.3 Wabi 2.2 for OpenLinux

TOC

11. Financial Software

- 11.1 BB Stock Pro and BB Stock Tool
- 11.2 TimeClock

12. Libraries

- 12.1 FontScope
- 12.2 INTERACTER
- 12.3 Matrix<LIB> - C++ Math Matrix Library
- 12.4 PKWARE Data Compression Library for Linux
- 12.5 readyBase
- 12.6 SIMLIB IG

13. Mathematics

- 13.1 Maple V Release 4 - The Power Edition
- 13.2 MATCOM and MATCOM MATH LIBRARY
- 13.3 Mathematica 3.0
- 13.4 MATLAB and Simulink

14. Multimedia

- 14.1 Peter Lipa and his Journeys
- 14.2 Lucka Vondrackova and her Journeys
- 14.3 MpegTV Player 1.0
- 14.4 Peter Nagy and his Journeys
- 14.5 Xaudio

15. Network Servers

- 15.1 Critical Angle X.500 Enabler
- 15.2 DNEWS News Server
- 15.3 Aventail Internet Policy Manager
- 15.4 Aventail VPN

- 15.5 WANPIPE
- 15.6 Zeus Web Server

16. Office Tools

- 16.1 Corel WordPerfect 8
- 16.2 The American Heritage Dictionary Deluxe
- 16.3 Applixware Office Suite for Linux
- 16.4 D.M.S. Document Management System
- 16.5 HotWire EasyFAX
- 16.6 NExS, the Network Extensible Spreadsheet
- 16.7 Axene Office
- 16.8 Projector and Projector/Net
- 16.9 The Virtual Office System
- 16.10 Axene XAllWrite
- 16.11 Axene Xclamation
- 16.12 Axene XQuad

17. Text Processing

- 17.1 Edith Pro for X11
- 17.2 TeraSpell 97 for Emacs

18. System Administration Tools

- 18.1 Host Factory
- 18.2 PerfectBACKUP+
- 18.3 Venus

19. X Windows Related Products

- 19.1 Accelerated-X Display Server
- 19.2 BXwidgets
- 19.3 BXwidgets/DB
- 19.4 Laptop, Accelerated-X Display Server

- 19.5 MaXimum cde Developer's Edition v1.0
- 19.6 Multi-headed, Accelerated-X Display Server
- 19.7 OpenGL, Accelerated-X Display Server
- 19.8 OSF-Certified Motif

20. Other Software

- 20.1 ABACUS 4
- 20.2 BBBS
- 20.3 Clustor
- 20.4 FootPrints
- 20.5 Aladdin Ghostscript
- 20.6 Magician
- 20.7 journyx WebTime
- 20.8 LanSafe
- 20.9 LjetMgr
- 20.10 Synchronize/CyberScheduler

21. Free Software for Commercial Hardware

- 21.1 Stallion Technologies Multiport Serial Boards

---

1. About this document

This is the Linux Commercial HOWTO. It contains a listing of commercial software which is available for Linux. The Linux Commercial HOWTO doesn't contain any information on Linux distributions -- this is covered by the Distribution HOWTO.

If you contact any companies or purchase any products listed in this document, please mention the Linux Commercial HOWTO.

This document was originated by Harald Milz <Harald.Milz@Linux.org>. and then maintained by Mr. Poet

It is now maintained by Mr. Poet AND LinuxPorts <poet@linuxports.com>.

If you need to know more about the Linux Documentation Project or about Linux HOWTO's, feel free to contact the supervisor Tim Bynum <linux-howto@metalab.unc.edu>.

Tim Bynum will post the listing to several national and international newsgroups on a monthly basis. In addition, the Linux Commercial HOWTO can be found on the World Wide Web at http://commercial.cyrius.com. New versions of the Linux Commercial HOWTO are always placed at this site first, so please be sure to check if the copy you are reading is still up to date!

The Linux Commercial HOWTO is not a forum for product announcements or marketing hype; it is a service for potential customers and the whole Linux community. Resellers will not be listed; the list is for companies who produce their software themselves. Two main goals are being aimed at:

- It shall help companies who want to run Linux to find software solutions and applications. The international distribution of this list will enhance the contact opportunity.
- It is meant to prove the commercial useability of Linux and thus to encourage other vendors to port their software as well.

Companies and developers who are offering their products for Linux and interested in joining the Linux Commercial HOWTO are invited to fill out the following form and contact me at poet@linuxports.com.

This HOWTO contains tabular entries for each product (example follows). The entry format is similar to the Linux Software Map (LSM) entry (field/stanza lengths are arbitrary). If you want me to add your entry please *keep short*, otherwise I'll have to shorten your data. Furthermore, please send me plain ASCII data; no HTML, and no PostScript.

```
Category:

        Databases, Data Visualisation, Development tools, Financial
        Software, Mathematics, Multimedia, Network Management, Text
        Processing, X Windows or Other Software.

Name:

The name of your application.

Description:
```

Short description of the package, just the basic functionality.

Distribution media:

Licensing policy:

Whatever applies. Is there a free demo or shareware version available via FTP or WWW? Where?

OS provisions:

Kernel version, XFree86 version, Motif version, RAM, harddisk usage, etc.

Documentation:

Printed documentation, page number, online help, language.

Extra features and add-Ons:

(and their prices)

Price range:

Whatever it costs.

Vendor:

Address:
Phone: (U.S. and Canada: if you like to be reachable internationally, please don't enter only a +1 800 or +1 900 number)
Fax:
EMail:
URL:
Contact:

---

2. Copyright Information

This HOWTO is Copyright © 1996, 1997, 1998 by Mr. Poet.

A verbatim copy may be reproduced or distributed in any medium physical or electronic without permission of the author. Translations are similarly permitted without express permission if it includes a notice on who translated it.

Short quotes may be used without prior consent by the author. Derivative work and partial distributions of the Linux Commercial HOWTO must be accompanied with either a verbatim copy of this file or a pointer to the verbatim copy.

Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

We further want that *all* information provided in the HOWTOs is disseminated. If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@metalab.unc.edu.

3. Disclaimer

This HOWTO is not actually a HOWTO in the sense of the Linux Documentation Project. Instead, it is an instrument to investigate the commercial Linux opportunity and to list applications which were already ported and marketed in a native Linux version. As a software vendor, you probably know that you can alternatively offer Linux users a statically linked SCO version of your application which would probably run under the iBCS2 emulator (albeit with a small performance penalty and higher memory requirements). Such applications will not be listed here.

I will not select nor deselect any particular product. Instead, everyone who wants to have her product included will be serviced. However, I reserve the right to shorten individual entries to keep things in shape.

If you don't find a particular product or vendor in this list, this is probably due to one of the following reasons:

- I never heard of that product or vendor and thus didn't try to get in contact.
- I did get in contact, but the vendor didn't answer yet.
- I did get in contact, but the vendor stated positively that he doesn't sell his product for Linux (yet).

In any case, please get in contact if you feel someone's missing; also if you discover any errors in the file.

Sometimes two vendor's addresses are mentioned in the ``Vendor:'' field. In these cases I received the information from the German subsidiary/distributor. The original manufacturer's address is always mentioned first.

4. Related Information

There's another document which covers commercial Linux software. It is maintained by Alan Cox <Alan.Cox@Linux.org> and can be found at http://www.uk.linux.org/LxCommercial.html.

In addition, feel free to visit the Linux Mall where you can order most products presented in this HOWTO.

5. Linux for the Commercial Market?

Copyright © 1996 iX Multiuser Multitasking Magazin
Courtesy of iX Multiuser Multitasking Magazin!

People keep asking me ``When is Linux going to be ready for the *commercial market*". I guess the first thing to discuss is what is meant by "commercial" in this context.

Some CD-ROM vendors have put the word "commercial" in their name, only to have the technical people think their product is good only for use by banks and insurance companies. Other people look at their product with disdain and say that ``Linux is not *commercial quality*", because it is missing some feature they need, or they feel it is unstable (usually without ever trying it even one time).

To me, the word ``commercial" has lost as much meaning in the marketplace as some of the other buzzwords:

- ``Open" vs ``Proprietary" (People are now saying UNIX is ``Proprietary")
- ``Workstation" vs ``PC" (What is a workstation, anyway?)

In the old days of computing the commercial market was banks, insurance companies and business-oriented facilities where the use of COBOL or RPG was the mainstream. The technical market was scientific, engineering and manufacturing where FORTRAN and assembly language was used. Somewhere along the way the term ``commercial" seems to have gotten twisted around to mean ``ready for the mass market", versus ``ready for hackers".

For the purpose of this article I will take the second meaning, and address whether I think Linux is ready for commercial purposes rather than the hobbyist and hacker market, and ready for the mass market rather than limited markets.

For those of you who hate reading long articles, or who are short on time, let me give you my conclusion right now. Then you can go out and drink beer or other fun activities:

``Yes, Linux is ready for the commercial market...in some cases".

In order for an operating system to be ready for the mass market it must have several attributes:

- have lots of applications
- be relatively easy to install
- have lots of applications
- be relatively easy to maintain
- have lots of applications
- be relatively easy to use
- have lots of applications
- not crash (much)

TOC

- have lots of applications
- be economical
- have lots of applications

But you can eliminate all of these considerations in today's mass market if only one thing is true:

You have lots of applications.

after all, there would not be 170,000,000 DOS systems in the world if any of the others *had* to be true.

I almost added that is has to be economical, but history has actually proven me wrong on that. If people added up the total cost of ownership, then Apple would certainly have won over the PC. But people ignore the human costs of someone else (or even themselves) beating their head against the wall trying to get something to work, or the system crashing repeatedly, or the fact that the one keystroke they can hit the easiest (through practice) is

```
<CTRL><ALT><DEL>
```

In the old days people were content to spend several hundreds of dollars on a simple ASCII text editor, or deal with a simple spreadsheet. And it took an act of mangement to get them, with lots of Purchase Orders. Today, they want multi-media integrated with their operating system, and have all the applications available that their neighbor (or boss, or compatriot) has available on *their* system. And they want to get these applications easily, certainly no harder than to call up on the phone to order them through a catalog, or go down to their corner store to get them.

Now what causes this plethora of applications for an operating system? Ease of programming? Good software development tools? Features inside the operating system? Stability of the interfaces over time?

The answer is ``none of these". While all these attributes may help convince an application developer to port, the one overriding issue is volume of the operating system platform. Again, if MSDOS were compared to MacOS, or even to UNIX and volume were not taken into account, we know which two operating systems would have the most applications, and they would not be from Microsoft.

While it is true that several Linux vendors are working on getting these applications for the mass market (read this " your mother and father"), the number of applications that run on Microsoft platforms have been estimated as high as 35,000. SunOS has an estimated 10,000 applications, with other `commercial UNIX" systems (including Solaris 2.x) much lower in number. It will take the Linux vendors a long time to get the number of applications necessary to hit the really large mass market, particularly if they did not depend on iBCS2 and DOS/Windows compatibility (which could supply a fair number of current applications), but depended on ``native" Linux applications.

So applications are king (and queen) for the mass market, and installed base (volume) or the promise of explosive growth (volume) is the key to these. But is the mass market the only ``commercial" market? The answer is ``no". The mass market is a subset (albeit very large one) of the commercial market. So let's look at what the rest of the commercial market needs. We will look at this by segmenting the market into:

- turnkey systems
- large end-user customers

- specialized markets

5.1 Turnkey systems

When I speak of turnkey systems I typically mean a computer system that has one specific (or not so specific) application that runs on it. Examples of turnkey systems are point-of-sale terminals, reservation systems, CAD systems, etc. But in a larger sense, other applications such as Web servers, nameservers (such as BIND), etc. could also be considered ``turnkey'', since they have only a few necessary programs that have to run on the system.

Usually turnkey systems are ones that an Independent Software Vendor (ISV) or Value Added Reseller (VAR) will chose a hardware system, an operating system, port an application to it, then duplicate that system 500 to 1000 times without change to the basic application. These ISVs and VARs will try to chose the lowest cost solution to fit their customer's needs.

Linux is perfect for these types of applications. The operating system is stable enough for the developer to port their application and test the application fully. Once it is fully tested and stable, the entire package is ``frozen'' and duplicated any number of times for the end customer.

Since the operating system may be freely copied, and it runs on inexpensive hardware, their variable costs are minimal. Even a developer who is not familiar with the Linux system (so they need help getting it running on a platform) will quickly pay back the porting and system programmer costs they accrue by not paying $200-$500. per license for the operating system. Plus they have all the source code for the entire system, in case they run into trouble later on. You can buy a lot of Linux support for $200-$500K.

As I said before, I include Internet Service Providers (ISPs) as part of this ``turnkey'' environment, for both external internet and internal internet. Why overload your expensive, high-powered, highly complex general-purpose server to do Web serving when a smaller, simpler box can offload it? Why not run your NIS slaves on a Linux box? Or perhaps your BIND server?

In the early days of Digital UNIX (known then as DEC OSF/1) we did not have very many applications. In fact we had none. The marketing staff came to me with sad faces asking if it was possible to sell an operating system that had no applications. I invented a term called ``Turbocharging'', which allowed a Digital UNIX system using the speed and power of the Alpha processor (as well as the throughput of our networking devices) to offload NIS, NFS, BIND and other services from people's overloaded, slower SPARC machines. We also showed people how they could use the rsh(1) command to allow the Alpha to do a portion of their very CPU intensive processing while delivering the result back to the SPARCs on their desk. This allowed the SPARCs to work more on applications and less on the other ``system administration'' tasks that they were performing. We sold lots of Digital UNIX systems based solely on executing those tasks. Today, of course, Digital UNIX has a lot more applications, and particularly very large memory databases that are extremely fast. But the same principle applies. The database engine runs on the Alpha system, supplying data to the slower SPARC engines as a ``Turbocharger''. I could see Linux systems headed in the same direction, following the same path.

TOC

5.2 Large end-user customers

Very large customers often have their own home-grown applications which they need to deploy across a wide network of people. Or they can have management dictate a certain suite of applications, which then can be ported to Linux. Since these customers are so large, their operating system costs are huge, and utilizing the savings using the Linux operating system they may completely cover the expenses of porting their software.

Or these very large customers may ``influence'' their layered product providers to port to the Linux platform. Finally, they may even change some of their computing habits (to use existing programs) if the cost savings are enough to warrant it.

Companies like Caldera are creating a suite of applications and approaching these very large customers to show them the operating system savings that they can achieve if they switch to Linux. While it is true that every application the customer could ever conceive of running may not run on Linux, by using the native applications, the iBCS2 applications, the DOSEMU applications, and applications that run under WABI, a nice suite of applications could be built to solve their needs.

5.3 Specialized markets

Finally there are what I call ``specialized markets''. Markets that might buy Linux simply because it is Linux, and not because of the application suites that it provides.

In the education field there are three main markets:

- administrative
- ``campus computing''
- computer science education

The administrative part is the ``business'' aspect of the market. They are looking for easy-to-use systems that can also handle complex administrative tasks that might cover a community the size of a small city.

The ``campus computing'' is the supply of computing power and service for majors of all types, web services and research into non-computer science (for example, molecular modeling) research.

Finally there is computer science education, both on the undergraduate and graduate level, as well as research into computer science.

While the administrative sub-market typically relies more on shrink-wrapped applications, the other two rely on them to a lesser extent (with the computer science education market relying the least). The other two markets can utilize a lot more of the freeware and shareware applications that are already ported to Linux. This gives them a very low-cost (from a software perspective) platform while allowing them to see and (often) modify the source code for the applications they use.

More importantly, in the computer science research area, the results of the research can be freely distributed to others working in the field, or even published as source code to illustrate the results. This can not be done with ``commercial'' operating systems.

TOC

Some universities are utilizing Linux more and more to run their campus. From a ``commercial'' standpoint, their needs are the same as many large businesses. Students graduating from college will know about Linux, and bring the word to their future employers.

Finally, there is the computer hobbyist and software developer market. I relate this market to the amateur radio market. In the amateur radio market the radio is often used to simply talk to other people, but at the same time the users investigate new ways of using radio, and improving it. Many electrical engineers started out as amateur radio users. So it can be with Linux, since for the first time both the prices of the hardware and the prices of the operating system source code are within the reach of mortal people.

In conclusion, I feel that Linux does have the items needed for several types of ``commercial'' uses:

- stability and quality
- low variable costs for turnkey applications
- explosive growth in volume to attract ISVs

What Linux really needs is for the ``commercial'' community to understand what is going on, and to embrace it where it will be useful. This will increase the volume numbers even more, which will attract more applications.

Along these lines I would like to ``advertise'' a joint effort of USENIX and Linux International to happen in January of 1997 in Anaheim, California of the United States. There will be a joint USENIX/Linux development conference, and while a certain part of the Linux conference will be oriented towards the development of the Linux operating system, the bulk of the conference will be oriented towards application developers and marketing people, to better understand the Linux operating system and how to sell their applications and services into the Linux market. We hope to show ISVs, VARs, resellers and distributors how they can make money by selling their applications and services on top of the Linux operating system.

5.4 Biography

Jon ``maddog'' Hall is a Senior Leader in the Digital Equipment Corporation UNIX group. He has been in the computer industry for twenty-five years, UNIX for sixteen years and has guided the emergence of six operating systems, including Alpha Linux. He has an MS in Computer Science.

---

6. Website Development

6.1 ASWedit, HTML editor

Description:

ASWedit is a commercial, comprehensive and easy to use HTML and text editor for X Window System and Motif. It offers three independent modes: a plain text editing mode and two context-sensitive, validating modes for authoring of HTML documents as used on the Internet and Intranets. The two HTML modes are: standard and experimental.

Distribution media:

3.5" diskettes, 4-mm DAT, 1/4" and 8mm tapes, Internet (FTP).

Licensing policy:

Per machine basis. The number of users that can run the software on the licensed computer is unlimited.

A version of the program, called asWedit, is available for free for students and staff in education and charitable non-profit organizations, and for free evaluation by individuals and commercial organizations. It is available via FTP from many archives. See http://www.advasoft.com/asWedit.html for details.

OS provisions:

Linux 1.2.13 or higher (ELF), X11R6, Motif 2.0 (not required if the statically linked version is used), 5 MB of RAM, 1.5-3.5 MB hard disk usage.

Documentation:

Printed documentation, online help, language: User's Guide (44 pages), HTML 3.2 extended, Reference Manual (89 pages), HTML 3.2 experimental, Reference Manual (106 pages). Online, context sensitive, hypertext help - 560 KB. Localized resource files are available for the following languages: English (default), Czech, Danish, Dutch, French, German, Polish, Portuguese, Spanish and Swedish. Commands and messages are localized for each language but the online help and documentation are only available in English.

Product support:

The license includes free product upgrades by FTP for a period of one year.

Extra features and add-ons:

Can work with external Unix filters. Highly customizable. Supports four different browsers for previewing.

Available since:

July, 1995.

Countries with distribution:

World wide.

Price range:

US$149. Quantity discounts are available.

Vendor:

AdvaSoft Ltd.

```
30 Hatch Road
London SW16 4PN
England
```

Phone:

+44 181 251 0033

Fax:

+44 181 251 0011

EMail:

as@advasoft.com

URL:

http://www.advasoft.com

Contact:

Andre Stochniol

Last modified:

August 30, 1997.

6.2 Empress DataWEB

Description:

Empress DataWEB allows users to easily and rapidly build dynamic, interactive, database-fed web applications. No special programming language needs to be learned; developers of applications simply can use HTML with the Empress extensions for accessing the RDBMS.

Distribution media:

CD-ROM.

Licensing policy:

Please contact vendor for evaluation copies.

OS provisions:

16 MB of RAM; 60 MB harddisk space (additional requirements: web browser required).

Documentation:

Online help.

Product support:

Full technical support available, priced separately.

Extra features and add-ons:

Extra features such as an HTTP server and other tools to facilitate the creation of web applications come with the software package.

Available since:

April, 1996.

Countries with distribution:

World wide.

Price range:

Please contact vendor.

Vendor:

Empress Software Inc.

```
6401 Golden Triangle Drive
Greenbelt, MD 20770
U.S.A.

3100 Steeles Avenue East
Markham, ON L3R 8T3
Canada
```

Phone:

+1 301 220 1919 (USA), +1 905 513 8888 (Canada)

Fax:

+1 301 220 1919 (USA), +1 905 513 1668 (Canada)

EMail:

sales@empress.com

URL:

http://www.empress.com

Contact:

Dick Naedel

Last modified:

August 4, 1997.

6.3 EZ-EDIT

Description:

EZ-EDIT is an online HTML editor, which allows users to completly manage their web site through a web browser (edit, create, upload, rename/move/copy, create and remove directories)! EZ-EDIT is the only editor which features the "File Filter" which allows you to specify what file types are allowed on your system. Create the look you want by editing EZ-EDIT's 16 template files with over 80 tags. Also allows you to set disk space limits! Supports form based file uploads, also includes a Java page creater. All administration is also done through a web browser.

Distribution media:

3.5" diskettes, Internet (FTP and WWW).

Licensing policy:

A free demo is available from our web site. Demo version is unlocked to full version with registration key.

OS provisions:

Linux/Intel.

Documentation:

Online HTML user and admin manuals (English only).

Product support:

Update service, support (via EMail).

Available since:

September 17, 1997.

Countries with distribution:

World wide.

Price range:

US$149.95.

Vendor:

Relative Web

```
P. O. Box 351
Saylorsburg, PA 18353-0351
```

U.S.A.

Phone:

+1 610 381 3072

Fax:

+1 610 381 3072

EMail:

ez-edit@relative-web.com

URL:

http://www.relative-web.com/dynamic/ez-edit/

Contact:

John Bergeron

Last modified:

October 6, 1997.

6.4 LinkScan

Description:

LinkScan operates on Unix an NT servers on both Internets and Intranets, LinkScan can test over 40,000 links per hour because it is the only link checker that uses multi-threaded simultaneous processing. LinkScan has been tested on web sites with over 45,000 pages and more than 80,000 links. LinkScan also produces two revolutionary types of maps of web sites. LinkScan's SiteMap enables the user to produce a site map that includes every link on a web site arranged in a hierarchical format that resembles a book's table of contents. LinkScan's TapMap is an expandable and collapsible site map that allows viewers to tap down through the various and multiple levels of a web site to quickly and easily navigate and explore the web site by tapping on a few control icons.

Distribution media:

Internet (WWW).

Licensing policy:

A license is required for each server on which the product is used. Free evaluation copies of LinkScan may be downloaded from our web site.

OS provisions:

Requires Perl 5 or higher.

Documentation:

Complete documentation and a comprehensive FAQ may be read at our web site and/or

TOC

downloaded.

Product support:

This product is continuously updated and maintained. Prompt responses to all inquiries and problems via EMail or telephone as required. No fees for support.

Available since:

January 7, 1997.

Countries with distribution:

World wide.

Price range:

US$495.

Vendor:

Electronic Software Publishing Corporation

```
1504 #8-00200 Main Street
Gardnerville, NV 89410-5273
U.S.A.
```

EMail:

ken@elsop.com

URL:

http://www.elsop.com

Contact:

Ken Churilla

Last modified:

November 6, 1997.

6.5 TalentSoft Web+ (WebPlus)

Description:

TalentSoft Web+ is a development tool dedicated to developing web-based client/server applications without writing low level CGI programs. Web+ enables rapid and easy creation of highly functional web pages which integrates with databases, file systems, EMail, Java applets, your legacy applications (EXEs, DLLs), and communicates with other TCP/IP applications using sockets. Web+ works with all popular web severs, databases, and operating systems and integrates closely with Netscape web servers via NSAPI and CGI. Web+ also acts as a multi-threaded web middleware that integrates the web servers with databases, EMail, TCP/IP sockets, and other applications. Please check out the "Teach Me Web+" link on our web site for the coolest

interactive tutorial with hands on exercises. We are proud to be the first to provide dynamic cc interpretation on the Internet! Now you may start writing and running your own web applicati without buying or installing Web+.

Distribution media:

3.5" diskettes, CD-ROM and Internet (FTP and WWW).

Licensing policy:

Free evluation copy available at http://www.TalentSoft.com. Trial out version notices displayed by will not time out.

OS provisions:

16 MB RAM, 5 MB HD.

Documentation:

Both printed documentation (about 200 pages) and HTML online help. English only.

Product support:

Free EMail, web+ conference, and phone support. Training available for a fee.

Available since:

May 1, 1997.

Countries with distribution:

USA, UK, Hong Kong, China, Singapore.

Price range:

US$195 to $1295.

Vendor:

TalentSoft / Talent Information Management, LLC.

```
900 Nicollet Mall, Suite 700
Minneapolis, MN 55402
U.S.A.

P.O. Box 2997
Minneapolis, MN 55402
U.S.A.
```

Phone:

+1 612 338 8900

Fax:

+1 612 904 0010

EMail:

info@TalentSoft.com

URL:

http://www.TalentSoft.com

Contact:

Victor Tong, Ian Gorrie, Jeff Persche, Tony Tong

Last modified:

August 8, 1997.

6.6 VirtuFlex 1.1

Description:

VirtuFlex 1.1 is a web application builder for adding dynamic functionality to a web site. VirtuFlex provides the power to transform web sites into live applications by integrating databases, fax, EMail and pagers with the web. VirtuFlex provides sophisticated functionality that can be added to web sites by any HTML developer. VirtuFlex is reusable, modular, easy to use, high performance and provides DB connectivity almost any database. The componenets of VirtuFlex are a macro language, macro processor, DB server and pre-built template packs.

Distribution media:

Internet (WWW).

Licensing policy:

VirtuFLex is licensed on a per domain basis. A free evaluation copy is available on our web site.

OS provisions:

Linux ELF binary format (1.2.x kernels or higher). VirtuFlex runs on standard Unix workstations with 8 MB of RAM minimum, 16 MB recommended.

Documentation:

Available for download from our web site, English.

Product support:

Basic support four hours. Other support options available - call for details.

Extra features and add-ons:

Pre-built Template-Paks come free with VirtuFlex for shopping carts, threaded discussion groups, database application builder, quizzes and surveys, web spiders and banner rotators. Additional Template-Paks are added on a regular basis.

Available since:

1996.

Countries with distribution:

World wide through UniDirect, Soft Export and the Internet.

Price range:

US$995, educational discounts available.

Vendor:

VirtuFlex Software Corp.

```
930 Massachusetts Avenue
Cambridge, MA 02139
U.S.A.
```

Phone:

+1 617 497 8006

Fax:

+1 617 492 0486

EMail:

comments@virtuflex.com

URL:

http://www.virtuflex.com

Contact:

Dan Housman

Last modified:

March 17, 1997.

6.7 Visual prolog

Description:

One of the worlds strongest prolog development environments.

Distribution media:

CD-ROM and Internet (WWW).

Documentation:

Manuals supplied on CD-ROM, online help and intro.html lots of examples.

Product support:

EMail based.

Available since:

1984.

Countries with distribution:

World wide.

Vendor:

Prolog Development Center

```
H.J. Holstvej 3-5A
DK-2605 Broendby
Dankmark
```

Phone:

+45 36 72 10 22

Fax:

+45 36 72 02 69

EMail:

sales@pdc.dk

URL:

http://www.pdc.dk

Contact:

Claus Witfelt <witfelt@pdc.dk>

Last modified:

November 28, 1997.

6.8 Web Crossing

Description:

Online conferencing server for the Intranet and Extranet and world wide web providing discussion forums and chat rooms. Web Crossing is a groupware application server, accessible with any web browser, via most web servers. It makes communication more efficient and productive than newsgroups or EMail mailing lists.

Distribution media:

Internet.

Licensing policy:

commercial; fully-functional demo available.

Price range:

US$995, unlimited users.

Vendor:

Lundeen & Associates

```
P.O. Box 2900
Alameda, CA 94501
U.S.A.
```

Phone:

+1 510 521 5855

Fax:

+1 510 522 6647

EMail:

sales@lundeen.com

URL:

http://webcrossing.com

Last modified:

August 16, 1997.

6.9 ThreadTrack and WebTailor from Webthreads.

Description:

ThreadTrack and WebTailor are lightweight browser and server independent CGI script packages, developed under Linux, that add state to web servers. ThreadTrack is used for tracking the activity of individual visitors to a web site, and WebTailor is used to dynamically modify the content of a web site in response to a visitor's profile or actions.

WebTailor uses a simple server-side scripting language to modify the site's content. The language, targeted to non-technical web designers, is easy to learn and use. For the more technical, the CGI interface has been expanded to enable parameter passing on a per visitor basis between scripts running on different pages.

ThreadTrack tags individual visitors to a web site with a unique identifier that remains with them for their visit. Each visitor's session is recorded click-by-click, so a database of aggregate and individual activity is available for reports. Comprehensive reporting is included, and the data (dBase III) is easily transferrable to custom reporting packages.

Distribution media:

Internet (WWW).

Licensing policy:

30 Day fully functional eval available from http://www.webthreads.com

OS provisions:

Any version of Linux on x86, a.out and ELF.

Documentation:

Web site and online provided with the package.

Product support:

EMail support.

Extra features and add-ons:

Msql interface. Registration site management addition.

Available since:

June, 1996.

Countries with distribution:

World wide.

Price range:

ThreadTrack starts at US$295. WebTailor starts at US$895.

Vendor:

Webthreads, L.L.C.

```
1919 Gallows Road, 10th floor
Vienna, VA 22182
U.S.A.
```

Phone:

+1 703 848 9027

Fax:

+1 703 848 2444

EMail:

info@webthreads.com

URL:

http://www.webthreads.com

Contact:

Gavin Sutcliffe

Last modified:

March 18, 1997.

7. Databases

7.1 c-tree Plus

Description:

Based on advanced B+tree (balanced) algorithm, c-tree Plus API handles all aspects of database I/O. Program single user or multi-user non-server applications royalty free and migrate existing c-tree Plus applications to the FairCom Server by recompiling.

Distribution media:

3.5" diskettes and CD-ROM.

Licensing policy:

c-tree Plus is licensed on a per programmer basis. Single-user, multi-user and multi-user non-server royalty-free distribution. Contact FairCom for possible restrictions: general purpose database and application development systems prohibited.

OS provisions:

3 MB hard drive space; 128 KB of RAM.

Documentation:

Printed manuals distributed with product; full online documentation with CD-ROM, available in English and Japanese.

Product support:

Three months of full technical support from purchase date. Unlimited technical support and product updates available thereafter with c-tree Plus maintenance program.

Available since:

October, 1995.

Countries with distribution:

World wide.

Price range:

US$895 includes source.

Vendor:

FairCom

```
4006 W. Broadway
```

```
Columbia, MO 65203-0100
U.S.A.
```

Phone:

+1 573 445 6833

Fax:

+1 573 445 9698

EMail:

Faircom@faircom.com

URL:

http://www.faircom.com

Contact:

Tamra Brown <tami@faircom.com>

Vendor:

FairCom Europe

```
Via Patrioti 6
I-24021 Albino
Italy
```

Phone:

+39 35 773 464

Fax:

+39 35 773 806

EMail:

Europe@faircom.com

Vendor:

FairCom Japan

```
Ikeda Bldg. #3 4F, 112-5
Komei-chou, Tsu-city MIE 514
Japan
```

Phone:

+81 059 229 7504

Fax:

+81 059 249 723

Vendor:

FairCom do Brasil Ltda.

Phone:

+55 11 3872 9802

Fax:

+55 11 3862 1752

EMail:

Brazil@faircom.com

Last modified:

August 7, 1997.

7.2 Empress

Description:

Empress for Linux is a multimedia RDBMS for members of the rapidly growing Linux developer community. Key components of the package include the powerful Empress RDBMS as well as dynamic SQL, Empress 4GL, Empress GUI Builder for rapidly developing graphical front-ends to Empress applications, a WWW HTML toolkit, and a grahical point and click interface to the Empress RDBMS development environment. A streamlined, single-user version of this product, Personal Empress for Linux, is available also.

Distribution media:

3.5" diskettes.

Licensing policy:

Free demos are available via FTP. Must contact vendor.

OS provisions:

Empress GUI Builder requires OSF Motif version 1.2.4 or 2.0. 16 MB RAM. 80 MB Disk Space. 486 CPU or better.

Documentation:

Printed documentation.

Product support:

Full technical support available, priced separately.

Extra features and add-ons:

Other features which are components of the package include shared libraries, shared memory, math library functions and a C language interface.

Available since:

December, 1995.

Countries with distribution:

World wide.

Price range:

Product is priced by number of concurrent users. Please contact vendor.

Vendor:

Empress Software Inc.

```
6401 Golden Triangle Drive
Greenbelt, MD 20770
U.S.A.

3100 Steeles Avenue East
Markham, ON L3R 8T3
Canada
```

Phone:

+1 301 220 1919 (USA), +1 905 513 8888 (Canada)

Fax:

+1 301 220 1919 (USA), +1 905 513 1668 (Canada)

EMail:

sales@empress.com

URL:

http://www.empress.com

Contact:

Dick Naedel

Last modified:

August 4, 1997.

7.3 Essentia

Description:

Database Engine.

Some Features: Remote databases, client/server, automatic consistency check, incremental backup, mirroring, shadowing, distributable database, journaling, versions, RISE, object oriented DBMS, implements relational model three tier client/server architecture, cooperative servers, language

TOC

independent user-configurable (English, Spanish, Portuguese).

Distribution media:

4/8 mm. DAT, 150/525 MB tape. Academic version available from:

- ftp://ftp.inter-soft.com
- http://www.inter-soft.com/html/products/essentia

Licensing policy:

Commercial. Free version for Linux available.

OS provisions:

10 MB disk space.

Documentation:

Available in PostScript and HTML.

Product support:

Contact essentia-info@inter-soft.com for more information.

Extra features and add-ons:

SQL Server, ODBC Interface for Windows, JDBC Interface, User servers.

Available since:

1993.

Countries with distribution:

Argentina, Brazil, Russia, Mexico, Venezuela and USA.

Price range:

Linux version for free (with some restrictions). Other prices available on request.

Vendor:

InterSoft Argentina S.A.

```
Córdoba 883 9th. Floor
Capital Federal (1054)
Argentina
```

Phone:

+54 1 318 8900

Fax:

+54 1 318 8997

EMail:

info@inter-soft.com

URL:

http://www.inter-soft.com

Last modified:

August 4, 1997.

7.4 FairCom Server

Description:

High performance, multi-threaded, transaction processing server. Features include: industrial quality transaction processing, including full commit and rollback; intermediate save points and complete logging; automatic log management; restart/disaster recovery; user passwords; access security and online administration; deadlock detection/resolution; read/write locks at the record/individual key level; more.

Distribution media:

3.5" diskettes and CD-ROM.

Licensing policy:

The FairCom Server is licensed on a per machine basis. Contact FairCom for specific licensing questions.

OS provisions:

2 MB of RAM.

Documentation:

Printed manuals distributed with product; full online documentation with CD-ROM, available in English and Japanese.

Product support:

Three months of full technical support from purchase date. Unlimited technical support and product updates available thereafter with Server maintenance program.

Available since:

October, 1995.

Countries with distribution:

World wide.

Price range:

US$445-$6795 depending upon platform and number of users. Special licensing and OEM agreements available.

Vendor:

FairCom

```
4006 W. Broadway
Columbia, MO 65203-0100
U.S.A.
```

Phone:

+1 573 445 6833

Fax:

+1 573 445 9698

EMail:

Faircom@faircom.com

URL:

http://www.faircom.com

Contact:

Tamra Brown <tami@faircom.com>

Vendor:

FairCom Europe


```
Via Patrioti 6
I-24021 Albino
Italy
```

Phone:

+39 35 773 464

Fax:

+39 35 773 806

EMail:

Europe@faircom.com

Vendor:

FairCom Japan


```
Ikeda Bldg. #3 4F, 112-5
Komei-chou, Tsu-city MIE 514
Japan
```

Phone:

+81 059 229 7504

Fax:

+81 059 249 723

Vendor:

FairCom do Brasil Ltda.

Phone:

+55 11 3872 9802

Fax:

+55 11 3862 1752

EMail:

Brazil@faircom.com

Last modified:

August 7, 1997.

7.5 INFORMIX-SE

Description:

INFORMIX-SE is an SQL-based database engine for small- to medium-range applications. It is the ideal solution for businesses that want the power of SQL without complex database administration requirements. With over a decade of proven reliability, INFORMIX-SE has a strong track record among Unix database servers.

INFORMIX-ESQL/C: Productivity, flexibility, portability, and standards are what programmers can achieve with the use of INFORMIX-ESQL for C. Developers can continue to use these familiar third-generation languages (3GLs) for developing applications and gain the advantage of using SQL to access data from those applications. Developers don't have to build their own database functions saving development time and effort.

Distribution media:

CD-ROM and Internet (FTP).

Licensing policy:

Free development license. Runtime Licenses must be purchased separately. Information on both is available at http://www.informix.com/informix/solutions/linux/lx.html.

OS provisions:

The database server process requires 1.2 MB to run, and can be installed with 6 MB of disk space.

Documentation:

Online manuals.

Product support:

Annual technical support and software maintenance contracts.

Extra features and add-ons:

TBD.

Available since:

July 22, 1998.

Countries with distribution:

World wide.

Price range:

To go along with your free development license, Informix offers attractively priced maintenance contracts through Informix-Assurance: http://www.informix.com/informix/services/csp/assurance/assurance.htm

Vendor:

Informix Software, Inc.

EMail:

sales@informix.com, http://www.informix.com/cgi-bin/contact.pl

URL:

http://www.informix.com

Last modified:

September 21, 1998.

7.6 Just Logic/SQL

Description:

Relational Database Management System

Distribution media:

3.5" diskettes.

Licensing policy:

Unlimited runtime included.

Documentation:

Printed manual, examples.

Product support:

EMail, fax or phone.

Extra features and add-ons:

- client-server option: lets run applications on several computers running Windows or Linux,

that access a central database on a Linux system.

- web-enabling option: passthrough between an HTTP Web server and a Just Logic/SQL database. No C or Perl required. SQL commands are embedded directly within HTML files. Compatible with Netscape, Apache, NCSA and all other CGI-compliant HTTP servers.

Available since:

1993.

Countries with distribution:

World wide.

Price range:

US$149-$395.

Vendor:

Just Logic Technologies

```
P.O. Box 63050, 40 Commerce Street
Nun's Island, Montreal, QC H3E 1V6
Canada
```

Phone:

+1 514 761 6887

Fax:

+1 514 642 6480

EMail:

sales@justlogic.com

URL:

http://www.justlogic.com

Contact:

Luc Vallieres

Last modified:

August 5, 1997.

7.7 KE Texpress

Description:

KE Texpress is a high-speed client server database engine that supports object-oriented, relational and free text data structures and operations. It is particularly suited to applications with large data sets, complex operations and large numbers of concurrent users. KE Texpress is used for a wide

variety WWW database applications. Vertical applications include collections management, library systems, vital statistics, archives, text retrieval and records management.

Distribution media:

Internet (FTP and WWW) and tape.

Licensing policy:

Commercial product licensed by number of concurrent users. A 30 day free trial is available at our web site.

OS provisions:

Linux ELF and a.out libraries. Requires about 30 MB disk space. Runs on over 20 other varieties of Unix and Windows NT.

Documentation:

Printed and HTML documentation is available.

Product support:

Annual technical support (EMail, fax and phone) and software maintenance contracts.

Extra features and add-ons:

- Texhtml WWW module - publish KE Texpress databases on the web
- Texql - SQL-like structured query language
- TexAPI - applications programming interface with client libraries for Windows (C, C++ and VB), Unix and Macintosh System 7
- TexODBC - ODBC drivers for Windows

KE Software has an extensive consulting service assisting clients to develop KE Texpress database applications.

Available since:

1984.

Countries with distribution:

USA, Canada, Australia, Hong Kong, Japan and Malaysia.

Price range:

US$2,000 to $100,000+.

Vendor:

KE Software Inc.

```
303-601 West Broadway
Vancouver, BC V5Z 4C2
Canada
```

Phone:

+1 604 877 1960

Fax:

+1 604 877 1961

EMail:

info@kesoftware.com

URL:

http://www.kesoftware.com

Contact:

Andrzej Kowalski

Last modified:

August 4, 1997.

7.8 Qddb

Description:

Qddb is fast, powerful and flexible database software that runs on Unix. Some of its features include: Tcl/Tk programming interface, easy to use, you can have a DB application completely up and, running in about 5 minutes, using nxqddb. CGI interface for quick and easy online databases and guestbooks. Fast, and powerful searching capability. Report generator. Barcharts and graphs. Mass mailings with EMail, letters and postcards.

Distribution media:

3.5" diskettes, tape and Internet (FTP and WWW).

Licensing policy:

GNU and Commercial versions available.

OS provisions:

GNU version comes with source code. Binary packages available for Linux, FreeBSD, and BSD/OS(BSDI) --- RPM and buildkit format.

Documentation:

Online documentation and PostScript files available. Printed manuals also available upon request.

Product support:

User and programmer support available along with upgrade contracts.

Extra features and add-ons:

Tcl/Tk programming interface, CGI interface, report generator, many useful free and commercial applications built with Qddb.

Available since:

1996.

Countries with distribution:

World wide.

Price range:

See http://www.hsdi.com/orders

Vendor:

Herrin Software Development, Inc.

```
41 South Highland Avenue
Prestonsburg, KY 41653
U.S.A.
```

Phone:

+1 606 886 8202

Fax:

+1 606 277 3239

EMail:

info@hsdi.com

URL:

http://www.hsdi.com

Contact:

Eric Herrin

Last modified:

November 4, 1997.

7.9 Raima Database Manager++

Description:

Low-level high performance database engine with C API and C++ class library, for embedding in applications. This database is used in thousands of leading commercial applications. Includes source for C++ class library that encapsulates database navigation and object storage and retrieval into C++ classes, adding an object-oriented interface. Supports multiple database models, including relational, network model, and combined.

Distribution media:

3.5" diskettes and tape.

Licensing policy:

Pay for development license, distribute runtime copies freely.

Documentation:

Extensive documentation available from Raima Corporation, dealing with all aspects of database, C API and C++ class library.

Product support:

Available from Raima Corporation on annual basis, raining available.

Extra features and add-ons:

Windows GUI Report Writer.

Available since:

1984, originally called db_VISTA.

Countries with distribution:

Direct in the USA, use distributors internationally - Germany, England, France, Italy, Spain, Netherlands, Russia, Estonia, Argentina, Columbia, Singapore, Malaysia, Australia, Finland, Taiwan, other countries.

Price range:

RDM++ Database Module is US$995 for single user, US$2,395 for multi-user. System is US$1,395 for single user, US$3,395 for multi-user (System includes RDM++ database, QUERY SQL query tool and REVISE database restructuring tool). All versions available with source code for extra cost.

Depends on machine class; lowest is US$3,995 without system utilities; US$6,195 with system utilities.

Vendor:

Raima Corporation

```
4800 Columbia Center
701 5th Avenue
Seattle, WA 98104
```

Phone:

+1 800 327 2462, +1 206 515 9477

Fax:

+1 206 748 5200

EMail:

sales@raima.com

URL:

http://www.raima.com

Contact:

Dave Morse

Last modified:

March 18, 1997.

7.10 Empress Embedded RDBMS

Description:

The Empress Embedded RDBMS is an embedded systems developer's toolkit. The RDBMS engine is fast, compact and easy-to-embed. Additionally, this package possesses superior bulk object handling capabilities. Embedded Empress RDBMS is Internet ready with a JDBC/ODBC bridge available.

Distribution media:

3.5" diskettes.

Licensing policy:

Please contact vendor for evaluations copies.

OS provisions:

32 MB of RAM, 60 MB disk disk space.

Documentation:

Printed documentation.

Product support:

Full technical support available, priced separately.

Extra features and add-ons:

Extra features included with the toolkit are Empress Report Writer and enhanced Internet capabilities (ability to use Java applets, etc.) via the HTML toolkit.

Available since:

January, 1997.

Countries with distribution:

World wide.

Price range:

Starting at US$1000 for PCs, US$4000 for typical workstations, US$16,000 for mid-range servers.

Vendor:

Empress Software Inc.

```
6401 Golden Triangle Drive
Greenbelt, MD 20770
U.S.A.

3100 Steeles Avenue East
Markham, ON L3R 8T3
Canada
```

Phone:

+1 301 220 1919 (USA), +1 905 513 8888 (Canada)

Fax:

+1 301 220 1919 (USA), +1 905 513 1668 (Canada)

EMail:

sales@empress.com

URL:

http://www.empress.com

Contact:

Dick Naedel

Last modified:

August 4, 1997.

7.11 SOLID Server

Description:

SOLID Server is a database engine for new applications and products. It is extremely easy to set up, and has a small footprint. SOLID Server is standards-compliant and full of power. It is perfectly suited for distributed use in countless copies. Its maintenance is care-free, requiring minimal or no administrator attention.

Distribution media:

3.5" diskettes and Internet (FTP and WWW).

Licensing policy:

Copy-protected.

OS provisions:

At least 2 MB RAM, recommended are 8 MB; harddisk about 3-4 MB.

Documentation:

English manuals and online help. WWW.

Product support:

Available.

Available since:

1994.

Countries with distribution:

World wide.

Price range:

SOLID Desktop US$99, SOLID Server US$199/seat, SOLID Web Engine US$495.

Vendor:

Solid Information Technology Ltd

```
Huovitie 3
FIN-00400 Helsinki
Finland
```

Phone:

+358 9 477 4730

Fax:

+358 9 477 47 390

EMail:

info@solidtech.com

URL:

http://www.solidtech.com

Last modified:

August 4, 1997.

7.12 Velocis Database Server

Description:

Velocis Database Server is designed for database application developers who are looking for a high performance client/server or web database engine. Velocis is a scaleable SQL client/server database engine that provides a rich set of architectural choices and APIs including ANSI SQL, SQL C-API, low-level C-API, C++ class libraries, and support for custom APIs. Unlike typical

TOC

relational client/server database products, Velocis supports both relational and pointer-based network model databases in any combination as well as processing on either side of the client/server equation. The choices of multiple operating platforms, APIs, processing locality (client or server), and database model can be mixed and combined to satisfy the performance requirements of virtually any application.

Distribution media:

3.5" diskettes and tape.

Licensing policy:

Pay for development license. Runtimes are requires to distribute applications.

Documentation:

Extensive documentation available from Raima Corporation, dealing with all aspects of database, C API and C++ class library.

Product support:

Available from Raima Corporation on annual basis, raining available.

Extra features and add-ons:

Windows GUI Report Writer, Raima Object Manager.

Available since:

1993, originally called Raima Database Server.

Countries with distribution:

World wide.

Price range:

1-8 users: US$1,995, 1-25 users: US$3,995, unlimited: US$8,995.

Vendor:

Raima Corporation

```
4800 Columbia Center
701 5th Avenue
Seattle, WA 98104
```

Phone:

+1 800 327 2462, +1 206 515 9477

Fax:

+1 206 748 5200

EMail:

sales@raima.com

URL:

http://www.raima.com

Contact:

Dave Morse

Last modified:

November 4, 1997.

7.13 Yard SQL

Description:

The YARD company offers the following products:

- YARD-SQL -- Relational SQL client/server database with compliance to X/Open XPG4 and ANSI SQL 92 including network support for accessing remote databases
- YARD-ESQLC -- Embedded SQL for C
- YARD-ODBC -- ODBC interface for MS Windows clients
- YARD-X -- Motif client for database access (No development tool)
- YARD-JDBC -- Java Interface

Distribution media:

CD-ROM and Internet (FTP).

Licensing policy:

License number and activation key for each product and installation with user dependent licenses. A Private Edition (limited to 1 user and 5 MB database) for non commercial private use and for evaluation is available at ftp://ftp.yard.de.

OS provisions:

Linux ELF version; RAM usage: 1 MB (minimum) for shared memory, 500 KB per user. Disk usage: YARD-SQL 10 MB, YARD-ESQLC 1 MB, YARD-ODBC 1.5 MB and YARD-X 5 MB.

Documentation:

Printed 600 pages reference and users guide in German. English documentation is available only as PostScript file.

Product support:

Upon request.

Extra features and add-ons:

All products also available for other widely distributed Unix systems (e.g. SCO Unix, SPARC Solaris, IBM, HP, SGI).

Available since:

January, 1994.

Countries with distribution:

Contact YARD Software GmbH for information about resellers.

Price range:

Upon request.

Vendor:

YARD Software GmbH

```
Wikingerstr. 18
D-51107 Köln
Germany
```

Fax:

+49 221 98664 99

EMail:

yard@yard.de

URL:

http://www.yard.de

FTP:

ftp://ftp.yard.de

Contact:

Thomas Schonhoven <thomass@yard.de>

Last modified:

August 4, 1997.

---

8. Data Visualisation and CAD

8.1 IDL (Interactive Data Language)

Description:

IDL is powerful software for data analysis, visualization, and application development. IDL's features include flexible I/O, object-oriented programming, 2D plotting, 3D graphics, volume rendering, image processing, mathematics, statistics, a cross-platform GUI toolkit, plus a high-level, array-oriented programming language. Use IDL for visual data analysis, rapid prototyping, or application development. IDL programs, including their graphical user interfaces, are portable across Linux, Windows 3.11, Windows 95, Windows NT, Mac, PowerMac, Unix and VMS.

The IDL-Student Version is a functionality-limited version of IDL 5.0. It is designed to compliment IDL-teaching lab environments and give students access to software specifically designed for technical curriculums including physics, astronomy, engineering, earth sciences, medical sciences and computer science. For more information or to place an order, visit our web site.

Distribution media:

CD-ROM and Internet (FTP).

Licensing policy:

Free demo CD-ROM available or download via FTP.

OS provisions:

IDL 5.0 for Linux is built using the Linux 2.0.18 kernel and the Red Hat version 4.0 Linux distribution.

Documentation:

3000+ pages of documentation on paper and hypertext online help.

Product support:

Customer support service, maintenance/update service, training and consulting are available. In addition, a Usenet group is devoted to IDL.

Extra features and add-ons:

ENVI: remote sensing image processing application, DataMiner: ODBC database access, HDF, CDF, netCDF support, animation, volume slicer, high resolution mapping.

Available since:

May, 1997.

Countries with distribution:

Ask vendor.

Price range:

Contact vendor/distributor.

Vendor:

Research Systems, Inc.

```
2995 Wilderness Place
Boulder, CO 80301
U.S.A.
```

Phone:

+1 303 786 9900

Fax:

+1 303 786 9909

EMail:

info@rsinc.com

URL:

http://www.rsinc.com

FTP:

ftp://ftp.rsinc.com

Last modified:

November 5, 1997.

8.2 Megahedron

Description:

A flexible and powerful 3D graphics engine controlled by a high-level interpreted language called SMPL. With it, you can learn about 3D, write your own interactive simulations, animations, and ray-tracings. You can even write your own shaders, customize procedural objects and motions, and use up to 64 computers at once for net rendering. Includes executables for Windows 95, Windows NT x86 and Alpha, SGI Irix and Linux.

Distribution media:

CD-ROM, ISO-9660 and Rock Ridge.

Licensing policy:

Very flexible. The owner is allowed to run as many copies as they can, on any platforms. Dem version, full documentation and sample images and scripts available on our web site.

OS provisions:

Supports kernel 1.2.8 or 1.3.15. Non-ELF, a.out executable (QMAGIC). 16 MB of RAM minimum recommended. 7 MB free disk space recommended.

Documentation:

In HTML on CDROM.

Product support:

Private news server, EMail. Patches on web site.

Extra features and add-ons:

Everything included on one CD-ROM.

Available since:

August, 1996.

Countries with distribution:

World wide, direct and through dealers.

Price range:

US$99

Vendor:

Syndesis Corporation

```
235 South Main Street
Jefferson, WI 53549
U.S.A.
```

Phone:

+1 414 674 5200

Fax:

+1 414 674 6363

EMail:

syndesis@threedee.com

URL:

http://www.threedee.com

Contact:

John Foust <jfoust@threedee.com>

Last modified:

November 3, 1997.

8.3 Tecplot 7.0

Description:

Tecplot is interactive data visualization software for XY plotting, 2D and 3D mesh, contour, vector, scatter, and shade plots. For more information, see the Amtec Engineering web site.

Distribution media:

CD-ROM and Internet (FTP).

Licensing policy:

Either personal licenses (node locked) or network licenses. Evaluation copies available on CD-ROM or by anonymous FTP.

OS provisions:

12 MB minimum, Linux 1.2 or newer (available in both statically linked and dynamically linked versions). Use the statically linked version if your system does not have Motif 1.2 or newer installed. 32 MB RAM is recommended.

Documentation:

Printed User's and Reference Manuals, full online help. English.

Product support:

Update and technical support included with license for first three months. Extra charge (ask for EUSS) after that. Training courses available.

Available since:

V7 since September 1996, V6 since August 1993.

Countries with distribution:

Many, see Amtec web site.

Price range:

US$1795 for personal license, US$2395 for single-user network license, US$1795 for each additional user. All prices the USA and Canada only.

Vendor:

Amtec Engineering, Inc

```
PO Box 3633
Bellevue, WA 98009-3633
U.S.A.
```

Phone:

+1 800 676 7568, +1 425 827 3304

Fax:

+1 425 827 3989

EMail:

tecplot@amtec.com

URL:

http://www.amtec.com

Last modified:

August 4, 1997.

8.4 VariCAD

Description:

Mechanical CAD system available for Linux and Windows 95 and Windows NT.

The system is equipped with all basic resources necessary both for constructors' and designers' work in engineering. It includes functions for 3D modeling, 2D drawing, editing, transformations, working with user defined objects, block, group and symbol creating capabilities. The whole system is very customizable and widely open.

Distribution media:

CD-ROM, Internet (WWW).

Licensing policy:

Free demo version available on our web site.

OS provisions:

Kernel version 2.0 or higher, XFree86 version 3.1+, RAM 16 MB, harddisk 20 MB, libc 5.4.

Documentation:

Online help, English.

Product support:

New version about every three months.

Extra features and add-ons:

New versions during first year free.

Available since:

1991.

Countries with distribution:

World wide.

Price range:

US$dollar;199-$499.

Vendor:

VariCAD

```
931 Greenbriar Avenue
Ottawa, ON K2C 0J8
Canada

P. O. Box 38
Liberec 2
460 02
Czech Republic
```

Phone:

+1 613 723 5319

Fax:

+1 613 723 5319

EMail:

mail@varicad.com

URL:

http://www.varicad.com

Contact:

Petr Placek

Last modified:

August 15, 1998.

8.5 VARKON

Description:

A high level development tool for CAD and Product modeling applications. Interactive parametric modelling in 2D and 3D with object oriented database. High level CAD and modelling language MBS included.

Distribution media:

Internet (WWW).

Licensing policy:

Free binary for Linux.

OS provisions:

Requires X but not Motif. Needs 3 MB of harddisk for basic installation and 3MB of RAM to run.

Documentation:

400 pages of documentation included in the free version for Linux.

Product support:

Service with continous updates and free consultation available for US $75/month.

Extra features and add-ons:

Additional plotterdrivers available free of charge.

Available since:

July, 1996.

Countries with distribution:

World wide.

Price range:

Linux version is free of charge.

Vendor:

Microform AB

```
Henningsholmsgatan 4
S-703 69 Orebro
Sweden
```

Phone:

+46 19314932

Fax:

+46 19314969

EMail:

info@microform.se

URL:

http://www.microform.se

Contact:

Johan Kjellander

Last modified:

September 30, 1997.

8.6 XVScan

Description:

Image scanning and manipulation software for HP ScanJet scanners.

Distribution media:

Internet (EMail and FTP). 4mm DDS2 DAT tape or 3.5 diskettes (additional US$15 in the USA, US$25 international).

Licensing policy:

XVScan is based on XV and is not available in demo version due to licensing restrictions. Distributed with full source code.

OS provisions:

Linux, tested with 1.2.x and 2.0.x, requires a recent generic SCSI driver support to be built in (no earlier than 1.1.79). Motif is not required. Any XFree version (X11R5, X11R6).

Documentation:

Online, WWW, English.

Product support:

Updates free for first year.

Extra features and add-ons:

Also available for HP-UX, BSD/OS 2.2 and 3.0, SunOS 4.1.x, Solaris 2.x, and FreeBSD.

Available since:

May, 1995.

Countries with distribution:

World wide.

Price range:

US$50 for FTP or EMail shipping. Additional US$15 for media in the United States, US$15, internationally. Mastercard, Visa and Discover, and American Express Credit Cards accepted.

Vendor:

tummy.com, ltd.

```
3506 Stratton Drive
Fort Collins, CO 80525-2722
```

U.S.A.

Phone:

+1 970 223 8215

Fax:

+1 408 490 2728

EMail:

xvscan@tummy.com

URL:

http://www.tummy.com/xvscan

Contact:

Sean Reifschneider <jafo@tummy.com> or Evelyn Mitchell <efm@tummy.com>

Last modified:

August 4, 1997.

---

9. Development Tools

9.1 ACUCOBOL-GT

Description:

ACUCOBOL-GT is an ANSI-85 COBOL development system with compiler, runtime, debugger, support utilities and documentation. It offers single source hardware independence, GUI COBOL, client/server capabilities, multithreading support, and data source flexibility. Additionally, ACUCOBOL-GT is year 2000 compliant.

Distribution media:

currently 3,5" diskettes (A CD-ROM is planned for 2Q/98).

Licensing policy:

Runtime license for each installed application including license fees for all products free of charge 30 day evaluation copy available on our web site.

OS provisions:

Linux a.out and ELF. Tested on Red Hat Linux 4.1 with kernel version 2.0.30.

Documentation:

Printed dccumentation for all products (english), online documentation for web evaluation copy.

Product support:

Update service (annual fee is 20% of product list price), training courses for core products (DM 600/day at scheduled dates), technical support included in update service.

Available since:

February, 1995.

Countries with distribution:

World wide.

Price range:

US$1.500 for single user development system (Linux), US$300 for each additional developer, runtime fees on request (price is depending on contract form).

Vendor (Germany):

Acucobol Deutschland GmbH

```
Otto-Hahn-Str. 9
```

D-61381 Friedrichsdorf
Germany

Phone:

+49 6175 93310

Fax:

+49 6175 1429

EMail:

aschmidt@acucobol.de

Contact:

Annette Schmidt

Vendor (USA):

Acucobol, Inc.

7950 Silverton Avenue, Suite #201
San Diego, CA 92126
U.S.A.

Phone:

+1 800 COBOL 85, +1 619 6897220

Fax:

+1 619 566 3071

EMail:

info@acucobol.com

URL:

http://www.acucobol.com

Contact:

Jeff Freedman

Last modified:

November 1, 1997.

9.2 Amzi! Prolog & Logic Server

Description:

Amzi! enables the easy integration of intelligent components with conventional applications allowing you to add logic-bases that give advice, configure and tune systems, diagnose problems,

apply business rules, monitor processes and parse documents. Your applications access a logic-base of rules just as a database server accesses records. The rules are expressed in Prolog which has powerful, built-in search and pattern matching capabilities. The Amzi! Logic Server is encapsulated as a C++ Class and C API Interface. You can add your own Prolog functions in C/C++. Includes: compiler, listener, debugger, linker, EXE, generator, call-in/call-out Logic Server API, full documentation, comprehensive Prolog tutorial and lots of sample code.

Distribution media:

Internet.

Licensing policy:

The Professional Edition includes an unlimited, royalty-free license. The Personal Edition is limited to distributing applications for non-commercial use only.

OS provisions:

About 350 KB of RAM for typical small application, 3-4 MB disk.

Documentation:

HTML format, includes full Prolog tutorial.

Product support:

Subscription Plus service provides automatic updates for a full year, $198. Custom development services available.

Linux Support:

No Linux system included. For Amzi! software, free tech support for registered users by phone, fax and EMail.

Available since:

November 3, 1995.

Countries with distribution:

World wide.

Price range:

Personal (Shareware), $49. Professional, $298.

Vendor:

Amzi! inc.

```
40 Samuel Prescott Drive
Stow, MA 01775
U.S.A.
```

Phone:

+1 508 897 7332

Fax:

+1 508 897 2784

EMail:

info@amzi.com

URL:

http://www.amzi.com

Contact:

Mary Kroening

Last modified:

August 4, 1997.

9.3 Basmark QuickBASIC

Description:

The Basmark QuickBASIC Compiler is a multi-user IBM-PC BASICA, MBASIC and Microsoft QuickBASIC Compiler designed to provide performance and consistency across a variety of machines (e.g. i386 and i486, Pentium, SPARC, RS/6000, HP PA-RISC) under Unix, AIX, SunOS, Linux, HP-UX and Xenix.

Distribution media:

Internet.

Licensing policy:

Per machine, no run-time restrictions.

OS provisions:

GNU GAS and LD must be installed.

Documentation:

400+ page manual, release and installation notes available in hardcopy form for an additional US$50 (plus shipping).

Product support:

Updates available for US$139 less shipping. Maintenance contracts available.

Extra features and add-ons:

C-ISAM (Informix Inc.) Interface in C source code form. Cost is US$35.

Available since:

December, 1993.

Countries with distribution:

World wide.

Price range:

US$195.

Vendor:

Basmark Corporation

```
P.O. Box 40450
Cleveland, OH 44140
U.S.A.
```

Phone:

+1 216 871 8855

Fax:

+1 216 871 9011

EMail:

jgo@basmark.com (for orders)

URL:

http://www.basmark.com

Contact:

Joseph O'Toole (for orders)

Last modified:

November 4, 1997.

9.4 Critical Mass CM3

Description:

Systems development compiler and runtime.

Distribution media:

Internet (WWW).

Licensing policy:

Commercial with library source, evaluation licenses available for download at http://www.cmass.com/cm3.

OS provisions:

Linux/ELF.

Documentation:

Online.

Product support:

Optional.

Extra features and add-ons:

See http://www.cmass.com/cm3

Available since:

May, 1996.

Countries with distribution:

USA.

Price range:

US$479/seat.

Vendor:

Critical Mass, Inc.

```
Critical Mass, Inc.
225R Concord Avenue
Cambridge, MA 02138
U.S.A.
```

Phone:

+1 617 354 6277

Fax:

+1 617 354 5027

EMail:

info@cmass.com

URL:

http://www.cmass.com

Last modified:

May 19, 1997.

9.5 Dynace

Description:

Dynace (pronounced like ``dynasty'' without the ``t'') is a preprocessor, include files and a library which extends the C language with advanced object oriented capabilities, automatic garbage

collection and multiple threads. Dynace is designed to solve many of the problems associated C++ while being easier to learn and containing more flexable object oriented facilities. Dynac able to add facilities previously only available in languages such as Smalltalk and CLOS with out all the overhead normally associated with those environments.

Distribution media:

Internet (WWW); including full C source code.

Licensing policy:

One license per programmer; Applications are royalty free. Full system is on http://www.edge.net/algorithms. Free for non-commercial use.

OS provisions:

Any Linux.

Documentation:

310 pages; in Postscript, HP PCL, or TeX DVI.

Product support:

Dynace-support@edge.net

Extra features and add-ons:

Complete Windows development system.

Available since:

December, 1993.

Price range:

US$599.

Vendor:

Algorithms Corporation

```
3020 Liberty Hills Drive
Franklin, TN 37067
U.S.A.
```

Phone:

+1 800 566 8991, +1 615 791 1636

Fax:

+1 615 791 7736

EMail:

blake@edge.net

URL:

http://www.edge.net/algorithms

Contact:

Blake McBride

Last modified:

August 4, 1997.

9.6 Absoft Fortran 77

Description:

VAX/VMS compatible ANSI Fortran 77 compiler and debugger.

Distribution media:

3.5" diskettes.

Licensing policy:

Single user license; multi-user packages also available.

OS provisions:

Linux 1.2.13.

Documentation:

300+ pages hard copy documentation.

Product support:

No-charge technical support.

Available since:

March, 1996.

Countries with distribution:

World wide.

Price range:

US$525-$2,000.

Vendor:

Absoft Corporation

```
2781 Bond Street
Rochester Hills, MI 48309
U.S.A.
```

Phone:

+1 248 853 0050

Fax:

+1 248 853 0108

EMail:

sales@absoft.com

URL:

http://www.absoft.com

Contact:

Wood Lotz

Last modified:

August 7, 1997.

9.7 Finesse

Description:

OSF/Motif GUI for shell scripts.

Distribution media:

Internet (FTP).

Licensing policy:

Finesse is a fully commercial product for other Unixes where nodelocked licenses are possible. The Linux version is freely available via FTP. Floppy medium is DM 90.

OS provisions:

ELF Libraries. X11R6. No Motif required.

Documentation:

PostScript File.

Product support:

No support for free version.

Available since:

May, 1995.

Countries with distribution:

World wide.

Price range:

Free demo. Commercial version with support on request.

Vendor:

science + computing GmbH

```
Hagellocher Weg 71
D-72070 Tübingen
Germany
```

Phone:

+49 7071 9457 0

Fax:

+49 7071 9457 27

EMail:

info@science-computing.de

URL:

http://www.science-computing.de

FTP:

ftp://ftp.science-computing.de/pub/finesse.

Contact:

Olaf Flebbe

Last modified:

February 25, 1997.

9.8 ISE Eiffel

Description:

ISE Eiffel is a seamless object-oriented development environment. ISE Eiffel provides an integrated solution for software developers through pure object-oriented methods, from analysis and design through code generation, maintenance, and reverse engineering.

The components of ISE Eiffel include EiffelBench, EiffelBase, EiffelBuild, EiffelVision, EiffelLex, EiffelParse, EiffelNet, EiffelStore, ObjEdit, EiffelCase, EiffelMath, EiffelWeb, DLE (Dynanic Linking in Eiffel) and SCOOP (Distribution/Concurrency mechanism).

Distribution media:

CD-ROM and Internet (FTP and WWW)

Licensing policy:

No run-time fees are required for products, commercial or otherwise, developed with ISE's technology. In the case of commercial products we simply require the product and its

TOC

documentation to acknowledge ISE Eiffel clearly. The copyright holder must acknowledge the product's use/incorporation of ISE Eiffel by (a) Featuring "ISE Eiffel" prominently in product documentation, and (b) Featuring an "ISE Eiffel" software window or pop-up message clearly visible during the copyrighted product's installation process.

OS provisions:

Linux ELF or a.out.

Documentation:

Printed documentation. Additional online help.

Product support:

Update service, maintenance, training available.

Available since:

August, 1994.

Countries with distribution:

World wide.

Price range:

Starting at US$99.

Vendor:

Interactive Software Engineering, Inc.

```
ISE Building
270 Storke Road, 2nd Floor
Goleta, CA 93117
U.S.A.
```

Phone:

+1 805 685 1006

Fax:

+1 805 685 6869

EMail:

info@eiffel.com

URL:

http://www.eiffel.com

Last modified:

October 3, 1997.

9.9 EiffelBench

TOC

Description:

Object-oriented CASE Workbench.

Distribution media:

CD-ROM and Internet (FTP and WWW).

Licensing policy:

No run-time fees are required for products, commercial or otherwise, developed with ISE's technology. In the case of commercial products we simply require the product and its documentation to acknowledge ISE Eiffel clearly. The copyright holder must acknowledge the product's use/incorporation of ISE Eiffel by (a) Featuring "ISE Eiffel" prominently in product documentation, and (b) Featuring an "ISE Eiffel" software window or pop-up message clearly visible during the copyrighted product's installation process.

OS provisions:

Linux ELF or a.out.

Documentation:

Eiffel: The Environment. Language References.

Product support:

Update service, maintenance, training available.

Available since:

August, 1994.

Countries with distribution:

World wide.

Price range:

Starting at US$99.

Vendor:

Interactive Software Engineering, Inc.

```
ISE Building
270 Storke Road, 2nd Floor
Goleta, CA 93117
U.S.A.
```

Phone:

+1 805 685 1006

Fax:

+1 805 685 6869

EMail:

info@eiffel.com

URL:

http://www.eiffel.com

Last modified:

October 3, 1997.

9.10 C-Forge IDE

Description:

C-Forge is a highly scalable, multi-user integrated development environment that provides full project management and complete edit, compile and debug cycle support. Includes its own full-featured editor, revision control tool and diff tool. Drag and drop is enabled throughout the environment. Allows easy configuration and use of external tools. Currently supports C/C++ -- more languages on the way.

Distribution media:

Internet (FTP).

Licensing policy:

Single user and floating multi user licenses available. Free demo available at our web site.

OS provisions:

Works with Linux kernels 2.0.x and 2.1.x. Installations available in tar.gz and RPM formats.

Documentation:

Online.

Product support:

By EMail.

Extra features and add-ons:

We add the features you want to future versions of C-Forge.

Available since:

February, 1998.

Countries with distribution:

World wide.

Price range:

US$30 for single user version. Contact for corporate pricing policies. Educational discounts available.

Vendor:

Code Forge Inc.

EMail:

consult@codeforge.com

URL:

http://www.codeforge.com

Contact:

Yuri Mironoff

Last modified:

February 8, 1998.

9.11 IdeaFix

Description:

The InterSoft Development Environment for Applications in Unix, or IdeaFix, is a set of programming tools and utilities designed to provide an integrated environment for programmers as well as end-users. The aim of IdeaFix is to maximize performance and productivity for both.

For the end-user, IdeaFix offers a user-friendly interface and an online help system. IdeaFix provides the developer with tools to simplify development of such design criteria as relational databases and modular structured programming.

Distribution media:

4/8 mm. DAT, 150/525 MB tape. Academic version available from:

- ftp://ftp.inter-soft.com/pub/ideafix
- http://www.inter-soft.com/html/products/ideafix

Licensing policy:

Commercial.

OS provisions:

20 MB disk space, and GNU C/C++ compiler.

Documentation:

Available in PostScript.

Product support:

Contact ideafix-info@inter-soft.com for more information.

Extra features and add-ons:

SQL server, Dali - Development Environment, Cracker for Windows.

Available since:

1986.

Countries with distribution:

Argentina, Brazil, Russia, Mexico, Venezuela and USA.

Price range:

Linux version for free (with some restrictions). Other prices available on request.

Vendor:

InterSoft Argentina S.A.

```
Córdoba 883 9th. Floor
Capital Federal (1054)
Argentina
```

Phone:

+54 1 318 8900

Fax:

+54 1 318 8997

EMail:

info@inter-soft.com

URL:

http://www.inter-soft.com

Last modified:

August 4, 1997.

9.12 j-tree

Description:

Harness the power of Java clients while maintaining a legacy database on wide variety of O/S. Java API with c-tree Plus' ISAM functionality gives Java functionality through native methods/RMI. j-tree utilizes the power/flexibility of FairCom's full line of database Servers.

Distribution media:

3.5" diskettes.

Licensing policy:

A licensed copy of c-tree Plus, thereafter royalty free.

OS provisions:

2 MB of RAM.

Documentation:

Full online documentation with CD-ROM, available in English and Japanese.

Product support:

Three months of full technical support from purchase date.

Available since:

August, 1997.

Countries with distribution:

World wide.

Price range:

Upon request.

Vendor:

FairCom

```
4006 W. Broadway
Columbia, MO 65203-0100
U.S.A.
```

Phone:

+1 573 445 6833

Fax:

+1 573 445 9698

EMail:

Faircom@faircom.com

URL:

http://www.faircom.com

Contact:

Tamra Brown <tami@faircom.com>

Vendor:

FairCom Europe

```
Via Patrioti 6
```

```
I-24021 Albino
Italy
```

Phone:

+39 35 773 464

Fax:

+39 35 773 806

EMail:

Europe@faircom.com

Vendor:

FairCom Japan

```
Ikeda Bldg. #3 4F, 112-5
Komei-chou, Tsu-city MIE 514
Japan
```

Phone:

+81 059 229 7504

Fax:

+81 059 249 723

Vendor:

FairCom do Brasil Ltda.

Phone:

+55 11 3872 9802

Fax:

+55 11 3862 1752

EMail:

Brazil@faircom.com

Last modified:

August 9, 1997.

9.13 KAI C++

Description:

The KAI C++ compiler provides conformance to the latest draft standard, high performance, low abstraction penalty, identical syntax and libraries on all supported platforms and superior customer

support. No other compiler is as close to the draft standard.

Distribution media:

Internet (FTP and WWW).

Licensing policy:

Commercial. 30 day free trial available at http://www.kai.com/kcc_howto.shtml

OS provisions:

Red Hat Linux 2.1 or later. Other Software: gcc 2.7.2.1. Disk space: 15 MB.

Documentation:

Complete online documentation is supplied.

Product support:

No-charge technical support. Annual Support service provides automatic updates for a full year, US$79.

Available since:

May, 1997.

Countries with distribution:

World wide.

Price range:

US$395 single processor; US$545 multiprocessor.

Vendor:

Kuck & Associates, Inc.

```
1906 Fox Drive
Champaign, IL  61820-7345
U.S.A.
```

Phone:

+1 217 356 2288

Fax:

+1 217 356 5199

EMail:

kai@kai.com

URL:

http://www.kai.com/kcc_howto.shtml

Contact:

Bruce Leasure <bruce@kai.com>

Last modified:

July 14, 1997.

9.14 Khoros Pro 2.1

Description:

Khoros Pro 2.2 is a software development environment with extensive image processing, software development, and data visualization capabilities.

Distribution media:

CD-ROM.

Licensing policy:

Single user license comes with Khoros Pro CD-ROM. For software developers, independent software developer licenses are available.

OS provisions:

Linux 2.0.18, gcc 2.7.2, Fortran compiler f2c 19951025+ AT&T Bell Labs.

Documentation:

Printed documentation; Installation Guide and User's Guide come with the CD-ROM. Five volume Developer's Manual set available from KRI.

Product support:

Training in Software Development and Digital Image Processing scheduled through 1998. See http://www.khoral.com/training/training.html.

Maintenance, technical and engineering support may be purchased from KRI on a negotiated basis.

Available since:

August, 1996.

Countries with distribution:

World wide. Addison Wesley selling to educational market.

Price range:

Single user Khoros Pro 2.2 is US$549. Independent software developer licenses start at US$5,000 and go up depending on platforms and organization structure. Royalty agreements can be negotiated.

Vendor:

Khoral Research Inc.

6001 Indian School NE Suite 200
Albuquerque, NM 87110
U.S.A.

Phone:

+1 505 837 6500

Fax:

+1 505 881 3842

EMail:

info@khoral.com

URL:

http://www.khoral.com

FTP:

ftp://ftp.khoral.com

Contact:

Annie MacFarlane <annie@khoral.com>

Last modified:

August 4, 1997.

9.15 MetaCard

Description:

MetaCard is a GUI development and multimedia authoring tool compatible with Apple Corporation's HyperCard. Anyone can use MetaCard to build GUI applications and hypermedia documents using a powerful, direct manipulation editor and an easy-to-learn scripting language.

MetaCard goes beyond HyperCard by including support for color controls and images, vector graphics, scrollbars, and dialog boxes. MetaCard's scripting language has support for arrays, custom (user defined) object properties, and is based on high-perforance "virtual compiler" technology. Stacks developed with MetaCard are portable among all popular Unix platforms and Windows 95/NT and can be distributed with without licensing fees or royalties.

Distribution media:

3.5" diskettes and Internet (FTP and EMail).

Licensing policy:

MetaCard can be licensed to a single, named individual, but can be used on any machine or combination of machines by that user. Multiple user packages are also available.

The save-disabled distribution is available via anonymous FTP from

ftp://ftp.metacard.com/MetaCard and ftp://ftp.uu.net/vendor/MetaCard.

OS provisions:

The Linux engine is built on a 1.2.13 ELF system. Minumum 8 MB RAM for runtime, 16MB RAM for development. Minimum 640x480x8 screen resolution for runtime, 800x600x8 for development. Development system requires about 5MB disk space. GUI is Motif-compliant, but does not require Motif libraries.

Documentation:

Complete online documentation is supplied. Printed documentation is an extra-cost option.

Product support:

Free EMail technical support, phone support available at extra cost.

Extra features and add-ons:

A library version of MetaCard that can be linked directly to C programs (Embedded MetaCard) is available at extra cost.

Available since:

June, 1992.

Countries with distribution:

World wide.

Price range:

US$995 for single-user all-platform development license.

Vendor:

MetaCard Corporation

```
4710 Shoup pl.
Boulder, CO 80303
U.S.A.
```

Phone:

+1 303 447 3936

Fax:

+1 303 499 9855

EMail:

info@metacard.com

URL:

http://www.metacard.com

Last modified:

August 4, 1997.

9.16 ObjectManual Release 3.0

Description:

Automated documentation generator from C++ programs. Generates doucmentation in various formats including HTML, RTF, MIF along with Java classes.

Distribution media:

Internet (FTP and WWW).

Licensing policy:

Single, Multiple, Site and Floating Licenses.

OS provisions:

Linux ELF.

Documentation:

Provided in PostScript along with the distribution.

Product support:

Free for 30 days. Then a contract based on six month to a one year cycle.

Extra features and add-ons:

New release are released every quarter.

Available since:

1995.

Countries with distribution:

USA.

Price range:

US$2295 for most flavours of Unix. Free of charge for Linux users that work in a non-commercial environment.

Vendor:

ObjectSoftware, Inc.

```
3519 Misty Meadow Drive
Dallas, TX 75287
U.S.A.
```

Phone:

+1 214 373 2021

Fax:

+1 972 662 0756

EMail:

objsoft@netcom.com

URL:

http://www.obsoft.com

Contact:

Bobby Sardana

Last modified:

September 30, 1997.

9.17 Critical Mass Reactor

Description:

Distributed application development environment.

Distribution media:

CD-ROM.

Licensing policy:

Commercial with library source, evaluation licenses available for purchase.

OS provisions:

Linux/ELF.

Documentation:

Extensive documentation included (printed and online)

Product support:

Optional.

Extra features and add-ons:

See http://www.cmass.com/reactor/overview/index.html#reactor_features

Available since:

May, 1996.

Countries with distribution:

USA.

Price range:

US$479-$3500/seat. Discounts are available for non-commercial and academic use.

Vendor:

Critical Mass, Inc.

```
Critical Mass, Inc.
225R Concord Avenue
Cambridge, MA 02138
U.S.A.
```

Phone:

+1 617 354 6277

Fax:

+1 617 354 5027

EMail:

info@cmass.com

URL:

http://www.cmass.com

Last modified:

May 19, 1997.

9.18 Resource Standard Metrics

Description:

A source code metrics and quality analysis tool for C and C++. This creates standard metrics like Lines of Code, Cylcomatic complexity and functional analysis. This tool is portable across most all popular operating systems. The quality analysis checks for semantic errors that most compilers miss and enforces stardard software development standards.

Distribution media:

Software can be downloaded from our

web site or EMailed or shipped on floppy disk in ZIP and tar format.

Licensing policy:

Shareware licensing for the demo version which processes one file. Commercial single user and network licenses will process any number of files of any size from wild cards at the command line, from a list or recursively descend a directory tree.

OS provisions:

Linux 1.2.

TOC

Documentation:

Comes with full online documentation or printed text file. English language only.

Product support:

1 Email support. 1 year software maintenance available.

Extra features and add-ons:

Some license come with unmodifiable source code for compiling on your specific OS. Each license comes with a pretty printing utility and a DOS to Unix conversion utility.

Available since:

March, 1997.

Countries with distribution:

World wide.

Price range:

US$99.99 for single user non-source code version to US$999.95 for the network source code license.

Vendor:

M Squared Technolgies

```
2128 Hidden Pine Lane
Apopka, FL 32712
U.S.A.
```

Phone:

+1 407 880 2627

Fax:

+1 407 880 2627

EMail:

m2tech@reachus.com

URL:

http://www.m2tech.net

Last modified:

September 25, 1998.

9.19 r-tree

TOC

Description:

r-tree report generator in C source code.

Provides complex, multi-line reports by handling virtually every aspect of report generation. The only programming requirement is to call the r-tree report function, which reads c-tree data files, performs calculations, monitors control breaks and accumulators and produces a formatted report. Complete with C source code. Requires c-tree Plus.

Distribution media:

3.5" diskettes and CD-ROM.

Licensing policy:

r-tree Plus is licensed on a per programmer basis. Royalty-free distribution. Contact FairCom for possible restrictions.

Documentation:

Printed manuals distributed with product; full online documentation with CD-ROM, available in English and Japanese.

Product support:

Three months of full technical support from purchase date. Unlimited technical support and product updates available thereafter with r-tree maintenance program.

Available since:

October, 1995.

Countries with distribution:

World wide.

Price range:

US$445.

Vendor:

FairCom

```
4006 W. Broadway
Columbia, MO 65203-0100
U.S.A.
```

Phone:

+1 573 445 6833

Fax:

+1 573 445 9698

EMail:

Faircom@faircom.com

URL:

http://www.faircom.com

Contact:

Tamra Brown <tami@faircom.com>

Vendor:

FairCom Europe

```
Via Patrioti 6
I-24021 Albino
Italy
```

Phone:

+39 35 773 464

Fax:

+39 35 773 806

EMail:

Europe@faircom.com

Vendor:

FairCom Japan

```
Ikeda Bldg. #3 4F, 112-5
Komei-chou, Tsu-city MIE 514
Japan
```

Phone:

+81 059 229 7504

Fax:

+81 059 249 723

Vendor:

FairCom do Brasil Ltda.

Phone:

+55 11 3872 9802

Fax:

+55 11 3862 1752

EMail:

Brazil@faircom.com

Last modified:

August 7, 1997.

9.20 sdoc (Source Documenter)

Description:

This program helps to create a complete, indexed documentation from your sources (C, tcl, Perl currently available, other will follow soon).

You add documentation information to your scripts using pod (Plain old documentation), the documentation format also used in Perl. The documentation options provided by pod are simple and easy to learn but powerfull enough to create nice and good looking documents.

sdoc uses this information to create a pod-document which in turn may be transformed to HTML, LaTeX (and PostScript), nroff or just plain text.

The preview function of tdoc lets you create documentation interactively - add or modify your documentation to the source file and check the output for correctness.

Here are some highlights:

- Documentation is included in source file, no separate files needed
- Powerfull formatting options combined with easy learning
- Good looking documentation can be created without hassle
- Automated indexing
- Output may be in HTML, PostScript and a lot of other formats (have a look at the perlpod manual page or at available pod converters).

The Tcl Documenter is also available as a commercial version. This version adds the following features:

- Significantly faster
- May optionally use Netscape to display preview pages.
- Includes the package concept to further automate file selection.
- Direct connection to emacs for file editing
- Direct creation of postscript output
- Full support, upgrades
- Complete documentation

Distribution media:

3.5" diskettes and Internet (EMail and WWW).

Licensing policy:

You can download the public domain version of sdoc from the neosoft tcl archive:

- http://www.neosoft.com/tcl/ftparchive/sorted/development
- ftp://ftp.neosoft.com/pub/tcl/sorted/development

OS provisions:

You need an ELF based system with X11.

Documentation:

Context sensitive help and complete documentation in Pod format. Available as PostScript on request.

Product support:

Installation support included, additional support available.

Available since:

July, 1996.

Countries with distribution:

World wide.

Price range:

Contact softWorks for pricing details. Discounts for students and educational institutions available.

Vendor:

softWorks, Richard Schwaninger

```
Theodor-Körnerstr. 173
A-8010 Graz
Austria
```

Phone:

+43 316 686590

Fax:

+43 316 686590

EMail:

risc@ping.at

URL:

http://members.ping.at/risc

Contact:

Richard Schwaninger

Last modified:

August 7, 1997.

9.21 SEDIT, S/REXX

Description:

SEDIT is a powerful Unix text editor patterned after IBM's XEDIT editor. It operates with a GUI under X windows or in character mode from a tty device. S/REXX is a full Unix implementation of IBM's SAA procedural language except that the numeric digit specification is limited to 15 digits. S/REXX functions as an imbedded macro language for SEDIT as well as providing a powerful modern programming language alternative to shell scripting languages. SEDIT and S/REXX may be purchased in a bundle or individually. See our web site for more information including pricing for Linux, etc. For additional information, please contact one of the local distributors listed on the web site or mail us at sedit@dialup.FranceNet.fr

Distribution media:

4mm, 8mm, 1/4" or 3.5" diskettes and Internet (FTP).

Licensing policy:

SEDIT and S/REXX are licensed products. See the above WWW site for details. For demo purposes, a short term license key will be provided on request to anyone who obtains the product via ftp.

OS provisions:

We believe these products will work with any recent stable Linux kernel since 1.1.18 supporting a.out format. Motif is fully bound with SEDIT and SEDIT is distributed with XFree86 3.1 shared libraries for use if your system is not at that level.

Documentation:

Documentation is furnished in PostScript form, in extensive online help files, and in optional printed manuals.

Product support:

Technical support is available for all licensed users. For maintenance, see the above URLs.

Extra features and add-ons:

A REXX interactive graphical debugger is available as an optional feature as are printed manuals, physical media, and on going maintenance. See the above URLs.

Available since:

The original Linux version has been available since 3rd quarter 1995.

Countries with distribution:

SEDIT and S/REXX are available worldwide. The URLs should be consulted for current information.

Price range:

Pricing information should be requested from the local distributor responsible for your country. An economical 2 user license is available for Linux without media or printed documentation with other options for more users, S/REXX, hardcopy manuals, etc.

Vendor:

Benaroya

```
31 Rue de Constantinople
F-75008 Paris
France
```

Phone:

+33 1 47 22 22 13

Fax:

+33 1 47 22 06 17

EMail:

sedit@dialup.FranceNet.fr

URL:

http://www.sedit.com

Contact:

Robert Benaroya

Last modified:

August 4, 1997.

9.22 SNiFF+

Description:

SNiFF+ is an open, scalable and multiplatform programming environment for C/C++, Java, CORBA IDL and Fortran. The main goal in developing SNiFF+ was to create an efficient and portable programming environment with a comfortable user interface and special support for object oriented programming.

Distribution media:

CD-ROM and Internet (FTP).

Licensing policy:

Trial license avialable.

OS provisions:

Linux 1.2 or 2.0.

Documentation:

Printed documentation and HTML version.

Extra features and add-ons:

Extensions for other programming languages available.

Vendor:

TakeFive Software GmbH

```
Jakob-Haringer-Straße 8
A-5020 Salzburg
Austria
```

Phone:

+43 662 4579150

Fax:

+43 662 4579156

EMail:

info@takefive.co.at

URL:

http://www.takefive.co.at

Last modified:

November 3, 1997.

9.23 ST/X (Smalltalk/X)

Description:

ST/X is a complete implementation of the Smalltalk programming language with development environment providing all of the advantages you would expect from Smalltalk: a fully object oriented programming language, a graphical development environment with editors, browsers, symbolic debugger, incremental compiler and just-in-time compilation to machine code as well as an extensive class library.

Distribution media:

CD-ROM.

Licensing policy:

TOC

There are non-commercial, evaluation and commercial licences available. There is also demo version on the Internet, although it is quite old, this should be updated at some time in 1997.

OS provisions:

ST/X requires 32 MB or more of memory and 100 MB hard disk space.

Documentation:

The ST/X documentation is available online as hypertext (HTML).

Product support:

eXept Software AG provides mentoring, training, and consulting. Onsite support is also available. For the commercial licence there is an update service and maintenance support.

Extra features and add-ons:

ST/X includes a supplementary stand-alone machine code compiler and the possibility to embed C code directly into Smalltalk methods. This allows the exploitation and use of existing C libraries or a performance improvement in the speed of time critical operations. An integral part of ST/X is an HTML browser which is fully implemented in Smalltalk. It supports network loadable Smalltalk applets and smalltalk scripts embedded in HTML pages. This allows the development of Internet and Intranet applications analogous to Java.

Available since:

ST/X has been available as a beta version on the Internet for quite a while. eXept Software AG has been further developing and marketing ST/X since 1996. Claus Gittinger (the developer from ST/X) is a founding member of eXept.

Countries with distribution:

World wide.

Price range:

There are non-commercial, evaluation and commercial licences available. The commercial licence costs US$3,500. A six month evaluation licence costs US$350. The non-commercial license is available for US$150 plus shipping and handling.

Vendor:

eXept Software AG

```
Besigheimerstr. 1
D-74369 Löchgau
Germany
```

Phone:

+49 7143 870045

Fax:

+49 7143 870048

EMail:

info@exept.de

URL:

http://home.t-online.de/home/exept

Contact:

David Queeney

Last modified:

August 5, 1997.

9.24 tdb (Tcl Debugger)

Description:

This is a debugger for tcl scripts. It allows interactive debugging of tcl/tk programs and speeds up your development cycle.

Here are some highlights:

- Step by step execution with breakpoints
- Display and manipulation of local and global variables
- On-the-fly source code modifications and fast programm restarts
- Graphical user interface, easy use
- Easy configuration and installation

The commercial version adds the following features:

- Full screen context display
- Hot link to editor (currently emacs)
- Scripting and user extensibility (tcl)
- Plugin support (script, popup)
- A lot faster
- Full support, upgrades
- Complete documentation

Distribution media:

3.5" diskettes and Internet (EMail and WWW)

Licensing policy:

You can download the public domain version of tdb from the neosoft archive:

- http://www.neosoft.com/tcl/ftparchive/sorted/development
- ftp://ftp.neosoft.com/pub/tcl/sorted/development

TOC

OS provisions:

ELF based system with X11.

Note that tdb requires changes to the base tcl C-libraries and also depends on dynamic loading. Please check out the info package for further details. You don' need tcl installed to use the debugger, all necessary code is included.

The commercial version may be purchased in source form and support for porting to other platforms is available. Please contact softWorks for details.

Documentation:

Context sensitive help and complete documentation in HTML format. Available as PostScript on request.

Product support:

Installation support included, additional support available.

Extra features and add-ons:

Debugger-PlugIns for speacialized debug tasks (e.g. to debug tk widget hierarchies are also available).

Available since:

July, 1996.

Countries with distribution:

World wide.

Price range:

US$450 for a binary version, US$750 for the source code, with a 10% discount for students and educational institutions.

Vendor:

softWorks, Richard Schwaninger

```
Theodor-Körnerstr. 173
A-8010 Graz
Austria
```

Phone:

+43 316 686590

Fax:

+43 316 686590

EMail:

risc@ping.at

URL:

http://members.ping.at/risc

Contact:

Richard Schwaninger

Last modified:

August 7, 1997.

9.25 tprof (Tcl Profiler)

Description:

This is a profiler for tcl scripts. It allows the creation of profiling data from your tcl programms and later analysis and detailed inspection through a graphical frontend.

This tool is especially well suited if you encounter performance problems with your tcl scripts. Instead of rewriting the whole application in C you can isolate time-critical procedures and speed them up (if necessary using C).

Here are some highlights:

- Call flow display
- Barcharts for number of calls, CPU time and real time
- Tabular display of profiling data set
- Analyse both tcl procs and commands written in C
- Graphical user interface, easy use
- Simple aquisition of profiling data
- Easy configuration and installation

tprof is available both as a public domain and as a commercial version. The commercial version adds the following features:

- *Zoom in* on interesting subsets of the data
- Printout of data sets
- PostScript (eps) images of charts and tables
- Merging of multiple data sets
- Exporting data sets in different formats to inspect them further (eg. sc/xspread, Wingz, DBase, text)
- A lot faster
- Full support, upgrades
- Complete documentation

Distribution media:

3.5" diskettes, Internet (EMail and WWW).

Licensing policy:

You can download the public domain version of tprof from sunsite.unc.edu and it's mirrors.

OS provisions:

You need an ELF based system with X11.

Documentation:

Context sensitive help and complete documentation in HTML format. Available as PostScript on request.

Product support:

Installation support included, additional support available.

Available since:

October, 1996.

Countries with distribution:

World wide.

Price range:

Contact softWorks for details and discounts.

Vendor:

softWorks, Richard Schwaninger

```
Theodor-Körnerstr. 173
A-8010 Graz
Austria
```

Phone:

+43 316 686590

Fax:

+43 316 686590

EMail:

risc@ping.at

URL:

http://members.ping.at/risc

Contact:

Richard Schwaninger

Last modified:

August 7, 1997.

9.26 View Designer/X (VDX)

Description:

Motif interface builder that generates C and C++ code. VDX provides WYSIWYG view, widget tree browser, resource editor, widget templates and more tools to design user interfaces.

Distribution media:

Internet (WWW).

Licensing policy:

Free demo version. Purchase of license key that removes restrictions of the demo version. Three types of licenses are available: company license, host license, student license.

OS provisions:

Linux, ELF, X11R6, Libc.5, 5 MB on harddisk.

Documentation:

Online help, English. PostScript manual, English. A German version of the manual is available.

Product support:

Minor update (1.x to 1.x) free. Free EMail support.

Available since:

December, 1996.

Countries with distribution:

World wide.

Price range:

DM 800 (approximately US$470), Major update (1.x to 2.x) DM 500 (approximately US$295).

Vendor:

Bredex GmbH

```
Fallersleber-Tor-Wall 23
D-38100 Braunschweig
Germany
```

Phone:

+49 531 24 33 00

Fax:

+49 531 24 33 099

EMail:

info@bredex.de

URL:

http://www.bredex.de

Contact:

Regina Heine

Last modified:

October 13, 1997.

9.27 XBasic

Description:

XBasic is a comprehensive visual program development environment that integrates a powerful editor, advanced 32/64-bit BASIC compiler, debugger, GuiDesigner and GraphicsDesigner. XBasic is written entirely in XBasic.

Distribution media:

Internet (FTP).

Licensing policy:

Each copy is licensed on a per machine basis.

OS provisions:

Linux (ELF 1.2.13 or higher).

Documentation:

300+ page electronic manual (Word format).

Product support:

US$3 per minute.

Available since:

January, 1996.

Countries with distribution:

World wide,

Price range:

US$195.

Vendor:

Basmark Corporation

P.O. Box 40450
Cleveland, OH 44140
U.S.A.

Phone:

+1 216 871 8855

Fax:

+1 216 871 9011

EMail:

jgo@basmark.com

URL:

http://www.basmark.com

Contact:

Joseph G. O'Toole

Last modified:

November 4, 1997.

9.28 XMove 4.0 for Linux

Description:

XMove is a software package for design, prototyping and testing the graphic user interface of a software system dealing dynamically changing values. This conception shall be partially reflected in the name XMove, which is also the abbreviation of "X Window System Meter Object Visualization and Editing Tool".

X Window System: XMove is based completely on the standard X Windows System. XMove was specifically designed to support this standard for graphics user interfaces. In many applications XMove can be used complementary to OSF/Motif.

Meter Object: Meter Object is the reference to the dynamic features of XMove. Using XMove the user can provide dynamic attributes for any graphics object in addition to the definition of static behavior.

Visualizing: means that the displaying and managing of the application-specific graphics, which can be designed interactively usign XMove, is done by XMove during of the application.

Editing Tool: stands for the interactive possibilities of creating the static and dynamic parts of an application-specific graphyc usign the XMove editors.

XMove generate the necesary C++ code for X Windows System to develop the application in a very sample way. XMove is an object oriented graphics developing system.

Distribution media:

CD-ROM, Internet (EMail and WWW).

Documentation:

PostScript manuals are included with the package.

Available since:

March, 1997.

Countries with distribution:

World wide.

Price range (Europe, Asia, Australia):

- US$6,000 (including the editor and libraries).
- US$3,500 (only the editor)
- US$3,500 (only the library)

Price range (America):

- US$5,000 (including the editor and libraries).
- US$3,000 (only the editor)
- US$3,000 (only the library)

Vendor:

Future Technologies

```
Via B. Cairoli, 1
I-33170 Pordenone
Italy
```

Phone:

+39 434 20 91 07

Fax:

+39 434 20 95 10

EMail:

info@futuretg.com

URL:

http://www.futuretg.com

Contact:

Giovanni A. Orlando

Last modified:

August 12, 1997.

10. Emulation

10.1 Emulus

Description:

Emulus is an X-Windows/Motif application which uses TCP/IP to establish a connection to an IBM mainframe host, emulating a 3270 terminal. Emulus supports mainframe graphics, including 3270 graphics datastream extension used by mainframe SAS Software. Other features include customization through dialogs, a scripting facility to record and playback host interaction, EasyPads (customized windows of buttons), and cut and paste between Emulus and other X applications. Emulus also comes with Helplus, an hypertext help viewer modeled after WinHelp. It is used to provide online help for Emulus, but can also be used to develop Help for other X applications.

Distribution media:

CD-ROM.

Licensing policy:

Licensing is per CPU.

OS provisions:

Linux 1.2 or higher, 8-16 MB RAM, 5 MB hard disk; Motif statically linked.

Documentation:

Printed book, man pages, online help, English.

Product support:

Technical Support from SAS Institute.

Price range:

US$99; includes hardcopy documentation, and Technical Support for one year.

Vendor:

SAS Institute Inc.

```
SAS Campus Drive
Cary, NC 27513
U.S.A.
```

Phone:

+1 919 677 8000

Fax:

+1 919 677 8166

EMail:

saspjh@unx.sas.com

URL:

http://www.sas.com

Contact:

Phil Herold

Last modified:

August 4, 1997.

10.2 Executor 2

Description:

Executor 2 allows your Linux system to run many Macintosh applications, including commercial software like Adobe Photoshop and Illustrator, Quark XPress, Microsoft Word and Excel and Quicken, as well as lots of Macintosh shareware, freeware and demoware. A compatibility database is available at our web site. Executor 2 also allows you to read and write Macintosh formatted media without losing important meta-data information like the file type and file creator.

Distribution media:

CD-ROM.

Licensing policy:

Commercial software with a time limited demo available from our web site.

OS provisions:

Linux/X and Linux/SVGAlib.

Documentation:

30 page mini-manual in CD-ROM jewel-box case, additional tutorial information available on our web site and in the Usenet group comp.emulators.mac.executor.

Extra features and add-ons:

Each Executor 2 CD-ROM contains Executor/DOS and Executor/Linux. You can switch between them, but are allowed to only run one at a time.

Price range:

Suggested Retail: US$249. Educational Institutions: US$149, Full-time Students: US$65.

Vendor:

ARDI

```
Suite 4-101
1650 University Boulevard, NE
Albuquerque, NM 87102
U.S.A.
```

Phone:

+1 505 766 9115

Fax:

+1 505 766 5153

EMail:

questions@ardi.com

URL:

http://www.ardi.com

Last modified:

November 4, 1997.

10.3 Wabi 2.2 for OpenLinux

Description:

Caldera licensed and ported SunSoft's Wabi technology to enable end users to run popular Windows 3.1 applications on Linux-based system software. Channel Partners and customers can utilize Caldera Wabi 2.2 for Linux running on Caldera OpenLinux, an open source environment, to remotely manage Windows 3.1 applications at home, in the office or on the road. Channel Partners who create customized turn-key solutions based on environments like SCOR OpenServer 5 or Windows NT can now use the Wabi/COL solution to increase revenues by lowering overall cost and system requirements. The suggested retail price for Wabi 2.2 for Linux is US$199. The product requires Linux on a 386 (or higher) Intel-based processor, 16 MB RAM (24 MB recommended), VGA-quality video and 10 MB disk space.

Distribution media:

CD-ROM.

Licensing policy:

One user per licensed copy. No shareware or FTP.

OS provisions:

Runs on Caldera Network Desktop and OpenLinux.

Documentation:

TOC

Online and printed manual.

Product support:

30-days installation support via EMail and phone.

Available since:

November 18, 1996.

Countries with distribution:

47 countries worldwide; see our web site for local reseller.

Price range:

US$199 with additional licenses at US$179. Educational pricing is US$149 with additional licenses at US$129.

Vendor:

Caldera, Inc.

```
633 S. 550 E.
Provo, UT 84606
U.S.A.
```

Phone:

+1 800 850 7779, +1 801 377 7687

Fax:

+1 801 377 8752

EMail:

orders@caldera.com

URL:

http://www.caldera.com

Contact:

Nathan Hatch

Last modified:

November 14, 1997.

---

11. Financial Software

11.1 BB Stock Pro and BB Stock Tool

Description:

BB Stock Pro and BB Stock Tool are powerful stock charting, tracking and analysis tools for investors. Features include stock charting, technical analysis, market timing, portfolio management. In addition, it has customized alerts, automatic stock split detection and management, stock screening, most active issues and big price movers, personalized watch list, Auto-Run, and builtin Easy Stock Update via Internet with either direct connection or behind firewall via proxy server.

Distribution media:

Internet (FTP and WWW).

Licensing policy:

Per machine license. 30 day free trial is available at our web site.

OS provisions:

Linux 2.0 ELF.

Documentation:

Context sensitive online help, and User's Guide in HTML, viewed with web browser.

Product support:

Free technical support. Frequently asked questions and answers available at our web site.

Available since:

Sun SPARC version available since December 1993. Linux version available since April 1995.

Countries with distribution:

USA, Canada, Sweden, Germany, Singapore, Malaysia, Taiwan.

Price range:

Linux version is US$99.

Vendor:

Falkor Technologies

```
P.O. Box 14201
Fremont, CA 94539
```

U.S.A.

Phone:

+1 510 505 0700

EMail:

skyline@falkor.com

URL:

http://www.falkor.com

Contact:

Henry Chen

Last modified:

March 18, 1997.

11.2 TimeClock

Description:

Employee time and attendance package for small to medium sized businesses. Includes custom report module, interface to barcode/magnetic card readers (Employee ID), and data export module.

Distribution media:

3.5" diskettes.

Licensing policy:

One CPU, unlimited users.

Documentation:

Printed manual and online HTML manual included with software.

Product support:

Free, usually 20 minute response time.

Extra features and add-ons:

Barcode/magnetic card interface.

Available since:

1992 XENIX and MS-DOS version, 1994 AIX version, October 1996 Linux version.

Countries with distribution:

USA.

Price range:

US$149.99.

Vendor:

Quality Software Solutions, Inc.

```
1322 Dodds Avenue
Chattanooga, TN 37404
U.S.A.
```

Phone:

+1 888 423 5757, +1 423 821 5757

Fax:

+1 423 629 0744

EMail:

dbryson@tclock.com

URL:

http://www.tclock.com

Contact:

Donald Bryson

Last modified:

August 4, 1997.

---

12. Libraries

12.1 FontScope

Description:

Type 1 Font Rasterizer Library. Allows users to arbitrarily scale, skew or rotate characters in Type 1 fonts and get back either a bitmap or an outline. Product includes full sources.

Distribution media:

3.5" diskettes.

Licensing policy:

Single user license with product. Since product includes full sources and our resources are limited, it is sold with *no warranty* and *no support*. Support contracts may be purchased at additional cost. OEM licenses, site licenses and enterprise-wide licenses available.

Documentation:

A manual, nicely typeset in 2-column format, is available with the distribution as a PostScript file.

Product support:

Support contracts may be purchased at additional cost.

Available since:

October 15, 1997.

Countries with distribution:

World wide.

Price range:

US$39.95. California residents subject to sales tax. Price subject to change without notice.

Vendor:

CurveSoft, Inc.

```
2053 Grant Road, Suite 555
Los Altos, CA 94024
U.S.A.
```

Phone:

+1 800 563 0843, +1 510 843 6485 (Sales only)

Fax:

+1 650 254 0900

EMail:

info@curvesoft.com

URL:

http://www.curvesoft.com

Contact:

Munagala V. S. Ramanath

Last modified:

October 24, 1997.

12.2 INTERACTER

Description:

INTERACTER is a portable user-interface and graphics subroutine library for Fortran software developers. INTERACTER-based programs are portable to a wide range of Fortran compilers running on Unix, VMS, Windows, DOS. The Linux version supports f2c and g77 (ELF). Functions include menus, text windows, forms, graphics primitives, presentation graphics, hardcopy (many formats) and operating system interface.

Distribution media:

3.5" diskettes and 4mm DAT.

Licensing policy:

INTERACTER-based programs can be distributed royalty free. Linux demo program available at http://www.demon.co.uk/issltd.

OS provisions:

Linux version supports X Windows (via Xlib), graphics terminals or Linux console (colour text-mode). Requires g77 or f2c.

Documentation:

Two-volume English language printed manual.

Product support:

Free lifetime technical support. New releases every six months. Chargeable software updates.

Extra features and add-ons:

Optional INTERACTER Toolkit provides form designer, graphics file viewer, online help, hardcopy post-processor and set-up file editor.

Available since:

October, 1995.

Countries with distribution:

Distributors in North America, Europe and Australia.

Price range:

Linux version: US$675 (UK £450).

Vendor:

Interactive Software Services Ltd.

```
Westwood House
Littleton Drive
Huntington
Staffs WS12 4TS
United Kingdom
```

Phone:

+44 1543 503611

Fax:

+44 1543 574566

EMail:

support@issltd.demon.co.uk

URL:

http://www.demon.co.uk/issltd

Last modified:

October 1, 1997.

12.3 Matrix<LIB> - C++ Math Matrix Library

Description:

Matrix<LIB> is a Matlab Compatible C++ Matrix Class Library, designed for development of advanced scientific high-level C++ code. Our main design objective was to keep the C++ source code clear and similar to the equivalent Matlab code, making your shift from Matlab to C++ easier. If you know how to program Matlab, you already know how to use our Matrix<LIB> C++ Math Library. The code was carefully optimized for maximal performance, too. Matrix<LIB> is also included in the MATCOM V2 distribution.

The library includes complex math, binary and unary operators, powerful indexing capabilities, signal processing, file I/O, linear algebra, string operations and graphics. Over 300 mathematical functions are included in Matrix<LIB>.

MS Windows and many Unix platform are supported with a variety of C++ compilers. An evaluation version (fully functional, 30 days limited) is available for free download from our web site.

Distribution media:

Internet (WWW).

Licensing policy:

From single license to site license. Contact The MathTools for details.

OS provisions:

Linux 1.2, gcc 2.6.3 or later, 8 MB of RAM, 15 MB on disk.

Documentation:

Users manual and library reference in various formats (MS Word, MS Write, plain text, PostScript).

Product support:

A fast product support is given by EMail only. The support section in the web page give immediate access to a wealth of resources, technical support, answers to frequently asked questions, technical documents and more.

Extra features and add-ons:

MathTools Accelerator: provides significant performance gains for Matrix<LIB> programs using Linear Algebra and matrix multiplication, on a Pentium with the Visuall C++ compiler.

Available since:

November, 1996.

Countries with distribution:

World wide.

Price range:

US$499 (US$249 for academic institutes).

Vendor:

MathTools, Ltd.

```
P.O. Box 855
Horsham, PA 19044-0855
U.S.A.
```

Phone:

+1 212 208 4476

Fax:

+1 888 628 4866, +1 215 957 1719

EMail:

info@mathtools.com

URL:

http://www.mathtools.com

Contact:

Robert G. Ford

Last modified:

August 7, 1997.

12.4 PKWARE Data Compression Library for Linux

Description:

The library does lossless compression and consists of compression, an extraction, and an error checking routine. To maintain flexibility and keep the library's code size at a minimum, the PKWARE Data Compression Library does not create .ZIP compatible archives.

Distribution media:

3.5" diskettes.

Licensing policy:

Per machine with no runtime royalties.

OS provisions:

Intel x86, 36 KB RAM for compressing and 13 KB RAM for uncompressing, 1 MB on hard disk.

Documentation:

Printed 33 page manual and separate installation notes in English. Example code provided on disk.

Product support:

No-charge. Technical support via phone, fax or EMail.

Extra features and add-ons:

Also compatible with FreeBSD.

Available since:

September 14, 1995.

Countries with distribution:

Australia, Brazil, France, Germany, India, Italy, Germany, The Netherlands, Japan, and the UK.

Price range:

US$450. Educational discount available.

Vendor:

PKWARE, Inc.

```
9025 N. Deerwood Drive
Brown Deer, WI 53223-2480
U.S.A.
```

Phone:

+1 414 354 8699

Fax:

+1 414 354 8559

EMail:

info@pkware.com

URL:

http://www.pkware.com

Contact:

Bob Gorman

Last modified:

August 5, 1997.

12.5 readyBase

Description:

A dynamic embeddable database library with a C API, Perl module, and Java interface.

Distribution media:

CD-ROM and Internet (FTP and WWW).

Licensing policy:

Commercial license. A free 30 day demo is available.

OS provisions:

Kernel 2.0.x. 1MB RAM.

Documentation:

Online documentation.

Product support:

Update service, maintenance, training and prices thereof.

Available since:

March, 1998.

Countries with distribution:

USA.

Price range:

US$1295 for 1 developer and 50 end-user license

Vendor:

Dynamical Systems, Ltd.

```
175 W. Jackson Boulevard, Suite A253
Chicago, IL 60604
U.S.A.
```

Phone:

+1 312 341 1755

Fax:

+1 312 341 1891

EMail:

info@dynamical-systems.com

URL:

http://www.dynamical-systems.com

Contact:

Gary Kuper

Last modified:

April 22, 1998.

12.6 SIMLIB IG

Description:

SIMLIB IG is a C library. It enables the user to communicate with an Evans & Sutherland image generator (Liberty and ESIG Systems) using a very efficient raw Ethernet protocol. There is no need for using opcodes, since SIMLIB IG provides an API to the functionality of the image generators.

Distribution media:

4mm DAT tar format and 3.5" diskettes.

Licensing policy:

Commercial.

OS provisions:

Linux 2.0.

Documentation:

Printed manual in English, source code examples on media.

Price range:

US$2,500.

Vendor:

KNIENIEDER Simulationstechnik

```
Technologiezentrum Tirol
Eduard Bodemgasse 5
A-6020 Innsbruck
Austria
```

Phone:

+43 512 390415 650

Fax:

+43 512 364000 20, +43 512 364000 30

EMail:

office@knienieder.co.at

Contact:

Stephan Haidacher

Last modified:

September 29, 1997.

---

13. Mathematics

13.1 Maple V Release 4 - The Power Edition

Description:

Maple V Release 4 is the powerful computer algebra system used by mathematicians, engineers, and scientists for teaching and research. Maple V lets you perform a wide range of symbolic and numeric computations quickly and accurately. It includes more than 2500 built-in math functions, 2D and 3D graphics, animation, and more. New features in Release 4 include outlining, styles, and hyperlinking, typeset math notation in input, output, and text, full text search of Help Database and new computational and plotting facilities.

Distribution media:

CD-ROM.

Licensing policy:

License for unlimited use in time of the bought version, floating license with user based pricing.

OS provisions:

Linux kernel 2.0.0 or above, X11 R6.1 or above (for running X11 GUI components), 12 MB of RAM suggested. The binaries are available in ELF format only.

Documentation:

3 books: "Maple V - Learning Guide", "Maple V - Programming Guide", "The Maple V Handbook". Also available in German: "Einführung in Maple V" und "Programmieren mit Maple V".

Documentation and help in English, a variety of books available in different languages.

Product support:

Updates/maintenance depending on license, please call the distributor or WMSI.

Extra features and add-ons:

Share library via EMail and FTP. Free mailing list available (maple_group@daisy.waterloo.edu).

Available since:

September, 1994.

Countries with distribution:

USA/Canada, Europe and Asia.

Vendor:

Waterloo Maple Inc.

```
450 Phillip Street
Waterloo, ON N2L 5J2
Canada
```

Phone:

+1 519 747 2373

Fax:

+1 519 747 5284

EMail:

info@maplesoft.com

Vendor in Germany:

Scientific Computers GmbH

```
Franzstr. 106
D-52064 Aachen
Germany
```

Phone:

+49 241 47075 0

Fax:

+49 241 44983

EMail:

maple@scientific.de

Contact:

Andreas Himmeldorf

Last modified:

August 4, 1997.

13.2 MATCOM and MATCOM MATH LIBRARY

Description:

MATCOM is a Matlab to C++ compiler and C++ MATRIX MATH LIBRARY. MATCOM compiles Matlab source files (M-Files) to C++ source code. The resulting code is linked with the supplied C++ Matrix class library to create standalone executables or MEX files.

The C++ code and library can be integrated in products, royalty free, saving the need to translate

the algorithm prototype.

Compiled code runs significantly faster than the original interpreted source.

Matrices of doubles, floats, ints and chars are supported, providing lower memory usage for many applications, especially signal and image processing.

MS Windows and many Unix platform are supported with a variety of C++ compilers. An evaluation version (fully functional, 30 days limited) is available for free download from our web site.

Distribution media:

Internet (WWW).

Licensing policy:

From single license to site license. Contact The MathTools for details.

OS provisions:

Linux 1.2, gcc 2.6.3 or later, 8 MB of RAM, 15 MB on disk.

Documentation:

Users manual and library reference in various formats (MS Word, MS Write, plain text, PostScript).

Product support:

A fast product support is given by EMail only. The support section in the web page give immediate access to a wealth of resources, technical support, answers to frequently asked questions, technical documents and more.

Extra features and add-ons:

MathTools Accelerator: provides significant performance gains for MATCOM programs using Linear Algebra and matrix multiplication, on a Pentium with the Visual C++ compiler.

Available since:

May, 1995.

Countries with distribution:

World wide.

Price range:

US$499 ($249 for academic institutes).

Vendor:

MathTools, Ltd.

```
P.O. Box 855
Horsham, PA 19044-0855
```

U.S.A.

Phone:

+1 212 208 4476

Fax:

+1 888 628 4866, +1 215 957 1719

EMail:

info@mathtools.com

URL:

http://www.mathtools.com

Contact:

Robert G. Ford

Last modified:

August 7, 1997.

13.3 Mathematica 3.0

Description:

Mathematica 3.0--Known for delivering quick, accurate numeric and symbolic solutions, Mathematica is ideal for creating interactive technical reports and presentations that include text, active formulas, graphics, and customizable buttons and palettes. Breakthrough new features, such as an innovative typesetting system that can do math, now make Mathematica even easier to use.

Distribution media:

CD-ROM.

Licensing policy:

Commercial, 30 day fully functional demo CD-ROM.

OS provisions:

Kernel 1.2 or higher.

Documentation:

The Mathematica Book 1400 pages. Complete book and additional help files online.

Product support:

Three service levels, see http://www.wolfram.com/service for details.

Extra features and add-ons:

About 15 application packages.

Available since:

October, 1996.

Countries with distribution:

World wide.

Price range:

US$1295, academic and student discounts, site licenses available.

Vendor:

Wolfram Research Europe Ltd

```
10 Blenheim Office Park
Lower Road
Long Hanborough
Oxfordshire
OX8 8LN
UK
```

Phone:

+44 1993 883400

Fax:

+44 1993 883800

EMail:

info@wolfram.co.uk

URL:

http://www.wolfram.co.uk

Last modified:

May 25, 1998.

13.4 MATLAB and Simulink

Description:

MATLAB is a high performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB supports the development of large complex applications. Features include the M-file Performance Profiler for algorithm optimization, the integrated visual editor and debugger, and powerful interactive tools for signal processing and control system design.

Simulink is a powerful, interactive software package for modeling, analyzing, and simulating

dynamic systems. It provides extremely accurate simulations of nonlinear continuous-time sys and systems with multiple modes of operation. Features include support for conditionally exec subsystems; state-of-the-art solvers; usability improvements such as signal and port labeling; and new blocks.

Distribution media:

CD-ROM and Internet (FTP).

Licensing policy:

Using GlobeTrotter (FLEXlm) CPU locked or floating network options. Contact The MathWorks for details.

OS provisions:

Linux 2.0.18, 32 MB of RAM, 40 MB disk space (60 MB to include all help files and documentation).

Documentation:

Printed documentation, online help, MATLAB Reference Guide (HTML).

Product support:

The MathWorks supports a variety of Internet-based services to provide product information and services 24 hours a day. These services give immediate access to a wealth of resources, product data sheets, technical support, answers to frequently asked questions, user-contributed materials, open-forum discussions, and more.

Extra features and add-ons:

MATLAB Extensions: MATLAB Compiler, MATLAB C/C++ Math Library.

Toolboxes: Fuzzy Logic, Signal Processing, Image Processing, Spline, Symbolic Math, Neural Network, Control System, Robust Control, Mu-Analysis & Synthesis, System Identification, Optimization, Statistics, Financial, Frequency Domain System Identification, Higher-Order Spectral Analysis, Communication, LMI Control, Model Predicitve Control, NAG Foundation, Neural Network, Partial Differential Equation, QFT Control Design, Symbolic Math and Extended Symbolic Math, Wavelet.

Simulink Extensions: Simulink Accelerator, Real-Time Workshop.

Blocksets: Fixed-Point, DSP, Nonlinear Control Design.

Available since:

June, 1995.

Countries with distribution:

World wide.

Price range:

Upon request.

Vendor:

The MathWorks, Inc.

```
24 Prime Park Way
Natick, MA 01760
U.S.A.
```

Phone:

+1 508 647 7000 x7322

Fax:

+1 508 647 7002

EMail:

efroio@mathworks.com, brian@mathworks.com

URL:

http://www.mathworks.com

Contact:

Enza Froio (Marketing), Brian Bourgault (Technical)

Last modified:

March 27, 1997.

---

14. Multimedia

14.1 Peter Lipa and his Journeys

Description:

This is a multimedia CD-ROM about world-known jazzman Peter Lipa. It is the first product from edition Journeys of Groups and Singers from KDK Company to contain Linux version of the program directly on the CD-ROM. The CD-ROM contains more than 8 hours of near-CD quality music (101 songs in full length), more than 30 minutes of videos, 2 hours of spoken word, more than 120 photos, 10 karaoke songs and a lot of other information (Bratislava Jazz Days, various performances all over the world, TV and theatre performances, ...). The CD-ROM is multilingual: English, German and Slovak.

Distribution media:

CD-ROM.

Licensing policy:

Program is free, data licensed per CPU.

OS provisions:

Linux kernel 2.0.27 or higher (may work with 1.2), XFree86 3.2 (may work with older X11R6 versions), 16 MB of RAM, 10-30 MB of disk space, no Motif, Intel Linux ELF.

Documentation:

Online and short description of controls in CD-ROM booklet (English, German and Slovak).

Available since:

October 14, 1997.

Countries with distribution:

World wide.

Price range:

US$30 plus shipping, includes unlimited free program upgrades (via FTP).

Vendor:

KDK Company, Ltd.

```
Vodna 8
040 01 Kosice
Slovakia
```

Phone:

+421 95 6233335

Fax:

+421 95 6233336

EMail:

linux@kdk.sk

URL:

http://www.kdk.sk, http://obchod.kdk.sk (virtual store).

Contact:

Ivan Schreter

Last modified:

November 1, 1997.

14.2 Lucka Vondrackova and her Journeys

Description:

This is a multimedia CD-ROM about Lucka Vondrackova, young Czech singer and actress. This CD-ROM is multilingual: English, Slovak and Czech. The product does not yet contain Linux version of the program directly on the CD-ROM (The Linux version of the program is supplied on an extra diskette or can be downloaded via FTP). The CD-ROM contains more than three hours of music (all Lucka's songs in full length), 9 videoclips, all lyrics to Lucka's songs with notes and guitar accords, more than 130 photos, a lot of interesting interviews, 10 karaoke songs and a lot of other information about Lucka.

Distribution media:

Multimedia data on CD-ROM, program on diskette or via FTP.

Licensing policy:

Program is free, data licensed per CPU.

OS provisions:

Linux kernel 2.0.27 or higher (may work with 1.2), XFree86 3.2 (may work with older X11R6 versions), 16 MB of RAM, 10-30 MB of disk space, no Motif, Intel Linux ELF.

Documentation:

Online and short description of controls in CD-ROM booklet (English and Slovak).

Available since:

October 14, 1997.

Countries with distribution:

World wide.

Price range:

US$34 plus shipping, includes unlimited free program upgrades (via FTP).

Vendor:

KDK Company, Ltd.

```
Vodna 8
040 01 Kosice
Slovakia
```

Phone:

+421 95 6233335

Fax:

+421 95 6233336

EMail:

linux@kdk.sk

URL:

http://www.kdk.sk, http://obchod.kdk.sk (virtual store).

Contact:

Ivan Schreter

Last modified:

November 1, 1997.

14.3 MpegTV Player 1.0

Description:

Real-time MPEG Video player (with audio) and Video-CD player.

Distribution media:

Internet.

Licensing policy:

Shareware for personal and non-profit use. Commercial licenses required for commercial and governmental use.

Documentation:

Online help and HTML manual on web site.

Extra features and add-ons:

Can stream video from networks.

Price range:

Shareware registration fee is US$10. Commercial licenses: please contact MpegTV at sales@mpegtv.com.

Vendor:

MpegTV LLC

```
90 Divisadero Street, #15
San Francisco, CA 94117
U.S.A.
```

Phone:

+1 415 864 6466

EMail:

info@mpegtv.com

URL:

http://www.mpegtv.com

Contact:

Tristan Savatier

Last modified:

December 23, 1997.

14.4 Peter Nagy and his Journeys

Description:

This is a multimedia CD-ROM about Peter Nagy, Slovak pop singer. This CD-ROM is available in Slovak only and does not yet contain the Linux version of the program directly on the CD-ROM (The Linux version of the program is supplied on an extra diskette or can be downloaded via FTP). The CD-ROM contains more than four hours of music (154 song samples, 38 songs in full length), 9 videoclips, 222 song lyrics, 93 song notes with guitar accords, 120 photos, 8 interviews, citations from the book Musicross, information about all albums of Peter Nagy and a lot of other information.

Distribution media:

Multimedia data on CD-ROM, program on diskette or via FTP.

Licensing policy:

Program is free, data licensed per CPU.

OS provisions:

Linux kernel 2.0.27 or higher (may work with 1.2), XFree86 3.2 (may work with older X11R6 versions), 16 MB of RAM, 10-30 MB of disk space, no Motif, Intel Linux ELF.

Documentation:

Online and short description of controls in CD-ROM booklet (Slovak only).

Available since:

October 14, 1997.

Countries with distribution:

World wide.

Price range:

US$34 plus shipping, includes unlimited free program upgrades (via FTP).

Vendor:

KDK Company, Ltd.

```
Vodna 8
040 01 Kosice
Slovakia
```

Phone:

+421 95 6233335

Fax:

+421 95 6233336

EMail:

linux@kdk.sk

URL:

http://www.kdk.sk, http://obchod.kdk.sk (virtual store).

Contact:

Ivan Schreter

Last modified:

November 1, 1997.

14.5 Xaudio

TOC

Description:

MP3 Player.

Distribution media:

Internet.

Licensing policy:

Free for personal and non-profit use. Commercial licenses required for commercial and governmental use.

Documentation:

Online help and HTML manual on web site.

Extra features and add-ons:

Can stream CD-quality audio from networks.

Price range:

Free for personal and non-profit use. Commercial licenses: please contact MpegTV at sales@mpegtv.com.

Vendor:

MpegTV LLC

```
90 Divisadero Street, #15
San Francisco, CA 94117
U.S.A.
```

Phone:

+1 415 864 6466

EMail:

xaudio@mpegtv.com

URL:

http://www.xaudio.com

Contact:

Gilles Boccon-Gibod

Last modified:

December 23, 1997.

---

15. Network Servers

15.1 Critical Angle X.500 Enabler

Description:

The Critical Angle X.500 Enabler allows an LDAP-only directory server to be integrated into an X.500 directory service, such as the PARADISE international white pages directory or an enterprise messaging directory service. It supports all X.500(88) operations, including authentication.

Distribution media:

Internet (WWW).

Licensing policy:

Licensed per server.

OS provisions:

Reference platform is Red Hat Linux 3 or later for Intel.

Documentation:

English documentation is included.

Product support:

An upgrade subscription is also available.

Countries with distribution:

World wide.

Price range:

A single server license is US$495.

There is also (in 1997) a special version, PARADISE X.500 Enabler, exclusively for universities. This version does not include the support for authentication or modification operations; it is intended only for searching public information. A single server license of PARADISE X.500 Enabler is currently US$95.

Vendor:

Critical Angle Inc.

```
4815 W. Braker Lane #502-385
Austin TX 78759
U.S.A.
```

EMail:

info@critical-angle.com

URL:

http://www.critical-angle.com

Contact:

Mark Wahl

Last modified:

August 5, 1997.

15.2 DNEWS News Server

Description:

DNEWS is an advanced, full featured NNTP based news server. It is suitable for ISP's and Corporate Intranets. DNEWS can be used as a replacement for INN or cnews and will generally result in improved performance and much lower administration. Features include: easy installation and management, flexible expire settings, flexible access restrictions, EMail confirmation of posts, XOVER extentions implemented, streaming extentions built in, dynamic sucking feeds option, news to web gateway included.

Distribution media:

Internet (WWW).

Licensing policy:

Commercial, DNEWS is try before you buy software and can be downloaded from our web site for free trial.

OS provisions:

No special requirements

Documentation:

A user manual is provided in HTML format. Online help included.

Product support:

Supported by Netwin LTD.

Extra features and add-ons:

DNEWSWEB - A news to web gateway. Allows local and Usenet news groups to be added to ordinary HTML pages and allows users to read and post to local and Usenet news groups directly using any web browser.

Available since:

1995.

Countries with distribution:

World wide.

Price range:

US$485 (Free registration for schools and universities).

Vendor:

Netwin LTD

```
P.O Box 27574
MT Roskill
Auckland
New Zealand
```

Fax:

+64 9 6300 689

EMail:

netwin@netwinsite.com

URL:

http://netwinsite.com

Contact:

Stephen Pugmire

Last modified:

August 4, 1997.

15.3 Aventail Internet Policy Manager

Description:

Commerical SOCKS v5 server for Linux with support for multiple authentication methods, sophisticated access controls, datastream filtering, and intelligent auditing capabilities.

Distribution media:

CD-ROM and Internet (FTP).

Licensing policy:

Commercial, free demo availalbe from http://www.aventail.com/download.html

OS provisions:

Linux 2.x, 32 MB of RAM recommended, administration tool requires X11R6 or later.

Approximately 14 MB of hard disk space required (less if you have Perl 5.003 and Perl/Tk installed).

Documentation:

English documentation available in PDF or Microsoft Word format.

Product support:

90 days free technical support. Yearly maintenance is 20% of your total licence cost. Maintenance includes product upgrades.

Extra features and add-ons:

Various authentication and filtering plugins to add functionality at run-time.

Available since:

November, 1996.

Countries with distribution:

World wide. VPN products avaiable only in 40 bit encryption strength outside of the USA or Canada due to export restrictions.

Price range:

IPM starting at US$6.495. VPN starting at US$7.995.

Vendor:

Aventail Corporation

```
808 Howell Street, 2nd Floor
Seattle, WA 98101
U.S.A.
```

Phone:

+1 888 762 5785, +1 206 215 1111

Fax:

+1 206 215 1120

EMail:

info@aventail.com

URL:

http://www.aventail.com

Last modified:

June 23, 1998.

15.4 Aventail VPN

TOC

Description:

Aventail VPN runs on Linux, Unix and NT platforms. Aventail VPN features multiple authentication and encryption methods, sophisticated access controls, intelligent filtering and makes no modifications to the underlying client or server operating system. Authentication modules include username and password, CHAP, RADIUS, SSL, digital certificates and token-based security. Encryption modules include DES, Triple DES and MD 4/5. Aventail VPN's access controls allow user permission to be grated based on variables such as the user's role within a company, group association, intended destination, type of application, type of encryption and/or authentication, or even more customized information. Aventail VPN is the only product to work on top of PPTP, IPSec, and L2TP.

Distribution media:

CD-ROM.

Licensing policy:

60 day evaluation period; pricing based on server connections and quantity of clients.

Product support:

All products ship with an automatic 90 day warranty and dial-up telephone technical support. To extend your coverage, you may purchase the Annual Support Program. The Support Program is 20% of your total license cost, annually renewable.

Available since:

May, 1997.

Countries with distribution:

World wide.

Price range:

Starts at US$6,995; Pricing based on connections.

Vendor:

Aventail Corporation

```
808 Howell Street, 2nd Floor
Seattle, WA 98101
U.S.A.
```

Phone:

+1 206 215 1111

Fax:

+1 206 215 1120

EMail:

info@aventail.com

URL:

http://www.aventail.com

Contact:

Deanna Leung

Last modified:

June 23, 1998.

15.5 WANPIPE

Description:

WANPIPE integrates into the Linux routing stack to provide IP and IPX routing over Frame Relay, PPP or X.25 networks at line speeds to 2.048Mbps. WANPIPE has either V.35, EIA530, X.21 or RS232 interfaces, or a built in T1 and Fractional T1 DSU/CSU. Extensive debugging and monitoring features make installation and management easy.

Distribution media:

Internet (FTP).

Licensing policy:

Linux source freely distributed.

OS provisions:

All versions of Linux.

Documentation:

Provided with source for Linux.

Product support:

Supported by Sangoma, unlimited.

Extra features and add-ons:

Optional integrated T1/Ft1 DSU/CSU, choice of V.35, X.21, EIA530 or RS232 cabling.

Available since:

1995.

Countries with distribution:

US, UK, Israel, Poland, Canada, Germany, Korea, China.

Price range:

US$589-$859.

Vendor:

Sangoma Technologies Inc.

```
1001 Denison Street, Suite 101
Markham, ON L3R 2Z6
Canada
```

Phone:

+1 905 474 1990, +1 800 388 2475

Fax:

+1 905 474 9223

EMail:

dm@sangoma.com

URL:

http://www.sangoma.com

FTP:

ftp://ftp.sangoma.com/pub/linux

Contact:

David Mandelstam

Last modified:

May 8, 1998.

15.6 Zeus Web Server

Description:

Scalable web application server for ISP hosting, Ontranets and secure e-Commerce. Full Frontpage support, 128 bit SSL3 world wide, Apache compatibility, web-centric delegatable management interface.

Application development in ISAPI, Java-Servlets and CGI. Distributed server-side Java gives high performance database connectivity.

Unlimited virtual servers with groupable configuration, real-time statistics, bandwidth throttling and extensive logging.

Distribution media:

Internet (WWW).

Licensing policy:

One License per machine running the software. Free 30 day evaluations are available from the site.

Documentation:

Online help, available in Germany and Spanish.

Price range:

Academic: £50. Commercial: £999.

Vendor:

Zeus Technology Limited

```
St Johns Innovation Centre, Cowley Road
Cambridge, CB4 4WS
UK
```

Phone:

+44 1223 42 17 27

Fax:

+44 1223 42 17 31

EMail:

sales@zeustech.net

URL:

http://www.zeustech.net

Contact:

Bryan Amesbury <bames@zeustech.net>

Last modified:

December 23, 1997.

---

16. Office Tools

16.1 Corel WordPerfect 8

Description:

WordPerfect 8 for Linux Personal Edition is a world-class word processor, native on Linux. It offers the same outstanding tools, intuitive graphical interface and file formatas the Windows version. WordPerfect 8 for Linux imports over 40 document types (including Microsoft Word97), and supports over 1,000 different printers.

WordPerfect 8 for Linux Server Edition includes everything from the Personal Edition, plus a character terminal binary and several powerful utilities, such as background merge and print. Server Edition also offers concurrent licensing for business use.

WordPerfect 8 for Linux (both editions) offer localized versions in English (US), English (UK), English (Canada), English (Australia), French (national), French (Canada), German, Italian, Spanish, and Dutch.

Distribution media:

CD-ROM and Internet (WWW).

Licensing policy:

WordPerfect 8 for Linux Personal Edition: licensing is per machine. WordPerfect 8 for Linux Server Edition: licensing is concurrent use.

OS provisions:

Kernel 2.0.x and later stable versions; libc. For X Windows, 9 MB of RAM for first user, 2.5 MB RAM for each additional user, 43-104 MB hard disk space (depending on install options). For Character terminals, 2 MB RAM for first user, 1 MB RAM for each additional user, 31-40 MB hard disk space (depending on install options).

Documentation:

Printed manuals: GUI binary, character terminal binary (approximately 750 pages each). Online (Adobe Acrobat format) manuals available for GUI binary.

Product support:

WordPerfect 8 for Linux Personal Edition: self-support (knowledge base, discussion groups, etc.), EMail support, telephone toll support. WordPerfect 8 for Linux Server Edition: 30 day up-and-running telephone support, EMail support; additional support and maintenance packages are available.

Extra features and add-ons:

100's of Language modules (provided by Corel, these provide keyboard templates, spelling dictionaries, and other tools needed to create documents in other languages). Medical spelling dictionaries (provided by third parties--contact info@sdcorp.com for details).

Available since:

September, 1998.

Countries with distribution:

World wide.

Price range:

US$39.95-$495.

Vendor:

Software Development Corporation

```
512 East 1860 South
Provo, UT 84606
U.S.A.
```

Phone:

+1 801 370 9600

Fax:

+1 801 370 9696

EMail:

info@sdcorp.com

URL:

http://www.sdcorp.com

Contact:

John Gleave

Last modified:

August 15, 1998.

16.2 The American Heritage Dictionary Deluxe

Description:

This long-awaited release is ideal for technical writers, authors, and researchers. Previously available only on PC's, the product offers the ultimate reference tool for Linux users, with full definitions for well over 200,000 words, plus pronunciations, derivations, proper usage, idioms and sample sentences. The world reknowned Roget's 500,000 word Thesarus is fully integrated

with the AHD. Browse, word and topic search, anagrams and more. Single user, network, academic and site licenses are available for Linux and key Unix platforms.

Distribution media:

CD-ROM, tapes, Internet (FTP and WWW).

Licensing policy:

Single user node locked or server license (shared). A free unclockable demo is available at our web site.

OS provisions:

Motif GUI, 8-16 MB RAM, 15 MB harddisk or run from the CD-ROM.

Documentation:

Documentation as a help text file. English language.

Product support:

Maintenance available. Call DUX for details.

Extra features and add-ons:

Addlitional reference volumes and additional language support to be announced.

Available since:

February, 1997.

Countries with distribution:

Currently sold in the USA and Germany. However, sales are made internationally via unclockable FTP/WWW copies and sold with authorized major credit card.

Price range:

US$49.95 for node lock single user. Call DUX or an authorized reseller for sliding scale or site license pricing.

Vendor:

DUX Software Corporation

```
425 Sherman Avenue Ste 330
Palo Alto, CA 94306
U.S.A.
```

Phone:

+1 800 543 4999, +1 415 473 1800

Fax:

+1 415 462 8723

EMail:

sales@dux.com

URL:

http://www.dux.com

Contact:

Bob Adams, Rob DuFrane

Last modified:

August 7, 1997.

16.3 Applixware Office Suite for Linux

Description:

Fully-graphical, fully-integrated office suite for Linux. Includes word processor, spreadsheet, presentation graphics, mail, database client, and HTML authoring software. Also includes Builder, a graphical rapid application development program.

Distribution media:

CD-ROM.

Licensing policy:

Per concurrent user license.

OS provisions:

Kernel 1.2.13 or better, XFree86 3.1.2 or better, 16 MB RAM, 150 MB hard disk.

Documentation:

Make-It-Happen Manual. Online documentation in English, French, and German.

Product support:

Paid support avaailable. Users can subscribe to mailing list applixware-list@applix.com, and see archives and FAQ's at http://linux.applixware.com.

Extra features and add-ons:

French and German interface, international dictionaries.

Available since:

July, 1996.

Countries with distribution:

North America, Europe, Japan, Australia.

Price range:

US$99, upgrades for US$79.

Vendor:

Applix, Inc.

```
112 Turnpike Road
Westboro, MA 01581
U.S.A.
```

Phone:

+1 508 870 0300

EMail:

mrlinux@applix.com

URL:

http://linux.applixware.com

Contact:

Richard Manly <rmanly@applix.com>

Last modified:

October 1, 1998.

16.4 D.M.S. Document Management System

Description:

The system collects documents in Tiff-g4, PostScript and JPEG and organizes them in folders with record information like names and descriptions. The system can be access using any web broswer: viewing, modify, full text retrieval and faxing are provided by a web interface. New documents are made by printing to printer spooler.

Licensing policy:

The system is installed by vendor on your Linux box with a server licence, unlimited clients, unlimited document number.

OS provisions:

Linux 2.0.

Documentation:

HTML.

Extra features and add-ons:

SQL server integrationm, print capture from accounting software.

Price range:

Upon request.

Vendor:

Studio LEADER Pro

```
Via Pietrastretta, 76
I-38100 Trento
Italy
```

Phone:

+39 461 828229

Fax:

+39 461 829826, +39 461 829877

EMail:

DMS@leader.it

URL:

http://www.leader.it

Contact:

Guido Brugnara <brugnara@leader.it>

Last modified:

August 19, 1997.

16.5 HotWire EasyFAX

Description:

Fully functional GUI Fax Management Program.

Distribution media:

Internet (WWW).

Licensing policy:

Free fully functional eval from WWW site..

OS provisions:

5.5 MB Disk, 4 MB RAM.

Documentation:

Online context-sensitive help.

Product support:

Via EMail and Usenet only.

Available since:

January 1, 1997.

Countries with distribution:

USA/Canada (Unisource Systems, Inc.), Germany/Europe (Delix Computer, GmbH).

Price range:

US$99.

Vendor:

Unisource Systems, Inc.

```
1409 N. Cove Boulevard
Longwood, FL 32750
U.S.A.
```

Phone:

+1 407 834 1973

Fax:

+1 407 834 8013

EMail:

sales@unisrc.com

URL:

http://www.unisrc.com

Contact:

Gary Heller

Last modified:

October 3, 1997.

16.6 NExS, the Network Extensible Spreadsheet

Description:

NExS is a full-featured graphical spreadsheet package which is designed to take advantage of the advanced features of Unix and the X Window system. It includes more than 230 computational functions and 13 2D and 3D graph types for engineering, scientific and business applications. NExS is extensible through a network aware API that allows user-defined "plug-ins" to add new features and functions in a variety of programming languages. NExS software plug-in bus includes 63 "slots" for remote processes to connect and extend NExS' basic functionality, allowing for the

development of sophisticated networked applications.

Distribution media:

Internet (FTP and WWW).

Licensing policy:

The NExS Personal Edition license is for a single user and single platform type. The owner may use the license on multiple machines. The Professional Edition license is a floating network license which allows a pool of licenses to be used on any supported platform in your network. Without a license, NExS runs in "demo mode" with limited functionality.

OS provisions:

NExS runs on Linux kernels which support ELF binaries. There are two flavors for Linux, one which requires Motif 2.0 shared libraries, and one which includes Motif 2.0 statically linked.

Documentation:

The NExS User Guide is available in PostScript (letter format, 431 pages, or A4 format, 403 pages), and in HTML. API and tclNExS documentation are available separate from the User Guide, also in PostScript and HTML format. Printed documentation is also available. All documentation is in English.

Product support:

EMail support is standard at no cost. Phone-in support is available for a charge. Bug fix and minor version releases (e.g., 1.2.x -> 1.3.0) are no cost. Major releases are substantially discounted for existing licensees.

Extra features and add-ons:

NExS plug-ins and user-contributed code are available at http://www.xess.com/plugins.html.

Available since:

1994.

Countries with distribution:

World wide.

Price range:

- US$149 ($99 academic) for Personal Edition
- US$249 ($125 academic) for Professional Edition

Vendor:

X Engineering Software Systems (XESS) Corp.

```
P.O. Box 33091
Raleigh, NC 27636
U.S.A.
```

Phone:

+1 800 961 7840, +1 919 387 0076

Fax:

+1 919 387 1302

EMail:

info@xess.com

URL:

http://www.xess.com

FTP:

ftp://ftp.vnet.net/pub/users/xess/NExS

Contact:

Dave Van den Bout <devb@xess.com>

Last modified:

March 15, 1997.

16.7 Axene Office

Description:

Axene Office suite includes Xclamation (Desktop Publishing), XQuad (Spreadsheet), XAllWrite (Word Processor) and XMayday (HTML Documentation browser). Available in English, French, German and Spanish.

Distribution media:

CD-ROM, tape, Internet (FTP).

Licensing policy:

Workstation or floating licenses. Downloadable demo version by FTP.

OS provisions:

8-16 MB RAM, 10 MB harddisk or run from the CD-ROM. a.out or ELF version.

Documentation:

Online HTML help and printed documentation.

Available since:

1998.

Countries with distribution:

World wide.

Price range:

US$49.

Vendor:

AXENE, Inc.

```
30 Montgomery street, suite 604
Jersey City, NJ 07302-3821
U.S.A.
```

Phone:

+1 201 434 4244

EMail:

info@axene.com

URL:

http://www.axene.com/english/showroom.html

FTP:

ftp://ftp.axene.com/pub

Last modified:

February 8, 1998.

16.8 Projector and Projector/Net

Description:

Produces project schedules and Gantt charts using critical path analysis from list of activities (tasks) with their associated durations and prerequisites. Projector produces sleek printed output and Projector/Net produces HTML for web clients.

Distribution media:

3.5" diskettes.

Licensing policy:

Single user and server licenses. There is a demo version available for downloading from our web site.

OS provisions:

PC/Linux. Projector requires GhostScript or PostScript printer. Projector/Net requires only a client-side web browser capable of displaying graphics (Netscape and IE are fine). RAM requirements for both versions are (very roughly) load module size plus 44 bytes/activity plus 32 bytes per prerequisite activity specification plus size of input project description file (excluding embedded comments). Tested only with Red Hat Linux 4.2. Projector/Net tested only with

Linux/Apache server with Internet Explorer and Netscape on the client side.

Documentation:

All contained in the product distribution README and sample project description file (English).

Countries with distribution:

World wide.

Price range:

US$200 for Projector. US$2500 for Projector/Net.

Vendor:

IEDU

```
Morris Dovey
1001 Office Park Road, Suite 300
West Des Moines, IA 50265
U.S.A.
```

Phone:

+1 515 225 1636

EMail:

mrdovey@iedu.com

URL:

http://www.iedu.com/project

Contact:

Morris Dovey

Last modified:

May 8, 1998.

16.9 The Virtual Office System

Description:

Internet capable client/server, enterprise document imaging and management. Scan directly to Linux host from Window client without using EMail or FTP.

Distribution media:

3.5" diskettes, tape and Internet (FTP).

Licensing policy:

Free Windows client demo available from our web site. 30 day evaluation of enterprise server

available upon request.

OS provisions:

Server: Linux 2.0. Client: Windows 95 and Windows NT Desktop.

Documentation:

Online help and printed administrator's manual.

Extra features and add-ons:

Windows file archiving.

Price range:

US$4995 - 5 user, 1 scanner, enterprise edition.

Vendor:

CONTEXT Information Systems, Ltd.

```
314 S. 2nd Street
Oakland, MD 21550
U.S.A.
```

Phone:

+1 301 334 6550

EMail:

sales@contextis.com

URL:

http://www.contextis.com

Last modified:

June 13, 1998.

16.10 Axene XAllWrite

Description:

The word-processing application Axene XAllWrite permits the user to create excellent quality documents with ease, whether a simple letter or a large, complex publication. It features tools for multi-document management, bitmap or line-art image integration, various font capabilities, and HTML format exportation. Available in English, French, German and Spanish.

Distribution media:

CD-ROM, tape, Internet (FTP).

Licensing policy:

Workstation or floating licenses. Downloadable demo version by FTP.

OS provisions:

8-16 MB RAM, 10 MB harddisk or run from the CD-ROM. a.out or ELF version.

Documentation:

Online HTML help and printed documentation.

Available since:

1998.

Countries with distribution:

World wide.

Price range:

US$25.

Vendor:

AXENE, Inc.

```
30 Montgomery street, suite 604
Jersey City, NJ 07302-3821
U.S.A.
```

Phone:

+1 201 434 4244

EMail:

info@axene.com

URL:

http://www.axene.com/english/xallwrite.html

FTP:

ftp://ftp.axene.com/pub

Last modified:

February 8, 1998.

16.11 Axene Xclamation

Description:

Xclamation has multidocument capabilities, multi-column and gutters capabilities, magnetic ruler marks, integration of text, bitmap or vectorized images within polymorphous frames, complex editing operations on frames (including logical, zooming, outlining, rotation, background

transparency, etc.), and text exportation in HTML format. Available in English, French, Spani and German.

Distribution media:

CD-ROM, tape, Internet (FTP).

Licensing policy:

Workstation or floating licenses. Downloadable demo version by FTP.

OS provisions:

8-16 MB RAM, 10 MB harddisk or run from the CD-ROM. a.out or ELF version.

Documentation:

Online HTML help and printed documentation.

Available since:

1996.

Countries with distribution:

World wide.

Price range:

US$25.

Vendor:

AXENE, Inc.

```
30 Montgomery street, suite 604
Jersey City, NJ 07302-3821
U.S.A.
```

Phone:

+1 201 434 4244

EMail:

info@axene.com

URL:

http://www.axene.com/english/xclamation.html

FTP:

ftp://ftp.axene.com/pub

Last modified:

February 8, 1998.

16.12 Axene XQuad

TOC

Description:

XQuad features all the necessary functions for scientific and financial calculations and for graphical presentation of numerical data, including over 100 mathematical, logical, and string manipulation functions, various types of graphs (from histograms to radar), importation of external data files (text, Excel, etc.), production tools which allow text in cells to have automatic effect on graphical presentation. Available in English, French, German and Spanish.

Distribution media:

CD-ROM, tape, Internet (FTP).

Licensing policy:

Workstation or floating licenses. Downloadable demo version by FTP.

OS provisions:

8-16 MB RAM, 10 MB harddisk or run from the CD-ROM. a.out or ELF version.

Documentation:

Online HTML help and printed documentation.

Available since:

1996.

Countries with distribution:

World wide.

Price range:

US$25.

Vendor:

AXENE, Inc.

```
30 Montgomery street, suite 604
Jersey City, NJ 07302-3821
U.S.A.
```

Phone:

+1 201 434 4244

EMail:

info@axene.com

URL:

http://www.axene.com/english/xquad.html

FTP:

ftp://ftp.axene.com/pub

Last modified:

February 8, 1998.

---

17. Text Processing

17.1 Edith Pro for X11

Description:

Edith Pro is a powerful and easy-to-use text editing package, for editing plain text such as EMail messages, HTML, LaTeX and programming languages. It is aimed at people who want a flexible editing tool but who are not prepared to spend much time into learning and configuring their application. It includes a fast text-only web browser with access to the system manual and info.

Distribution media:

Internet (WWW).

Licensing policy:

You are only allowed to download Edith from the ZFC web pages for *personal use*. Without a licence, Edith will regularly issue messages stressing that the copy used is not licenced. Personal use without a licence is allowed, but organizations are required to buy a licence.

OS provisions:

ELF binary. 1MB is sufficient for editing normally sized text.

Documentation:

Online HTML help covers the printed manual completely. The printed manual contains illustrations not in the online help.

Product support:

Free updates can be downloaded from the ZFC web pages. Licence holders will be notified of updates. Licence holders can contact ZFC by EMail for immediate support.

Available since:

January 1, 1997.

Countries with distribution:

World wide directly from ZFC.

Price range:

NLG 70 (approximately US$35) single host, NLG 400 (approximately US$200) full site licence. 50%50 discount for educational organizations and students.

Vendor:

ZFC

P.O. Box 15813
1001 NH Amsterdam
The Netherlands

Phone:

+31 20 4 208 248

EMail:

zfc@zfc.nl

URL:

http://www.zfc.nl

Contact:

Annius V. Groenink

Last modified:

August 4, 1997.

17.2 TeraSpell 97 for Emacs

Description:

TeraSpell 97 for Emacs is the user-friendly spell checker for Emacs which incorporates visual highlights for misspelled words and on-the-fly spell checking. TeraSpell 97 for Emacs sets a new standard of quality in spelling correction software by providing the technically most advanced spelling correction system available on the market.

TeraSpell eliminates the traditional tedious spelling dialog and speeds up the process of spell checking your documents. Visual highlights let you efficiently identify all spelling errors in your documents at once and ignore all non-words that you may have used in your document. TeraSpell can spell check your document as you type. As soon as you type an incorrect word, it becomes highlighted. By clicking right on a highlighted word, a menu of suggestions appears.

Distribution media:

Internet (EMail and WWW).

Licensing policy:

Evaluation version available for free from our web site. Single user license, 3, 5, 10, 20 users license. Site-wide and University wide licenses,

OS provisions:

Emacs 19.29.1 or higher. XEmacs 19.14 or higher.

Documentation:

Online documentation.

Product support:

EMmail support. WWW Updates. FAQs on WWW.

Extra features and add-ons:

Additional languages available.

Available since:

August, 1997.

Countries with distribution:

World wide.

Price range:

US$29.99 single user student Linux license. US$89 single user Linux, Solaris, Irix, HPUX.

Vendor:

Teragram Corporation

```
236 Huntington Avenue
Boston MA 02115-4701
U.S.A.
```

Phone:

+1 617 369 0100

Fax:

+1 617 369 0101

EMail:

info@teragram.com

URL:

http://www.teragram.com

Contact:

Yves Schabes

Last modified:

August 16, 1997.

18. System Administration Tools

18.1 Host Factory

Description:

Host Factory is a system to control the entire software contents of deployed Unix machines. Host Factory lets you customize one prototype Unix system and use it to build hundreds of similar systems. You can deliver incremental changes to running systems and roll them back if they fail. Changes are tracked in a version-control filesystem integral to the product. Host Factory provides an environment to do configuration management and version control of complete hosts.

Licensing policy:

Commercial and GPL.

Documentation:

Printed documentation.

Product support:

EMail.

Available since:

1996.

Countries with distribution:

World wide.

Price range:

Upon request. Special prices for educational institutions.

Vendor:

Working Version

```
31 Shea Road
Cambridge, MA 02140
U.S.A.
```

EMail:

bb@wv.com

URL:

http://www.wv.com

Contact:

Brian Bartholomew

Last modified:

July 22, 1997.

18.2 PerfectBACKUP+

Description:

PerfectBACKUP+ is the Unix/Linux world's premier Backup/Restore utility. It is the worlds fastest backup/restore for Unix and has been since 1985. Previously sold in the US as FASTBACK PLUS for Unix, the counterpart to the DOS FASTBACK PLUS, it now has character and Motif interfaces, networking support, compression, the best verification and recovery, scheduling and is compatible with both tar and cpio. Backs up all files, devices, partitions etc. and can backup Windows, Netware and Windows NT drives too.

Distribution media:

3.5" diskettes, Internet (WWW), 4/8mm DAT and soon CD-ROM.

Licensing policy:

Free evaluation copy from our web site. Free online manual from same the same address. Price per CPU, unlimited users.

OS provisions:

Kernels 1.2.x and 2.0.x, X11R6, Motif 1.2, 3 MB of RAM, 3 MB disk.

Documentation:

Printed (188 pages), PostScript online, and online help. English language documentation, but help by phone and EMail also avilable in Spanish.

Product support:

Unlimited free EMail support and phone support.

Extra features and add-ons:

Comes with free PC program to read tar/cpio diskettes onto PC. AutoChanger add-on with tape recycling for high-end shops.

Available since:

1996.

Countries with distribution:

USA, Canada, Germay, UK, Switzerland, Singapore, Malaysia, Germany.

Price range:

UD$79 to US$499. Linux off the net (no media) is US$79 and HP/UX is US$499. Always full

product.

Vendor:

Unisource Systems, Inc.

```
1409 North Cove Boulevard
Longwood, FL 32750
U.S.A.
```

Phone:

+1 407 834 1973

Fax:

+1 407 834 1973

EMail:

sales@unisrc.com

URL:

http://www.unisrc.com

Contact:

Gary Heller

Last modified:

October 3, 1997.

18.3 Venus

Description:

Venus is an distributed administration tool for any network built of Unix workstations from different hardware vendors. A Venus network presents itselfs to the user as a homogeneous computer system and guarantees transparent access to any network resource. It provides software tools for major cluster management tasks: configuration management, software distribution, user and filesystem administration.

Licensing policy:

Commercial.

OS provisions:

Currently build on top of the S.u.S.E distribution.

Documentation:

Printed doumentation, online help.

Product support:

TOC

Hotline, EMail, courses.

Available since:

1994.

Countries with distribution:

World wide.

Price range:

Upon request. Special prices for educational institutions.

Vendor:

science + computing GmbH

```
Hagellocher Weg 71
D-72070 Tübingen
Germany
```

Phone:

+49 7071 9457 0

Fax:

+49 7071 9457 27

EMail:

info@science-computing.de

URL:

http://www.science-computing.de

Contact:

Olaf Flebbe

Last modified:

February 25, 1997.

---

19. X Windows Related Products

19.1 Accelerated-X Display Server

Description:

Commercial grade X display server with workstation-class features like multiple visuals, overlays, and gamma correction for FreeBSD, and Linux. The AcceleratedX server drives the most popular graphics hardware, including the latest cards from ATI, Diamond, Matrox, NeoMagic, Number 9, and S3. In all, more than 490 different cards are supported.

Distribution media:

CD-ROM.

Licensing policy:

Per CPU.

OS provisions:

No known problems with any Linux distribution.

Documentation:

A 100 page perfect bound manual is included.

Product support:

Unlimited support in included.

Available since:

1994.

Countries with distribution:

World wide.

Price range:

Contact your local reseller, or Xi Graphics.

Vendor:

Xi Graphics

```
1801 Broadway, Suite 1710
Denver, CO 80202
U.S.A.
```

Phone:

+1 303 298 7478

Fax:

+1 303 298 1406

EMail:

sales@xig.com

URL:

http://www.xig.com

Contact:

Kyle Fink

Last modified:

August 14, 1997.

19.2 BXwidgets

Description:

BXwidgets is a supplementary widget set to develop especially commercial Motif applications. It includes special widgets for tabulars, masks, formatted input field and the full support for a complete help system. All has Motif look and feel and is platform independent.

Distribution media:

3.5" diskettes.

Licensing policy:

Source code licence.

OS provisions:

Any kernel, X11R5 or X11R6, Motif 1.1 or Motif 1.2.

Documentation:

About 250 pages of documenation with examples. Several demo programs.

Product support:

Support and update service is available (DM 1200; approximately US$750)

Available since:

1992.

Countries with distribution:

Germany, UK and USA.

Price range:

DM 1600 (approximately US$1000) for a full source code licence.

Vendor:

BREDEX GmbH

```
Fallersleber-Tor-Wall 23
D-38100 Braunschweig
Germany
```

Phone:

+49 531 24 33 0 0

Fax:

+49 531 24 33 0 99

EMail:

info@bredex.de

URL:

http://www.bredex.de

Last modified:

November 3, 1997.

19.3 BXwidgets/DB

Description:

BXwidgets/DB is a supplementary widget set to access relational databases. Data can be automatically queried form the database, displayed in various forms, changed and stored in the database. All applications built with BXwidgets/DB are database independent. The database access is performed by transaction-servers. Transaction-Server and database may reside on other hotsts in the network.

Distribution media:

CD-ROM.

Licensing policy:

Per OS license, no run-time restrictions. Evaluation license available.

OS provisions:

X11R5 and Motif 1.2 must be installed.

Documentation:

Printed manual in English.

Product support:

TOC

Support and update service is available.

Available since:

1996.

Price range:

DM 4900 (approximately US$2900).

Vendor:

BREDEX GmbH

```
Fallersleber-Tor-Wall 23
D-38100 Braunschweig
Germany
```

Phone:

+49 531 24 33 0 0

Fax:

+49 531 24 33 0 99

EMail:

info@bredex.de

URL:

http://www.bredex.de

Last modified:

November 3, 1997.

19.4 Laptop, Accelerated-X Display Server

Description:

Commercial grade X display server with workstation-class features like multiple visuals, overlays, and gamma correction for FreeBSD and Linux. The AcceleratedX server drives the most popular laptop hardware, including the latest cards from Dell, IBM, Toshiba, and Neo Magic.

Distribution media:

CD-ROM.

Licensing policy:

Per CPU.

OS provisions:

No known problems with any Linux distribution.

Documentation:

A 100 page perfect bound manual is included.

Product support:

Unlimited support in included.

Available since:

1994.

Countries with distribution:

World wide.

Price range:

Contact your local reseller, or Xi Graphics.

Vendor:

Xi Graphics

```
1801 Broadway, Suite 1710
Denver, CO 80202
U.S.A.
```

Phone:

+1 303 298 7478

Fax:

+1 303 298 1406

EMail:

sales@xig.com

URL:

http://www.xig.com

Contact:

Kyle Fink

Last modified:

August 14, 1997.

19.5 MaXimum cde Developer's Edition v1.0

TOC

Description:

A complete networked desktop that enables administrators to integrate industry-standard PCs into existing Unix networks with better internationalization and the same functionality as more expensive workstations. Also includes developer's tools.

Distribution media:

CD-ROM.

Licensing policy:

Per CPU.

OS provisions:

No known problems with any Linux distribution.

Documentation:

A 100 page perfect bound manual the Accelerated-X server and a 300 page perfect bound manual for CDE is included.

Product support:

Unlimited support in included.

Available since:

1997.

Countries with distribution:

World wide.

Price range:

Contact your local reseller, or Xi Graphics.

Vendor:

Xi Graphics

```
1801 Broadway, Suite 1710
Denver, CO 80202
U.S.A.
```

Phone:

+1 303 298 7478

Fax:

+1 303 298 1406

EMail:

TOC

sales@xig.com

URL:

http://www.xig.com

Contact:

Kyle Fink

Last modified:

August 14, 1997.

19.6 Multi-headed, Accelerated-X Display Server

Description:

Commercial grade X display server with workstation-class features like multiple visuals, overlays, and gamma correction for FreeBSD and Linux. The AcceleratedX server drives the most popular graphics hardware, including the latest cards from Colorgraphics, STB, and Matrox. In all, more than 10 different cards are supported.

Distribution media:

CD-ROM.

Licensing policy:

Per CPU.

OS provisions:

No known problems with any Linux distribution.

Documentation:

A 100 page perfect bound manual is included.

Product support:

Installation support in included.

Available since:

1994.

Countries with distribution:

World wide.

Price range:

Contact your local reseller, or Xi Graphics.

Vendor:

Xi Graphics

1801 Broadway, Suite 1710
Denver, CO 80202
U.S.A.

Phone:

+1 303 298 7478

Fax:

+1 303 298 1406

EMail:

sales@xig.com

URL:

http://www.xig.com

Contact:

Kyle Fink

Last modified:

August 14, 1997.

19.7 OpenGL, Accelerated-X Display Server

Description:

Commercial grade X display server with workstation-class features like multiple visuals, overlays, and gamma correction for FreeBSD and Linux. The AcceleratedX server drives the most popular graphics hardware, including the latest cards from ATI, Diamond, Matrox, NeoMagic, Number 9, and S3. The lastest version of OpenGL from Silicon Graphics Inc. In all, more than 450 different cards are supported.

Distribution media:

CD-ROM.

Licensing policy:

Per CPU.

OS provisions:

No known problems with any Linux distribution.

Documentation:

A 100 page perfect bound manual is included.

Product support:

Installation support in included.

Available since:

1994.

Countries with distribution:

World wide.

Price range:

Contact your local reseller, or Xi Graphics.

Vendor:

Xi Graphics

```
1801 Broadway, Suite 1710
Denver, CO 80202
U.S.A.
```

Phone:

+1 303 298 7478

Fax:

+1 303 298 1406

EMail:

sales@xig.com

URL:

http://www.xig.com

Contact:

Kyle Fink

Last modified:

August 14, 1997.

19.8 OSF-Certified Motif

Description:

A complete OSF-Certified Motif runtime and development environment.

Distribution media:

CD-ROM.

Licensing policy:

TOC

Per CPU.

OS provisions:

No known problems with any Linux distribution.

Documentation:

Release note that include references to well-written commerical reference manuals.

Product support:

Installation support in included.

Available since:

1995.

Countries with distribution:

World wide.

Price range:

Contact your local reseller, or Xi Graphics.

Vendor:

Xi Graphics

```
1801 Broadway, Suite 1710
Denver, CO 80202
U.S.A.
```

Phone:

+1 303 298 7478

Fax:

+1 303 298 1406

EMail:

sales@xig.com

URL:

http://www.xig.com

Contact:

Kyle Fink

Last modified:

August 14, 1997.

20. Other Software

20.1 ABACUS 4

Description:

The ABACUS 4 process control system is a block based software system for the real time control of industrial and other continuous and batch processes.

Distribution media:

Available at http://www.arosnet.se/bbab/abacus-4.980708.tar.gz

Licensing policy:

Per machine license.

OS provisions:

Tested on Slackware 3.4, expected to run on any Linux distribution for Linux.

Documentation:

Reference Manual is in HTML and included with the product. Can be seen at http://www.arosnet.se/bbab/abacusmanual/index.html

Product support:

EMail support.

Available since:

July 1998.

Countries with distribution:

World wide.

Price range:

US$1995 per CPU, discounts for quantity.

Vendor:

McHedlund Associates

```
c/o Bristol Babcock AB
Fallhammargatan 2a
S-72133 Västerås
Sweden
```

Phone:

+46 21 350450

Fax:

+46 21 350455

EMail:

abacusabacus@hotmail.com, abacus4@geocities.com

URL:

http://www.geocities.com/Eureka/Enterprises/2275/abacus4.html

Contact:

Frank McNamara

Last modified:

July 12, 1998.

20.2 BBBS

Description:

Complete BBS package with full internal FidoNet and Internet support. ISDN, mailer, mail processor, TICK, AllFix, telnet, FTP, finger, news, EMail, gateway, offline, VT320, ANSI, multilingual, multinode, Mg, full screen editor, multitask, CD-ROM, HYDRA, Zmodem, Kermit, BZLink, TCP/IP, voice, FAX, charsets, accounts, IRC, plus much much more. Everything you need to run a BBS!

Distribution media:

3.5" diskette, Internet (FTP and WWW).

Licensing policy:

Per node basis. Shareware version can be downloaded from http://www.bbbs.net.

OS provisions:

Current version, 3.33 How, is for libc5. Next version and current beta is for libc6.

Documentation:

ASCII and HTML documentation, English.

Extra features and add-ons:

Lot of 3rd party utilities.

Countries with distribution:

World wide.

Price range:

Two node license US$60.

Vendor:

Foobar Oy

```
Paavolankatu 3 D 34
FIN-20240 Turku
Finland
```

Phone:

+358 2 240 1513

Fax:

+358 2 240 7755

EMail:

b@bbbs.net

URL:

http://www.bbbs.net

Contact:

Kim Heino

Last modified:

August 2, 1998.

20.3 Clustor

Description:

Clustor is a program for managing large computational tasks. Clustor greatly simplifies a common computationally intensive activity - running the same program code numerous times with different inputs. Clustor provides increased performance by distributing jobs over a network of computers and improved task management through a friendly user interface.

Targeted at users who run computationally demanding tasks, Clustor supports all phases of running such a task: task preparation, job generation and job execution. With Clustor, users who do not have extensive knowledge of programming parallel applications, such as scientists, engineers, researchers, are able to utilize the power of networked computers.

Distribution media:

Internet (WWW). Optional CD-ROM.

Licensing policy:

A free demo licenses with limited functionality, 30 days evaluation licenses with complete

functionality from our web site.

OS provisions:

Linux 1.2 or higher.

Documentation:

Web site, optional printed documentation.

Product support:

EMail support.

Extra features and add-ons:

Features for load monitoring and resource sharing, network installation program.

Available since:

February, 1997.

Countries with distribution:

World wide

Price range:

Starting at US$495.

Vendor:

Active Tools Inc.

```
246 First Street, Suite 310
San Francisco, CA 94105
U.S.A.
```

Phone:

+1 415 882 7062

Fax:

+1 415 680 2369

EMail:

info@activetools.com

URL:

http://www.activetools.com

Contact:

Rok Sosic, Sergij Foski

Last modified:

August 7, 1997.

20.4 FootPrints

TOC

Description:

The first web-based helpdesk software system. Tracks problems/solutions, trouble tickets -- whatever information you need to manage -- and makes it available via the World Wide Web to your customers or users. Resides on Unix or NT Web server; platform independent access using any web browser.

Distribution media:

Internet (FTP and WWW). Also available: 3.5" diskettes and tape.

Licensing policy:

Free eval available from http://www.unipress.com/cgi-bin/free_evals or ftp://ftp.unipress.com/pub/free_evals.

Pricing is on a per-user basis.

OS provisions:

Runs with any Linux running a web server.

Documentation:

Online manual/help.

Product support:

15% of software purchase price charged for maintenance and updates. Technical support via web and phone is free.

Available since:

January 1, 1997.

Countries with distribution:

World wide.

Price range:

US$1995 per starter pack (server software plus 3 user licenses). Licenses may be Group (a number of users can access your helpdesk, e.g. all your customers use one Group license) or Individual (helpdesk engineers and administrators would each require one Individual license).

Vendor:

UniPress Software, Inc.

```
2025 Lincoln Highway
Edison, NJ 08817
U.S.A.
```

Phone:

+1 800 222 0550, +1 732 287 2100

Fax:

+1 732 287 4929

EMail:

info@unipress.com

URL:

http://www.unipress.com/footprints

Contact:

Sue Glassberg <sue@unipress.com>

Last modified:

August 4, 1997.

20.5 Aladdin Ghostscript

Description:

PostScript, PDF, PCL5e (2Q97), PCL5c (2Q97), and PCL XL interpreter. Provides X Windows previewing, conversion to half a dozen raster file formats, and output to dozens of printers. Runs in all OS environments, not just Linux, and on any processor (32- or 64-bit) with a C compiler. Can drive black-and-white, RGB, and CMYK devices, both bilevel and continuous-tone. For more information, see http://www.cs.wisc.edu/~ghost/index.html. Note that Aladdin Ghostscript is an OEM product only: end-user licensing is not available (and not necessary, since the product is free for end users), nor is end-user support.

Distribution media:

3.5" diskettes.

Licensing policy:

Normally per-CPU. License normally includes source code, upgrades, and support.

OS provisions:

X11R5 or later is required for X Windows previewing. OS-specific modules are available for Unix (including Linux), VMS, MS Windows (including Windows NT), OS/2, and Macintosh. Porting to other OSs is simple: all OS dependencies are segregated in a single file of typically less than 100 lines of code.

Documentation:

Online documentation only, in English. Currently in plain ASCII text; texinfo and HTML versions are in preparation.

Product support:

Normally included in license; by EMail, phone, and fax.

Extra features and add-ons:

Each input language is priced separately, with a discount for multiple languages.

Available since:

1988 as free software, 1991 commercially.

Countries with distribution:

World wide.

Price range:

Negotiated for each customer.

Vendor:

Artifex Software Inc.

```
454 Las Gallinas Avenue, Suite 108
San Rafael, CA 94903
U.S.A.
```

Phone:

+1 415 492 9861

Fax:

+1 415 492 9862

EMail:

info@arsoft.com

URL:

http://www.cs.wisc.edu/~ghost/index.html

Last modified:

September 30, 1997.

20.6 Magician

Description:

Magician is a high performance Java interface to native OpenGL implementations allowing developers to write sophisticated OpenGL applications in Java.

Distribution media:

CD-ROM and Internet (WWW).

Licensing policy:

TOC

Free for personal use only. However, personal users may acquire a licence affording faster technical support and the software for all platforms on CD and a printed manual. Commercial users are required to licence the product for use. Source code licences are also available.

OS provisions:

Runs on Linux (libc5 and glibc) based systems.

Documentation:

Freely downloadable javadoc API reference. Freely downloadable comprehensive Programmer's Guide.

Product support:

Support for Magician is available via EMail, phone and fax. Mailing lists exist for inter-developer discussions. The purchase of licences (personal or commercial) gives you higher priority support.

Extra features and add-ons:

Support for all OpenGL 1.1 GL and GLU calls. Many features of Aux and GLUT have also been translated to work within a Java environment. Built-in, easy-to-use multiple simultaneous view support. Multi-threaded rendering. Support for direct OpenGL calls or polymorphic methods.

Available since:

June 10, 1998.

Countries with distribution:

World wide.

Price range:

US$35 for a Personal Runtime Licence. US$200 per set for Commercial Runtime Licences. Source licences pricing available on request.

Vendor:

Arcane Technologies Ltd.

```
P.O. Box 3738
Glasgow, G41 4YD
Scotland
UK
```

Phone:

+44 141 423 3449

Fax:

+44 141 423 3449

EMail:

magician-sales@arcana.co.uk

URL:

http://www.arcana.co.uk/products/magician

Contact:

Alligator Descartes

Last modified:

July 12, 1998.

20.7 journyx WebTime

Description:

A web based time and attendance tracking software product. You can record and report on employee time worldwide using the power of the Internet, or across your entire enterprise on your Intranet.

Distribution media:

3.5" diskettes and Internet (FTP).

Licensing policy:

Free for first 60 days of use, US$1000 plus US$25 per user thereafter.

OS provisions:

Runs on any version of Linux.

Documentation:

All documentation is in HTML and included online with the product.

Product support:

A maintenance contract is 15% of the purchase price per annum.

Extra features and add-ons:

Works with any browser including text based browsers such as lynx. Easy export of data to Microsoft Excel or Quickbooks. Free webserver and SQL database engine included in the package.

Available since:

August 1, 1997.

Countries with distribution:

World wide.

Price range:

U$1000-$10,000.

Vendor:

journyx

```
6716 Beauford Drive
Austin, TX 78750
U.S.A.
```

Phone:

+1 512 345 8282

Fax:

+1 512 342 9379

EMail:

info@journyx.com

URL:

http://journyx.com

Contact:

Curt Finch

Last modified:

October 24, 1997.

20.8 LanSafe

Description:

LanSafe III is a UPS Power Management application. It provides automatic orderly shutdown functionality in case of an extended power failure that should outlast the UPS battery run time. LanSafe III enables broadcast messages and EMail to be sent according to user defined power condition changes.

LanSafe III is bundled with the following Exide Electronics UPSs: OneUPS Plus, NetUPS, PowerWare Prestige, PowerWare Profile, PowerWare Plus 5xx. It also ships with FPS Power Systems UPSs: PowerRite Plus, PowerRite Max, PowerWorks A30, PowerWorks A40, Series 9000 and Series 10000.

It is also possible to purchase a separate software license to use with a previous UPS model or an other manufactures UPS.

Distribution media:

CD-ROM.

OS provisions:

Linux kernel 2.0. Both X/Motif based and character based user interfaces provided.

TOC

Documentation:

Online help. Online manual.

Product support:

Available.

Extra features and add-ons:

Support for other manufacturer's UPS.

Available since:

May, 1997.

Countries with distribution:

World wide.

Price range:

US$149.

Vendor (USA):

Exide Electronics

```
8609 Six Forks Road
Raleigh, NC 27615
U.S.A.
```

Vendor (Europe/Middle East/Africa):

Exide Electronics S.A.

```
MPL House, Prescott Road
Poyle, Colnbrook
Berkshire, SL3 0AE
UK
```

Phone:

+1 800 554 3448, +1 919 872 3020, +44 1753 686200

Fax:

+1 800 75 EXIDE, +44 1753 686827

EMail:

info@exide.com, info@deltecpower.com, sales_support@exide.co.uk

URL:

http://www.exide.com, http://www.deltecpower.com, http://www.fpsUPS.com

Last modified:

October 12, 1997.

20.9 LjetMgr

Description:

Printers from Hewlett-Packard (both laser and ink variants) have an extensive set of configurable options. These options are normally set on the front panel of the printer.

The most notable options are:

- Ccono-mode to save toner
- Powersave mode
- Page format selection
- Font/typeface and language selection

With LjetManager you can directly modify these settings on your screen. It is especially usefull for printers that lack front panel configurability (like the LaserJet 5L) or if you have more than one printer at your disposal.

LjetManager sports a graphical user interface, is fully localizable and comes with documentation and help pages in HTML format.

Distribution media:

3.5" diskettes, Internet (EMail and WWW).

Licensing policy:

You can download a demo version version of LjetMgr from the following location and the corresponding mirror sites: ftp://sunsite.unc.edu/pub/Linux/system/Printing Just have a look for ljetmgr-2.6.README and the accompanying files.

OS provisions:

Needs a printer that supports PJL. Available for Linux on Intel.

Documentation:

Complete context sensitive Help and documentation in HTML format. A PostScript version is available on request.

Product support:

Prices include installation support and one year of free upgrades. Additional support available on request.

Available since:

May, 1996.

Countries with distribution:

World wide.

Price range:

Please contact softWorks for current prices. Discounts for educational institutions and students available.

Vendor:

softWorks, Richard Schwaninger

```
Theodor-Körnerstr. 173
A-8010 Graz
Austria
```

Phone:

+43 316 686590

Fax:

+43 316 686590

EMail:

risc@ping.at

URL:

http://members.ping.at/risc

Contact:

Richard Schwaninger

Last modified:

August 7, 1997.

20.10 Synchronize/CyberScheduler

Description:

Calendaring and scheduling software for LANS, WANs and Intranets.

Distribution media:

Internet (FTP).

Licensing policy:

Per user.

OS provisions:

Linux.

Documentation:

TOC

Take a look at our web site.

Available since:

1989.

Countries with distribution:

World wide.

Price range:

US$100 per user.

Vendor:

CrossWind Technologies Inc.

```
1505 Ocean Street, Suite 1
Santa Cruz, CA 95060
U.S.A.
```

Phone:

+1 408 469 1780

Fax:

+1 408 469 1750

EMail:

info@crosswind.com

URL:

http://www.crosswind.com

Contact:

Anne Becker

Last modified:

November 13, 1997.

---

21. Free Software for Commercial Hardware

This section covers free Linux software (e.g. drivers) for specialized hardware.

21.1 Stallion Technologies Multiport Serial Boards

Description:

Stallion provides Linux support for its EasyIO and EasyConnection range of multiport cards. EasyIO is available with either 4 asynchronous serial ports with RJ45 connectors, or 8 asynchronous serial ports with RJ45 or DB25 connectors. EasyIO supports data rates of up to 145 Kbits/sec or 460 Kbits/sec per port (depending on the model) and full modem signaling. Drivers are available for most Intel-based operating systems, including Linux. EasyIO is available for the ISA bus.

EasyConnection is an expandable multiport solution that provides from 8 to 64 asynchronous serial ports in 8 or 16 port increments. Both RJ45 and DB25 connectors are available. Features include full surge suppression, full modem signaling, and data rates of up to 145 Kbits/sec or 460 Kbits/sec per port (depending on the panel version). Drivers are available for most Intel-based operating systems, including Linux. EasyConnection is available for the ISA, EISA and PCI bus.

Distribution media:

Internet (FTP).

Licensing policy:

The Linux driver is available at no charge and is copyrighted under the GNU General Public License. Source code is provided.

OS provisions:

The most current driver supports all 2.0.x kernel versions. The drivers are updated regularly to keep pace with kernel changes.

Documentation:

Hardware reference documentation provided with the boards. The Linux driver package contains a README file with driver installation and usage information.

Product support:

Stallion Technologies now fully supports the Stallion board Linux drivers. Software updates are available from the Internet. Stallion also provides technical support for the hardware. Contact Stallion support for assistance.

Available since:

October, 1994.

Countries with distribution:

World wide distribution. Available in most countries.

Price range:

Prices for the serial boards vary from country to country. Contact your nearest Stallion Technologies office for local availability and pricing. The driver package contains a file with contact information for all Stallion Technologies offices.

Vendor:

Stallion Technologies Pty Ltd.

```
2880 Research Park Drive, Suite 160
Soquel, CA 95073
U.S.A.
```

Phone:

+1 800 347 7979

Fax:

+1 408 477 0444

EMail:

info@stallion.com

URL:

http://www.stallion.com

FTP:

ftp://ftp.stallion.com

Last modified:

September 30, 1997.

Configuration HOWTO

By Guido Gonzato, guido@ibogeo.df.unibo.it

v1.2.6, 19 January 1999

This HOWTO aims at making the fine--tuning of your newly installed Linux box quicker and easier. Here you will find a set of configurations for the most common applications, so you can start to work with a well-usable system.

1. Introduction

- 1.1 Why This HOWTO
- 1.2 What We Will Be Configuring

2. General System Setup

- 2.1 Keyboard
- 2.2 Kernel Matters
- 2.3 Sendmail Lock
- 2.4 Hard Disk Performance
- 2.5 Parallel Port Zip Drive
- 2.6 Device Drivers
- 2.7 Login Messages
- 2.8 Hostname
- 2.9 Mouse
- 2.10 Mount Points
- 2.11 lilo(8) and LOADLIN.EXE
- 2.12 Mail Capabilities
- 2.13 Printer Configuration
- 2.14 SVGATextMode

3. Software Configuration

- 3.1 bash(1)
- 3.2 ls(1)
- 3.3 less(1)
- 3.4 emacs(1)
- 3.5 joe(1)
- 3.6 jed(1)
- 3.7 efax(1)
- 3.8 TeX and Friends
- 3.9 Avoid PPProblems!
- 3.10 POP Client
- 3.11 X Window System (XFree86)
- 3.12 Users' Configurations
- 3.13 Upgrading

4. Configuration Software

5. The End

- 5.1 Copyright
- 5.2 Feedback
- 5.3 Disclaimer

1. Introduction

1.1 Why This HOWTO

I have installed Linux on many PCs and noted that current distributions are terrific but, annoyingly, lack some basic configuration. Most applications will work out of the box, but others won't. Moreover, I have noted that the same questions crop up on c.o.l.setup over and over again.

To try and remedy this situation, and to have a memorandum for fresh installations, I wrote a do--this--and--that list that I later expanded to this HOWTO. Here you will find a handful of configuration examples for the most common applications, programs, and services, which should save you a fair amount of time and work.

A few of the examples outlined in this HOWTO depend somehow on the distribution. I only have access to Red Hat, Caldera OpenLinux, and S.u.S.E. installations, so don't take any of my tips as gospel if you have Slackware, Debian or other distributions. In any case, reading documentation and the HOWTOs always pays off, so you're advised to do so anyway. My reference distribution is Red Hat; to avoid repetitions, ``AT2YD'' stands for ``Adapt This to Your Distribution''.

A final note: I hope that this HOWTO becomes obsolete as soon as possible. The current trend in Linux distributions is to provide configuration tools rather than leave the sysadm alone with a pile of docs. Therefore, whenever possible I'll point you to the right tool.

The most recent revision of this document, including some translations, is available on `http://sunsite.unc.edu/mdw/HOWTO`.

1.2 What We Will Be Configuring

There can be endless hardware configurations for a PC, but in my experience one is quite common: a PC fitted with a large HD split into three partitions (one for DOS/Windows, one for Linux, one for the swap), sound card, modem, CD--ROM drive, printer, mouse. A parallel port Zip Drive is also very common.

This is the hardware I'll assume you want to configure, but it's easy to adapt the following tips to different configurations. It's implicitly assumed that you'll be root when editing/fixing/hacking.

And now, lads, sleeves up.

2. General System Setup

2.1 Keyboard

First of all, how to configure the keyboard. If you missed this step during installation or have changed your keyboard, you'll have to:

- choose a suitable key table from /usr/lib/kbd/keytables/; for example, `it.map` selects the Italian keyboard;
- edit the file /etc/sysconfig/keyboard so as it reads: `KEYTABLE="/usr/lib/kbd/keytables/it.map"`;
- to set up the keyboard repeat rate and delay time, add this line to /etc/rc.d/rc.sysinit (Red Hat) or /etc/rc.d/rc.boot (Caldera) or /etc/rc.d/boot (S.u.S.E.):

```
/sbin/kbdrate -s -r 16 -d 500  # or whatever you like
```

To load the key table, issue /etc/rc.d/init.d/keytable start (AT2YD). Other special keys will be considered in the following sections.

To enable NumLock on by default, add these lines to /etc/rc.d/rc.sysinit (AT2YD):

```
for tty in /dev/tty[1-9]*; do
  setleds -D +num < $tty
done
```

2.2 Kernel Matters

IMHO, the first thing to do next is build a kernel that best suits your system. It's very simple to do but, in any case, refer to the `README` file in /usr/src/linux/ or the Kernel HOWTO. Hints:

- consider carefully your needs. Choosing a kernel configuration, applying the patches, and compiling it once and for all is more productive than reconfiguring and recompiling each month; this is especially true if your Linux box is a server. Don't forget to include support for all the hardware you might likely add in the future (e.g. SCSI, Zip, network cards, etc);
- if your PC is based on a Cyrix CPU that is not properly recognised as such by the kernel, apply appropriate patches to improve its performance. Information on `http://www.linuxhq.com/patch/20-p0591.html` ;
- notebook users will want to improve their LCD screen legibility. I used to suggest that a kernel patch should be applied, but `SVGATextMode` (see Section SVGATextMode) is probably the best solution. Anyway, the Big Cursor kernel patch is on `http://www.linuxhq.com/patch/20-p0239.html` ;
- again for notebook users, if you plan to use a PCMCIA modem/fax *don't* compile serial support as a module; compile it in the kernel, otherwise your PCMCIA modem won't work;
- if you use PCMCIA cards, remember to compile those modules as well; also, don't forget to edit /etc/pcmcia/network.opts if you use a PCMCIA network card;

- if you get those pesky messages from `modprobe` indicating that some modules can't be located, that means that your /etc/conf.modules needs mending. For instance, if you don't use the ipx and appletalk modules, add these lines:

```
alias net-pf-4 off
alias net-pf-5 off
```

- to save time the next time you reconfigure and recompile the kernel, it's a good idea to save your configuration on a file and keep it in a safe place. Beware: if you upgrade your kernel this config file is bound to be useless, because it doesn't contain the features of the new kernel.

2.3 Sendmail Lock

On some systems, `sendmail` hangs the machine for a couple of minutes at boot time. There are two cases: 1) the machine is not directly connected to the Internet, 2) the machine is connected, and has a permanent IP address.

Quick fix for the first case: make sure your /etc/hosts contains a line that reads

```
127.0.0.1   localhost
```

For the second case: the lock is caused by /etc/hosts containing a line like

```
127.0.0.1   localhost   your_host_name
```

which you will split this way:

```
127.0.0.1    localhost
w.x.y.z      your_host_name
```

See also Section Hostname.

2.4 Hard Disk Performance

Your (E)IDE hard disk's performance can be greatly enhanced by *carefully* using `hdparm`(8). If your Linux distribution doesn't include it, you'll find on `ftp://sunsite.unc.edu/pub/Linux/system/hardware` ; look for a file called hdparm-X.Y.tar.gz.

Since many details depend on your hard disk and HD controller, I can't give you a general recipe. As you risk to toast your filesystem, please *read the man page carefully* before using some of the options. At its simplest, you could add the following line to /etc/rc.d/rc.sysinit:

```
/sbin/hdparm -c1 /dev/hda  # first IDE drive assumed
```

which enables (E)IDE 32-bit I/O support. As for the `-m' option, this is what `hdparm` author Mark Lord emailed me:

> (...) if your system uses components from the past couple of years [< 1997], it will be fine. Older than that, there *may* be a problem (unlikely). The really buggy chips were the CMD0646 and RZ1000 chips, used *extensively* on 486 and (early) 586 motherboards about 2-3 years ago.

TOC

2.5 Parallel Port Zip Drive

To use the parallel port version of the Zip drive you can use the default driver that comes with recent (2.x.x) kernels. During kernel configuration, make sure that SCSI support and SCSI disk support are enabled. Remember, there can be conflicts between the printer and the Zip drive on the same parallel port, so you will want to use kernel modules.

Zip disks are sold preformatted on partition /dev/sda4. To enable the Zip, all you have to do is issue

```
#~ chmod 666 /dev/sda4  # everyone can access the Zip Drive
#~ insmod ppa
```

and the Zip can now be mounted as usual (better write the last line in /etc/rc.d/rc.sysinit). You also access the Zip drive via `mtools` adding this line to your /etc/mtools.conf:

```
drive z: file="/dev/sda4" exclusive
```

There's a better `ppa` driver than the standard one, though: have a look at `http://www.torque.net/~campbell` .

2.6 Device Drivers

Devices in `/dev` (or better, links to the actual device drivers) may be missing. Check what devices your mouse, modem, and CD--ROM drive correspond to, then do what follows:

```
~# cd /dev
/dev# ln -s ttyS0 mouse
/dev# ln -s ttyS1 modem
/dev# ln -s hdb cdrom
```

Tip: in some notebooks the mouse device is /dev/psaux: take this into account when configuring X11.

If you want, do `chmod 666` to such devices as `/dev/cdrom`, `/dev/floppy`, `/dev/modem`, etc. to make them fully accessible by every user. Security problems ahead! Alternatively, create the ``cdrom" group, make `/dev/cdrom` owned by `root.cdrom`, then add users to this group. A similar procedure applies to other devices.

2.7 Login Messages

If you want to customise the login messages, check whether your /etc/rc.d/rc.local overwrites /etc/issue and /etc/motd. (RedHat does.) If so, get on with your editor.

2.8 Hostname

Issuing the command `hostname new_host_name` may not be enough. To avoid the dreaded `sendmail` lock, follow these steps (only valid for a stand--alone machine):

- edit /etc/sysconfig/network and change the hostname therein (e.g. `new_host_name.localdomain`);

- edit /etc/HOSTNAME appropriately;
- append the new hostname in the line in /etc/hosts:

```
127.0.0.1       localhost  new_host_name.locadomain
```

2.9 Mouse

`gpm` mouse services are useful to perform cut and paste in tty mode, and to use the mouse in some applications. For Red Hat, check that you have a file called /etc/sysconfig/mouse and that it reads:

```
MOUSETYPE="Microsoft"
XEMU3=yes
```

Moreover, you must have a file /etc/rc.d/init.d/gpm, where you'll put additional command line parameters. Mine reads:

```
...
       daemon gpm -t $MOUSETYPE -d 2 -a 5 -B 132 # two-button mouse
...
```

Obviously, make sure this configuration is right for your mouse type. In most notebooks, `MOUSETYPE` is ``PS/2".

For Caldera, all you have to do is append this line to /etc/rc.d/rc.boot:

```
/usr/bin/gpm
```

On S.u.S.E., `gpm` options go in /etc/rc.config; on Debian, you'll edit /etc/gpm.conf.

If you like to use menus in console with Ctrl-button, then configure `gpm-root`. Edit the default menu in /etc/gpm-root.conf , then launch `gpm-root` from /etc/rc.d/rc.local (AT2YD).

2.10 Mount Points

It's handy to have mount points for the floppy, other devices and NFS-exported directories. For example, you can do the following:

```
~# cd /mnt
/mnt# mkdir floppy; mkdir cdrom; mkdir win; mkdir zip ; mkdir server
```

This creates mount points for a DOS/Win floppy, an ext2 floppy, the CD--ROM, the Windows partition, the parallel port Zip drive, and a NFS directory.

Now edit the file /etc/fstab and add the following entries:

```
/dev/fd0        /mnt/floppy     vfat            user,noauto 0 1
/dev/cdrom      /mnt/cdrom      iso9660         ro,user,noauto 0 1
/dev/sda4       /mnt/zip        vfat            user,noauto,exec 0 1
/dev/hda1       /mnt/win        vfat            user,noauto 0 1
```

```
server:/export  /mnt/server     nfs             defaults
```

Obviously, you must use the correct device in the first field. Recent kernels support fat32 partitions natively; for older ones, there's a kernel patch on http://bmrc.berkeley.edu/people/chaffee/fat32.html . `man mount` for further information.

You're strongly advised to install `mtools` for accessing DOS/Windows floppies!

2.11 lilo(8) and LOADLIN.EXE

Many users run both Linux and DOS/Windows on their PC, and want to choose at boot time which os to use; this should be done at install time, but in case, do what follows. Let's suppose that /dev/hda1 contains DOS/Windows and that /dev/hda2 contains Linux.

```
~# fdisk
Using /dev/hda as default device!

Command (m for help):a
Partition number (1-4): 2

Command (m for help):w
~#
```

This makes the Linux partition bootable. Then write this basic /etc/lilo.conf file:

```
boot = /dev/hda2
compact
delay = 50
# message = /boot/bootmesg.txt  # write your own, if you will
root = current
image = /boot/vmlinuz  # boot linux by default as this entry comes first
  label = linux
  read-only
other = /dev/hda1
  table = /dev/hda
  label = dos
```

Now issue /sbin/lilo and you're done. Being `lilo` a crucial part of your installation, you're strongly advised to read its documentation anyway.

To boot Linux from DOS/Windows without resetting, put `LOADLIN.EXE` in a directory (in the DOS partition!) included in the DOS path; then copy your kernel to, say, `C:\TEMP\VMLINUZ`. The following simple `.BAT` file will boot Linux:

```
rem   linux.bat
smartdrv /C
loadlin c:\temp\vmlinuz root=/dev/hda2 ro
```

If you use Windows 9x, set the properties of this `.BAT` so as it starts in MS--DOS mode.

TOC

Security Tip

Making a backup copy of your MBR before installing Linux is a safe move. Use `restorrb` (included in the `FIPS` package) before installation, or a Linux rescue floppy to issue this command:

```
rescue:~# dd if=/dev/hda of=MBR bs=512 count=1
```

then make at least two copies of the file `MBR` on floppies. Should disaster strike, you'll be able to restore your old MBR by issuing:

```
rescue:~# dd if=/mnt/MBR of=/dev/hda bs=446 count=1
```

assuming that a floppy containing `MBR` is mounted under `/mnt`. Alternatively, use a DOS rescue floppy to issue `FDISK /MBR`.

2.12 Mail Capabilities

You will want to be able to read mail messages written in HTML or containing exotic file formats. Make sure you have two files: /etc/mime.types and /etc/mailcap. The first one lists file types and related extensions, like:

```
application/postscript          ps eps
image/jpeg                      jpe jpeg jpg
text/html                       html
```

while the second one tells the mail client how to display that file type.

You may receive mail from people who use Microsoft Outlook, whose messages are in multi--part MIME format. These two lines, put in etc/mailcap, should let you read those messages:

```
text/plain; less %s; needsterminal
text/html; lynx -force_html %s; needsterminal
```

2.13 Printer Configuration

All distributions I know have a configuration tool for setting up the printer (`printtool`, `yast`, or `magicfilter`); if you don't have it, this is a basic manual configuration.

Let's suppose you have a non--PostScript (non ``Windows-only'' too!) printer you want to use to print raw text (e.g., C source files) and PostScript files via Ghostscript, which is assumed to be already installed.

Setting up the printer involves a few steps:

- find out which one the parallel print device is: try

  ```
  ~# echo "hello, world" > /dev/lp0
  ~# echo "hello, world" > /dev/lp1
  ```

  and take note which one works.
- make two spool directories:

```
~# cd /var/spool/lpd
/var/spool/lpd/# mkdir raw ; mkdir postscript
```

- if your printer exibits the ``staircase effect'' (most inkjets do), you'll need a filter. Try to print two lines with

  ```
  ~# echo "first line" > /dev/lp1 ; echo "second line" > /dev/lp1
  ```

  if the output is like this:

  ```
  first line
            second line
  ```

  then save this script as /var/spool/lpd/raw/filter:

  ```
  #!/bin/sh
  # This filter does away with the "staircase effect"
  awk '{print $0, "\r"}'
  ```

  and make it executable with `chmod 755 /var/spool/lpd/raw/filter`.
- make a filter for PostScript emulation. Write the following filter as /var/spool/lpd/postscript/filter:

  ```
  #!/bin/sh

  DEVICE=djet500
  RESOLUTION=300x300
  PAPERSIZE=a4
  SENDEOF=

  nenscript -TUS -ZB -p- |
  if [ "$DEVICE" = "PostScript" ]; then
          cat -
  else
          gs -q -sDEVICE=$DEVICE \
                  -r$RESOLUTION \
                  -sPAPERSIZE=$PAPERSIZE \
                  -dNOPAUSE \
                  -dSAFER \
                  -sOutputFile=- -
  fi

  if [ "$SENDEOF" != "" ]; then
          printf "\004"
  fi
  ```

  (in this example an HP DeskJet printer is assumed. Fix it to suit your printer).
- finally, add the following entries in /etc/printcap:

  ```
  # /etc/printcap
  lp|ps|PS|PostScript|djps:\
          :sd=/var/spool/lpd/postscript:\
          :mx#0:\
  ```

```
                :lp=/dev/lp1:\
                :if=/var/spool/lpd/postscript/filter:\
                :sh:
        raw:\
                :sd=/var/spool/lpd/raw:\
                :mx#0:\
                :lp=/dev/lp1:\
                :if=/var/spool/lpd/raw/filter:\
                :sh:
```

For more complex or exotic printing configurations, the Printing-HOWTO awaits you.

If you use `printtool`, be aware that the GSDEVICE chosen by Printtool will work, but not necessarily at its best for your printer. You may consider fiddling a bit with the file `postscript.cfg`; for instance, I changed GSDEVICE from `cdj500` to `djet500` and now my prints come out much quicker.

2.14 SVGATextMode

This utility, available on `ftp://tsx-11.mit.edu/pub/linux/sources/sbin` , can be used to change the console screen resolution, font, and cursor shape. Users whose language include accented characters will be able to use them in console applications, while notebook users may change the cursor shape to make it more visible.

Edit /etc/TextConfig or /etc/TextMode, starting with the default VGA definition. Europeans should be happy with this ``LoadFont'' section:

```
Option "LoadFont"
FontProg "/usr/bin/setfont"
FontPath "/usr/lib/kbd/consolefonts"
FontSelect "lat1u-16.psf"   8x16 9x16 8x15 9x15
FontSelect "lat1u-14.psf"   8x14 9x14 8x13 9x13
FontSelect "lat1u-12.psf"   8x12 9x12 8x11 9x11
FontSelect "lat1u-08.psf"   8x8  9x8  8x7  9x7
```

Once you're done, try your configuration with a command like

```
~# SVGATextMode "80x34x9"
```

and if everything appears to be working fine, remove the warnings from /etc/TextMode and include this line in etc/rc.d/rc.sysinit (AT2YD):

```
# SVGATextMode
/usr/sbin/SVGATextMode "80x34x9"
```

Please note that the block cursor only works with some modes; on one of my machines, "80x32x9".

3. Software Configuration

These are the the configuration files we are going to custimise: /etc/profile /etc/bashrc `.bashrc` `.bashrc` `.bash_profile` `.bash_logout` `.inputrc` `.less` `.lessrc` `.xinitrc` `.fvwmrc` `.fvwm2rc95` `.Xmodmap` `.Xmodmap.num` `.Xdefaults` `.jedrc` `.abbrevs.sl` `.joerc` `.emacs` . Don't add users until you have completed your system configuration; you'll put the dot files in /etc/skel.

3.1 bash(1)

Arguably, the most important piece of software after the kernel. To tailor `bash`'s behaviour, these are the main files to edit:

- /etc/bashrc contains system wide aliases and functions;
- /etc/profile contains system wide environment stuff and startup programs;
- `$HOME/.bashrc` contains user aliases and functions;
- `$HOME/.bash_profile` contains user environment stuff and startup programs;
- `$HOME/.inputrc` contains key bindings and other bits.

Examples of these files are shown below. First, the most important: /etc/profile. It's used to configure a lot of features in your Linux box, as you will see in the following sections. Please look out for reverse quotes!

```
# /etc/profile

# System wide environment and startup programs
# Functions and aliases go in /etc/bashrc

# This file sets up the following features and programs:
#
#   o path
#   o prompts
#   o a few environment variables
#   o colour ls
#   o less
#   o rxvt
#
# Users can override these settings and/or add others in their
# $HOME/.bash_profile

# set a decent path (including Kde)
PATH="$PATH:/usr/X11R6/bin:/opt/kde/bin:$HOME/bin:."

# notify the user: login or non-login shell. If login, the prompt is
# coloured in blue; otherwise in magenta. Root's prompt is red.
# See the Colour-ls mini HOWTO for an explanation of the escape codes.
USER=`whoami`
if [ $LOGNAME = $USER ] ; then
  COLOUR=44  # blue
```

TOC

```
else
  COLOUR=45  # magenta
fi

if [ $USER = 'root' ] ; then
  COLOUR=41  # red
  PATH="$PATH:/usr/local/bin"
fi

ESC="\033"
STYLE=';1m' # bold; choose which one to use
# STYLE='m' # plain
PS1="\[$ESC[$COLOUR;37$STYLE\]$USER:\[$ESC[37;40$STYLE\]\w\\$ "
PS2="> "

# no core dumps, please
ulimit -c 0

# set umask
if [ `id -gn` = `id -un` -a `id -u` -gt 14 ]; then
  umask 002
else
  umask 022
fi

# a few variables
USER=`id -un`
LOGNAME=$USER
MAIL="/var/spool/mail/$USER"
NNTPSERVER=news.iol.it # put your own here
VISUAL=jed
EDITOR=jed
HOSTNAME=`/bin/hostname`
HISTSIZE=1000
HISTFILESIZE=1000
export PATH PS1 PS2 USER LOGNAME MAIL NNTPSERVER
export VISUAL EDITOR HOSTNAME HISTSIZE HISTFILESIZE

# enable colour ls
eval `dircolors /etc/DIR_COLORS -b`
export LS_OPTIONS='-s -F -T 0 --color=yes'

# customize less
LESS='-M-Q'
LESSEDIT="%E ?lt+%lt. %f"
LESSOPEN="| lesspipe.sh %s"
LESSCHARDEF=8bcccbcc13b.4b95.33b. # show colours in ls -l | less
export LESS LESSEDIT LESSOPEN VISUAL LESSCHARDEF

# fix the backspace key in rxvt
if [ "$COLORTERM" != "" ] ; then
  stty erase ^H  # alternative: ^H
fi
```

TOC

```
for i in /etc/profile.d/*.sh ; do
  if [ -x $i ]; then
    . $i
  fi
done

# call fortune, if available
if [ -x /usr/games/fortune ] ; then
  echo ; /usr/games/fortune ; echo
fi
```

This is a sample /etc/bashrc:

```
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile

# For some unknown reason bash refuses to inherit
# PS1 in some circumstances that I can't figure out.
# Putting PS1 here ensures that it gets loaded every time.

USER=`whoami`
if [ $LOGNAME = $USER ] ; then
  COLOUR=44  # blue
else
  COLOUR=45  # magenta
fi

if [ $USER = 'root' ] ; then
  COLOUR=41  # red
fi

ESC="\033"
STYLE=';1m'  # bold
# STYLE='m'    # plain
export PS1="\[$ESC[$COLOUR;37$STYLE\]$USER:\[$ESC[37;40$STYLE\]\w\\$ "
export PS2="> "
export CDPATH="$CDPATH:~"

alias which="type -path"
alias ls="ls $LS_OPTIONS"
```

This is a sample `.bashrc`:

```
# $HOME/.bashrc
# Source global definitions
```

```
if [ -f /etc/bashrc ]; then
  . /etc/bashrc
fi

# this is needed to notify the user that they are in non-login shell
if [ "$GET_PS1" = "" ] ; then
  COLOUR=45; ESC="\033"; STYLE=';1m';  # STYLE='m'
  USER=`whoami`
  export PS1="\[$ESC[$COLOUR;37$STYLE\]$USER:\[$ESC[37;40$STYLE\]\w\\$ "
fi

# aliases
alias cp='cp -i'
alias l=less
alias lyx='lyx -width 900 -height 700'
alias mv='mv -i'
alias rm='rm -i'
alias x=startx

# A few useful functions

inst() # Install a .tar.gz archive in the current directory.
{ tar -zxvf $1 }

cz() # List the contents of a .zip archive.
{ unzip -l $* }

ctgz() # List the contents of a .tar.gz archive.
{
  for file in $* ; do
    tar -ztf ${file}
  done
}

tgz() # Create a .tgz archive a la zip.
{
  name=$1 ; tar -cvf $1 ; shift
  tar -rf ${name} $*
  gzip -S .tgz ${name}
}
```

This is a sample `.bash_profile`:

```
# $HOME/.bash_profile

# User specific environment and startup programs
# This file contains user-defined settings that override
# those in /etc/profile

# Get user aliases and functions
```

```
if [ -f ~/.bashrc ]; then
  GET_PS1="NO"  # don't change the prompt colour
  . ~/.bashrc
fi

# set a few `default' directories
export CDPATH="$CDPATH:$HOME:$HOME/text:$HOME/text/geology"
```

This is a sample `.inputrc`:

```
# $HOME/.inputrc

# key bindings
"\e[1~": beginning-of-line
"\e[3~": delete-char
"\e[4~": end-of-line
# (F1 .. F5) are "\e[[A" ... "\e[[E"
"\e[[A": "info \C-m"

set bell-style visible          # please don't beep
set meta-flag On                # allow 8-bit input (i.e, accented letters)
set convert-meta Off            # don't strip 8-bit characters
set output-meta On              # display 8-bit characters correctly
set horizontal-scroll-mode On   # scroll long command lines
set show-all-if-ambiguous On    # after TAB is pressed
```

To make the backspace and delete keys work correctly in `xterm` and other X11 applications, the following is also needed:

- put this in your `.xinitrc`:

  ```
  usermodmap=$HOME/.Xmodmap
  xmodmap $usermodmap
  ```

- then your `.Xmodmap` will contain:

  ```
  keycode 22 = BackSpace
  keycode 107 = Delete
  ```

  this fixes the console. To fix `xterm`:

- put this in your `.Xdefaults`:

  ```
  xterm*VT100.Translations: #override <Key>BackSpace: string(0x7F)\n\
          <Key>Delete:        string(0x1b) string("[3~")\n\
          <Key>Home:          string(0x1b) string("[1~")\n\
          <Key>End:           string(0x1b) string("[4~")\n\
          Ctrl<Key>Prior:     string(0x1b) string("[40~")\n\
          Ctrl<Key>Next:      string(0x1b) string("[41~")

  nxterm*VT100.Translations: #override <Key>BackSpace: string(0x7F)\n\
          <Key>Delete:        string(0x1b) string("[3~")\n\
  ```

```
<Key>Home:          string(0x1b) string("[1~")\n\
<Key>End:           string(0x1b) string("[4~")\n\
Ctrl<Key>Prior:     string(0x1b) string("[40~")\n\
Ctrl<Key>Next:      string(0x1b) string("[41~")
```

`rxvt` is a wee bit more complicated, as some compile--time options influence its behaviour. See the above /etc/profile.

More info in `bash`(1) and `readline`(3) man pages.

Don't expect every application to work correctly! If you run `joe` in `xterm`, for instance, some keys won't work; the same holds for versions of `rxvt` older than 2.4.5.

3.2 ls(1)

`ls` can display directory listings using colours to highlight different file types. To enable this feature, you just need a couple of lines in /etc/profile as seen above. However, this won't work with `rxvt` older than v. 2.21; use some flavour of `xterm` instead. It looks like `rxvt` has a bug that prevents it from inheriting the environment correctly in some circumstances. `rxvt` 2.4.5 upwards is OK.

Caldera's `ls` doesn't have colours, but there's an equivalent `color-ls`. Add this in /etc/bashrc:

```
alias ls="color-ls $LS_OPTIONS"
```

3.3 less(1)

With this excellent pager you can browse not only plain text files, but also gzip compressed, tar and zip archives, man pages, and what have you. Its configuration involves a few steps:

- to use it with the movement keys, have this plain ASCII file `.lesskey` in your home directory:

  ```
  ^[[A   back-line
  ^[[B   forw-line
  ^[[C   right-scroll
  ^[[D   left-scroll
  ^[OA   back-line
  ^[OB   forw-line
  ^[OC   right-scroll
  ^[OD   left-scroll
  ^[[6~  forw-scroll
  ^[[5~  back-scroll
  ^[[1~  goto-line
  ^[[4~  goto-end
  ^[[7~  goto-line
  ^[[8~  goto-end
  ```

  then run the command `lesskey`. (These are escape sequences for vt100-like terminals.) This creates a binary file `.less` containing the key bindings.
- write the following file as /usr/local/bin/lesspipe.sh:

  ```
  #!/bin/sh
  # This is a preprocessor for 'less'.  It is used when this environment
  # variable is set:   LESSOPEN="|lesspipe.sh %s"
  ```

```
lesspipe() {
  case "$1" in
  *.tar) tar tf $1 2>/dev/null ;; # View contents of .tar and .tgz files
  *.tgz|*.tar.gz|*.tar.Z|*.tar.z) tar ztf $1 2>/dev/null ;;
  *.Z|*.z|*.gz) gzip -dc $1  2>/dev/null ;; # View compressed files correctly
  *.zip) unzip -l $1 2>/dev/null ;; # View archives
  *.arj) unarj -l $1 2>/dev/null ;;
  *.rpm) rpm -qpil $1 2>/dev/null ;;
  *.cpio) cpio --list -F $1 2>/dev/null ;;
  *.1|*.2|*.3|*.4|*.5|*.6|*.7|*.8|*.9|*.n|*.man) FILE=`file -L $1`
    FILE=`echo $FILE | cut -d ' ' -f 2`
    if [ "$FILE" = "troff" ]; then
      groff -s -p -t -e -Tascii -mandoc $1
    fi ;;
  *) file $1 | grep text > /dev/null ;
    if [ $? = 1 ] ; then # it's not some kind of text
      strings $1
    fi ;;
  esac
}

lesspipe $1
```

then make it executable with `chmod 755 lesspipe.sh`.

- put the necessary variables in /etc/profile as seen above.

3.4 emacs(1)

I rarely use `emacs`, so I have only a couple of tips for you. Some `emacs` distributions don't come preconfigured for colours and syntax highlighting. Put this in your `.emacs`:

```
(global-font-lock-mode t)
(setq font-lock-maximum-decoration t)
```

This only works in X11. Moreover, to enable accented characters you'll add this line:

```
(standard-display-european 1)
```

I'll leave it to you to peruse all of `emacs`' documentation to find out how to tailor it to your needs---potentially, it can take months of hacking. The Dotfile generator (Section Configuration Software) is a good helping hand.

3.5 joe(1)

Some versions of `joe` don't work with colours in console, and some special keys don't work either. To my knowledge, no one has found a solution to these small nags. A quick and dirty (and inelegant) solution to the former problem is this:

```
~$ export TERM=vt100
~$ joe myfile
   (edit your file)
~$ export TERM=linux
```

Configure `joe` editing `.joerc`, `.jstarrc` or your favourite emulation; you can start from the system-wide conf files in /usr/lib/joe.

TOC

3.6 jed(1)

This is my favourite editor: it does what I need, it's lighter and easier to configure than `emacs`, and emulates other editors quite well. Many users at my university use `jed` to emulate `EDT`, VMS' system editor.

`jed`'s configuration files are `.jedrc` and /usr/lib/jed/lib/*; the former can be adapted from `jed.rc` in the latter directory.

- to make `jed` use the special keys correctly, write the file /usr/lib/jed/lib/defaults.sl whose only line reads:

```
() = evalfile("linux");
```

- if `xjed` apparently doesn't recognise the DEL key, add these lines to your `.jedrc`:

```
#ifdef XWINDOWS
  x_set_keysym (0xFFFF, 0, "\e[3~");
  setkey (``delete_char_cmd'', "\e[3~");
#endif
```

- edit /usr/lib/jed/lib/linux.sl to specify `Info_Directory = "/usr/info";` and `/bin/mail` after `/UCB_Mailer = "/bin/mail";`;
- to make `jed` emulate `EDT` (or other editors) all you have to do is edit a couple of lines in `.jedrc`. If you want the numeric keypad `+' to delete words instead of a single character, add this in `.jedrc`:

```
unsetkey("\eOl");
unsetkey("\eOP\eOl");
setkey("edt_wdel", "\eOl");
setkey("edt_uwdel", "\eOP\eOl");
```

  after the line that reads `() = evalfile("edt")` (or similar);
- to make `xjed` use the numeric keypad for `EDT` emulation, insert the following in `.Xmodmap`:

```
keycode 77  = KP_F1
keycode 112 = KP_F2
keycode 63  = KP_F3
keycode 82  = KP_F4
keycode 86  = KP_Separator
```

- colour customization for `xjed` is done adding lines like these in `.Xdefaults`:

```
xjed*Geometry: 80x32+150+50
xjed*font: 10x20
xjed*background: midnight blue
# and so on...
```

- the ``abbreviation'' feature is an invaluable timesaver. Write a file like the following as `$HOME/.abbrevs.sl`:

```
create_abbrev_table ("Global", "0-9A-Za-z");
define_abbrev ("Global", "GG", "Guido Gonzato");
create_abbrev_table ("TeX", "\\A-Za-z0-9");
define_abbrev ("TeX", "\\beq", "\\begin{equation}");
define_abbrev ("TeX", "\\eeq", "\\end{equation}");
```

```
% and so on...
```

and type `ESC x abbrev_mode` to enable it. To have the abbreviation enabled by default, add entries like t to your `.jedrc`:

```
define text_mode_hook ()
{
  set_abbrev_mode (1);
}
%
define fortran_hook ()
{
  set_abbrev_mode (1);
  use_abbrev_table ("Fortran");
}
% and so on...
```

3.7 efax(1)

This package is probably the most convenient for simple sending/receiving of faxes. You'll have to tailor the script /usr/bin/fax; easy job, but a couple of quirks caused me quite an headache:

- DIALPREFIX: chances are that simply putting `T' or `P' won't work in some countries (in Italy, at least). Put `ATDT' or `ATDP' instead;
- INIT and RESET: these strings contain the initialisers `-i' and `-k', needed by `efax`. If you want to add an AT command, add it to the appropriate string leaving out `AT' and preceding the rest with either `-i' or `-k'. Example: to add the `ATX3' command to INIT, you'll append `-iX3'.

3.8 TeX and Friends

I'll assume you have the teTeX distribution. Just a couple of things here:

- to configure the hyphenation pattern for your language, edit the file /usr/lib/texmf/texmf/tex/generic/config/language.dat, then do:

  ```
  ~# texconfig init ; texconfig hyphen
  ```

- if you add a LaTeX package, after adding the files under /usr/lib/texmf/texmf/tex/latex/ run the command `texhash` so that teTeX recognises the new package;
- to tailor `dvips`, the file to edit is /usr/lib/texmf/texmf/dvips/config/config.ps. Be aware that the fields regarding the default resolution also affect `xdvi`'s behaviour; if you experience annoying attempts to create fonts each time you run it, put the line

  ```
  XDvi*mfmode:
  ```

  in `.Xdefault`. This should help.

3.9 Avoid PPProblems!

I'll take it for granted that your kernel has PPP + TCP/IP support compiled in, that loopback is enabled, and that you already have the `pppd` package correctly installed and, if you will, suid root. Obviously, your ISP must support PPP.

There are now two ways to get PPP to work: a) manual configuration, and b) a configuration program that automagically

sees to it. Whichever option you choose, have the following information on hand:

- your ISP's telephone number;
- your ISP's name, mail and news server address;
- your ISP's domain;
- your username and password.

Manual configuration is a drudgery. It's about editing files and writing scripts; not too much work, but it's easy to make mistakes and newcomers are often intimidated. The PPP HOWTO is there for you. Alternatively, there are tools that ask for the information above and do all the work.

You'll be surely better off if get hold of one of the following nice tools:

- an X11--based tool is EzPPP, whose home page is `http://www.serv.net/~cameron/ezppp/index.html` . Very easy to use, almost self--explanatory, but requires you to write the connection script;
- for tty--based connections, theere are tools available on `ftp://sunsite.unc.edu/pub/Linux/system/network/serial/ppp`. One of the finest is pppsetup-X.XX.tar.gz. Script required here;
- the simplest configuration tools to date are surely `wvdial` and `eznet`. You feed them your ISP's phone number, your username, your password, and you're in business. Their home pages are at `http://www.worldvisions.ca/wvdial` and `http://www.hwaci.com/sw/eznet`. My preference goes to the latter.

A Quick Start with eznet

First of all, create an /etc/resolv.conf like this:

```
nameserver w.x.y.z
```

where you'll insert the address of your ISP's nameserver. To create an account with `eznet`, issue the following command:

```
#~ eznet add service=YOUR_ISP user=NAME password=PASSWORD phone=PHONE
```

then try to dial your ISP with `eznet up YOUR_ISP`. If the modem waits for the dial tone and won't connect, then try this command:

```
#~ eznet change YOUR_ISP init0=atx3
```

To hang up, the command is `eznet down`. That's all!

3.10 POP Client

To retrieve your mail from a POP server, you use a POP client like `fetchpop` or `fetchmail`. The latter is more advanced, and is probably the only option if your ISP's PPP server can't deal with the command LAST. They're available on `ftp://sunsite.unc.edu/pub/Linux/system/mail/pop`.

To configure these clients:

- `fetchpop`: the first time you run it, you'll be prompted for some information. Answer the questions and you're set.
- `fetchmail`: adapt this sample `.fetchmailrc`:

```
# $HOME/.fetchmailrc
poll mbox.myisp.com with protocol pop3;
  user john there with password _Loo%ny is john here
```

One user reported that adding ``smtphost localhost'' to the second line improved performance dramatically. You must set the permissions to this file with the command `chmod 600 .fetchmailrc`, otherwise `fetchmail` will rightly refuse to start. This example is very basic and assumes that you have a running `sendmail`; there are endless possibilities of configuration. Check out on http://www.ccil.org/~esr/fetchmail.

3.11 X Window System (XFree86)

Setting Up the X Server

Come on, it's not difficult as it used to be... All major distributions include a tool for setting up X11 (e.g. `XConfigurator`, `sax`, `XF86Setup`, or at least `xf86config`). X configuration is virtually automatic these days, but a few video cards may refuse to work. Each time I experience problems setting up X on a system, I resort to a simple method that has always worked:

- make sure that the plain VGA server is installed;
- go to `ftp://ftp.XFree86.org/pub/XFree86/current/binaries` , cd to the proper Linux subdirectory, and download the archives `X_version_bin.tgz`, `X_version_set.tgz`, and all the servers. Amongst other programs, the first one contains the most up-to-date `SuperProbe`;
- unpack `X_version_bin.tgz` to a temporary directory, cd to it, and run `./SuperProbe`. If your video card is recognised, chances are that you'll be able to set it up. Otherwise, hard luck;
- install the servers and `X_version_set.tgz` from /usr/X11R6/, then run `XF86Setup`.

This has always worked for me, but your mileage may vary. Please note that most times X11 won't configure because the specs you choose for your monitor are too low! Start with conservative settings, i.e. 800x600 and 256 colours, then pump it up. *Warning*: these operations are dangerous and your monitor might be damaged!

Keypad

We have seen above how to make a few special keys work. The sample file `.Xmodmap` works well if you want to use Xjed, but it makes the keypad unusable. You'll then need another config file, which we'll call `.Xmodmap.num`:

```
! Definitions can be found in <X11/keysymdef.h>

keycode 77  = Num_Lock
keycode 112 = KP_Divide
keycode 63  = KP_Multiply
keycode 82  = KP_Subtract
keycode 86  = KP_Add
keycode 79  = KP_7
keycode 80  = KP_8
keycode 81  = KP_9
keycode 83  = KP_4
keycode 84  = KP_5
keycode 85  = KP_6
keycode 87  = KP_1
keycode 88  = KP_2
keycode 89  = KP_3
```

```
keycode 90  = KP_0
keycode 91  = KP_Decimal
```

Make sure that your /etc/X11/XF86Config does not contain these three lines:

```
ServerNumLock
Xleds
XkbDisable
```

and in case, comment them out. To re-enable the keypad, you'll issue the command `xmodmap .Xmodmap.num`.

Graphical Login with xdm

To be greeted by a graphical login, edit the file /etc/inittab, which should include a line like this:

```
x:5:respawn:/usr/bin/X11/xdm -nodaemon
```

where 5 is the runlevel corresponding to X11 (S.u.S.E. uses 4). Modify the line that defines the default runlevel (usually 2 or 3), changing it as above:

```
id:5:initdefault:
```

The number of colours is specified in /etc/X11/xdm/Xserver (AT2YD):

```
:0 local /usr/X11R6/bin/X -bpp 16  # 65k colours
```

If you already have `.xinitrc`, copy it to `.xsession` and make the latter executable with `chmod +x .xsession`. Now issue the command `telinit 5` and you're in business.

Window Manager

Once you've managed to make X work, there are endless possibilities of configuration; it depends on the window manager you use - there are tens to choose from. Mostly, it's all down to editing one or more ASCII files in your home directory; in other cases you don't have to edit a thing, and use an applet or even a menu.

Some examples:

- **the fvwm family**: copy /etc/X11/fvwm/system.fvwmrc (or similar) to your home using the appropriate name, browse it and start experimenting. You may waste a lot of time before you get the precise look and feel you like;
- **WindowMaker**: it has several config files that live under `$HOME/GNUstep`, and a cool configuration applet;
- **KDE**: nothing to edit manually here: everything can be done via the menu.

In short: if you don't mind editing config file, choose something like `icewm`, `fvwm*`, `blackbox` etc; if you do mind, the choice is currently restricted to KDE and WindowMaker.

It's important to have a good `.xinitrc`. An example:

```
#!/bin/sh
# $HOME/.xinitrc

usermodmap=$HOME/.Xmodmap
xmodmap $usermodmap

xset s noblank  # turn off the screen saver
```

```
xset s 300 2    # screen saver start after 5 min
xset m 10 5     # set mouse acceleration

rxvt -cr green -ls -bg black -fg white -fn 7x14 \
  -geometry 80x30+57+0 &

if [ "$1" = "" ] ; then  # default
  WINMGR=wmaker
else
  WINMGR=$1
fi

$WINMGR
```

Although it doesn't appear to be strictly required, make it executable with `chmod +x .xinitrc`.

This `.xinitrc` lets you choose the window manager: try

```
$ startx startkde # or other w.m.
```

(it doesn't work with some S.u.S.E. versions, though).

3.12 Users' Configurations

When you're done editing the dot files, copy them to /etc/skel as seen in Section Software Configuration. Note that `.pinerc` can't be fully tailored; make sure that at least the fields `user-domain`, `smtp-server`, and `nntp-server` are properly set up.

3.13 Upgrading

If you upgrade your machine, do your backup as usual and remember to save a few additional files. Some could be /etc/X11/XF86Config, /usr/bin/fax, all the stuff in /usr/local, the kernel configuration, the whole /etc, and all the mail in /var/spool/mail.

---

4. Configuration Software

There are several programs that make Linux easy to setup and configure. Some are becoming sort of standard: Red Hat, Caldera and other distributions ship with apps like `printtool`, `netcfg`, `usertool`, etc, while S.u.S.E. ships with a comprehensive configuration program called YAST. Other useful programs are:

- **The Dotfile Generator**: fine X app with modules to configure packages like `emacs`, `bash`, `procmail` and more. Its page is on `http://www.imada.ou.dk/~blackie/dotfile` ;
- **Linuxconf**: the ultimate configuration tool. It can do everything, both in console and under X. Go to http://www.solucorp.qc.ca/linuxconf at once.

5. The End

5.1 Copyright

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at tjbynum@sunsite.unc.edu via email.

5.2 Feedback

Perhaps even more than other HOWTOs, this one needs and welcomes your suggestions, criticisms, and contributions. Not only is feedback welcome: it's necessary. If you think something is missing or wrong, please email me. If you have a distribution other than Red Hat or Caldera and your config files are different or placed in other directories, please tell me and I'll include your tips. My aim is making life with Linux as easy as possible.

Linux has a huge number of packages, so it's impossible to include directions for all of them. Please keep your requests/suggestions pertinent to the ``most reasonable'' programs---I'll leave it to your common sense.

5.3 Disclaimer

``Configuration HOWTO'' was written by Guido Gonzato, `guido@ibogeo.df.unibo.it`. Many thanks to all other HOWTO authors and man pages writers/maintainers, whose work I've shamelessly pilfered; and to all people who provided me with feedback.

This document is provided ``as is''. I put great effort into writing it as accurately as I could, but you use the information contained in it at your own risk. In no event shall I be liable for any damages resulting from the use of this work.

I hope you'll find this work useful, though. Whenever I install a new Linux box, I actually do...

Enjoy,

Guido =8-)

Linux Consultants HOWTO

Mr. Poet <poet@linuxports.com>

v9.46, 3 July 1999

This document contains a listing of companies providing commercial Linux related support. If you want to find a Linux consultant or consulting firm in your area, this listing will probably be of help for you. It is maintained by Mr. Poet poet@linuxports.com. This document is also completely indexed and databased at the LinuxPorts website. If you would like information on advertising on the Consultants database pages please contact us at poet@linuxports.com.

Note to the reader:

This is the first edition of the new format. We hope you like it. Please send comments to Mr. Poet poet@linuxports.com.

Mr. Poet

About this document

This is the Linux Consultants HOWTO. It is a listing of companies providing commercial Linux related support. If you contact any companies listed in this document, please mention the Linux Consultants HOWTO.

If you need to know more about the Linux Documentation Project or about Linux HOWTO's, feel free to contact the supervisor Tim Bynum linux-howto@metalab.unc.edu.

Linux Consultants HOWTO can be found on the World Wide Web at http://www.linuxports.com/. New versions of the Linux Consultants HOWTO are always placed at this site first, so please be sure to check if the copy you are reading is still up to date!

This site also contains all information needed to submit to this howto or search this howto via a database.

Companies providing Linux support are invited to fill out the form at LinuxPorts Please note that we have temporarily stopped accepting 8-bit characters (German umlauts, etc).

Copyright Information

This HOWTO is Copyright © 1999 Mr. Poet and LinuxPorts

A verbatime copy may be reproduced or distributed in any electronic medium without permission of the author. Physical medium copies, reproductions or distributions for commercial use must have explicit

permission of the author. Translations are similarly permitted without express permission if it includ notice on who translated it.

Short quotes may be used without prior consent by the author. Derivative work and partial distributions of the Linux Consultants HOWTO must be accompanied with either a verbatime copy of this file or a pointer to the verbatime copy.

Commercial redistribution is allowed and encouraged; however, the author must be notified of any such distributions and all copyright information must be intact.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

We further want that *all* information provided in the HOWTOs is disseminated. If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@metalab.unc.edu.

1. Consultant Listings

- 1.1 Consultant ID: Argentina, BASystem
- 1.2 Consultant ID: Argentina, CompuSoft
- 1.3 Consultant ID: Argentina, CyberWare Information Technology
- 1.4 Consultant ID: Argentina, Generic Soft
- 1.5 Consultant ID: Argentina, Jorge Forte
- 1.6 Consultant ID: Argentina, NIXE S.R.L.
- 1.7 Consultant ID: Argentina, Nixe S.R.L.
- 1.8 Consultant ID: Australia, Computer Clinic (WA) Pty Ltd
- 1.9 Consultant ID: Australia, Cybersource Pty. Ltd.
- 1.10 Consultant ID: Australia, Cyberspace Corporation
- 1.11 Consultant ID: Australia, DataSource Pty. Ltd
- 1.12 Consultant ID: Australia, EMUSYS Unix Consulting
- 1.13 Consultant ID: Australia, Echo Labs
- 1.14 Consultant ID: Australia, Farrow Norris
- 1.15 Consultant ID: Australia, Hinterlands Consultancy I.T.
- 1.16 Consultant ID: Australia, John Pearson
- 1.17 Consultant ID: Australia, LINSUP.COM
- 1.18 Consultant ID: Australia, Linux Garden
- 1.19 Consultant ID: Australia, Studio of Arts And Sciences (SAAS)
- 1.20 Consultant ID: Australia, TereDonn Computer Engineering

- 1.21 Consultant ID: Australia, WWWalker Web Development
- 1.22 Consultant ID: Australia, linsup.com
- 1.23 Consultant ID: Austria, Bitsniz Data Systems
- 1.24 Consultant ID: Austria, Bitsniz Data Systems
- 1.25 Consultant ID: Austria, Horus GmbH
- 1.26 Consultant ID: Austria, LINUX@work
- 1.27 Consultant ID: Austria, LinzNet
- 1.28 Consultant ID: Austria, communication.center.one
- 1.29 Consultant ID: Austria, g.a.m.s. edv-dienstleistungen gmbh
- 1.30 Consultant ID: Austria, iDAS
- 1.31 Consultant ID: Belgium, Alexandre Dulaunoy, Consultance OpenSource
- 1.32 Consultant ID: Belgium, BARTH Group's
- 1.33 Consultant ID: Belgium, Better Access NV
- 1.34 Consultant ID: Belgium, CompuSense bvba
- 1.35 Consultant ID: Belgium, CoolNet
- 1.36 Consultant ID: Belgium, Double Barrel Consultancy and Productions
- 1.37 Consultant ID: Belgium, SPIER bvba
- 1.38 Consultant ID: Belgium, VirgoPlus
- 1.39 Consultant ID: Belgium, information technology partners
- 1.40 Consultant ID: Brazil, 4Web Internet
- 1.41 Consultant ID: Brazil, Abner Graham Jacobsen
- 1.42 Consultant ID: Brazil, Fernando Mauro Martins
- 1.43 Consultant ID: Brazil, Pragana, Filhos E Cia Ltda
- 1.44 Consultant ID: Brazil, TELEMATICA Coml. de Teleinformática Ltda.
- 1.45 Consultant ID: Brazil, iSNet Serviços de Informatica
- 1.46 Consultant ID: Bulgaria, University of Mining and Geology, Dept. Computer S
- 1.47 Consultant ID: Canada, 4 Office Automation
- 1.48 Consultant ID: Canada, 8D Technologies inc.
- 1.49 Consultant ID: Canada, ADR Computing
- 1.50 Consultant ID: Canada, Affinity Systems Inc.
- 1.51 Consultant ID: Canada, Best Computers
- 1.52 Consultant ID: Canada, CAL Consultants Inc.
- 1.53 Consultant ID: Canada, Centurion Services

TOC

- 1.54 Consultant ID: Canada, Computing And Networking Centre
- 1.55 Consultant ID: Canada, DFC International Computers Inc.
- 1.56 Consultant ID: Canada, DFC International Computing Inc.
- 1.57 Consultant ID: Canada, DPD Software Ltd
- 1.58 Consultant ID: Canada, Diminishing Networks
- 1.59 Consultant ID: Canada, Gedris Linux Consulting
- 1.60 Consultant ID: Canada, Global Proximity Corporation
- 1.61 Consultant ID: Canada, Hard Data Ltd.
- 1.62 Consultant ID: Canada, JDP Computer Systems Inc.
- 1.63 Consultant ID: Canada, LeadSource Consultants Inc.
- 1.64 Consultant ID: Canada, Logisoft Technologies inc
- 1.65 Consultant ID: Canada, MOE Online Enterprises
- 1.66 Consultant ID: Canada, Montage IT Services
- 1.67 Consultant ID: Canada, MostlyLinux
- 1.68 Consultant ID: Canada, Net Direct Inc.
- 1.69 Consultant ID: Canada, Pilkington Software Inc.
- 1.70 Consultant ID: Canada, Pryor and Pryor Inc.
- 1.71 Consultant ID: Canada, Quist Consulting
- 1.72 Consultant ID: Canada, Roaring Penguin Software Inc.
- 1.73 Consultant ID: Canada, TeleDynamics Communications Inc
- 1.74 Consultant ID: Canada, The People's Linux
- 1.75 Consultant ID: Canada, VL
- 1.76 Consultant ID: Canada, Webcon, Inc.
- 1.77 Consultant ID: Colombia, Linux Colombia Ltda
- 1.78 Consultant ID: Colombia, Skina Ltda.
- 1.79 Consultant ID: Czech Republic, Martin Knotek, poskytovani software
- 1.80 Consultant ID: Denmark, Dansk Data Elektronik A S
- 1.81 Consultant ID: Denmark, Plomus
- 1.82 Consultant ID: Egypt, AB2
- 1.83 Consultant ID: Finland, Netsol Network Solutions Oy
- 1.84 Consultant ID: Finland, SOT Finnish Sofware Engineering Ltd.
- 1.85 Consultant ID: Finland, Tmi Netics Konsultointi
- 1.86 Consultant ID: France, ALCOVE

- 1.87 Consultant ID: France, CRAO
- 1.88 Consultant ID: France, EBC Consulting
- 1.89 Consultant ID: France, NTCD
- 1.90 Consultant ID: France, Paralline
- 1.91 Consultant ID: France, Paul Boyer Consultants Sarl
- 1.92 Consultant ID: France, Thorvax
- 1.93 Consultant ID: Germany, Andreas Spengler EDV-Service
- 1.94 Consultant ID: Germany, Bigge and Langanke GbR
- 1.95 Consultant ID: Germany, Call-a-Server
- 1.96 Consultant ID: Germany, Gecko GmbH
- 1.97 Consultant ID: Germany, Gerd Aschemann
- 1.98 Consultant ID: Germany, ID-Pro GmbH
- 1.99 Consultant ID: Germany, Jochen Laser IT-Services
- 1.100 Consultant ID: Germany, LINUXHAUS
- 1.101 Consultant ID: Germany, Linux-Systemhaus Schulz
- 1.102 Consultant ID: Germany, Manfred Kunde EDV-Beratung
- 1.103 Consultant ID: Germany, Matthias Stolte Datentechnik
- 1.104 Consultant ID: Germany, PiN - Präsenz im Netz Ges. f. Informationstechnolo
- 1.105 Consultant ID: Germany, Simulina GmbH
- 1.106 Consultant ID: Germany, Syslab.com oHg
- 1.107 Consultant ID: Germany, WebArtists Internet Services
- 1.108 Consultant ID: Germany, Wesemann Software and Consulting
- 1.109 Consultant ID: Germany, Wolfgang Kirch
- 1.110 Consultant ID: Germany, WorNet Internetdienste
- 1.111 Consultant ID: Germany, ariadne_net
- 1.112 Consultant ID: Germany, b.i.t. beratungsgesellschaft für informations-tech
- 1.113 Consultant ID: Germany, conIT GmbH
- 1.114 Consultant ID: Germany, dev consulting GmbH
- 1.115 Consultant ID: Germany, frontsite AG
- 1.116 Consultant ID: Germany, intraDAT GmbH
- 1.117 Consultant ID: Germany, ipm intranet project management gmbh
- 1.118 Consultant ID: Germany, loth systemtechnik
- 1.119 Consultant ID: Germany, netServe oHG Münster

- 1.120 Consultant ID: Germany, ralf geschke internet consulting
- 1.121 Consultant ID: Germany, unitegs - Unix Systems Technologies
- 1.122 Consultant ID: Hong Kong, Cyber Channel Int'l Ltd
- 1.123 Consultant ID: Hong Kong, Hong Kong Terminal
- 1.124 Consultant ID: Hong Kong, Linux Center (Hong Kong) Ltd.
- 1.125 Consultant ID: Hong Kong, Uniforce System Ltd.
- 1.126 Consultant ID: Hong Kong, Vortex Ltd.
- 1.127 Consultant ID: Hong Kong, Vortex Ltd.
- 1.128 Consultant ID: Hong Kong, Yoric Network Ltd.
- 1.129 Consultant ID: India, C and B Consulting
- 1.130 Consultant ID: India, Connect Solutions
- 1.131 Consultant ID: India, DSF Internet Services
- 1.132 Consultant ID: India, Netangle Com Pvt ltd
- 1.133 Consultant ID: India, Shah Micro System
- 1.134 Consultant ID: Indonesia, Netindo Pusaka Prima pt.
- 1.135 Consultant ID: Indonesia, PT Cakram DataLingga Duaribu
- 1.136 Consultant ID: Israel, Breakthrough LTD
- 1.137 Consultant ID: Israel, vPrise
- 1.138 Consultant ID: Italy, Alessandro Rubini
- 1.139 Consultant ID: Italy, Francesco Patamia
- 1.140 Consultant ID: Italy, Luca Perugini
- 1.141 Consultant ID: Italy, Marco Michelino
- 1.142 Consultant ID: Italy, Mauro Barella
- 1.143 Consultant ID: Italy, PROSA Progettazione Sviluppo Aperto
- 1.144 Consultant ID: Italy, Panservice
- 1.145 Consultant ID: Italy, Studio LEADER Pro
- 1.146 Consultant ID: Japan, M and T
- 1.147 Consultant ID: Kuwait, Internet Universe - Network Solutions
- 1.148 Consultant ID: Malaysia, Linux Resources Sdn. Bhd.
- 1.149 Consultant ID: Malaysia, MAGNIFIX
- 1.150 Consultant ID: Mexico, Gustavo Cavazos
- 1.151 Consultant ID: Mexico, Linux Michoacan OpenSource
- 1.152 Consultant ID: Mexico, Servicios en Alta Tecnologia

- 1.153 Consultant ID: Mexico, Virtual.Net
- 1.154 Consultant ID: Morocco, ANNOUR Technologies
- 1.155 Consultant ID: Netherlands, Abe I.T. B.V.
- 1.156 Consultant ID: Netherlands, BitWizard B.V.
- 1.157 Consultant ID: Netherlands, CMG
- 1.158 Consultant ID: Netherlands, CistroN
- 1.159 Consultant ID: Netherlands, ComCare IT BV
- 1.160 Consultant ID: Netherlands, Le Reseau netwerksystemen BV
- 1.161 Consultant ID: Netherlands, Linvision
- 1.162 Consultant ID: Netherlands, Manna Automatisering
- 1.163 Consultant ID: Netherlands, Maxfort Consultancy
- 1.164 Consultant ID: Netherlands, Murphy Software BV
- 1.165 Consultant ID: Netherlands, Netland Internet Services
- 1.166 Consultant ID: Netherlands, Root Systems
- 1.167 Consultant ID: Netherlands, Stone IT Services BV
- 1.168 Consultant ID: New Zealand, Catalyst IT Ltd
- 1.169 Consultant ID: New Zealand, iOpen Technologies Ltd.
- 1.170 Consultant ID: Nigeria, CyberAge Systems
- 1.171 Consultant ID: Norway, AQS AS
- 1.172 Consultant ID: Norway, Initio IT-losninger
- 1.173 Consultant ID: Poland, AM ELEKTRONIK s. c.
- 1.174 Consultant ID: Poland, Internet Designers s.c.
- 1.175 Consultant ID: Poland, RG Studio
- 1.176 Consultant ID: Poland, Softomat Ltd.
- 1.177 Consultant ID: Portugal, COM2000 Sistemas Informaticos
- 1.178 Consultant ID: Portugal, S and F CONSULTORES LDA
- 1.179 Consultant ID: Puerto Rico, EZ Networks
- 1.180 Consultant ID: Singapore, Aeon Xe Technology Pte Ltd
- 1.181 Consultant ID: Singapore, LinuxAsia Pte Ltd
- 1.182 Consultant ID: Singapore, Webworks Pte Ltd
- 1.183 Consultant ID: South Africa, Neogenix
- 1.184 Consultant ID: South Africa, Obsidian Systems
- 1.185 Consultant ID: Spain, ACCESSUS - Proyectos y Consultores en Telecomunica

TOC

- 1.186 Consultant ID: Spain, LANGRE INGENIEROS S.L.
- 1.187 Consultant ID: Spain, Linux Sistemas
- 1.188 Consultant ID: Sweden, Ewing Data AB
- 1.189 Consultant ID: Sweden, Omicron Software Systems AB
- 1.190 Consultant ID: Sweden, Tarkus
- 1.191 Consultant ID: Sweden, Xantus AB
- 1.192 Consultant ID: Switzerland, 3T
- 1.193 Consultant ID: Switzerland, Born Consulting
- 1.194 Consultant ID: Switzerland, CONNEXT Communications
- 1.195 Consultant ID: Switzerland, ETV Software Engineering S.A.
- 1.196 Consultant ID: Switzerland, Erik Rossen
- 1.197 Consultant ID: Switzerland, Penta Group AG
- 1.198 Consultant ID: Switzerland, SciOS Scientific Operating Systems GmbH
- 1.199 Consultant ID: Switzerland, Technology Solutions
- 1.200 Consultant ID: Taiwan, WHINSTONE INFO. CO., LTD.
- 1.201 Consultant ID: Turkey, Gelecek A.S
- 1.202 Consultant ID: Turkey, YORE Elektronik Yayimcilik A.S.
- 1.203 Consultant ID: US, AL, The Net Effect, LLC
- 1.204 Consultant ID: US, AZ, Brindlesoft
- 1.205 Consultant ID: US, AZ, Jay Ts
- 1.206 Consultant ID: US, AZ, MagusNet, Inc.
- 1.207 Consultant ID: US, AZ, Tapestry Software
- 1.208 Consultant ID: US, CA, ASA Computers, Inc.
- 1.209 Consultant ID: US, CA, Aedra Associates, Inc.
- 1.210 Consultant ID: US, CA, David Fetter Consulting
- 1.211 Consultant ID: US, CA, Gregg Weber
- 1.212 Consultant ID: US, CA, Herlein Engineering
- 1.213 Consultant ID: US, CA, Hiverworld, LLC
- 1.214 Consultant ID: US, CA, KrystalBox Technologies, Inc.
- 1.215 Consultant ID: US, CA, Linuxcare
- 1.216 Consultant ID: US, CA, MK Information System Services
- 1.217 Consultant ID: US, CA, Mountain Ranch Design
- 1.218 Consultant ID: US, CA, Network Presence, LLC

- 1.219 Consultant ID: US, CA, Newyen Technology
- 1.220 Consultant ID: US, CA, Oration
- 1.221 Consultant ID: US, CA, Ozlin Network Designs
- 1.222 Consultant ID: US, CA, PSO Systems Inc
- 1.223 Consultant ID: US, CA, Predictive Science
- 1.224 Consultant ID: US, CA, Starshine Technical Services
- 1.225 Consultant ID: US, CA, Tesno Systems
- 1.226 Consultant ID: US, CA, U.S. Technical Services
- 1.227 Consultant ID: US, CA, US CustomLogic, LLC
- 1.228 Consultant ID: US, CA, Vennerable Consulting
- 1.229 Consultant ID: US, CA, interactivate consulting group
- 1.230 Consultant ID: US, CO, BDA Consulting LLC
- 1.231 Consultant ID: US, CO, Eklektix, Inc.
- 1.232 Consultant ID: US, CO, MW Software
- 1.233 Consultant ID: US, CO, Paramin Consulting Group
- 1.234 Consultant ID: US, CO, tummy.com, ltd.
- 1.235 Consultant ID: US, CT, DCA Online (Dutch Computer Association)
- 1.236 Consultant ID: US, CT, DownCity, LLC
- 1.237 Consultant ID: US, CT, GNUTEC
- 1.238 Consultant ID: US, CT, MANHATTAN GROUP
- 1.239 Consultant ID: US, CT, Tempest Harding, Inc.
- 1.240 Consultant ID: US, DC, BKP, Inc.
- 1.241 Consultant ID: US, DE, W. E. Hughes, INC
- 1.242 Consultant ID: US, FL, Anthony Awtrey Consulting
- 1.243 Consultant ID: US, FL, Mackintosh Web, Inc.
- 1.244 Consultant ID: US, FL, Richard J Kinch, PhD
- 1.245 Consultant ID: US, FL, Spectrum Enterprises
- 1.246 Consultant ID: US, FL, TSM Services
- 1.247 Consultant ID: US, FL, Yup Dot Com
- 1.248 Consultant ID: US, FL, eBuilders Network
- 1.249 Consultant ID: US, GA, Fly-By-Day Consulting, Inc.
- 1.250 Consultant ID: US, GA, John Meister
- 1.251 Consultant ID: US, HI, Cytronix Computer Services, Inc.

TOC

- 1.252 Consultant ID: US, IA, JEONET
- 1.253 Consultant ID: US, IL, Cobler Corporation
- 1.254 Consultant ID: US, IL, Dataline
- 1.255 Consultant ID: US, IL, Ossmann Consulting
- 1.256 Consultant ID: US, IL, Promethan Consulting
- 1.257 Consultant ID: US, IL, Remington*Microshare
- 1.258 Consultant ID: US, IL, WellThot Inc.
- 1.259 Consultant ID: US, IL, onShore, Inc
- 1.260 Consultant ID: US, IN, CLFINC
- 1.261 Consultant ID: US, IN, IndyBox Systems, Inc.
- 1.262 Consultant ID: US, IN, ValCom
- 1.263 Consultant ID: US, LA, Advanced Network Technologies
- 1.264 Consultant ID: US, LA, Connectivity Solutions, Inc.
- 1.265 Consultant ID: US, MA, CYWAYS, Incorporated
- 1.266 Consultant ID: US, MA, ComputeAmerica.com
- 1.267 Consultant ID: US, MA, DataProfit
- 1.268 Consultant ID: US, MA, LSC
- 1.269 Consultant ID: US, MA, NetInterface Consulting
- 1.270 Consultant ID: US, MA, New England Linux Resource
- 1.271 Consultant ID: US, MA, New England Linux resources
- 1.272 Consultant ID: US, MA, Worldmachine Technologies Corporation
- 1.273 Consultant ID: US, MD, JAMUX
- 1.274 Consultant ID: US, MD, RABA Technologies, Inc.
- 1.275 Consultant ID: US, ME, TransWeb ToolBox
- 1.276 Consultant ID: US, MI, Jim Willette
- 1.277 Consultant ID: US, MI, Nugent Telecommunications
- 1.278 Consultant ID: US, MI, Radiomind Studios
- 1.279 Consultant ID: US, MN, Anvi Corp
- 1.280 Consultant ID: US, MO, Firefly Enterprises, Inc.
- 1.281 Consultant ID: US, MO, Firefly Enterprises, Inc.
- 1.282 Consultant ID: US, MO, PhazeOne, Inc.
- 1.283 Consultant ID: US, NC, ApeX Systems Integration Corp.
- 1.284 Consultant ID: US, NC, Moongroup Consulting, Inc.

TOC

- 1.285 Consultant ID: US, NC, Network Computing Solutions
- 1.286 Consultant ID: US, NC, UTSL dot com
- 1.287 Consultant ID: US, NC, Whole Systems Solutions, Inc.
- 1.288 Consultant ID: US, NC, World Wide Wizdom
- 1.289 Consultant ID: US, NE, Paktronix Systems LLC
- 1.290 Consultant ID: US, NH, AKCP, Inc
- 1.291 Consultant ID: US, NJ, Etc Services
- 1.292 Consultant ID: US, NJ, Voortek Software, Inc
- 1.293 Consultant ID: US, NON United States, ASA Computers, Inc.
- 1.294 Consultant ID: US, NON United States, Coldweb Consulting
- 1.295 Consultant ID: US, NON United States, Emit Sp z o.o.
- 1.296 Consultant ID: US, NON United States, LinuxAsia Pte Ltd
- 1.297 Consultant ID: US, NON United States, Magiclink
- 1.298 Consultant ID: US, NON United States, Psand Limited
- 1.299 Consultant ID: US, NY, Aegis Information Systems, Inc.
- 1.300 Consultant ID: US, NY, Andrew Madloch
- 1.301 Consultant ID: US, NY, Donahue Consulting
- 1.302 Consultant ID: US, NY, JAP Consulting
- 1.303 Consultant ID: US, NY, Lrw.Net
- 1.304 Consultant ID: US, NY, RetiComp, Inc.
- 1.305 Consultant ID: US, NY, Templar Studios, Inc.
- 1.306 Consultant ID: US, NY, Zenez, Inc.
- 1.307 Consultant ID: US, OH, 800Linux
- 1.308 Consultant ID: US, OH, Advanced Unix and Network Consulting
- 1.309 Consultant ID: US, OH, Bradley M. Kuhn
- 1.310 Consultant ID: US, OH, Cleveland Internet Association
- 1.311 Consultant ID: US, OH, Computer Wizards Consulting
- 1.312 Consultant ID: US, OH, Essex Software Development, Inc.
- 1.313 Consultant ID: US, OH, LANshark Systems, Inc
- 1.314 Consultant ID: US, OH, Lightstream Internet, Inc.
- 1.315 Consultant ID: US, OH, TES, Inc.
- 1.316 Consultant ID: US, OH, The Computer Underground
- 1.317 Consultant ID: US, OH, The Computer Underground, Inc.

TOC

- 1.318 Consultant ID: US, OK, After Hours Computer Consulting,LLC.
- 1.319 Consultant ID: US, OR, 3D Technology, Inc.
- 1.320 Consultant ID: US, OR, Avalon Technology Group, Inc.
- 1.321 Consultant ID: US, OR, Fargus.Net
- 1.322 Consultant ID: US, OR, Fence-Post Technology
- 1.323 Consultant ID: US, OR, Hurrah Internet Services
- 1.324 Consultant ID: US, OR, Naked Ape Consulting
- 1.325 Consultant ID: US, PA, Associated Micro Systems, Inc.
- 1.326 Consultant ID: US, PA, JAV-Associates Consulting
- 1.327 Consultant ID: US, PA, Michael Jastremski
- 1.328 Consultant ID: US, PA, Reed Consulting
- 1.329 Consultant ID: US, PA, Rich Westcott
- 1.330 Consultant ID: US, PA, Rich Westcott
- 1.331 Consultant ID: US, RI, New England Computer
- 1.332 Consultant ID: US, SC, Crux
- 1.333 Consultant ID: US, SC, GAME.NET
- 1.334 Consultant ID: US, SC, GAME.NET, Inc.
- 1.335 Consultant ID: US, SC, Linux Empower
- 1.336 Consultant ID: US, TN, Michael Chaney Consulting Corporation
- 1.337 Consultant ID: US, TX, Aegis Data Systems
- 1.338 Consultant ID: US, TX, Computer Innovations International
- 1.339 Consultant ID: US, TX, Internet Power And Light
- 1.340 Consultant ID: US, TX, LEAP _ Linux Enterprise Application Products
- 1.341 Consultant ID: US, TX, PSW Technologies
- 1.342 Consultant ID: US, TX, ProSoft Plus, Inc.
- 1.343 Consultant ID: US, TX, Quinn Team Incorporated
- 1.344 Consultant ID: US, TX, R and B Consulting
- 1.345 Consultant ID: US, TX, Z Consulting
- 1.346 Consultant ID: US, UT, Open Source Consulting Group
- 1.347 Consultant ID: US, VA, 2RAD
- 1.348 Consultant ID: US, VA, Atlantis Consulting
- 1.349 Consultant ID: US, VA, Atlantis
- 1.350 Consultant ID: US, VA, Branch and Company

TOC

- 1.351 Consultant ID: US, VA, Brown Technolgies Network, Inc.
- 1.352 Consultant ID: US, VA, Linux and Web Professionals
- 1.353 Consultant ID: US, VA, NetSource of Virginia, Inc.
- 1.354 Consultant ID: US, VA, Noblestar Systems Corporation
- 1.355 Consultant ID: US, VA, Phaedo Consulting, Inc.
- 1.356 Consultant ID: US, VA, Signet Computers, Inc.
- 1.357 Consultant ID: US, VA, Spectrum Office Systems, Inc.
- 1.358 Consultant ID: US, VT, Everest Systems
- 1.359 Consultant ID: US, WA, Cheek Consulting
- 1.360 Consultant ID: US, WA, Foresight Technologies
- 1.361 Consultant ID: US, WA, K and D Development
- 1.362 Consultant ID: US, WA, PeakServ
- 1.363 Consultant ID: US, WA, Tacoma and Puget Sound Company
- 1.364 Consultant ID: US, WA, Veriteam, Inc.
- 1.365 Consultant ID: US, WA, buzzwerks, Inc.
- 1.366 Consultant ID: US, WI, Midnight Software
- 1.367 Consultant ID: US, WI, Network Integration Services
- 1.368 Consultant ID: US, WI, Perry Systems
- 1.369 Consultant ID: US, WI, Rindy Consulting
- 1.370 Consultant ID: US, WI, WhitePine Consulting
- 1.371 Consultant ID: United Arab Emirates, JM Infotech Solutions STPI
- 1.372 Consultant ID: United Kingdom, CiteCall Ltd.
- 1.373 Consultant ID: United Kingdom, Enterprise Management Consulting
- 1.374 Consultant ID: United Kingdom, FGM Technology LTD EuroDev Cybernetics LTD
- 1.375 Consultant ID: United Kingdom, J.S Unix Consultants Ltd
- 1.376 Consultant ID: United Kingdom, Jason Clifford
- 1.377 Consultant ID: United Kingdom, Michiel Dethmers
- 1.378 Consultant ID: United Kingdom, Philip Hands Computing Ltd
- 1.379 Consultant ID: United Kingdom, RedAmazon
- 1.380 Consultant ID: United Kingdom, Share Net International Ltd
- 1.381 Consultant ID: United Kingdom, iCode Systems Ltd
- 1.382 Consultant ID: Venezuela, Ernesto Hernández-Novich
- 1.383 Consultant ID: Venezuela, Grupo INABOX,C.A.

- 1.384 Consultant ID: Venezuela, Linux Sistemas, C.A.
- 1.385 Consultant ID: Yugoslavia, Berislav Todorovic

1. Consultant Listings

1.1 Consultant ID: Argentina, BASystem

```
Manuel ugarte 2328
Buenos Aires, NON United States 1428

Argentina
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

RedHat Not Applicable

Company Phone Number:

54-11 4781-9710

Company Fax:

54-11 4789-9228

Support Phone Number:

54-11 4781-9710

Contact Email Address:

clau@basystem.com.ar

Company URL:

http://www.basystem.com.ar

Primary Contact:

Norberto Mayo

1.2 Consultant ID: Argentina, CompuSoft

```
Drago 23, Piso 5, Oficina 12
Bahia Blanca, NON United States 8000

Argentina
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems System Administration System Security

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

N/A

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

N/A

Company Phone Number:

+54(291) 455-3570

Company Fax:

+54(291) 453-6735

Support Phone Number:

TOC

+54(291) 15 648-9284

Contact Email Address:

goburastero@arnet.com.ar

Company URL:

http://www.arnet.com.ar

Primary Contact:

Guillermo Burastero

1.3 Consultant ID: Argentina, CyberWare Information Technology

```
Zeballos 2774
Rosario, NON United States 2000

Argentina
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(54)(0341) 156-40-14

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

soporte@cyberware.com.ar

Company URL:

http://www.cyberware.com.ar

Primary Contact:

Javier Kohan

1.4 Consultant ID: Argentina, Generic Soft

```
Defensa 767 ofic. 20
Buenos Aires, NON United States 1065

Argentina
```

Consulting Specialties:

CIFS (Samba) Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

5411-4300-7294

Company Fax:

5411-4300-7294

Support Phone Number:

5411-4300-7294

Contact Email Address:

soporte@generic.findhere.com

Company URL:

http://generic.findhere.com

Primary Contact:

Pablo Lazaro

1.5 Consultant ID: Argentina, Jorge Forte

```
Calle 47 Nro 826 Dep 9
La Plata, NON United States 1900

Argentina
```

Consulting Specialties:

CIFS (Samba) E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

542214821914

Company Fax:

N/A

Support Phone Number:

N/A

Contact Email Address:

soporte@forte.com.ar

Company URL:

http://www.forte.com.ar

Primary Contact:

Jorge Forte

1.6 Consultant ID: Argentina, NIXE S.R.L.

```
Pte. Luis S. Peña 366 4 A
Buenos Aires, NON United States 1110

Argentina
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

Slackware

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

+54 (11) 4383-7831

Company Fax:

+54 (11) 4383-7831

Support Phone Number:

+54 (11) 15-4444-832

Contact Email Address:

support@nixe.com.ar

Company URL:

http://www.nixe.com.ar

Primary Contact:

Gustavo Gasparrini

1.7 Consultant ID: Argentina, Nixe S.R.L.

```
Pte. Luis Saenz Pe#a 366 4A
Buenos Aires, NON United States 1110

Argentina
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Slackware

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

N/A

Company Phone Number:

+54 (11) 4383-7831

Company Fax:

+54 (11) 4383-7831

Support Phone Number:

+54 (11) 4383-7831

Contact Email Address:

support@nixe.com.ar

Company URL:

http://N/A

Primary Contact:

Gustavo Gasparrini

1.8 Consultant ID: Australia, Computer Clinic (WA) Pty Ltd

```
4/226 Main St Osborne Park
Perth WA, NON United States 6017

Australia
```

Consulting Specialties:

Custom Systems

Main Distribution:

Caldera

Other Distribution:

TOC

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

HP

Linux Certifications:

Caldera

Company Phone Number:

08 94406611

Company Fax:

08 94406610

Support Phone Number:

08 94406611

Contact Email Address:

support@cclinic.com.au

Company URL:

http://www.cclinic.com.au

Primary Contact:

Colin or Garth Hewitt or Atkinson

1.9 Consultant ID: Australia, Cybersource Pty. Ltd.

```
Level 8, 140 Queen Street
Melbourne, NON United States 3000

Australia
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

+61 3 9642 5997

Company Fax:

+61 3 9642 5998

Support Phone Number:

+61 3 9642 5997

Contact Email Address:

rh-support@cyber.com.au

Company URL:

http://www.cyber.com.au

Primary Contact:

Con Zymaris

1.10 Consultant ID: Australia, Cyberspace Corporation

```
3A/426 Burwood Highway
Melbourne, NON United States 3152

Australia
```

TOC

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

IBM

Linux Certifications:

Not Applicable

Company Phone Number:

+61 (3) 9887 4700

Company Fax:

+61 (3) 9887 2756

Support Phone Number:

+61 (3) 9887 4700

Contact Email Address:

support@cyberspace.net.au

Company URL:

http://www.cyberspace.net.au

Primary Contact:

Keith Chamberlain

1.11 Consultant ID: Australia, DataSource Pty. Ltd

N/A
SYDNEY, NON United States 2570

Australia

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+61418675892

Company Fax:

N/A

Support Phone Number:

+6146556792

Contact Email Address:

Linux@sysmon.net

Company URL:

http://www.sysmon.net

Primary Contact:

Steve Deadman

1.12 Consultant ID: Australia, EMUSYS Unix Consulting

20 Ivanhoe Rd
Croydon, NON United States 2132

Australia

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+61 2 9798 7604

Company Fax:

+61 2 9798 2854

Support Phone Number:

0500 500 EMU

Contact Email Address:

support@emusys.com.au

Company URL:

http://www.emusys.com.au

Primary Contact:

Anthony Rumble

1.13 Consultant ID: Australia, Echo Labs

```
29 Weld St, Nedlands
Perth, NON United States 6009

Australia
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+61 41 356 0008

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

echo@iinet.net.au

Company URL:

http://www.iinet.net.au/~echo

Primary Contact:

Ralph Billes

1.14 Consultant ID: Australia, Farrow Norris

```
58 St Georges Pde
Sydney, NON United States 2220

Australia
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+61 4 1724 3183

Company Fax:

+61 2 9546 4468

Support Phone Number:

+61 4 1724 3183

Contact Email Address:

support@fn.com.au

Company URL:

http://http://www.fn.com.au/

Primary Contact:

James Farrow

1.15 Consultant ID: Australia, Hinterlands Consultancy I.T.

```
1 Crusader Rd
Galston NSW., NON United States 2159

Australia
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

Intel and Alpha

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+61 2 9653 2837

Company Fax:

+61 2 9653 2937

Support Phone Number:

+61 2 9653 2837

Contact Email Address:

support@hinterlands.com.au

Company URL:

http://www.hinterlands.com.au

Primary Contact:

Chris Blown

1.16 Consultant ID: Australia, John Pearson

```
PO Box 3412 Rundle Mall
Adelaide, NON United States 5000

Australia
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

TOC

Company Phone Number:

+61 407391169

Company Fax:

+61 883644231

Support Phone Number:

+61 407391169

Contact Email Address:

huiac@camtech.net.au

Company URL:

http://N/A

Primary Contact:

John Pearson

1.17 Consultant ID: Australia, LINSUP.COM

```
P. O. Box 689
St. Ives, NON United States 2075

Australia
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Debian

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+61 2 9144-6131

Company Fax:

+61 2 9144-6138

Support Phone Number:

+61 2 9144-6131

Contact Email Address:

info@linsup.com

Company URL:

http://www.linsup.com

Primary Contact:

Richard Ames

1.18 Consultant ID: Australia, Linux Garden

```
Unit 4, 142 Culloden Road, Marsfield
Sydney, NSW, NON United States 2122

Australia
```

Consulting Specialties:

Custom Programming Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

61 0417 869242

Company Fax:

N/A

Support Phone Number:

61 0417 869242

Contact Email Address:

shyam@linuxgarden.com

Company URL:

http://www.linuxgarden.com

Primary Contact:

Shyam Govardhan

1.19 Consultant ID: Australia, Studio of Arts And Sciences (SAAS)

```
PO Box 214 Mosman
Sydney, NON United States NSW 2088

Australia
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Darwin and all MAC

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+61-2-9453-9100

Company Fax:

+61-2-9453-9101

Support Phone Number:

+61-2-9453-9100

Contact Email Address:

saas@saas.nsw.edu.au

Company URL:

http://www.saas.nsw.edu.au

Primary Contact:

Roger Buck

1.20 Consultant ID: Australia, TereDonn Computer Engineering

```
Suite 1, 128 Bowen St, Spring Hill
BRISBANE, ZZ 4004

Australia
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration

System Security

Main Distribution:

Other

Other Distribution:

ALL

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+61 7 32369366

Company Fax:

+61 7 32369930

Support Phone Number:

+61 414 663053

Contact Email Address:

service@tdce.com.au

Company URL:

http://www.tdce.com.au

Primary Contact:

Terence Giufre-Sweetser

1.21 Consultant ID: Australia, WWWalker Web Development

```
PO Box 288
Wentworthville, NON United States 2145
```

TOC

Australia

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Internet (Web) Programming Netware Connectivity Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

61-412-405-727

Company Fax:

61-2-97772058

Support Phone Number:

61-412-405727

Contact Email Address:

dwight@zip.com.au

Company URL:

http://www.wwwalker.com.au

Primary Contact:

Dwight Walker

1.22 Consultant ID: Australia, linsup.com

TOC

P. O. Box 689
St. Ives, NON United States 2075

Australia

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

61 2 9144-6131

Company Fax:

61 2 9144-6138

Support Phone Number:

61 2 9144-6131

Contact Email Address:

info@linsup.com

Company URL:

http://www.linsup.com

Primary Contact:

Richard Ames

1.23 Consultant ID: Austria, Bitsniz Data Systems

```
Kreuzgasse 60
Wien, NON United States A-1180

Austria
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Network Administration System Administration

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+43 (0)1 406 50 98 0

Company Fax:

++43 (0)1 406 50 98

Support Phone Number:

Not Available

Contact Email Address:

support@bitsniz.at

Company URL:

http://www.bitsniz.at

Primary Contact:

Franz Antonicek

1.24 Consultant ID: Austria, Bitsniz Data Systems

```
Kreuzgasse 60
Wien, NON United States A-1180

Austria
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Network Administration System Administration

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+43 (0)1 406 50 98 0

Company Fax:

++43 (0)1 406 50 98

Support Phone Number:

Not Available

Contact Email Address:

support@bitsniz.at

Company URL:

http://www.bitsniz.at

Primary Contact:

Franz Antonicek

1.25 Consultant ID: Austria, Horus GmbH

```
Jakob-Haringer-Straße 8
Salzburg, NON United States A-5020

Austria
```

Consulting Specialties:

CIFS (Samba) Custom Programming E-Commerce Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

+43 (0)662 450363-0

Company Fax:

+43 (0)662 450363-30

Support Phone Number:

+43 (0)662 450363-0

Contact Email Address:

info@horus.com

Company URL:

http://www.horus.com

Primary Contact:

Thomas 'Dune' Freina

1.26 Consultant ID: Austria, LINUX@work

```
Hauptstr. 99
Hagenberg, NON United States 4232

Austria
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

TOC

+43 7236 2200

Company Fax:

+43 7236 2200 22

Support Phone Number:

+43 7236 2200 13

Contact Email Address:

support@linuxatwork.at

Company URL:

http://www.linuxatwork.at

Primary Contact:

Andreas Hupfau

1.27 Consultant ID: Austria, LinzNet

```
Floetzerweg 150
Linz, NON United States 4030

Austria
```

Consulting Specialties:

CIFS (Samba) Custom Programming E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+43 732 370700

Company Fax:

+43 732 376554

Support Phone Number:

+43 732 370700

Contact Email Address:

support@linznet.at

Company URL:

http://www.linznet.at

Primary Contact:

Rainer Skarke

1.28 Consultant ID: Austria, communication.center.one

```
Hetzendorfer Str.23/20
Vienna, ZZ A-1120

Austria
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+43-1-8045753

Company Fax:

+43-1-8031553

Support Phone Number:

+43-676-477-70-70

Contact Email Address:

support@ccone.at

Company URL:

http://www.ccone.at

Primary Contact:

Gerhard Beck

1.29 Consultant ID: Austria, g.a.m.s. edv-dienstleistungen gmbh

```
Stiegergasse 15-17/8
Vienna, NON United States 1150

Austria
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+43 1 895 84 99

Company Fax:

+43 1 895 84 99 60

Support Phone Number:

+43 1 895 84 99

Contact Email Address:

linux@gams.at

Company URL:

http://www.gams.at

Primary Contact:

Andreas Kainz

1.30 Consultant ID: Austria, iDAS

```
Gasgasse 13/5
Vienna, NON United States A-1150

Austria
```

Consulting Specialties:

CIFS (Samba) Custom Systems Internet (Web) Programming Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

++43-664-2630131

Company Fax:

++43-1-8939122-26

Support Phone Number:

++43-664-2630131

Contact Email Address:

office@iDAS.co.at

Company URL:

http://www.iDAS.co.at

Primary Contact:

Thomas Kirchtag

1.31 Consultant ID: Belgium, Alexandre Dulaunoy, Consultance OpenSource

```
4, rue d'Aineffe
Borlez Faimes, NON United States 4317

Belgium
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet

Server Development Internet (Web) Programming Netware Connectivity Network Administra
System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

self-made

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

3219566115

Company Fax:

3219567337

Support Phone Number:

3219566115

Contact Email Address:

adulau@unix.be.eu.org

Company URL:

http://http://unix.be.eu.org/adulau

Primary Contact:

Alexandre Dulaunoy

1.32 Consultant ID: Belgium, BARTH Group's

```
12, rue de l'Espoir
Liège, NON United States 4030
```

Belgium

Consulting Specialties:

Custom Programming

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

32 4 367.49.16

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

BARTH.groups@skynet.be

Company URL:

http://www.skynet.be

Primary Contact:

Wathelet Michel

1.33 Consultant ID: Belgium, Better Access NV

Geldenaakse Vest 6

Leuven, NON United States 3000

Belgium

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Caldera

Specialty Distribution:

Other

Other Specialty Distribution:

Custombuilt Linux So

Reseller Authorizations:

Compaq HP Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+32 16 29 80 45

Company Fax:

+32 16 29 80 46

Support Phone Number:

+32 16 29 80 45

Contact Email Address:

info@ba.be

Company URL:

http://www.ba.be

Primary Contact:

Jan Guldentops

1.34 Consultant ID: Belgium, CompuSense bvba

TOC

Mr Vorsselmanslaan 27
Kalmthout, NON United States B 2920

Belgium

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security System Administration System Security

Main Distribution:

RedHat

Other Distribution:

SuSe

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

Sparc Linux

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+32 75 503503

Company Fax:

+32 3 6666579

Support Phone Number:

+32 75 503503

Contact Email Address:

herman.willekens@skynet.be

Company URL:

http://www.skynet.be

Primary Contact:

Herman Willekens

1.35 Consultant ID: Belgium, CoolNet

TOC

```
rue Croix Claire 67
NANDRIN, NON United States 4550

Belgium
```

Consulting Specialties:

Custom Programming E-Commerce Firewalls and Internet Security Internet Server Development Network Administration

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+32 75 93 51 71

Company Fax:

+32 4 372 01 38

Support Phone Number:

N/A

Contact Email Address:

marc.brocha@coolnet.be

Company URL:

http://www.coolnet.be

Primary Contact:

marc brocha N/A

1.36 Consultant ID: Belgium, Double Barrel Consultancy and Productions

```
Sportstraat 28
Gent, ZZ 9000

Belgium
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+32-9-2227764

Company Fax:

+32-9-2224976

Support Phone Number:

+32-9-2227764

Contact Email Address:

help@double-barrel.be

Company URL:

http://www.double-barrel.be

Primary Contact:

Michael Vergallen

1.37 Consultant ID: Belgium, SPIER bvba

```
Knaptandstraat 96
Sint-Niklaas, NON United States B9100

Belgium
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq

Linux Certifications:

Not Applicable

Company Phone Number:

+ 32 3 765 90 61

Company Fax:

+ 32 3 765 90 62

Support Phone Number:

+ 32 3 765 90 61

Contact Email Address:

support@spier.be

Company URL:

http://www.spier.be

Primary Contact:

Jan Lybeert

1.38 Consultant ID: Belgium, VirgoPlus

```
Rue Dartois, 1
Liège, NON United States 4000

Belgium
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+32-4-253 00 59

Company Fax:

+32-4-253 00 49

Support Phone Number:

+32-4-253 00 85

Contact Email Address:

bruno@virgoplus.com

Company URL:

http://www.virgoplus.com

Primary Contact:

Christophe Ozer

1.39 Consultant ID: Belgium, information technology partners

```
Klokhof 2
Heusden-Zolder, NON United States 3550

Belgium
```

Consulting Specialties:

E-Commerce Internet Server Development Internet (Web) Programming Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

Suse/Caldera

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+32 11 53 77 53

Company Fax:

+32 11 53 77 17

Support Phone Number:

+32 11 53 77 53

Contact Email Address:

guy.dillen@skynet.be

Company URL:

http://www.skynet.be

Primary Contact:

Guy Dillen

1.40 Consultant ID: Brazil, 4Web Internet

```
R. James Watt, 142 - Cj 42
Sao Paulo, NON United States 04576-050

Brazil
```

Consulting Specialties:

E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

55-11-5506-6844

Company Fax:

55-11-5506-6844

Support Phone Number:

55-11-5506-6844

Contact Email Address:

esaito@4web.com.br

Company URL:

http://www.4web.com.br

Primary Contact:

Eduardo Saito

1.41 Consultant ID: Brazil, Abner Graham Jacobsen

```
Rua Bruno Becacici, 107
Vitoria, NON United States 29042-0320

Brazil
```

Consulting Specialties:

CIFS (Samba) E-Commerce Firewalls and Internet Security Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Conectiva

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

55 027 322-1837

Company Fax:

55 027 223-0562

Support Phone Number:

Not Available

Contact Email Address:

ajacobsen@interlink.com.br

Company URL:

http://www.interlink.com.br

Primary Contact:

N/A N/A

1.42 Consultant ID: Brazil, Fernando Mauro Martins

```
Rua Padre Justino, 44
São Paulo, NON United States 05580-090

Brazil
```

Consulting Specialties:

Custom Programming

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

none

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+55 11 814-5679

Company Fax:

na

Support Phone Number:

+55 11 814-5679

Contact Email Address:

lalo@webcom.com

Company URL:

http://www.webcom.com

Primary Contact:

Fernando Martins

1.43 Consultant ID: Brazil, Pragana, Filhos E Cia Ltda

```
CxPostal 721 CDC Aldeia
Camaragibe, NON United States 54792-990

Brazil
```

Consulting Specialties:

Custom Programming

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

55-81-459-1776

Company Fax:

55-81-459-1776

Support Phone Number:

N/A

Contact Email Address:

rpragana@acm.org

Company URL:

http://N/A

Primary Contact:

Rildo Pragana

1.44 Consultant ID: Brazil, TELEMATICA Coml. de Teleinformática Ltda.

```
R. Princesa Isabel, 211
Pelotas, NON United States 96015-590

Brazil
```

Consulting Specialties:

CIFS (Samba) Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

Conectiva Linux

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+55 532 27-7266

Company Fax:

+55 532 27-7972

Support Phone Number:

+55 532 27-7266

Contact Email Address:

telematica@conesul.com.br

Company URL:

http://N/A

Primary Contact:

Vagner Farias

1.45 Consultant ID: Brazil, iSNet Serviços de Informatica

```
Av. Othon Gama d'Eça, 900/704
Florianopolis, NON United States 88015-240

Brazil
```

Consulting Specialties:

E-Commerce

Main Distribution:

Other

Other Distribution:

Conectiva

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

+55 48 322-0249

Company Fax:

N/A

Support Phone Number:

N/A

Contact Email Address:

info@isnet.com.br

Company URL:

http://www.isnet.com.br

Primary Contact:

Paulo Santos

1.46 Consultant ID: Bulgaria, University of Mining and Geology, Dept. Computer S

```
Studentski Grad
```

Sofia, NON United States 1700

Bulgaria

Consulting Specialties:

Custom Programming Custom Systems Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+359 2 62581 /ext.56

Company Fax:

+359 2

Support Phone Number:

+359 2 62581 /ext.56

Contact Email Address:

support@mgu.bg

Company URL:

http://www.mgu.bg

Primary Contact:

Volin Karagiozov

1.47 Consultant ID: Canada, 4 Office Automation

TOC

```
375 Britannia Road East
Mississauga, ON L4Z 3E2

Canada
```

Consulting Specialties:

CIFS (Samba)

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

HP

Linux Certifications:

Not Applicable

Company Phone Number:

1-416-463-3423

Company Fax:

1-905-501-1515

Support Phone Number:

1-416-463-3423

Contact Email Address:

hyeung@4office.com

Company URL:

http://www.4office.com

Primary Contact:

Hillman Yeung

1.48 Consultant ID: Canada, 8D Technologies inc.

Pobox 778
Hudson, PQ J0P1H0

Canada

Consulting Specialties:

CIFS (Samba) Custom Programming E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

514.994.2645

Company Fax:

514.994.2654

Support Phone Number:

514.994.2645

Contact Email Address:

info@8D.com

Company URL:

http://www.8D.com

Primary Contact:

Patrick Bernard

1.49 Consultant ID: Canada, ADR Computing

```
5-1115 West 10th Avenue
Vancouver, BC V6H 1J2

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1 604 202 2800

Company Fax:

N/A

Support Phone Number:

1 604 202 2800

Contact Email Address:

support@adrcomp.com

Company URL:

http://www.adrcomp.com

Primary Contact:

Chris Airriess

1.50 Consultant ID: Canada, Affinity Systems Inc.

```
Box 10, RR1
Innisfail, AB T4G 1T6

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-403-886-4638

Company Fax:

1-403-886-2663

Support Phone Number:

1-403-886-4638

Contact Email Address:

support@affinity-systems.ab.ca

Company URL:

http://www.affinity-systems.ab.ca

Primary Contact:

James Bourne

1.51 Consultant ID: Canada, Best Computers

```
10610 170 St
Edmonton, AB T5S-1P3

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

SuSE, Debian

Specialty Distribution:

Other

Other Specialty Distribution:

Secured Linux

Reseller Authorizations:

HP

Linux Certifications:

Not Applicable

Company Phone Number:

780-413-9830

Company Fax:

780-413-9831

Support Phone Number:

780-413-9830

Contact Email Address:

support@best-comp.com

Company URL:

http://http://www.best-comp.com/

Primary Contact:

Kurt Seifried

1.52 Consultant ID: Canada, CAL Consultants Inc.

```
358 Danforth Ave., Suite 29
Toronto, ON M4K 3Z2

Canada
```

Consulting Specialties:

E-Commerce Internet Server Development Network Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP

Linux Certifications:

N/A

Company Phone Number:

01.416.461.7634

Company Fax:

01.416.461.6908

Support Phone Number:

01.416.461.7634

Contact Email Address:

support@cal.ca

Company URL:

http://www.cal.ca

Primary Contact:

Graydon Clipperton

1.53 Consultant ID: Canada, Centurion Services

```
9351 Blundell Road
Richmond, BC V6Y 1K5

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

IBM

Linux Certifications:

N/A

Company Phone Number:

1-604-279-1857

Company Fax:

1-604-279-1800

Support Phone Number:

1-604-720-1857

Contact Email Address:

jwalter@rogers.wave.ca

Company URL:

http://www.rogers.wave.ca

Primary Contact:

Mr. Jan Walter

1.54 Consultant ID: Canada, Computing And Networking Centre

```
POBOX 28117
Dartmouth/Halifax, NS B2W6E2

Canada
```

Consulting Specialties:

Network Administration

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

902-465-3734

Company Fax:

902-465-3734

Support Phone Number:

902-465-3734

Contact Email Address:

cnc98@sprint.ca

Company URL:

http://N/A

Primary Contact:

Ahmed Bentiba

1.55 Consultant ID: Canada, DFC International Computers Inc.

```
52 Mowatt Court
Thornhill, ON L3T 6V5

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq IBM

Linux Certifications:

RedHat Not Applicable

Company Phone Number:

(905)731-6449

Company Fax:

(905)731-7684

Support Phone Number:

(905)731-6449

Contact Email Address:

support@dfc.com

Company URL:

http://www.dfc.com

Primary Contact:

David Fitzerman

1.56 Consultant ID: Canada, DFC International Computing Inc.

```
52 Mowatt Court
Thornhill, ON L3T 6V5

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

N/A

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

RedHat

Company Phone Number:

(905)731-6449

Company Fax:

(905)731-7684

Support Phone Number:

(905)731-6449

Contact Email Address:

support@dfc.com

Company URL:

http://www.dfc.com

Primary Contact:

David Fitzerman

1.57 Consultant ID: Canada, DPD Software Ltd

```
1455 Waverly St
Winnipeg, MB R2W 0RY

Canada
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet

Server Development Internet (Web) Programming Netware Connectivity Network Administra
System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Slak

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

IBM

Linux Certifications:

RedHat

Company Phone Number:

204-489-4398

Company Fax:

204-489-4398

Support Phone Number:

204-489-4398

Contact Email Address:

kevind@dpdsoftware.mb.ca

Company URL:

http://www.dpdsoftware.mb.ca

Primary Contact:

Kevin Druet

1.58 Consultant ID: Canada, Diminishing Networks

```
5330 Entwhistle Drive
Nanaimo, BC V9V 1H2
```

Canada

Consulting Specialties:

Custom Systems Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-250-758-2217

Company Fax:

N/A

Support Phone Number:

1-250-758-2217

Contact Email Address:

tech@dim.org

Company URL:

http://www.dim.org

Primary Contact:

Adam Jordens

1.59 Consultant ID: Canada, Gedris Linux Consulting

183 St. Andrew Street
Ottawa, ON K1N 5G3

Canada

Consulting Specialties:

Custom Programming Internet Server Development Internet (Web) Programming System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(613)562-0628

Company Fax:

N/A

Support Phone Number:

(613)562-0628

Contact Email Address:

vic@worldchat.com

Company URL:

http://www.worldchat.com

Primary Contact:

Victor Gedris

1.60 Consultant ID: Canada, Global Proximity Corporation

RR 7
Woodstock, ON N4S 7W2

Canada

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

+1 519 469 3439

Company Fax:

+1 519 469 8653

Support Phone Number:

+1 519 469 3439

Contact Email Address:

Support@Global.Proximity.ON.CA

Company URL:

http://www.Global.Proximity.ON.CA

Primary Contact:

Chris Tyler

1.61 Consultant ID: Canada, Hard Data Ltd.

```
11060 - 166 Avenue
Edmonton, AB T5X 1Y3

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

ALL

Specialty Distribution:

Other

Other Specialty Distribution:

Hard Hat Linux

Reseller Authorizations:

Compaq HP

Linux Certifications:

Not Applicable

Company Phone Number:

01-780-456-9771

Company Fax:

01-780-456-9772

Support Phone Number:

01-780-456-1510

Contact Email Address:

support@harddata.com

Company URL:

http://www.harddata.com

Primary Contact:

Maurice Hilarius

1.62 Consultant ID: Canada, JDP Computer Systems Inc.

```
206 Finch Ave. East
Toronto, ON L9W 3S9

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-416-733-3655

Company Fax:

1-416-733-1839

Support Phone Number:

1-416-733-3655

Contact Email Address:

support@jdp.com

Company URL:

http://www.jdp.com

Primary Contact:

Gordon Gray

1.63 Consultant ID: Canada, LeadSource Consultants Inc.

```
1227 Mississauga Road
Mississauga, ON L5H 2J1

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

slackware

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

HoNet

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

905 278 6808

Company Fax:

905 278 6863

Support Phone Number:

905 278 6808

Contact Email Address:

support@leadsource.ca

Company URL:

http://http://www.leadsource.ca

Primary Contact:

Adam Mikolajewicz

1.64 Consultant ID: Canada, Logisoft Technologies inc

```
718 pl Denise-pelletier
Ste-Julie, PQ J3E 2G8

Canada
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Mandrake

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(514) 570-7638

Company Fax:

N/A

Support Phone Number:

(514) 580-0050

Contact Email Address:

support@logisoftech.com

Company URL:

http://www.logisoftech.com

Primary Contact:

Guillaume Bourque

1.65 Consultant ID: Canada, MOE Online Enterprises

```
50 Stephanie, Suite 1008
Toronto, ON M5T1B3

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+1 416 588 0672

Company Fax:

N/A

Support Phone Number:

n/a

Contact Email Address:

mozai@canada.com

Company URL:

http://ativan.netdesign.net/~moses/

Primary Contact:

Moses Moore

1.66 Consultant ID: Canada, Montage IT Services

```
350, 708 - 11 Ave. SW
Calgary, AB T2R0E4

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

All

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

01.403.265.4461

Company Fax:

01.403.265.4462

Support Phone Number:

N/A

Contact Email Address:

dboyd@cal.montage.ca

Company URL:

http://www.montage.ca

Primary Contact:

Darren Boyd

1.67 Consultant ID: Canada, MostlyLinux

```
N/A
Calgary, AB N/A

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Systems Network Administration System Administration System Security

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Small Business serve

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

403-285-1399

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Scott Barker

1.68 Consultant ID: Canada, Net Direct Inc.

```
557 Havelock Dr.
Waterloo, ON N2L 4Z1

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

None

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

519-579-5006

Company Fax:

519-745-9940

Support Phone Number:

519-579-5006

Contact Email Address:

support@netdirect.ca

Company URL:

http://www.net.direct.ca

Primary Contact:

John Van Ostrand

1.69 Consultant ID: Canada, Pilkington Software Inc.

1046 Barrington St
Halifax, NS B3H 2R1

Canada

Consulting Specialties:

Custom Programming Custom Systems

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

902 423 6033

Company Fax:

902 423 6033

Support Phone Number:

902 423 6033

Contact Email Address:

support@cpsoft.com

Company URL:

http://www.cpsoft.com

Primary Contact:

Charles Pilkington

1.70 Consultant ID: Canada, Pryor and Pryor Inc.

TOC

```
602 - 1230 Comox Street
Vancouver, BC V6E 1K7

Canada
```

Consulting Specialties:

Custom Programming

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

604-685-2621

Company Fax:

604-683-3488

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://http://www.pryor-and-pryor.com

Primary Contact:

Roger Pryor

1.71 Consultant ID: Canada, Quist Consulting

TOC

```
219 Donlea Drive
Toronto, ON M4G2N1

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Debian

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

UltraPenguin

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1.416.696.7600

Company Fax:

1.416.978.6620

Support Phone Number:

1.416.696.7600

Contact Email Address:

russ@quist.on.ca

Company URL:

http://www.quist.on.ca

Primary Contact:

Russell Suherland

1.72 Consultant ID: Canada, Roaring Penguin Software Inc.

```
986 Eiffel Avenue
Ottawa, ON K2C 0J2

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Network Administration System Administration

Main Distribution:

Other

Other Distribution:

Any distro

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

613 851-4379

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

David Skoll

1.73 Consultant ID: Canada, TeleDynamics Communications Inc

```
7 Forest Place RR#1
Sauble Beach, ON N0H 2G0

Canada
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

519-422-1150

Company Fax:

519-422-2723

Support Phone Number:

Not Available

Contact Email Address:

teledynamics@canada.com

Company URL:

http://http://www.teledyn.com

Primary Contact:

Gary Murphy

1.74 Consultant ID: Canada, The People's Linux

```
109-288 East 14th Avenue
Vancouver, BC V5T 2M6

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

604-709-9512

Company Fax:

604-709-9512

Support Phone Number:

604-709-9512

Contact Email Address:

fred@nuge.com

Company URL:

http://http://www.linuxnut.com

Primary Contact:

Bob Strasser

1.75 Consultant ID: Canada, VL

```
221 BRitannia
Ottawa, ON K2B - 5X1

Canada
```

Consulting Specialties:

System Security

Main Distribution:

RedHat

Other Distribution:

Caldera

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

1 (613) 292-9818

Company Fax:

1 (613) 829-0967

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Hung Vu

1.76 Consultant ID: Canada, Webcon, Inc.

```
70 Forest Hill Ave.
Ottawa, ON K2C 1P6

Canada
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+1 613 276 6206

Company Fax:

+1 613 276 8206

Support Phone Number:

+1 613 276 6206

Contact Email Address:

tech@webcon.net

Company URL:

http://www.webcon.net

Primary Contact:

Robert Hardy

1.77 Consultant ID: Colombia, Linux Colombia Ltda

```
Avenida 5AN ·23DN68 Oficina 2-118B
Cali, NON United States N/A

Colombia
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

57 2 6607908

Company Fax:

57 2 6676596

Support Phone Number:

57 2 6607908

Contact Email Address:

linuxcol@linuxcolombia.com.co

Company URL:

http://www.linuxcolombia.com.co

Primary Contact:

Juan Diego Bolaños Ramirez

1.78 Consultant ID: Colombia, Skina Ltda.

```
Calle 95 #30-61 int 8
Santafe de Bogota, NON United States NA

Colombia
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

571 256-5008

Company Fax:

571 256-5008

Support Phone Number:

571 256-5008

Contact Email Address:

skina@skina.com.co

Company URL:

http://www.skina.com.co

Primary Contact:

Ricardo Naranjo_Faccini

1.79 Consultant ID: Czech Republic, Martin Knotek, poskytovani software

```
Havelkova 23
Brno, NON United States 625 00

Czech Republic
```

Consulting Specialties:

Custom Programming Custom Systems System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

420 5 356605

Company Fax:

N/A

Support Phone Number:

420 5 356605

Contact Email Address:

knotekmartin@email.cz

Company URL:

http://www.email.cz

Primary Contact:

Martin Knotek

1.80 Consultant ID: Denmark, Dansk Data Elektronik A S

```
Herlev Hovedgade 199
Herlev, NON United States 2730

Denmark
```

Consulting Specialties:

Custom Systems

Main Distribution:

Other

Other Distribution:

Any well known Linux

Specialty Distribution:

Other

Other Specialty Distribution:

N/A

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+45 44572000

Company Fax:

+45 44572001

Support Phone Number:

+45 44572010

Contact Email Address:

support@dde.dk

Company URL:

http://www.dde.dk

Primary Contact:

Jens Knudsen

1.81 Consultant ID: Denmark, Plomus

```
Nøddelunden 110
Smørum, NON United States DK-2765
```

Denmark

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+45 44 66 24 16

Company Fax:

+45 44 66 25 16

Support Phone Number:

+45 44 66 25 16

Contact Email Address:

support@plomus.dk

Company URL:

http://www.plomus.dk

Primary Contact:

Claus Sørensen

1.82 Consultant ID: Egypt, AB2

TOC

```
29 Alexander the Great St.
Alexandria, NON United States 21131

Egypt
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

Slackware

Specialty Distribution:

Other

Other Specialty Distribution:

Intel

Reseller Authorizations:

Not Applicable

Linux Certifications:

N/A

Company Phone Number:

+20 3 482 4756

Company Fax:

+20 3 483 5030

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://www.ab2.com

Primary Contact:

Khalid Baheyeldin

1.83 Consultant ID: Finland, Netsol Network Solutions Oy

```
Hiihtotie 3 A
Vantaa, NON United States 01280

Finland
```

Consulting Specialties:

Custom Systems Firewalls and Internet Security Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP IBM Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+358 9 34090410

Company Fax:

+358 9 34090404

Support Phone Number:

+358 9 34090410

Contact Email Address:

support@netsol.fi

Company URL:

http://www.netsol.fi

Primary Contact:

Timo Virtaneva

1.84 Consultant ID: Finland, SOT Finnish Sofware Engineering Ltd.

```
Hermiankatu 8 E
TAMPERE, NON United States 33720

Finland
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

Best Linux

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+358-3-316 5544

Company Fax:

+358-3-316 5959

Support Phone Number:

0600-95549

Contact Email Address:

linux@bestlinux.net

Company URL:

http://www.sot.com

Primary Contact:

Santeri Kannisto Kai Valijärvi

1.85 Consultant ID: Finland, Tmi Netics Konsultointi

```
Eteläniityntie 6
MUSTASAARI, NON United States FIN-65610

Finland
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Network Administration System Administration

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

Other

Other Specialty Distribution:

planned with custome

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+358 50 501 2132

Company Fax:

N/A

Support Phone Number:

+358 50 501 2132

Contact Email Address:

juhani.puska@netics.fi

Company URL:

http://www.netics.fi

Primary Contact:

Juhani Puska

1.86 Consultant ID: France, ALCOVE

```
12 - 13 place Indira Gandhi
Gennevilliers, NON United States 92230

France
```

Consulting Specialties:

CIFS (Samba) Custom Programming E-Commerce Firewalls and Internet Security Internet Server Development System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+33 1 47 33 82 84

Company Fax:

+33 1 47 33 76 98

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Anne Collet

1.87 Consultant ID: France, CRAO

```
19, rue Forcrand
Montpellier, NON United States 34000

France
```

Consulting Specialties:

Custom Systems E-Commerce Internet Server Development Internet (Web) Programming

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+33 4 67 63 24 27

Company Fax:

N/A

Support Phone Number:

N/A

Contact Email Address:

crao@crao.net

Company URL:

http://crao.net

Primary Contact:

Arnaud Fontaine

1.88 Consultant ID: France, EBC Consulting

```
16, rue des sorbiers
SORBEY, NON United States 57580

France
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

client specific

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+33 (0) 3. 87.64.58.

Company Fax:

+33 (0) 3. 87.64.58.

Support Phone Number:

+33 (0) 3. 87.64.58.

Contact Email Address:

ebconsul@altavista.net

Company URL:

http://www.ebcip.fr

Primary Contact:

Eric Binger

1.89 Consultant ID: France, NTCD

```
11 rue A. Carrel
PARIS, ZZ 75019

France
```

Consulting Specialties:

Custom Programming Custom Systems Internet Server Development Internet (Web) Programming System Administration

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

Debian

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

33 1 40 40 77 09

Company Fax:

33 1 40 40 77 09

Support Phone Number:

none

Contact Email Address:

http://www.ntcd.com/hotline/index.htm

Company URL:

http://ntcd

Primary Contact:

Didier Blanchard

1.90 Consultant ID: France, Paralline

```
71, avenue des Vosges
STRASBOURG, NON United States 67000

France
```

Consulting Specialties:

Custom Systems

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

33 3 88 14 17 40

Company Fax:

33 3 88 14 17 41

Support Phone Number:

33 3 88 14 17 40

Contact Email Address:

support@paralline.com

Company URL:

http://www.paralline.com

Primary Contact:

Pierre BRUA

1.91 Consultant ID: France, Paul Boyer Consultants Sarl

```
20 rue d'Hauteville
Paris, NON United States 75010

France
```

Consulting Specialties:

Firewalls and Internet Security System Security

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+33 1 47 700 801

Company Fax:

+33 1 47 700 850

Support Phone Number:

+33 1 47 700 801

Contact Email Address:

paulboyer@usa.net

Company URL:

http://www.usa.net

Primary Contact:

Paul Boyer

1.92 Consultant ID: France, Thorvax

```
39, rue Armand Silvestre
Courbevoie, NON United States 92400

France
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+33 6 80 08 93 83

Company Fax:

+33 1 43 33 51 33

Support Phone Number:

+33 6 80 08 93 83

Contact Email Address:

Illaire@Club-Internet.fr

Company URL:

http://www.penguinpowered.com/~tyrann

Primary Contact:

Vincent Illaire

1.93 Consultant ID: Germany, Andreas Spengler EDV-Service

```
Gronauer Strasse 9
Karben, NON United States 61184

Germany
```

Consulting Specialties:

Custom Programming Custom Systems Firewalls and Internet Security Network Administration System Administration

Main Distribution:

Suse

Other Distribution:

RedHat

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49 6039 44401

Company Fax:

+49 6039 44401

Support Phone Number:

+49 6039 44401

Contact Email Address:

spengler@math.uni-frankfurt.de

Company URL:

http://N/A

Primary Contact:

Andreas Spengler

1.94 Consultant ID: Germany, Bigge and Langanke GbR

Corunnastr. 1
Iserlohn, NON United States 58636

Germany

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Internet Server Development Internet (Web) Programming

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

N/A

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

support@transdata.de

Company URL:

http://www.transdata.de

Primary Contact:

Frank Bigge

1.95 Consultant ID: Germany, Call-a-Server

TOC

```
N/A
Berlin, NON United States D-14129

Germany
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

DLD, RedHat and SuSE

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

N/A

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49 +30 +177 7170896

Company Fax:

+49 +30 8017423

Support Phone Number:

+49 +30 +177 7170896

Contact Email Address:

support@call-a-server.de

Company URL:

http://www.call-a-server.de

Primary Contact:

Joachim W.-D. von Thadden

1.96 Consultant ID: Germany, Gecko GmbH

Ringstr. 53
Bad Oeynhausen, NON United States 32549

Germany

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49.5731.741010

Company Fax:

+49.5731.741013

Support Phone Number:

Not Available

Contact Email Address:

support@lizard.de

Company URL:

http://www.lizard.de

Primary Contact:

Eike Sieker

1.97 Consultant ID: Germany, Gerd Aschemann

```
Osannstr. 49
Darmstadt, NON United States 64285

Germany
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Suse

Other Distribution:

Redhat, Caldera

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49/6151/16-2259

Company Fax:

+49/6151/16-3052

Support Phone Number:

+49/6151/16-2259

Contact Email Address:

Aschemann@Informatik.TU-Darmstadt.de

Company URL:

http://www.Informatik.TU-Darmstadt.de

Primary Contact:

Gerd Aschemann

1.98 Consultant ID: Germany, ID-Pro GmbH

```
Koenigswinterer Str. 116
Bonn, NON United States 53227

Germany
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

All major distributi

Specialty Distribution:

Other

Other Specialty Distribution:

N/A

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49-(0)228-42154-0

Company Fax:

+49-(0)228-42154-29

Support Phone Number:

+49-(0)228-42154-19

Contact Email Address:

support@id-pro.de

Company URL:

http://id-pro.de

Primary Contact:

Daniel Riek

1.99 Consultant ID: Germany, Jochen Laser IT-Services

```
Fliederweg 2a
Woebbelin, NON United States 19288

Germany
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Suse

Other Distribution:

others as well

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49 (0)38753 80181

Company Fax:

on request

Support Phone Number:

+49 (0)38753 80181

Contact Email Address:

on request

Company URL:

http://N/A

Primary Contact:

Jochen Laser

1.100 Consultant ID: Germany, LINUXHAUS

```
RINGBAHNSTR. 13
BERLIN, NON United States 10711

Germany
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

SuSE

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49 30 890 944 63

Company Fax:

+49 30 890 944 64

Support Phone Number:

Not Available

Contact Email Address:

support@linuxhaus.de

Company URL:

http://www.linuxhaus.de

Primary Contact:

Joerg Fruehbrodt

1.101 Consultant ID: Germany, Linux-Systemhaus Schulz

```
Hauptfeld 18
Dortmund, NON United States 44369

Germany
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Internet (Web) Programming Netware Connectivity Network Administration System Administration

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

none

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

0049 172 2362995

Company Fax:

N/A

Support Phone Number:

N/A

Contact Email Address:

schulz@linux-systemhaus.de

Company URL:

http://www.Linux-Systemhaus.de

Primary Contact:

Karsten Schulz

1.102 Consultant ID: Germany, Manfred Kunde EDV-Beratung

```
Hauptstrasse 70
Karben, NON United States 61184

Germany
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Internet Server Development Internet (Web) Programming

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49-(0)-06039-931064

Company Fax:

+49-(0)-06039-931065

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://http://www.Manfred-Kunde.de

Primary Contact:

Manfred Kunde

1.103 Consultant ID: Germany, Matthias Stolte Datentechnik

```
N/A
33613 Bielefeld, NON United States N/A

Germany
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

SuSE und andere...

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

N/A

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49-521-880056

Company Fax:

+49-521-880059

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

N/A N/A

1.104 Consultant ID: Germany, PiN - Präsenz im Netz Ges. f. Informationstechnolo

```
Leimbacher Str. 36
Wuppertal, NON United States 42281

Germany
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

Alle Distributionen

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49-202-2501164

Company Fax:

+49-202-2501165

Support Phone Number:

+49-202-2501164

Contact Email Address:

support@pinserve.de

Company URL:

http://www.pinserve.de

Primary Contact:

Andre Dressler

1.105 Consultant ID: Germany, Simulina GmbH

```
Trajanstraße  8
LADENBURG, NON United States DE-68526
```

Germany

Consulting Specialties:

Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

+49 6203 92 20 41

Company Fax:

+49 6203 92 20 42

Support Phone Number:

+49 6203 92 40 41

Contact Email Address:

hk@simulina.se

Company URL:

http://www.ncc-mannheim.net/simulina

Primary Contact:

Håkan Källberg

1.106 Consultant ID: Germany, Syslab.com oHg

TOC

Karl-Theodor Straße 66
München, NON United States 80803

Germany

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49 89 306358-90

Company Fax:

+49 89 306358-99

Support Phone Number:

+49 89 306358-92

Contact Email Address:

support@syslab.com

Company URL:

http://www.syslab.com

Primary Contact:

Manfred Lang

1.107 Consultant ID: Germany, WebArtists Internet Services

```
Steendammwisch 49
Hamburg, NON United States 22459

Germany
```

Consulting Specialties:

Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49-40-555 40 427

Company Fax:

+49-40-555 40 182

Support Phone Number:

0172/4032506

Contact Email Address:

support@linuxberatung.com

Company URL:

http://www.webartists.net

Primary Contact:

Markus Braasch

1.108 Consultant ID: Germany, Wesemann Software and Consulting

```
Heinrich-Plett-Allee 18
Bremen, NON United States D-28259

Germany
```

Consulting Specialties:

System Administration

Main Distribution:

Suse

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

N/A

Company Phone Number:

++49 421 579 579 3

Company Fax:

++49 421 579 58 28

Support Phone Number:

+49 421 579 579 3

Contact Email Address:

info@wscnet.de

Company URL:

http://www.wscnet.de

Primary Contact:

Andreas Wesemann

1.109 Consultant ID: Germany, Wolfgang Kirch

```
Gustav-Heinemann-Ring 51
Eisenberg, NON United States 67304

Germany
```

Consulting Specialties:

Firewalls and Internet Security

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

49 6351 398761

Company Fax:

49 6351 398762

Support Phone Number:

Not Available

Contact Email Address:

Kirch.Wolfgang@t-online.de

Company URL:

http://www.t-online.de

Primary Contact:

Wolfgang Kirch

1.110 Consultant ID: Germany, WorNet Internetdienste

```
Bürgermeister-Graf-Ring 28
Geretsried, NON United States D-82538

Germany
```

Consulting Specialties:

Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49 8171 41809-0

Company Fax:

+49 8171 41809-9

Support Phone Number:

Not Available

Contact Email Address:

wornet@wor.net

Company URL:

http://www.wor.net

Primary Contact:

Christian Eich

1.111 Consultant ID: Germany, ariadne_net

```
Köpenicker Str. 8
Berlin, NON United States 10997

Germany
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

0493061280438

Company Fax:

0493061280437

Support Phone Number:

0493061280438

Contact Email Address:

aduecker@t-online.de

Company URL:

http://www.t-online.de

Primary Contact:

Andreas Dücker

1.112 Consultant ID: Germany, b.i.t. beratungsgesellschaft für informations-tech

```
elisabethenstr. 62
darmstadt, NON United States 64283

Germany
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Internet Server Development Network Administration System Administration System Security

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49-6151-827575

Company Fax:

+49-6151-827576

Support Phone Number:

+49-6151-827575

Contact Email Address:

support@b-i-t.de

Company URL:

http://www.b-i-t.de

Primary Contact:

Andy Schmidt

1.113 Consultant ID: Germany, conIT GmbH

```
Industriering 7
Grosswallstadt, NON United States 63868

Germany
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49 (0)6022/262-082

Company Fax:

+49 (0)6022/262-101

Support Phone Number:

+49 (0)6022/262-102

Contact Email Address:

support@conIT.com

Company URL:

http://www.conIT.com

Primary Contact:

Arnold Weis

1.114 Consultant ID: Germany, dev consulting GmbH

```
Webereistr. 3
Bielefeld, NON United States 33602

Germany
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity

Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

Linux Server

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat Not Applicable

Company Phone Number:

N/A

Company Fax:

+49 521 1365803

Support Phone Number:

+49 521 1365800

Contact Email Address:

info@devcon.net

Company URL:

http://www.devcon.net

Primary Contact:

Winfried Motzkus

1.115 Consultant ID: Germany, frontsite AG

```
Eschollbrücker Straße 4a
Darmstadt, NON United States 64283

Germany
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

high avail. and ISDN

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49-6151-300-88-80

Company Fax:

+49-6151-300-88-80

Support Phone Number:

+49-6151-300-88-80

Contact Email Address:

info@frontsite.de

Company URL:

http://www.frontsite.de

Primary Contact:

Mark Semmler

1.116 Consultant ID: Germany, intraDAT GmbH

```
Wilhelm-Leuschner-Str. 9-11
Frankfurt/Main, NON United States D-60329
```

TOC

Germany

Consulting Specialties:

Custom Programming E-Commerce Firewalls and Internet Security Internet (Web) Programming Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49 +69-9740-5703

Company Fax:

+49 +69-9740-5705

Support Phone Number:

+49 +69-9740-5703

Contact Email Address:

email@intradat.com

Company URL:

http://www.intradat.com

Primary Contact:

Manfred Schmid

1.117 Consultant ID: Germany, ipm intranet project management gmbh

Huelchrather Strasse 17-23
Koeln, NON United States 50670

Germany

Consulting Specialties:

E-Commerce

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

49-221-97353217

Company Fax:

49-221-97353216

Support Phone Number:

49-221-97353217

Contact Email Address:

support@ipm-koeln.de

Company URL:

http://http://www.ipm-koeln.de

Primary Contact:

Dietmar Horn

1.118 Consultant ID: Germany, loth systemtechnik

```
Julius-Vosseler-Strasse 136
Hamburg, ZZ D-22527

Germany
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

N/A

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

N/A

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49 40 40192181

Company Fax:

+49 40 40192183

Support Phone Number:

+49 171 5406784

Contact Email Address:

support@loth.de

Company URL:

http://www.loth.de

Primary Contact:

Michael Loth

1.119 Consultant ID: Germany, netServe oHG Münster

```
Fresnostraße 10a
Münster, NON United States 48159

Germany
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

49-251-98725-0

Company Fax:

49-251-866977

Support Phone Number:

49-251-98725-21

Contact Email Address:

info@muenster.com

Company URL:

http://www.muenster.com

Primary Contact:

Torsten Mueller

1.120 Consultant ID: Germany, ralf geschke internet consulting

```
Roemerhofweg 1a
Erftstadt, NON United States 50374

Germany
```

Consulting Specialties:

Internet Server Development Internet (Web) Programming Network Administration System Administration

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49-177-5262345

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

support@geschke.nt

Company URL:

http://www.geschke.nt

Primary Contact:

Ralf Geschke

1.121 Consultant ID: Germany, unitegs - Unix Systems Technologies

```
Wacholderweg 11
Ilmenau, NON United States 98693

Germany
```

Consulting Specialties:

Custom Programming Custom Systems System Administration

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+49 3677 884620

Company Fax:

+49 3677 884621

Support Phone Number:

Not Available

Contact Email Address:

support@unitegs.de

Company URL:

http://www.unitegs.de

Primary Contact:

Gunar Schorcht

1.122 Consultant ID: Hong Kong, Cyber Channel Int'l Ltd

```
Room 1723 610 Nathan Road Kowloon
Hong Kong SAR, NON United States N/A

Hong Kong
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration Virtual Private Networking

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

IBM

Linux Certifications:

Not Applicable

Company Phone Number:

852 23754341

Company Fax:

852 23754148

Support Phone Number:

852 27823211

Contact Email Address:

support@cyberchn.com.hk

Company URL:

http://http://www.cyberchn.com.hk

Primary Contact:

Thomas Lee Paul Lee

1.123 Consultant ID: Hong Kong, Hong Kong Terminal

```
72A G/F Tokwawan Road,
Kowloon, NON United States N/A

Hong Kong
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

852-21420007

Company Fax:

852-27814919

Support Phone Number:

852-21420007

Contact Email Address:

support@zero.com.hk

Company URL:

http://http://zero.com.hk

Primary Contact:

Peter Wong

1.124 Consultant ID: Hong Kong, Linux Center (Hong Kong) Ltd.

```
Unit A, 5th Floor, Comet Comeercial Building , 42A
Kowloon, NON United States N/A

Hong Kong
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration

System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+852 2959 3675

Company Fax:

+852 2959 3672

Support Phone Number:

available to custome

Contact Email Address:

available to customers only

Company URL:

http://N/A

Primary Contact:

Chris Yung

1.125 Consultant ID: Hong Kong, Uniforce System Ltd.

```
Rm 903B, Kaiser Center, 18 Centre Street, Sai Ying
Hong Kong, NON United States N/A
```

TOC

Hong Kong

Consulting Specialties:

Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+852 24464316

Company Fax:

+852 21918185

Support Phone Number:

+852 24464316

Contact Email Address:

support@uniforce.net

Company URL:

http://www.uniforce.net

Primary Contact:

Francis Kam

1.126 Consultant ID: Hong Kong, Vortex Ltd.

TOC

```
Tsim Sha Tsui
Hong Kong, NON United States N/A

Hong Kong
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Network Administration System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(852) 2302-1988

Company Fax:

(852) 2302-1212

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Stephen Wong Elizabeth Hui

1.127 Consultant ID: Hong Kong, Vortex Ltd.

```
Tsim Sha Tsui
Hong Kong, NON United States N/A

Hong Kong
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Network Administration System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(852) 2302-1988

Company Fax:

(852) 2302-1212

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Stephen Wong Elizabeth Hui

1.128 Consultant ID: Hong Kong, Yoric Network Ltd.

```
Rm 801, On Hong Commerical Building, 145 Hennessy
Wanchai, NON United States N/A

Hong Kong
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Slackware

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

852 2391 4685

Company Fax:

852 2330 4818

Support Phone Number:

852 2391 4685

Contact Email Address:

support@yoric.com

Company URL:

http://www.yoric.com

Primary Contact:

Vincent Chan

1.129 Consultant ID: India, C and B Consulting

```
15/5 18th Cross
Bangalore, NON United States 560055

India
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+91 (80) 3440397

Company Fax:

+91 (80) 3341137

Support Phone Number:

+91 (80) 3440397

Contact Email Address:

linux-support@cbconsulting.com

Company URL:

http://www.cbconsulting.com

Primary Contact:

Atul Chitnis

1.130 Consultant ID: India, Connect Solutions

```
Not Applicable
Not Appicable, NON United States 10

India
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security

Main Distribution:

Suse

Other Distribution:

Easy, Debian, RedHat

Specialty Distribution:

Other

Other Specialty Distribution:

Testing all kinds of

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

not applicable

Company Fax:

not applicable

Support Phone Number:

not applicable

Contact Email Address:

sriram@connectsolutions.net

Company URL:

http://www.connectsolutions.net

Primary Contact:

Sriram V

1.131 Consultant ID: India, DSF Internet Services

```
C-29 Neeti Bagh
New Delhi, NON United States 110049

India
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

91-11-685-6158

Company Fax:

91-11-685-6157

Support Phone Number:

91-11-685-6158

Contact Email Address:

support@indoworld.com

Company URL:

http://www.indoworld.com

Primary Contact:

Anmol Taneja

1.132 Consultant ID: India, Netangle Com Pvt ltd

```
# 2 , Vishal Bhavan , Road no 12, Banjara hills
Hyderabad, ZZ 500034

India
```

Consulting Specialties:

CIFS (Samba)

Main Distribution:

RedHat

Other Distribution:

Debian

Specialty Distribution:

Other

Other Specialty Distribution:

RPM,TAR

Reseller Authorizations:

HP

Linux Certifications:

Not Applicable

Company Phone Number:

091-40-3300394

Company Fax:

091-40-3391412

Support Phone Number:

091-40-3328207

Contact Email Address:

goswami@netangle.us-inc.com

Company URL:

http://www.netangle.us-inc.com

Primary Contact:

satyakam goswami

1.133 Consultant ID: India, Shah Micro System

```
103, Orental House, 229/231, Samuel Street, Musjid
Bombay, NON United States 400003

India
```

Consulting Specialties:

CIFS (Samba) Custom Systems Internet Server Development Netware Connectivity Network Administration System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

All Linux Flavours

Specialty Distribution:

Other

Other Specialty Distribution:

All i386 Distributio

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+91-22-8862711

Company Fax:

+91-22-8868214

Support Phone Number:

+91-22-8862711

Contact Email Address:

dinesh@indiamal.com

Company URL:

http://members.tripod.com/~dineshah/

Primary Contact:

Dinesh Shah

1.134 Consultant ID: Indonesia, Netindo Pusaka Prima pt.

```
Duta Harapan Indah  Blok SS/10, Jl. Kapuk Muara
Jakarta, NON United States 14460

Indonesia
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

+6221 - 6631868

Company Fax:

+6221 - 6623606

Support Phone Number:

n/a

Contact Email Address:

support@netindo.co.id

Company URL:

http://www.netindo.co.id

Primary Contact:

Andy Luhur

1.135 Consultant ID: Indonesia, PT Cakram DataLingga Duaribu

```
Jln Gunung Batu 131
Bogor, NON United States 16610

Indonesia
```

Consulting Specialties:

CIFS (Samba) Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+62-251-332122

Company Fax:

+62-251-332122

Support Phone Number:

+62-251-332122

Contact Email Address:

linux-support@cdl2000.or.id

Company URL:

http://http://www.cdl2000.or.id

Primary Contact:

Maliana Harsoadi

1.136 Consultant ID: Israel, Breakthrough LTD

```
Hatzabar 18/9
Kiron, NON United States 55106

Israel
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet

Server Development Network Administration System Administration System Security

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

N/A

Company Phone Number:

972-52-442549

Company Fax:

972-3-6356434

Support Phone Number:

972-52-442549

Contact Email Address:

webmaster@breakt.co.il

Company URL:

http://www.breakt.co.il

Primary Contact:

Alex Rier N/A

1.137 Consultant ID: Israel, vPrise

```
50, Disengof
Tel-Aviv, NON United States N/A

Israel
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Internet Server Development Internet (Web) Programming Network Administration System Administration

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

972-54-654576

Company Fax:

972-151-54-654576

Support Phone Number:

972-54-654576

Contact Email Address:

support@vprise.com

Company URL:

http://www.vprise.com

Primary Contact:

Shai Almog

1.138 Consultant ID: Italy, Alessandro Rubini

```
Viale Golgi 72
PAvia, NON United States 27100
```

Italy

Consulting Specialties:

Custom Programming Custom Systems Firewalls and Internet Security Network Administration

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

ET-Linux (embedded 3

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+39 0382 529554

Company Fax:

N/A

Support Phone Number:

+39 0382 529554

Contact Email Address:

alessandro.rubini@linux.it

Company URL:

http://www.linux.it

Primary Contact:

Alessandro Rubini

1.139 Consultant ID: Italy, Francesco Patamia

Via G. Marrazzo, 2
Cropani Marina, NON United States N/A

Italy

Consulting Specialties:

CIFS (Samba) E-Commerce Internet Server Development Internet (Web) Programming Network Administration System Administration

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+390961-961486

Company Fax:

+390961-961155

Support Phone Number:

+390338-8165310

Contact Email Address:

patamia@biponline.com

Company URL:

http://http://www.biponline.com/fpnet

Primary Contact:

Francesco Patamia

1.140 Consultant ID: Italy, Luca Perugini

TOC

Via Furio Camillo 27
Passo Corese (RI), NON United States 02036

Italy

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

+390765488457

Company Fax:

N/A

Support Phone Number:

+3903473392355

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

N/A N/A

1.141 Consultant ID: Italy, Marco Michelino

```
Viale Colli Aminei 36
Napoli, NON United States 80131

Italy
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Network Administration System Administration System Security

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

N/A

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+390817418669

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

mmarco@hempseed.com

Company URL:

http://www.hempseed.com

Primary Contact:

Marco Michelino

1.142 Consultant ID: Italy, Mauro Barella

```
Via Roma, 8
UDINE, NON United States 33100

Italy
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

N/A

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+39 0432 512592

Company Fax:

N/A

Support Phone Number:

N/A

Contact Email Address:

N/A

Company URL:

http://www.maurobarella.org

Primary Contact:

Mauro Barella

1.143 Consultant ID: Italy, PROSA Progettazione Sviluppo Aperto

```
via Degli Zabarella, 3
Padova, NON United States 35121

Italy
```

Consulting Specialties:

Custom Programming

Main Distribution:

Debian

Other Distribution:

PROSA Debian GNU/Lin

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

+39 049 66 05 19

Company Fax:

+39 049 87 80 504

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Davide Barbieri

1.144 Consultant ID: Italy, Panservice

```
Via Nervi - CC LT Fiori - Torre 8
Latina, NON United States 04100

Italy
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

39 0773 410020

Company Fax:

39 0773 410020

Support Phone Number:

39 0773 410084

Contact Email Address:

helpdesk@panservice.it

Company URL:

http://http://www.panservice.it

Primary Contact:

Giuliano Peritore

1.145 Consultant ID: Italy, Studio LEADER Pro

```
Via Pietrastretta 76
TRENTO, NON United States 38100

Italy
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Network Administration System Security Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

Caldera RedHat

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

HAL91

Reseller Authorizations:

Not Applicable

Linux Certifications:

Caldera Not Applicable

Company Phone Number:

+39 461828229

Company Fax:

+39 461829826

Support Phone Number:

Not Available

Contact Email Address:

info@leader.it

Company URL:

http://www.leader.it

Primary Contact:

Guido Brugnara

1.146 Consultant ID: Japan, M and T

```
1-8-19 Toshimaku Minami Ikebukuro
Tokyo, NON United States 171

Japan
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+81 3 5396 5966

Company Fax:

+81 3 3983 4542

Support Phone Number:

+81 3 5396 5966

Contact Email Address:

fisher@m-t.com

Company URL:

http://www.m-t.com

Primary Contact:

Masahiro Otsuka

1.147 Consultant ID: Kuwait, Internet Universe - Network Solutions

```
Dasma
P.O.Box 2, NON United States non

Kuwait
```

Consulting Specialties:

Custom Systems Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

General Linux networ

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+965-262-5678

Company Fax:

+965-262-5679

Support Phone Number:

+965-262-5678

Contact Email Address:

support@kuwait-net.com

Company URL:

http://www.kuwait-net.com

Primary Contact:

Faisal Al-Abhoul

1.148 Consultant ID: Malaysia, Linux Resources Sdn. Bhd.

```
No. 33-1, Jalan 46A/26, Pusat Bandar Taman Sri Ram
Setapak, NON United States 53300

Malaysia
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

N/A

Company Phone Number:

+603-4139010

Company Fax:

+603-4139011

Support Phone Number:

+603-4139010

Contact Email Address:

drliwa@pc.jaring.my

Company URL:

http://www.ihsan.com/linux

Primary Contact:

Dr. Liwaudin Muhamad

1.149 Consultant ID: Malaysia, MAGNIFIX

```
2102 Wisma Rampai, Jln 34/26
Kuala Lumpur, NON United States 53300

Malaysia
```

Consulting Specialties:

Internet Server Development

Main Distribution:

RedHat

Other Distribution:

SUSE

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq

Linux Certifications:

RedHat

Company Phone Number:

603 412 1775

Company Fax:

603 412 1550

Support Phone Number:

603 412 1776

Contact Email Address:

info@magnifix.com.my

Company URL:

http://www.magnifix.com.my

Primary Contact:

Izauddin Isa

1.150 Consultant ID: Mexico, Gustavo Cavazos

```
N/A
Mexico City, NON United States N/A

Mexico
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System

Administration System Security

Main Distribution:

RedHat

Other Distribution:

Slackware

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

N/A

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

gcavazos@yahoo.com

Company URL:

http://N/A

Primary Contact:

Gustavo Cavazos

1.151 Consultant ID: Mexico, Linux Michoacan OpenSource

```
186 Gral Antonio Cabales
Morelia, NON United States 58000
```

Mexico

Consulting Specialties:

Custom Programming Custom Systems Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

CHARAL

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+5243 246824

Company Fax:

+5243 246824

Support Phone Number:

+5243 246824

Contact Email Address:

linux@ciencia.dcc.umich.mx

Company URL:

http://www.ciencia.dcc.umich.mx

Primary Contact:

Cederik De Leon

1.152 Consultant ID: Mexico, Servicios en Alta Tecnologia

Jose Ma. Chavez #420, Centro
Aguascalientes, NON United States 20000

Mexico

Consulting Specialties:

System Administration

Main Distribution:

RedHat

Other Distribution:

SUSE

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

52(49)15-70-26

Company Fax:

52(49)15-70-26

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Hector Vega

1.153 Consultant ID: Mexico, Virtual.Net

```
Martin Torres #200
Xalapa, Veracruz, NON United States 91020

Mexico
```

Consulting Specialties:

E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Slackware 4.0, Suse

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

N/A

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+52 (28) 135187

Company Fax:

+52 (28) 135187

Support Phone Number:

+52 (28) 249063

Contact Email Address:

soporte@virtual.net.mx

Company URL:

http://www.virtual.net.mx

Primary Contact:

Roman Torres

1.154 Consultant ID: Morocco, ANNOUR Technologies

```
89 Bd Anfa, Appt. 9 A,
CASABLANCA, NON United States 20000

Morocco
```

Consulting Specialties:

E-Commerce

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

HP

Linux Certifications:

RedHat

Company Phone Number:

212 2 48 11 34

Company Fax:

212 2 48 70 56

Support Phone Number:

212 2 48 68 09

Contact Email Address:

info@annour.com

Company URL:

http://www.annour.com

Primary Contact:

Karim BENABID Hicham BENABID

1.155 Consultant ID: Netherlands, Abe I.T. B.V.

```
Oude haagweg 295
Den Haag, NON United States 2552 EP

Netherlands
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+31704400368

Company Fax:

+31704400369

Support Phone Number:

+31704400368

Contact Email Address:

info@aboveit.nl

Company URL:

http://www.aboveit.nl

Primary Contact:

Alex Schreuder

1.156 Consultant ID: Netherlands, BitWizard B.V.

```
Van Bronckhorststraat 12
Delft, NON United States 2612 XV

Netherlands
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development System Security

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+31-15-2137555

Company Fax:

+31-15-2138217

Support Phone Number:

+31-15-2137555

Contact Email Address:

info@BitWizard.nl

Company URL:

http://www.BitWizard.nl

Primary Contact:

Roger Wolff

1.157 Consultant ID: Netherlands, CMG

```
Kralingseweg 241-249
Rotterdam, NON United States 3009 AN

Netherlands
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

RedHat

Company Phone Number:

+31.10.2537000

Company Fax:

+31.2537032

Support Phone Number:

N/A

Contact Email Address:

Henk.Drenth@cmg.nl

Company URL:

http://www.cmg.nl

Primary Contact:

Henk Drenth

1.158 Consultant ID: Netherlands, CistroN

```
38, Ondernemingsweg
Alphen aan den Rijn, NON United States 2404 HN

Netherlands
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+31-172-421071

Company Fax:

+31-172-430979

Support Phone Number:

n/a

Contact Email Address:

linux@cistron.nl

Company URL:

http://www.cistron.nl

Primary Contact:

Danny ter Haar

1.159 Consultant ID: Netherlands, ComCare IT BV

```
Papaverweg 34
Amsterdam, NON United States 1032 KJ

Netherlands
```

Consulting Specialties:

CIFS (Samba) E-Commerce Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

RedHat

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+31 20 4942165

Company Fax:

+31 20 4942164

Support Phone Number:

N/A

Contact Email Address:

N/A

Company URL:

http://www.comcare.nl

Primary Contact:

Sjoerd Venema

1.160 Consultant ID: Netherlands, Le Reseau netwerksystemen BV

```
Bieslookstraat 31, Unit A3
Groningen, NON United States NL-9731 HH

Netherlands
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

Secure Linux (own di

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

and LinuxRouter

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+31 50 549 2701

Company Fax:

+31 50 549 2310

Support Phone Number:

+31 50 549 4548

Contact Email Address:

support@reseau.nl

Company URL:

http://www.reseau.nl/

Primary Contact:

Arthur Donkers Rudo de Jong

1.161 Consultant ID: Netherlands, Linvision

```
Schoemakerstraat 95 A
Delft, NON United States 2628 AX

Netherlands
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

TOC

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

BeoWulf Clustering

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+31 15 260 04 33

Company Fax:

+31 15 260 05 04

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Geerten Schram

1.162 Consultant ID: Netherlands, Manna Automatisering

```
Meteorenstraat 116
Hilversum, NON United States 1223 EV

Netherlands
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Netware Connectivity Networ Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Advise

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

31356423980

Company Fax:

31356423981

Support Phone Number:

31356423982

Contact Email Address:

info@manna.nl

Company URL:

http://www.manna.nl

Primary Contact:

N/A N/A

1.163 Consultant ID: Netherlands, Maxfort Consultancy

```
Albardastraat 11
```

Schiedam, NON United States 3119 PA

Netherlands

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Alpha version

Reseller Authorizations:

Compaq

Linux Certifications:

RedHat

Company Phone Number:

0031102469224

Company Fax:

0031102469225

Support Phone Number:

N/A

Contact Email Address:

helpdesk@maxfort.com

Company URL:

http://www.maxfort.com

Primary Contact:

Arjan van Eersel

1.164 Consultant ID: Netherlands, Murphy Softwa BV

P.O. Box 285
Enschede, NON United States NL-7500 AG

Netherlands

Consulting Specialties:

Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

31 53 4320055

Company Fax:

31 53 5360448

Support Phone Number:

N/A

Contact Email Address:

office@murphy.nl

Company URL:

http://www.murphy.nl

Primary Contact:

Marc Roelofs

1.165 Consultant ID: Netherlands, Netland Internet Services

```
Kruislaan 419
Amsterdam, NON United States 1098 VA

Netherlands
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

our own

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+31 20 562 8282

Company Fax:

+31 20 562 8281

Support Phone Number:

+31 20 562 8280

Contact Email Address:

support@netland.nl

Company URL:

http://www.netland.nl

Primary Contact:

Ron Arts

1.166 Consultant ID: Netherlands, Root Systems

```
van Kinsbergenstraat 73
's Gravenhage, ZZ 2518 GW

Netherlands
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(+31) 70 3460010

Company Fax:

(+31) 70 3460118

Support Phone Number:

(+31) 703460126

Contact Email Address:

support@rootsystems.nl

Company URL:

http://www.rootsystems.nl

Primary Contact:

Conrad Langford

1.167 Consultant ID: Netherlands, Stone IT Services BV

```
Jan Steenlaan 3
Bilthoven, NON United States 3723 BS

Netherlands
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Netware Connectivity Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

S.u.S.E.

Reseller Authorizations:

Compaq

Linux Certifications:

Caldera RedHat

Company Phone Number:

+31 30 2287995

Company Fax:

+31 30 2287997

Support Phone Number:

+31 30 2294653

Contact Email Address:

support@stonebv.com

Company URL:

http://www.stonebv.com

Primary Contact:

M.J. Smit

1.168 Consultant ID: New Zealand, Catalyst IT Ltd

```
PO Box 10-225, Level 21, Morrison Kent House, 105
Wellington, NON United States 6000

New Zealand
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+64 (4) 499 2267

Company Fax:

+64 (4) 499 5596

Support Phone Number:

+64 (4) 499 2267

Contact Email Address:

linux@cat-it.co.nz

Company URL:

http://www.cat-it.co.nz

Primary Contact:

Andrew McMillan

1.169 Consultant ID: New Zealand, iOpen Technologies Ltd.

```
P.O. Box 13-917
Christchurch, NON United States 8001

New Zealand
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Modified RedHat

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

64-3-389-1686

Company Fax:

N/A

Support Phone Number:

64-3-389-1686

Contact Email Address:

support@iopen.co.nz

Company URL:

http://www.iopen.co.nz

Primary Contact:

Richard Waid

1.170 Consultant ID: Nigeria, CyberAge Systems

```
6, Rasaq Balogun Str,
Surulere, Lagos, NON United States N/A

Nigeria
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Caldera

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

Others

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

234 01 2602080

Company Fax:

234 01 2602080

Support Phone Number:

N/A

Contact Email Address:

cyberservice@eudoramail.com

Company URL:

http://N/A

Primary Contact:

Edwin Okugbo

1.171 Consultant ID: Norway, AQS AS

```
P.B. 121
Eidsvoll, NON United States 2080

Norway
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

47 63962296

Company Fax:

N/A

Support Phone Number:

N/A

Contact Email Address:

N/A

Company URL:

http://www.aqs.as

Primary Contact:

Martin Langsjøen

1.172 Consultant ID: Norway, Initio IT-losninger

```
Holtermannsv. 31B
Trondheim, NON United States 7031

Norway
```

Consulting Specialties:

Custom Programming Firewalls and Internet Security Internet Server Development Internet (W
Programming Network Administration System Administration System Security Virtual Privat
Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+47 - 73 83 71 71

Company Fax:

+47 - 73 83 71 70

Support Phone Number:

+47 - 73 83 71 71

Contact Email Address:

initio@initio.no

Company URL:

http://www.initio.no

Primary Contact:

Pal Loberg

1.173 Consultant ID: Poland, AM ELEKTRONIK s. c.

```
Biala 7
Gdansk, NON United States 80-435
```

Poland

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Network Administration System Administration System Security

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+48 58 3440061

Company Fax:

+48 58 3440061

Support Phone Number:

Not Available

Contact Email Address:

amelektr@itnet.pl

Company URL:

http://www.itnet.pl/amelektr/

Primary Contact:

Marek Michalkiewicz

1.174 Consultant ID: Poland, Internet Designers s


```
Rynek 58
Wroclaw, NON United States 50-116

Poland
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+48713410099

Company Fax:

+48713410099

Support Phone Number:

+48713460827

Contact Email Address:

support@id.pl

Company URL:

http://www.id.pl

Primary Contact:

Ryszard Lach

1.175 Consultant ID: Poland, RG Studio

Limanowskiego 23
Warszawa, NON United States 02-934

Poland

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Cobalt

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+48 22 651 66 38

Company Fax:

+48 22 651 66 38

Support Phone Number:

+48 22 651 66 38

Contact Email Address:

info@rgstudio.com.pl

Company URL:

http://www.rgstudio.com.pl

Primary Contact:

Jan Rychter

1.176 Consultant ID: Poland, Softomat Ltd.

```
Krynicka 1
Gdansk, NON United States 80-393

Poland
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Internet (Web) Programming Netware Connectivity Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

+48 58 5543 730

Company Fax:

+48 58 556 11 76

Support Phone Number:

+48 58 5543 731

Contact Email Address:

softomat@softomat.com.pl

Company URL:

http://www.softomat.com.pl

Primary Contact:

Jacek Kijewski

1.177 Consultant ID: Portugal, COM2000 Sistemas Informaticos


```
APARTADO 13098
LISBOA, NON United States 1019

Portugal
```


Consulting Specialties:

Internet Server Development

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

HP

Linux Certifications:

Not Applicable

Company Phone Number:

+351 1 8478838

Company Fax:

+351 1 8475438

Support Phone Number:

+351 1 8478838

Contact Email Address:

linux@com2000si.com

Company URL:

http://www.com2000si.com

Primary Contact:

N/A N/A

1.178 Consultant ID: Portugal, S and F CONSULTORES LDA

```
R. Prof. Fernando Fonseca, 16
Lisboa, NON United States 1600-618

Portugal
```

Consulting Specialties:

CIFS (Samba) E-Commerce Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

HP

Linux Certifications:

N/A

Company Phone Number:

351 1 757 17 18

Company Fax:

351 1 757 17 33

Support Phone Number:

351 1 7520017

Contact Email Address:

sf@sfconsultores.pt

Company URL:

http://www.sfconsultores.pt

Primary Contact:

José Ferradeira

1.179 Consultant ID: Puerto Rico, EZ Networks

```
PO Box 3534
Guaynabo, PR 00970

Puerto Rico
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

787-289-7801

Company Fax:

787-289-8779

Support Phone Number:

NA

Contact Email Address:

info@ez-networks.com

Company URL:

http://www.ez-networks.com

Primary Contact:

Jose Torres

1.180 Consultant ID: Singapore, Aeon Xe Technology Pte Ltd

```
38 Edgware Road
Singapore, NON United States 799781

Singapore
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security

Main Distribution:

Suse

Other Distribution:

Redhat, Slackware

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

654846834

Company Fax:

654842687

Support Phone Number:

Not Available

Contact Email Address:

support@aeonxe.com

Company URL:

http://www.aeonxe.com

Primary Contact:

Eddie Tan

1.181 Consultant ID: Singapore, LinuxAsia Pte Ltd

```
No. 2 Alexandra Road, #05-03 Delta House
Singapore, NON United States 159919

Singapore
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Internet Server Development Network Administration System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

65-272 5482

Company Fax:

65-234 2400

Support Phone Number:

65-272 5482

Contact Email Address:

support@linuxasia.com

Company URL:

http://www.linuxasia.com

Primary Contact:

Stephane Benoist

1.182 Consultant ID: Singapore, Webworks Pte Ltd

```
37, Lorong 23 Geylang, #03-04, Yu Li Industrial Bu
Singapore, NON United States 388371

Singapore
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Slackware

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq

Linux Certifications:

Not Applicable

Company Phone Number:

+65 741 9526

Company Fax:

+65 749 3806

Support Phone Number:

+65 741 9526

Contact Email Address:

support@webworks.com.sg

Company URL:

http://www.webworks.com.sg

Primary Contact:

Lon Voon Ong

1.183 Consultant ID: South Africa, Neogenix

```
Pineslopes Office Park; East Block; Witkoppen Road
Johannesburg, NON United States N/A

South Africa
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

TOC

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

IBM

Linux Certifications:

RedHat

Company Phone Number:

+27 (0)11 465-0798

Company Fax:

+27 (0)11 465-9688

Support Phone Number:

+27 (0)11 465-0796

Contact Email Address:

marius@neogenix.co.za

Company URL:

http://www.neogenix.co.za

Primary Contact:

Theunis Botha

1.184 Consultant ID: South Africa, Obsidian Systems

```
Unit 3, Boskruin Business Park, Boskruin
Johannesburg, NON United States 2156

South Africa
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+27-11-792-6500

Company Fax:

+27-11-792-6522

Support Phone Number:

+27-11-792-6500

Contact Email Address:

support@obsidian.co.za

Company URL:

http://www.obsidian.co.za

Primary Contact:

Anton de Wet

1.185 Consultant ID: Spain, ACCESSUS - Proyectos y Consultores en Telecomunica

Moctezuma, 4 - 1
Santander - Cantabria, NON United States 39003

Spain

Consulting Specialties:

CIFS (Samba) E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+34 942 31 80 55

Company Fax:

+34 942 31 80 56

Support Phone Number:

+34 942 31 80 55

Contact Email Address:

seguridad@grupoaccessus.com

Company URL:

http://www.grupoaccessus.com

Primary Contact:

Jose Antonio Revilla

1.186 Consultant ID: Spain, LANGRE INGENIEROS S.L.

Castilla 95
Santander, NON United States 39009

Spain

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Netware Connectivity Network Administration System Administration Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

HP

Linux Certifications:

Not Applicable

Company Phone Number:

34-942-33-56-33

Company Fax:

34-942-35-42-75

Support Phone Number:

34-942-33-56-33

Contact Email Address:

linux@langre.com

Company URL:

http://www.langre.com

Primary Contact:

Arturo Limon

1.187 Consultant ID: Spain, Linux Sistemas

```
C/ Emilio Carrere, 8 5izq
Madrid, ZZ N/A

Spain
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+34914459494

Company Fax:

N/A

Support Phone Number:

+34914459494

Contact Email Address:

vaceituno@hotmail.com

Company URL:

http://www.hotmail.com

Primary Contact:

Vicente Aceituno

1.188 Consultant ID: Sweden, Ewing Data AB

```
Götgatan 39
Stockholm, NON United States 11621

Sweden
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

RedHat, Debian, Cald

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+46 (0)8 643-1106

Company Fax:

+46 (0)8 643-1106

Support Phone Number:

+46 (0)8 643-1106

Contact Email Address:

admin@ewingdata.com

Company URL:

http://www.ewingdata.com

Primary Contact:

James Ewing

1.189 Consultant ID: Sweden, Omicron Software Systems AB

```
Borgarfjordsgatan 13
Kista, NON United States S-164 40

Sweden
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

ALL

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

46 8 7510180

Company Fax:

46 8 7039413

Support Phone Number:

46 8 7512940

Contact Email Address:

support@omicron.se

Company URL:

http://www.omicron.se

Primary Contact:

Johan Nordin

1.190 Consultant ID: Sweden, Tarkus

```
Flitiga Lisas v. 8
Västerås, NON United States 72246

Sweden
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

N/A

Company Fax:

+33 4 50602749

Support Phone Number:

+46 70 75 22246

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Carlos Vidal

1.191 Consultant ID: Sweden, Xantus AB

```
Bredgatan 10
Lund, NON United States S-222 21

Sweden
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+46 46 399200

Company Fax:

+46 46 399201

Support Phone Number:

+46 46 399200

Contact Email Address:

info@xantus.se

Company URL:

http://www.xantus.se

Primary Contact:

Henrik Carling

1.192 Consultant ID: Switzerland, 3T

```
Gassackerweg 47
Frenkendorf, NON United States CH-4402

Switzerland
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Suse

Other Distribution:

RedHat, Stampede

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+41 61 303 91 91

Company Fax:

+41 61 303 91 92

Support Phone Number:

+41 61 303 91 91

Contact Email Address:

info@t-t-t.com

Company URL:

http://www.t-t-t.com

Primary Contact:

Stefan Schaefer

1.193 Consultant ID: Switzerland, Born Consulting

```
Brunnadernstrasse 3
Berne, NON United States 3006

Switzerland
```

Consulting Specialties:

CIFS (Samba)

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

MkLinux and SPARC Li

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+41 (0)79 338 95 79

Company Fax:

-

Support Phone Number:

+41 (0)79 338 95 79

Contact Email Address:

flborn@spectraweb.ch

Company URL:

http://www.spectraweb.ch

Primary Contact:

Fritz Lorenz Born

1.194 Consultant ID: Switzerland, CONNEXT Communications

```
Lebrigstrasse 1
Goetighofen, NON United States 8583

Switzerland
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+41-71 640 0410

Company Fax:

+41-71 640 0411

Support Phone Number:

+41-71 640 0410

Contact Email Address:

support@connext.ch

Company URL:

http://www.connext.ch

Primary Contact:

Mauro Canal

1.195 Consultant ID: Switzerland, ETV Software Engineering S.A.

```
Ch. du Pavillon 2, PO Box 242
Grand Saconnex, NON United States 1218

Switzerland
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+41 22 717 64 00

Company Fax:

+41 22 717 64 01

Support Phone Number:

+41 22 717 64 18

Contact Email Address:

linux-support@software-engineering.ch

Company URL:

http://www.software-engineering.ch

Primary Contact:

Joel Margot

1.196 Consultant ID: Switzerland, Erik Rossen

```
Rue Centrale 26
Chavannes-Renens, NON United States CH-1022
```

TOC

Switzerland

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Internet (Web) Programming System Administration

Main Distribution:

Other

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

+41 21 636 42 80

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Erik Rossen

1.197 Consultant ID: Switzerland, Penta Group AG


```
Badenerstrasse 701
ZURICH, NON United States CH-8048

Switzerland
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Network Administration System Administration

Main Distribution:

Other

Other Distribution:

ALL (whatever Custom

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

Not Applicable

Company Phone Number:

+41 1 438 50 50

Company Fax:

+41 1 438 50 60

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Benedikt Hochstrasser

1.198 Consultant ID: Switzerland, SciOS Scientific Operating Systems GmbH

```
N/A
Islikon, NON United States 8546

Switzerland
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

++41-52-366-52-15

Company Fax:

++41-52-366-52-14

Support Phone Number:

++41-52-366-52-15

Contact Email Address:

support@scios.ch

Company URL:

http://www.scios.ch

Primary Contact:

N/A N/A

1.199 Consultant ID: Switzerland, Technology Solutions

```
N/A
Pfaffhausen, NON United States 8118

Switzerland
```

Consulting Specialties:

CIFS (Samba) E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Debian, Suse

Specialty Distribution:

Other

Other Specialty Distribution:

Linux/AXP

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+41 1 825 28 36

Company Fax:

+41 1 825 28 36

Support Phone Number:

Not Available

Contact Email Address:

linux-support@feigin.com

Company URL:

http://N/A

Primary Contact:

N/A N/A

1.200 Consultant ID: Taiwan, WHINSTONE INFO. CO., LTD.

```
N/A
TAIPEI, NON United States 105

Taiwan
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

886-2-27492626

Company Fax:

886-2-27468790

Support Phone Number:

886-2-27492626

Contact Email Address:

info@whinstone.com

Company URL:

http://www.whinstone.com

Primary Contact:

Anthony Hsia

1.201 Consultant ID: Turkey, Gelecek A.S

```
Perpa Ticaret Merkezi Kat:11 B Blok No:1932
Istanbul, NON United States N/A

Turkey
```

Consulting Specialties:

CIFS (Samba) E-Commerce Firewalls and Internet Security Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

EasyLinux

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+90 212 2107194

Company Fax:

+90 212 2100442

Support Phone Number:

N/A

Contact Email Address:

destek@gelecek.com.tr

Company URL:

http://www.gelecek.com.tr

Primary Contact:

Gorkem Cetin

1.202 Consultant ID: Turkey, YORE Elektronik Yayimcilik A.S.

```
Halaskargazi C 301K 5  Sisli
Istanbul, NON United States 80260

Turkey
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+90 212 234 0090

Company Fax:

+90 212 230 7142

Support Phone Number:

+90 212 234 0090

Contact Email Address:

destek@yore.com.tr

Company URL:

http://http://www.yore.net.tr/

Primary Contact:

Tunc Ozbilgin

1.203 Consultant ID: US, AL, The Net Effect, LLC

```
P O Box 885
Mobile, AL 36601

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

USA (334) 433 0196

Company Fax:

USA (334) 433 5371

Support Phone Number:

USA (334) 433 0196

Contact Email Address:

grs@theneteffect.com

Company URL:

http://www.theneteffect.com

Primary Contact:

Glenda Snodgrass

1.204 Consultant ID: US, AZ, Brindlesoft

```
PO Box 237
St. David, AZ 85630-0237

US
```

Consulting Specialties:

Custom Programming System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(520) 720-9431

Company Fax:

N/A

Support Phone Number:

(520) 720-9431

Contact Email Address:

thegrendel@theriver.com

Company URL:

http://www.theriver.com

Primary Contact:

Mendel Cooper

1.205 Consultant ID: US, AZ, Jay Ts

```
P.O. Box 410
Sedona, AZ 86339

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System

Administration System Security

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

N/A

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(888) 282-0549

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://http://jayts.cx

Primary Contact:

Jay Ts

1.206 Consultant ID: US, AZ, MagusNet, Inc.

```
1201 East Washington Ave.
Gilbert, AZ 85234

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

602-892-8186

Company Fax:

NA

Support Phone Number:

NA

Contact Email Address:

support@magusnet.com

Company URL:

http://www.magusnet.com

Primary Contact:

Jean Francois

1.207 Consultant ID: US, AZ, Tapestry Software

```
7990 E. Maguey Drive
Tucson, AZ 85750
```

TOC

US

Consulting Specialties:

Custom Programming Internet (Web) Programming System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

01-520-749-2728

Company Fax:

01-520-749-2885

Support Phone Number:

N/A

Contact Email Address:

sbw@tapestry.tucson.az.us

Company URL:

http://N/A

Primary Contact:

Steve Wampler

1.208 Consultant ID: US, CA, ASA Computers, Inc.

2354 Calle Del Mundo
Santa Clara, CA 95054

US

Consulting Specialties:

Custom Systems System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP

Linux Certifications:

RedHat

Company Phone Number:

+1-408-654-2901 x 20

Company Fax:

408-654-2910

Support Phone Number:

+1-408-654-2901 x 22

Contact Email Address:

tech@asacomputers.com

Company URL:

http://www.asacomputers.com

Primary Contact:

Kedar Rajadnya

1.209 Consultant ID: US, CA, Aedra Associates, In

TOC

N/A
San Jose, CA 95117

US

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

408-246-0875

Company Fax:

N/A

Support Phone Number:

408-246-0875

Contact Email Address:

support@aedra.com

Company URL:

http://www.aedra.com

Primary Contact:

Steve Otto

1.210 Consultant ID: US, CA, David Fetter Consulting

```
888 O'Farrell Street, Suite E1205
San Francisco, CA 94109

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Internet Server Development Internet (Web) Programming System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+1 415 567 2690

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

dfetter@best.com

Company URL:

http://www.best.com

Primary Contact:

David Fetter

1.211 Consultant ID: US, CA, Gregg Weber

```
N/A
El Sobrante, CA 94803

US
```

Consulting Specialties:

Internet Server Development Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

510 223-3310

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

N/A N/A

1.212 Consultant ID: US, CA, Herlein Engineering

```
2034 Filbert Street
San Francisco, CA 94123

US
```

Consulting Specialties:

Custom Programming Custom Systems Internet Server Development Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

N/A

Company Phone Number:

415-519-3650

Company Fax:

415-440-9015

Support Phone Number:

415-519-3650

Contact Email Address:

support@herlein.com

Company URL:

http://www.herlein.com

Primary Contact:

Greg Herlein

1.213 Consultant ID: US, CA, Hiverworld, LLC

```
1918 University Av e
Berkeley, CA 94704

US
```

Consulting Specialties:

System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

510.848.0740

Company Fax:

n/a

Support Phone Number:

510.848.0740

Contact Email Address:

support@hiverworld.com

Company URL:

http://www.hiverworld.com

Primary Contact:

Gretchen Flowers

1.214 Consultant ID: US, CA, KrystalBox Technologies, Inc.

```
555 Bryant Street #359
Palo Alto, CA 94301

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

650-327-8929

Company Fax:

650-327-8949

Support Phone Number:

N/A

Contact Email Address:

support@krystalbox.com

Company URL:

http://www.krystalbox.com

Primary Contact:

Travis Godbout

1.215 Consultant ID: US, CA, Linuxcare

```
650 Townsend Street
San Francisco, CA 94103

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

We support ALL distr

Specialty Distribution:

Other

Other Specialty Distribution:

We support ALL distr

Reseller Authorizations:

N/A

Linux Certifications:

Caldera RedHat

Company Phone Number:

888-546-4878

Company Fax:

415-701-7457

Support Phone Number:

888-546-4878

Contact Email Address:

support@linuxcare.com

Company URL:

http://www.linuxcare.com

Primary Contact:

David LaDuke

1.216 Consultant ID: US, CA, MK Information System Services

```
N/A
Roseville, CA N/A

US
```

Consulting Specialties:

Custom Programming Custom Systems Internet Server Development Internet (Web) Programming Network Administration System Administration

Main Distribution:

Suse

Other Distribution:

Caldera, RedHat

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(916)765-2253

Company Fax:

N/A

Support Phone Number:

N/A

Contact Email Address:

blaw@ns.net

Company URL:

http://N/A

Primary Contact:

Bob Law

1.217 Consultant ID: US, CA, Mountain Ranch Design

```
P.O. Box 679
Mountain Ranch, CA 95246

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

209-754-3108

Company Fax:

209-754-0273

Support Phone Number:

1-888-551-3003

Contact Email Address:

support@mtnranch.com

Company URL:

http://www.mtnranch.com

Primary Contact:

Mark Hansen

1.218 Consultant ID: US, CA, Network Presence, LLC

```
548 S Spring St, Ste 1015
Los Angeles, CA 90013

US
```

TOC

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

213-627-4356

Company Fax:

213-627-4357

Support Phone Number:

213-627-4356

Contact Email Address:

support@netpr.com

Company URL:

http://www.netpr.com

Primary Contact:

Robert Brown

1.219 Consultant ID: US, CA, Newyen Technology

3535 El Portal Dr C206
El Sobrante, CA 94803

US

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP Not Applicable

Linux Certifications:

N/A

Company Phone Number:

1-510-758-5950

Company Fax:

N/A

Support Phone Number:

1-510-758-5950

Contact Email Address:

sales@newyen.com

Company URL:

http://www.newyen.com

Primary Contact:

Kent Nguyen

1.220 Consultant ID: US, CA, Oration

TOC

```
7 West 41st Avenue, Suite 78
San Mateo, CA 94403

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

877-ORATION

Company Fax:

650-572-8315

Support Phone Number:

877-ORATION

Contact Email Address:

support@oration.com

Company URL:

http://www.oration.com

Primary Contact:

Pramod John

1.221 Consultant ID: US, CA, Ozlin Network Designs

```
600-A Anton Blvd. #302
Costa Mesa, CA 92626

US
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Netware Connectivity Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

714-546-4250

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

info@ozlin.com

Company URL:

http://www.ozlin.com

Primary Contact:

Mark Linn

1.222 Consultant ID: US, CA, PSO Systems Inc

```
120 Linden Street
Oakland, CA 94607

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

510 663 1156

Company Fax:

510 663 1157

Support Phone Number:

510 663 1156

Contact Email Address:

support@pso.com

Company URL:

http://www.pso.com

Primary Contact:

Hwajin Bae

1.223 Consultant ID: US, CA, Predictive Science

```
2008 Conquista Ave
Long Beach, CA 90815

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration

Main Distribution:

Other

Other Distribution:

RedHat and Slackware

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1.562.592.6884

Company Fax:

1.562.592.6884

Support Phone Number:

1.562.592.6884

Contact Email Address:

lganders@gte.net

Company URL:

http://www.gte.net

Primary Contact:

Leonard Anderson

1.224 Consultant ID: US, CA, Starshine Technical Services

```
903 Harriet Avenue
Campbell, CA 95008

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems Internet Server Development Internet (Web) Programming System Administration System Security

Main Distribution:

Other

Other Distribution:

PC-based but includi

Specialty Distribution:

Other

Other Specialty Distribution:

will aid requirement

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

800-938-4078

Company Fax:

408-374-LUGS

Support Phone Number:

(leave voice mail at

Contact Email Address:

consulting@starshine.org

Company URL:

http://www.starshine.org

Primary Contact:

Heather Stern N/A

1.225 Consultant ID: US, CA, Tesno Systems

```
3435 Ocean Park Bl Ste 112
Santa Monica, CA 90405

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

HP

Linux Certifications:

Not Applicable

Company Phone Number:

310 2047875

Company Fax:

310 5589221

Support Phone Number:

310 2047875

Contact Email Address:

info@tesnosystems.com

Company URL:

http://www.tesnosystems.com

Primary Contact:

Jerry Leahy

1.226 Consultant ID: US, CA, U.S. Technical Services

```
9121 Atlanta Ave. Suite 230
Huntington Beach, CA 92646

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(714) 374-6300

Company Fax:

(714) 242-1346

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

brian harvey

1.227 Consultant ID: US, CA, US CustomLogic, LLC

```
11292 Coloma Road, Suite C
Gold River, CA 95670

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Intel

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-916-853-4920

Company Fax:

1-916-853-4924

Support Phone Number:

1-916-853-4920

Contact Email Address:

info@customlogic.com

Company URL:

http://www.customlogic.com

Primary Contact:

Tom Henderson

1.228 Consultant ID: US, CA, Vennerable Consulting

```
1563 Solano Avenue, Suite 516
berkeley, CA 94707

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

510 528 8072

Company Fax:

510 528 8072

Support Phone Number:

510 528 8072

Contact Email Address:

support@vennerable.com

Company URL:

http://www.vennerable.com

Primary Contact:

jason venner

1.229 Consultant ID: US, CA, interactivate consulting group

```
2244b carmel valley road
del mar, CA 92014

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

red hat, slackware,

Specialty Distribution:

Other

Other Specialty Distribution:

custom

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

please use email

Company Fax:

619 793 4069

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

sean dreilinger

1.230 Consultant ID: US, CO, BDA Consulting LLC

```
758 English Sparrow Trail Suite 100
Highlands Ranch, CO 80126
```

TOC

US

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

firewalls

Reseller Authorizations:

Not Applicable

Linux Certifications:

N/A

Company Phone Number:

303-683-9252

Company Fax:

303-683-7439

Support Phone Number:

303-683-9252

Contact Email Address:

support@bdaconsulting.com

Company URL:

http://www.bdaconsulting.com

Primary Contact:

James Davis

1.231 Consultant ID: US, CO, Eklektix, Inc.

TOC

5624 Rim Rock Ct
Boulder, CO 80301-3553

US

Consulting Specialties:

Custom Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

All others

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

303 581 0750

Company Fax:

303 581 0385

Support Phone Number:

303 581 0750

Contact Email Address:

info@eklektix.com

Company URL:

http://www.eklektix.com

Primary Contact:

Elizabeth Coolbaugh

1.232 Consultant ID: US, CO, MW Software

TOC

P.O. Box 481762
Denver, CO 80248-1762

US

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

303-832-5595

Company Fax:

303-832-3579

Support Phone Number:

303-832-5595

Contact Email Address:

mwinter@mwsoftware.com

Company URL:

http://mwsoftware.com

Primary Contact:

Mark Winter

1.233 Consultant ID: US, CO, Paramin Consulting Group

```
6424 South Quebec Street
Englewood, CO 80111

US
```

Consulting Specialties:

E-Commerce

Main Distribution:

Other

Other Distribution:

Slakware

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

IBM

Linux Certifications:

N/A

Company Phone Number:

888 337 8227

Company Fax:

303 721 7522

Support Phone Number:

303 773 1361

Contact Email Address:

support@paramin.com

Company URL:

http://www.paramin.com

Primary Contact:

Amy Strauser

1.234 Consultant ID: US, CO, tummy.com, ltd.

```
5400 Fossil Court North
Fort Collins , CO 80525

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

Own Dist.

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

970 223-8215

Company Fax:

408 490-2728

Support Phone Number:

Not Available

Contact Email Address:

support@tummy.com

Company URL:

http://www.tummy.com

Primary Contact:

Evelyn Mitchell

1.235 Consultant ID: US, CT, DCA Online (Dutch Computer Association)

```
26 Tobacco Road
Weston, CT 06883-1612

US
```

Consulting Specialties:

CIFS (Samba) E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+1 203 222 8267

Company Fax:

+1 203 226 7037

Support Phone Number:

+1 203 222 8267

Contact Email Address:

support@dcaonline.com

Company URL:

http://www.dcaonline.com

Primary Contact:

Peter Leonard Krebs

1.236 Consultant ID: US, CT, DownCity, LLC

```
131 Broadway
Norwich, CT 06360

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-860-823-3000

Company Fax:

1-860-859-9401

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Robert Szarka

1.237 Consultant ID: US, CT, GNUTEC

```
1655 North Colony Road, Suite 3002
Meriden, CT 06450

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

All Distributions

Specialty Distribution:

Other

Other Specialty Distribution:

All Distributions

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-203-379-0125

Company Fax:

1-203-379-0126

Support Phone Number:

1-203-668-UNIX

Contact Email Address:

support@gnutec.com

Company URL:

http://http://www.gnutec.com/

Primary Contact:

Kyle Amon

1.238 Consultant ID: US, CT, MANHATTAN GROUP

```
925 WHITE PLAINS ROAD
TRUMBULL, CT 06611

US
```

Consulting Specialties:

Custom Programming Custom Systems Internet Server Development Internet (Web) Programming System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

203-445-0011

Company Fax:

203-445-0009

Support Phone Number:

203-445-0011

Contact Email Address:

SUPPORT@MANHATTANGROUP,COM

Company URL:

http://WWW.MANHATTANGROUP.COM

Primary Contact:

JOHN LOMBARDI

1.239 Consultant ID: US, CT, Tempest Harding, Inc.

```
322 Main Street, Suite 1204
Willimantic, CT 06226

US
```

Consulting Specialties:

Custom Systems

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

860-456-3639

Company Fax:

860-456-2814

Support Phone Number:

860-456-3639

Contact Email Address:

info@thinc.net

Company URL:

http://www.thinc.net

Primary Contact:

Paul Herbstritt

1.240 Consultant ID: US, DC, BKP, Inc.

```
P.O. Box 21026
Washington, DC 20009

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems Internet Server Development Netware Connectivity Network Administration System Administration

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(202) 332-5609

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

John Thomas N/A

1.241 Consultant ID: US, DE, W. E. Hughes, INC

```
1527 Vandyke Greenspring Road
Smyrna, DE 19977

US
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

N/A

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

wally@dca.net

Company URL:

http://members.dca.net/wally

Primary Contact:

Wally Hughes

1.242 Consultant ID: US, FL, Anthony Awtrey Consulting

```
781 Marsaille Dr Suite D
Indialantic, FL 32903

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming E-Commerce Firewalls and Internet Security Internet Server

Development Internet (Web) Programming Network Administration System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Slackware

Specialty Distribution:

Other

Other Specialty Distribution:

None

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

407-777-7164

Company Fax:

407-779-7269

Support Phone Number:

407-777-7164

Contact Email Address:

support@awtrey.com

Company URL:

http://www.awtrey.com

Primary Contact:

Anthony Awtrey

1.243 Consultant ID: US, FL, Mackintosh Web, Inc.

```
3318 Clark Rd
Sarasota, FL 34231
```

US

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-941-316-5691

Company Fax:

1-941-921-5148

Support Phone Number:

1-941-921-5148

Contact Email Address:

terry@terrym.com

Company URL:

http://www.terrym.com

Primary Contact:

Terry Mackintosh

1.244 Consultant ID: US, FL, Richard J Kinch, PhD

TOC

6994 Pebble Beach Ct
Lake Worth, FL 33467

US

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+1 (561) 966-8400

Company Fax:

+1 (561) 966-0962

Support Phone Number:

+1 (561) 966-8400

Contact Email Address:

kinch@truetex.com

Company URL:

http://www.truetex.com

Primary Contact:

Richard Kinch

1.245 Consultant ID: US, FL, Spectrum Enterprises

```
1609 Puritan Ave
Winter Park, FL 32792

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

1-407-257-1437

Company Fax:

1-407-257-1437

Support Phone Number:

N/A

Contact Email Address:

support@spectrumenterprises.com

Company URL:

http://www.spectrumenterprises.com

Primary Contact:

Paul Marble

1.246 Consultant ID: US, FL, TSM Services

```
108 Sunset Blvd
Boynton Beach, FL 33426

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

N/A

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

561-374-8874

Company Fax:

561-375-7877

Support Phone Number:

Not Available

Contact Email Address:

support@tsmservices.com

Company URL:

http://www.tsmservices.com

Primary Contact:

Steve Grevemeyer

1.247 Consultant ID: US, FL, Yup Dot Com

```
531 NE 2nd Street
Gainesville, FL 32601

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

352/379-8921

Company Fax:

N/A

Support Phone Number:

352/379-8921

Contact Email Address:

info@yup.com

Company URL:

http://yup.com

Primary Contact:

Daniel Butler

1.248 Consultant ID: US, FL, eBuilders Network

```
8204 Crystal Clear Lane Suite 1200
Orlando , FL 32809

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Internet Server Development Internet (Web) Programming

Main Distribution:

RedHat

Other Distribution:

Mandrake

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

407-857-6800

Company Fax:

407-857-2333

Support Phone Number:

407-857-6800

Contact Email Address:

sales@ebuildernet.com

Company URL:

http://www.ebuildernet.com

Primary Contact:

Pablo Bryan

1.249 Consultant ID: US, GA, Fly-By-Day Consulting, Inc.

```
4642 Bentley Place
Duluth, GA 30096

US
```

Consulting Specialties:

Custom Programming Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1 770 662 8321

Company Fax:

N/A

Support Phone Number:

1 770 662 8321

Contact Email Address:

bob@cavu.com

Company URL:

http://www.cavu.com

Primary Contact:

Bob Toxen

1.250 Consultant ID: US, GA, John Meister

```
2350 Cobb Parkway, #34J
Smyrna, GA 30080

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(770) 980-0665

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

meisterj@acm.org

Company URL:

http://www.acm.org

Primary Contact:

John Meister

1.251 Consultant ID: US, HI, Cytronix Computer Services, Inc.

```
350 Ward Ave. 106-288
Honolulu, HI 96814

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Internet Server Development Internet (Web) Programming Netware Connectivity Virtual Private Networking

Main Distribution:

Other

Other Distribution:

Custom In-House Dist

Specialty Distribution:

Other

Other Specialty Distribution:

Customer In-House Di

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(808) 383-0558

Company Fax:

N/A

Support Phone Number:

(800) 939-9247

Contact Email Address:

deven@cytronix.net

Company URL:

http://www.cytronix.net

Primary Contact:

Deven Phillips

1.252 Consultant ID: US, IA, JEONET

```
PO Box 1282
Iowa City, IA 52241

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

DNS and Firewall con

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

319-338-6353

Company Fax:

319-338-6353

Support Phone Number:

800 number for clien

Contact Email Address:

linuxsupport@jeo.net

Company URL:

http://www.jeo.net

Primary Contact:

Afan Ottenheimer

1.253 Consultant ID: US, IL, Cobler Corporation

```
3013 S. Wolf Rd.
Westchester, IL 60154

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

708-205-3313

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Kenneth Cobler

1.254 Consultant ID: US, IL, Dataline

```
105 South Wright St. Suite #309
Champaign, IL 61820

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet

Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

2173512647

Company Fax:

2173512647

Support Phone Number:

2173512647

Contact Email Address:

support@jay.cx

Company URL:

http://N/A

Primary Contact:

Jay Thomas

1.255 Consultant ID: US, IL, Ossmann Consulting

```
651 W. Belmont Ave.
Chicago, IL 60657
```

US

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

N/A

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

773-294-0250

Company Fax:

N/A

Support Phone Number:

773-294-0250

Contact Email Address:

mike@ossmann.com

Company URL:

http://www.ossmann.com

Primary Contact:

Michael Ossmann

1.256 Consultant ID: US, IL, Promethan Consultin

18 Skyview Drive
Springfield, IL 62702

US

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

Red Hat, Debian

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

217/698-6234

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Jeff Licquia

1.257 Consultant ID: US, IL, Remington*Microshare

```
1834 Walden Office Square, Suite 100
Schaumburg, IL 60173

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet (Web) Programming Netware Connectivity Network Administration System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

Not Applicable

Company Phone Number:

847-221-0200

Company Fax:

847-221-0399

Support Phone Number:

847-221-0200

Contact Email Address:

chrisd@remingtonltd.com

Company URL:

http://www.remingtonmicroshare.com

Primary Contact:

Jeremy Simon

1.258 Consultant ID: US, IL, WellThot Inc.

```
305 Van Damin Ave.
Glen Ellyn, IL 60137-5215

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

630-545-1470

Company Fax:

630-545-1471

Support Phone Number:

630-545-1470

Contact Email Address:

support@wellthot.com

Company URL:

http://wellthot.com

Primary Contact:

Mark Schwenk

1.259 Consultant ID: US, IL, onShore, Inc

```
1407 W Chicago Av
Chicago, IL 60622

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

Pacific HiTech HA

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

RedHat

Company Phone Number:

312 850-5200

Company Fax:

312 850-5208

Support Phone Number:

312 850-5200

Contact Email Address:

support-managers@onshore.com

Company URL:

http://www.onshore.com

Primary Contact:

Adam Di Carlo

1.260 Consultant ID: US, IN, CLFINC

```
P.O.Box 1271
Columbus, IN 47202-1271

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

812-372-1911

Company Fax:

N/A

Support Phone Number:

812-372-1911

Contact Email Address:

cfiene@iquest.net

Company URL:

http://www.iquest.net

Primary Contact:

Curtis Fiene

1.261 Consultant ID: US, IN, IndyBox Systems, Inc.

```
PO Box 3564
Carmel, IN 46082-3564

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Caldera RedHat

Company Phone Number:

01 317 8469762

Company Fax:

N/A

Support Phone Number:

+01 317 8469762

Contact Email Address:

support@indybox.com

Company URL:

http://www.indybox.com

Primary Contact:

Charles Martin

1.262 Consultant ID: US, IN, ValCom

```
8098 Woodland Drive
Indianapolis, IN 46278

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

Not Applicable

Company Phone Number:

(317)870-8020

Company Fax:

(317)870-8035

Support Phone Number:

(317)870-8020

Contact Email Address:

mreba@pccval.com

Company URL:

http://www.pccval.com

Primary Contact:

Matthew Reba

1.263 Consultant ID: US, LA, Advanced Network Technologies

```
10795 Mead Rd. #104
Baton Rouge, LA 70816

US
```

Consulting Specialties:

Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration

Main Distribution:

Other

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

225-296-5815

Company Fax:

N/A

Support Phone Number:

225-296-5815

Contact Email Address:

support@advanced-net-tech.com

Company URL:

http://www.advanced-net-tech.com

Primary Contact:

Benjamin Curtis

1.264 Consultant ID: US, LA, Connectivity Solutions, Inc.

```
365 Canal Street, Suite 2300
New Orleans, LA 70130

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

504-561-6540

Company Fax:

504-561-6694

Support Phone Number:

504-561-6540

Contact Email Address:

support@conn-sol.com

Company URL:

http://www.conn-sol.com

Primary Contact:

Kevin Wilson

1.265 Consultant ID: US, MA, CYWAYS, Incorporated

```
19 Westchester Road
Newton, MA 02468

US
```

TOC

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

+1 800 5-CYWAYS

Company Fax:

+1 617 796-8997

Support Phone Number:

+1 617 796-8995

Contact Email Address:

support@cyways.com

Company URL:

http://www.cyways.com

Primary Contact:

Peter Lemieux

1.266 Consultant ID: US, MA, ComputeAmerica.com

120C Broadmeadow Rd.
Marlborough, MA 01752

US

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Intel

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

Not Applicable

Company Phone Number:

877-672-2171

Company Fax:

508-624-0120

Support Phone Number:

508-460-8897

Contact Email Address:

support@computeamerica.com

Company URL:

http://www.computeamerica.com/WebDesign

Primary Contact:

Sonny parlin Steve Doherty

1.267 Consultant ID: US, MA, DataProfit

330 Whitney Ave
Holyoke, MA 01040

US

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

Not Applicable

Company Phone Number:

800-437-7634

Company Fax:

413-536-3354

Support Phone Number:

NA

Contact Email Address:

rcolumbia@dataprofit.com

Company URL:

http://www.dataprofit.com

Primary Contact:

Richard Columbia

1.268 Consultant ID: US, MA, LSC

```
205 West Granville RD
Tolland, MA 01034

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

413-258-4074

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

c5141@iname.com

Company URL:

http://www.iname.com

Primary Contact:

Richard Columbia

1.269 Consultant ID: US, MA, NetInterface Consulting

```
56 Houghton
Hudson, MA 01749

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1.978.567.0972

Company Fax:

1.978.567.0972

Support Phone Number:

1.978.567.0972

Contact Email Address:

support@netinterface.com

Company URL:

http://www.netinterface.com

Primary Contact:

Lenny Forziati

1.270 Consultant ID: US, MA, New England Linux Resource

```
N/A
Newton, MA 02468

US
```

Consulting Specialties:

E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

Caldera

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

617 332 0387

Company Fax:

N/A

Support Phone Number:

617 332 0837

Contact Email Address:

nelinuxresources@hotmail.com

Company URL:

http://www.nelinuxresourses.com

Primary Contact:

curt N/A

1.271 Consultant ID: US, MA, New England Linux resources

```
newton
Newton, MA 02468

US
```

Consulting Specialties:

System Administration

Main Distribution:

RedHat

Other Distribution:

caldera

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

617 332 0837

Company Fax:

none

Support Phone Number:

617 332 0837

Contact Email Address:

nelinuxresources@hotmail.com

Company URL:

http://www.nelinuxresources.com

Primary Contact:

curt ****

1.272 Consultant ID: US, MA, Worldmachine Technologies Corporation

```
44 Winter Street, 3rd Floor
Boston, MA 02108-4745

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

all

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(617) 357-4040

Company Fax:

(617) 357-4949

Support Phone Number:

(617) 357-4040

Contact Email Address:

support@worldmachine.com

Company URL:

http://www.worldmachine.com

Primary Contact:

Eric Hansen

1.273 Consultant ID: US, MD, JAMUX

```
4520 Yates Road
Beltsville, MD 20705-2629

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

1 301 595 2013

Company Fax:

N/A

Support Phone Number:

1 301 595 2013

Contact Email Address:

jam@jamux.com

Company URL:

http://www.jamux.com

Primary Contact:

John Martin

1.274 Consultant ID: US, MD, RABA Technologies, Inc.

```
10500 Little Patuxent Pkwy, Ste. 190
Columbia, MD 21044

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Caldera RedHat

Company Phone Number:

(410) 715-9399

Company Fax:

(410) 715-9397

Support Phone Number:

(410) 715-9399

Contact Email Address:

stuart.chandler@raba.com

Company URL:

http://www.raba.com

Primary Contact:

Stuart Chandler

1.275 Consultant ID: US, ME, TransWeb ToolBox

```
PO Box 788
Windham, ME 04062

US
```

Consulting Specialties:

N/A

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Caldera

Reseller Authorizations:

N/A

Linux Certifications:

RedHat

Company Phone Number:

207-450-7914

Company Fax:

207-892-8745

Support Phone Number:

207-450-7914

Contact Email Address:

support@transwebtools.com

Company URL:

http://www.transwebtools.com

Primary Contact:

Henri Brady

1.276 Consultant ID: US, MI, Jim Willette

```
18190 - 15 mile Rd.
Big Rapids, MI 49307

US
```

Consulting Specialties:

CIFS (Samba)

Main Distribution:

Other

Other Distribution:

Any

Specialty Distribution:

Other

Other Specialty Distribution:

Any

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(616) 592 - 0405

Company Fax:

N/A

Support Phone Number:

(616) 592-0405

Contact Email Address:

jimw@ctec.net

Company URL:

http://www.ctec.net

Primary Contact:

Jim Willette

1.277 Consultant ID: US, MI, Nugent Telecommunications

```
3081 Braeburn Circle
Ann Arbor, MI 48108
```

US

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

N/A

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

734-971-1076

Company Fax:

734-971-4529

Support Phone Number:

734-971-1076

Contact Email Address:

jjn@nuge.com

Company URL:

http://www.nuge.com

Primary Contact:

Jay Nugent

1.278 Consultant ID: US, MI, Radiomind Studios

TOC

213 Schuring Rd
Portage, MI 49024-5012

US

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

616-323-9417

Company Fax:

N/A

Support Phone Number:

N/A

Contact Email Address:

support@radiomind.com

Company URL:

http://www.radiomind.com

Primary Contact:

John Beimler

1.279 Consultant ID: US, MN, Anvi Corp

```
2610 Grand St. NE
Minneapolis, MN 55418

US
```

Consulting Specialties:

Custom Programming Custom Systems Internet Server Development Internet (Web) Programming System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Slackware

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(612) 782-1955

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://menon.com/

Primary Contact:

Kps Menon

1.280 Consultant ID: US, MO, Firefly Enterprises, Inc.

```
922 Rochdale Dr Suite 100
Kirkwood, MO 63122

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Internet (Web) Programming Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

314-835-1590

Company Fax:

314-835-1591

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Ross Sponholtz

1.281 Consultant ID: US, MO, Firefly Enterprises, Inc.

```
922 Rochdale Dr Suite 100
Kirkwood, MO 63122

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Internet (Web) Programming Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

314-835-1590

Company Fax:

314-835-1591

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Ross Sponholtz

1.282 Consultant ID: US, MO, PhazeOne, Inc.

```
3101 Broadway, Suite 605
Kansas City, MO 64111

US
```

Consulting Specialties:

Custom Programming E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(816)756-1300

Company Fax:

(816)756-1355

Support Phone Number:

(816)756-1300

Contact Email Address:

support@phazeone.com

Company URL:

http://www.phazeone.com

Primary Contact:

Steven Heutinck

1.283 Consultant ID: US, NC, ApeX Systems Integration Corp.

```
PO Box 338
Apex, NC 27502

US
```

Consulting Specialties:

Custom Systems Firewalls and Internet Security Network Administration System Administration

Main Distribution:

Suse

Other Distribution:

Red Hat, and others

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

919-468-8150

Company Fax:

919-468-5288

Support Phone Number:

NA

Contact Email Address:

apex@2boot.com

Company URL:

http://www.2boot.com

Primary Contact:

Joe Ferguson

1.284 Consultant ID: US, NC, Moongroup Consulting, Inc.

```
134 Grande Meadow Way
Cary, NC 27513

US
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

01-919-462-9426

Company Fax:

01-919-462-9427

Support Phone Number:

01-919-462-9426

Contact Email Address:

support@moongroup.com

Company URL:

http://www.moongroup.com

Primary Contact:

Chuck Mead

1.285 Consultant ID: US, NC, Network Computing Solutions

```
3733 National Dr.
Raleigh, NC 27612

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-919-510-6970

Company Fax:

1-919-510-6971

Support Phone Number:

1-919-510-6970

Contact Email Address:

linux-support@networks.com

Company URL:

http://www.networks.com

Primary Contact:

Pat Hughes

1.286 Consultant ID: US, NC, UTSL dot com

```
1007 North Buchanan Blvd.
Durham, NC 27701-1139

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Internet Server Development Internet (Web) Programming

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Debian

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

919-286-1441

Company Fax:

N/A

Support Phone Number:

919-286-1441

Contact Email Address:

biggers@utsl.com

Company URL:

http://www.utsl.com

Primary Contact:

Mark Biggers

1.287 Consultant ID: US, NC, Whole Systems Solutions, Inc.

```
2 Terrace Way, Suite A
Greensboro, NC 27403

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Any Distribution

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+1 336-297-4977

Company Fax:

+1 336-297-9787

Support Phone Number:

+1 336-297-4977

Contact Email Address:

support@wss.net

Company URL:

http://www.wss.net

Primary Contact:

Jay Eisenberg

1.288 Consultant ID: US, NC, World Wide Wizdom

```
104 Quartermaine Ct.
Cary, NC 27513-5152
```

US

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

919-818-1855

Company Fax:

919-468-4684

Support Phone Number:

919-468-4680

Contact Email Address:

support@wiZdom.net

Company URL:

http://www.wiZdom.net

Primary Contact:

David Boone

1.289 Consultant ID: US, NE, Paktronix Systems LLC

TOC

1506 North 59th Street
Omaha, NE 68104-4830

US

Consulting Specialties:

Custom Systems Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Trinux, Pretty Secur

Reseller Authorizations:

IBM

Linux Certifications:

N/A

Company Phone Number:

1-402-932-7250

Company Fax:

1-402-932-7258

Support Phone Number:

1-402-932-7250

Contact Email Address:

tech@paktronix.com

Company URL:

http://www.paktronix.com

Primary Contact:

Paul Greunke

1.290 Consultant ID: US, NH, AKCP, Inc

```
379 Amherst Street, Suite 182
Nashua, NH 03063

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

603-881-7343

Company Fax:

N/A

Support Phone Number:

603-881-7343

Contact Email Address:

info@kcpinc.com

Company URL:

http://www.kcpinc.com

Primary Contact:

Brad Klein

1.291 Consultant ID: US, NJ, Etc Services

```
31 N. Vivyen St.
Bergenfield, NJ 07621-1529

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

201-385-7113

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Scott Webster

1.292 Consultant ID: US, NJ, Voortek Software, Inc

```
1 Las Brisas Blvd
Voorhees, NJ 08043

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

Caldera

Company Phone Number:

David J Adamson

Company Fax:

609-346-3172

Support Phone Number:

888-346-4616

Contact Email Address:

voorteksoftware@erols.com

Company URL:

http://www.erols.com

Primary Contact:

David J Adamson

1.293 Consultant ID: US, NON United States, ASA Computers, Inc.

```
23
N/A, NON United States N/A

US
```

Consulting Specialties:

N/A

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

N/A

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

N/A N/A

1.294 Consultant ID: US, NON United States, Coldweb Consulting

```
2200 A Montclair Rd
Greensboro, NON United States 27410

US
```

Consulting Specialties:

E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Caldera

Other Distribution:

Redhat and Other

Specialty Distribution:

Other

Other Specialty Distribution:

Turbo Linux

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

336-323-0253

Company Fax:

N/A

Support Phone Number:

N/A

Contact Email Address:

mjezorek@usa.net

Company URL:

http://www.quack-addict.com

Primary Contact:

Matt Jezorek

1.295 Consultant ID: US, NON United States, Emit Sp z o.o.

```
ul. Koszarowa 48/9
Wroclaw, NON United States 51-149

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

Custom

Specialty Distribution:

Other

Other Specialty Distribution:

Single floppy

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+48713267026

Company Fax:

+48713289733

Support Phone Number:

+48713288967

Contact Email Address:

kontakt@emit.pl

Company URL:

http://www.emit.pl

Primary Contact:

Ian Carr-de Avelon

1.296 Consultant ID: US, NON United States, LinuxAsia Pte Ltd

```
N/A
N/A, NON United States N/A

US
```

Consulting Specialties:

N/A

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

N/A

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

N/A N/A

1.297 Consultant ID: US, NON United States, Magiclink

```
Av. ACM, Sala 304 Edf. Maxcenter - Pituba
Salvador - BA, NON United States 41858-900

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

+55-71-351-1615

Company Fax:

+55-71-353-2220

Support Phone Number:

+55-71-351-1615

Contact Email Address:

admin@magiclink.com.br

Company URL:

http://www.magiclink.com.br

Primary Contact:

Isamar Maia

1.298 Consultant ID: US, NON United States, Psand Limited

```
N/A
N/A, NON United States N/A

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming System Security Virtual Private Networking

Main Distribution:

Caldera

Other Distribution:

RedHat

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Caldera

Company Phone Number:

N/A

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://www.psand.net

Primary Contact:

Mike Harris

1.299 Consultant ID: US, NY, Aegis Information Systems, Inc.

P.O. Box 730
Hicksville, NY 11802

US

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

various

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

800-234-4700

Company Fax:

516-937-3999

Support Phone Number:

800-344-8165

Contact Email Address:

help@AegisInfoSys.com.

Company URL:

http://www.AegisInfoSys.com.

Primary Contact:

David Harari

1.300 Consultant ID: US, NY, Andrew Madloch

TOC

```
152 Madison Avenue, Suite 1104
New York, NY 10016

US
```

Consulting Specialties:

Custom Programming E-Commerce Internet (Web) Programming

Main Distribution:

RedHat

Other Distribution:

Slackware

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

+1 212 214 0558

Company Fax:

+1 212 214 0558

Support Phone Number:

+1 212 214 0558

Contact Email Address:

support@intermarket3d.com

Company URL:

http://www.intermarket3d.com

Primary Contact:

Andrew Madloch

TOC

1.301 Consultant ID: US, NY, Donahue Consulting

```
100 John St., #1410
New York, NY 10038

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

212.998.3158

Company Fax:

212.995.4120

Support Phone Number:

212.998.3158

Contact Email Address:

adam@donahueinc.net

Company URL:

http://www.donahueinc.net

Primary Contact:

Adam Donahue

1.302 Consultant ID: US, NY, JAP Consulting

```
82 Beach 2nd Street
Lawrence, NY 11559

US
```

Consulting Specialties:

Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration

Main Distribution:

Suse

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-516-371-6886

Company Fax:

1-516-371-6886

Support Phone Number:

1-516-371-6886

Contact Email Address:

support@japconsulting.com

Company URL:

http://www.japconsulting.com

Primary Contact:

Jason Pearsall

1.303 Consultant ID: US, NY, Lrw.Net

```
188 2nd Ave, #9
New York, NY 10003

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Suse

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-212-254-3551

Company Fax:

1-212-254-6175

Support Phone Number:

1-212-254-3551

Contact Email Address:

rw26@lrw.net

Company URL:

http://lrw.net

Primary Contact:

Randy Wright

1.304 Consultant ID: US, NY, RetiComp, Inc.

```
75 West End Ave. Suite R32C
New York, NY 10023

US
```

Consulting Specialties:

Firewalls and Internet Security Internet Server Development Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

212-957-6710

Company Fax:

212-957-6711

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://www.reticomp.com

Primary Contact:

Donn Morrill

1.305 Consultant ID: US, NY, Templar Studios, Inc.

```
157 Ludlow St.
New York, NY 10002

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Or any other...

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

212-982-9360

Company Fax:

212-982-9370

Support Phone Number:

212-982-9360

Contact Email Address:

dwood@templar.com

Company URL:

http://www.templar.com

Primary Contact:

David Wood

1.306 Consultant ID: US, NY, Zenez, Inc.

```
40 Paine Street
Lindenhurst, NY 11757

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

Not Applicable

Company Phone Number:

1-516-957-1907

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

ben@amvalue.com

Company URL:

http://N/A

Primary Contact:

Benjamin Suto

1.307 Consultant ID: US, OH, 800Linux

```
1357 West Lane Ave suite 211
Columbus, OH 43221

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

614-324-1610

Company Fax:

614-324-1617

Support Phone Number:

614-324-1610

Contact Email Address:

suppor

Company URL:

http://www.linux800.com

Primary Contact:

N/A N/A

1.308 Consultant ID: US, OH, Advanced Unix and Network Consulting

```
7652 Crimson Ct.
Mentor, OH 44060

US
```

Consulting Specialties:

Firewalls and Internet Security Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

(440) 255-0421

Company Fax:

(440) 255-0421

Support Phone Number:

none

Contact Email Address:

none

Company URL:

http://n/a

Primary Contact:

Dennis Dumont

1.309 Consultant ID: US, OH, Bradley M. Kuhn

```
P.O. Box 20042
Cincinnati, OH 45220

US
```

Consulting Specialties:

Custom Programming Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

Red Hat

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

513-475-0311

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

bkuhn@acm.org

Company URL:

http://www.ebb.org/bkuhn

Primary Contact:

Bradley Kuhn

1.310 Consultant ID: US, OH, Cleveland Internet Association

```
4876 East 85th
Cleveland, OH 44125

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

TOC

RedHat

Other Distribution:

Will consider any

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

216-552-4368

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Jeff Garvas

1.311 Consultant ID: US, OH, Computer Wizards Consulting

```
148 Apricot Ln
Dayton, OH 45433-1109

US
```

Consulting Specialties:

Custom Programming Firewalls and Internet Security Internet Server Development Internet (W
Programming Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

HP

Linux Certifications:

Not Applicable

Company Phone Number:

937-754-0458

Company Fax:

937-754-0194

Support Phone Number:

937-754-0458

Contact Email Address:

support@computerwizards.com

Company URL:

http://www.computerwizards.com

Primary Contact:

Jeff Lamb

1.312 Consultant ID: US, OH, Essex Software Development, Inc.

```
3452 Meadowbrook Blvd.
```

Cleveland Heights, OH 44118

US

Consulting Specialties:

CIFS (Samba) Custom Programming Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

216-932-1230

Company Fax:

216-397-3380

Support Phone Number:

216-932-1230

Contact Email Address:

Support@esdi.com

Company URL:

http://http://www.esdi.com

Primary Contact:

Brendan O'Haire

1.313 Consultant ID: US, OH, LANshark Systems, Inc

TOC

784 Morrison Road
Columbus, OH 43230

US

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

All others

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

614-751-1111

Company Fax:

614-751-1112

Support Phone Number:

614-751-1111

Contact Email Address:

support@lanshark.com

Company URL:

http://www.lanshark.com

Primary Contact:

Scott Sharkey

1.314 Consultant ID: US, OH, Lightstream Internet, Inc.

```
7700 Saint Clair
Mentor, OH 44060

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(440)/269-2352

Company Fax:

(440)/942-6387

Support Phone Number:

(440)/269-2352

Contact Email Address:

tech@lightstream.net

Company URL:

http://www.lightstream.net

Primary Contact:

Jon Changnon

1.315 Consultant ID: US, OH, TES, Inc.

```
2903 Ravogli Ave.
Cincinnati, OH 45211

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+1 513 661 3200

Company Fax:

TOC

+1 513 661 3732

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Larry Townsend

1.316 Consultant ID: US, OH, The Computer Underground

```
1357 West Lane Ave suite 210
Columbus, OH 43221

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

N/A

Company Phone Number:

614-485-0506

Company Fax:

N/A

Support Phone Number:

614-485-0506

Contact Email Address:

support@tcu-inc.com

Company URL:

http://http://www.tcu-inc.com

Primary Contact:

Mark Nielsen

1.317 Consultant ID: US, OH, The Computer Underground, Inc.

```
1357 West Lane Ave suite 210
Columbus, OH 43221

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

614-485-0506

Company Fax:

614-485-9617

Support Phone Number:

614-485-0506

Contact Email Address:

suport@tcu-inc.com

Company URL:

http://www.tcu-inc.com

Primary Contact:

Mark Nielsen

1.318 Consultant ID: US, OK, After Hours Computer Consulting,LLC.

```
2 West 6th Street, Suite 450
Tulsa, OK 74119

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

918.583.1004

Company Fax:

918.587.2422

Support Phone Number:

918.583.1004

Contact Email Address:

support@ahcctulsa.com

Company URL:

http://www.ahcctulsa.com

Primary Contact:

Kenneth Hess

1.319 Consultant ID: US, OR, 3D Technology, Inc.

```
9150 SW Pioneer Ct. Suite E
Portland, OR 97070

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet (Web) Programming Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

HP

Linux Certifications:

RedHat

Company Phone Number:

877-333-2677

Company Fax:

503-582-9102

Support Phone Number:

503-582-1190

Contact Email Address:

support@3dtech.net

Company URL:

http://www.3dtech.net

Primary Contact:

Kerry Lafferty

1.320 Consultant ID: US, OR, Avalon Technology Group, Inc.

```
0607 SW Idaho St.
Portland, OR 97201

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet

Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

HP

Linux Certifications:

Not Applicable

Company Phone Number:

(503) 246-3630

Company Fax:

(503) 246-3124

Support Phone Number:

(503) 246-3630

Contact Email Address:

support@avalongroup.net

Company URL:

http://www.avalongroup.net

Primary Contact:

Timothy Grant

1.321 Consultant ID: US, OR, Fargus.Net

```
326 sw 7th street
Corvallis, OR 97333
```

US

Consulting Specialties:

Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

1 541 754 6231

Company Fax:

N/A

Support Phone Number:

N/A

Contact Email Address:

nick@fargus.net

Company URL:

http://www.fargus.net

Primary Contact:

Nick Rusnov

1.322 Consultant ID: US, OR, Fence-Post Technology

5117 NE 31st Ave
Portland, OR 97211

US

Consulting Specialties:

Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

503-287-5244

Company Fax:

503-335-9306

Support Phone Number:

503-287-5244

Contact Email Address:

info@fence-post.com

Company URL:

http://www.fence-post.com

Primary Contact:

Art Opiela-Young

1.323 Consultant ID: US, OR, Hurrah Internet Services


```
28 SW 1st Ave.
Portland, OR 97204

US
```

Consulting Specialties:

Custom Programming E-Commerce Internet Server Development Internet (Web) Programming System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-503-380-3218

Company Fax:

1-503-223-3140

Support Phone Number:

1-503-380-3218

Contact Email Address:

support@hurrah.com

Company URL:

http://www.hurrah.com

Primary Contact:

Tracy Camp David Elkins

1.324 Consultant ID: US, OR, Naked Ape Consulting

```
4404 SE Nehalem St.
Portland, OR 97206

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

None

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+1 503 788 4688

Company Fax:

+1 503 788 4688

Support Phone Number:

+1 503 788 4688

Contact Email Address:

wcooley@nakedape.navi.net

Company URL:

http://nakedape.navi.net

Primary Contact:

W. Reilly Cooley

1.325 Consultant ID: US, PA, Associated Micro Systems, Inc.

```
140 South Village Avenue Suite 20
Exton, PA 19341

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

610-524-9816

Company Fax:

610-524-9821

Support Phone Number:

610-524-9816

Contact Email Address:

mmarcini@amicrosys.com

Company URL:

http://www.amicrosys.com

Primary Contact:

Mike Marciniszyn

1.326 Consultant ID: US, PA, JAV-Associates Consulting

```
239 South West Street
York, PA 17404

US
```

Consulting Specialties:

Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

7178540425

Company Fax:

3058322662

Support Phone Number:

7178540425

Contact Email Address:

tech@src-router.tytek.net

Company URL:

http://src-router.tytek.net

Primary Contact:

John Vorchak

1.327 Consultant ID: US, PA, Michael Jastremski

```
922 South 46ths
Philadelphia, PA 19143

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Any

Specialty Distribution:

Other

Other Specialty Distribution:

Our Own Distro :)

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

215-681-8316

Company Fax:

215-555-1212

Support Phone Number:

215-681-8316

Contact Email Address:

support@westphila.net

Company URL:

http://http://westphila.net

Primary Contact:

Michael Jastremski

1.328 Consultant ID: US, PA, Reed Consulting

```
26 courtney st
pittsburgh, PA 15202

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

412-766-2373

Company Fax:

N/A

Support Phone Number:

N/A

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Joel Reed

1.329 Consultant ID: US, PA, Rich Westcott

```
4053 Boone St
Philadeplphia, PA 19127

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

slackware

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

215-487-7529

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://www.ctlinux.ecogchair.org

Primary Contact:

Rich Westcott N/A

1.330 Consultant ID: US, PA, Rich Westcott

```
4053 Boone St
Philadeplphia, PA 19127

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

slackware

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

215-487-7529

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://www.ctlinux.ecogchair.org

Primary Contact:

Rich Westcott N/A

1.331 Consultant ID: US, RI, New England Computer

```
847 Mendon Rd.
Woonsocket, RI 02895

US
```

Consulting Specialties:

CIFS (Samba)

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

4017629980

Company Fax:

4017650031

Support Phone Number:

4017629980

Contact Email Address:

jaxon6@home.com

Company URL:

http://www.home.com

Primary Contact:

Patrick Paul

1.332 Consultant ID: US, SC, Crux

```
729 Martin Raod
Starr, SC 29684

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

864-375-3077

Company Fax:

N/A

Support Phone Number:

864-375-0377

Contact Email Address:

carlk@carol.net

Company URL:

http://www.carol.net

Primary Contact:

Carl A. Kolasky N/A

1.333 Consultant ID: US, SC, GAME.NET

```
1209 Midvale Avenue
Charleston, SC 29412
```

TOC

US

Consulting Specialties:

Internet Server Development Internet (Web) Programming Network Administration System Administration

Main Distribution:

Other

Other Distribution:

slackware

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-843-762-7131

Company Fax:

1-843-762-7131

Support Phone Number:

1-843-762-7131

Contact Email Address:

info@game.net

Company URL:

http://game.net

Primary Contact:

Jim Sims

1.334 Consultant ID: US, SC, GAME.NET, Inc.

TOC

1209 Midvale Avenue
Charleston, SC 29412

US

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

SlackWare

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+1-843-762-7131

Company Fax:

fax is for losers -

Support Phone Number:

+1-843-762-7131

Contact Email Address:

support@game.net

Company URL:

http://www.game.net

Primary Contact:

Rick Bauman

1.335 Consultant ID: US, SC, Linux Empower

```
N/A
Sumter, SC 29040

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

N/A

Company Fax:

N/A

Support Phone Number:

N/A

Contact Email Address:

consulting@linuxempower.com

Company URL:

http://www.linuxempower.com

Primary Contact:

Rob Smith

1.336 Consultant ID: US, TN, Michael Chaney Consulting Corporation

```
2624 Oak Forest Drive
Antioch, TN 37013

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Internet (Web) Programming Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+1 615 361 3254

Company Fax:

+1 615 361 1244

Support Phone Number:

+1 615 361 3254

Contact Email Address:

mdchaney@michaelchaney.com

Company URL:

http://www.michaelchaney.com/serious

Primary Contact:

Michael Chaney

1.337 Consultant ID: US, TX, Aegis Data Systems

```
PO Box 28554
Dallas, TX 75228

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

214.394.4213

Company Fax:

972.857.9909

Support Phone Number:

214.394.4213

Contact Email Address:

operations@aegisdata.com

Company URL:

http://www.aegisdata.com

Primary Contact:

Mark Stingley

1.338 Consultant ID: US, TX, Computer Innovations International

```
2145 NW Military Highway, Suite 103
San Antonio, TX 78213

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Caldera

Specialty Distribution:

Other

Other Specialty Distribution:

Intel/Sparc/Alpha

Reseller Authorizations:

Compaq HP IBM

Linux Certifications:

Caldera

Company Phone Number:

210-348-0850 or 210-

Company Fax:

210-348-0851

Support Phone Number:

210-348-0850 or 210-

Contact Email Address:

techsales@txcii.com

Company URL:

http://www.txcii.com

Primary Contact:

Waldemar Vega

1.339 Consultant ID: US, TX, Internet Power And Light

```
19009 Preston Rd #215-240
Dallas, TX 75252-2496

US
```

Consulting Specialties:

Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+1 972 407 0727

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Phil Howard

1.340 Consultant ID: US, TX, LEAP _ Linux Enterprise Application Products

```
P.O. Box 850905
Richardson, TX 75085-0905

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Linux Host with ASCI

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+1.972.475.8440

Company Fax:

+1.972.463.2998

Support Phone Number:

+1.972.475.8440

Contact Email Address:

adcoker@leap4linux.com

Company URL:

http://www.leap4linux.com

Primary Contact:

Dr. Stephen F. Wheeler

1.341 Consultant ID: US, TX, PSW Technologies

```
6300 Bridgepoint Parkway, Bldg 3, Ste 200
Austin, TX 78730

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming System Administration System Security Virtual Private Networking

Main Distribution:

TOC

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat Not Applicable

Company Phone Number:

512-343-6666

Company Fax:

512-343-9650

Support Phone Number:

512-343-6666

Contact Email Address:

linux@pswtech.com

Company URL:

http://www.psw.com

Primary Contact:

Satish Rao

1.342 Consultant ID: US, TX, ProSoft Plus, Inc.

```
11113 Cherry Park
San Antonio, TX 78249

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Network Administration System

Administration Virtual Private Networking

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

IBM

Linux Certifications:

Caldera

Company Phone Number:

(210)694-0801

Company Fax:

(210) 699-9416

Support Phone Number:

Not Available

Contact Email Address:

prosoft@satx.net

Company URL:

http://N/A

Primary Contact:

Ed Valentine

1.343 Consultant ID: US, TX, Quinn Team Incorporated

```
12941 North Freeway
Houston, TX 77060
```

US

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Mandrake

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

281.875.3883 #211

Company Fax:

281.875.3890

Support Phone Number:

Not Available

Contact Email Address:

linux@quinnteam.com

Company URL:

http://www.quinnteam.com

Primary Contact:

Micah Quinn

1.344 Consultant ID: US, TX, R and B Consulting

TOC

```
14019 SW FWY #301  PMB 411
Sugar land, TX 77478

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1-281-494-UNIX

Company Fax:

1-281-494-4329

Support Phone Number:

N/A

Contact Email Address:

Support@TexasComputers.com

Company URL:

http://www.TexasComputers.com

Primary Contact:

Brian Walters

1.345 Consultant ID: US, TX, Z Consulting

```
Route 1 Box 68 A-1
Hockley, TX 77447

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

409-931-2838

Company Fax:

N/A

Support Phone Number:

409-931-2838

Contact Email Address:

wczimmer@earthlink.net

Company URL:

http://www.earthlink.net

Primary Contact:

Chris Zimmerman

1.346 Consultant ID: US, UT, Open Source Consulting Group

```
5136 W. 6435 S.
West Jordan, UT 84084

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

Any

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

Any

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(801) 966-1297

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

info@oscg.net

Company URL:

http://www.oscg.net

Primary Contact:

Mark Whitley

1.347 Consultant ID: US, VA, 2RAD

```
14402 Haystack Court
Centreville, VA 20120

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

703 502-0640

Company Fax:

703 502 8870

Support Phone Number:

703 502-0640

Contact Email Address:

rad@2rad.net

Company URL:

http://www.2rad.net

Primary Contact:

Mark Rorabaugh

1.348 Consultant ID: US, VA, Atlantis Consulting

```
206B 88th Street
Virginia Beach, VA 23451

US
```

Consulting Specialties:

Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

Other

Other Specialty Distribution:

None

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

757 425-5950

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

mskar@zasoftware.com

Company URL:

http://www.zasoftware.com

Primary Contact:

Marc Skarshinski

1.349 Consultant ID: US, VA, Atlantis

```
206B 88th Street
Virginia Beach, VA 23451

US
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

Slackware

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

757 425-5950

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

mskar@zasoftware.com

Company URL:

http://www.zasoftware.com

Primary Contact:

Marc Skarshinski

1.350 Consultant ID: US, VA, Branch and Company

```
P.O. Box 547
Powhatan, VA 23139

US
```

Consulting Specialties:

Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

804-598-2153 USA

Company Fax:

804-225-8023 USA

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Carlisle Branch

1.351 Consultant ID: US, VA, Brown Technolgies Network, Inc.

```
43953 Bruceton Mills Circle
Ashburn, VA 20147-4806

US
```

Consulting Specialties:

Custom Programming Custom Systems Firewalls and Internet Security Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

703-858-0642

Company Fax:

703-858-0642

Support Phone Number:

703-858-0642

Contact Email Address:

pbrown@btechnet.com

Company URL:

http://www.btechnet.com

Primary Contact:

Paul Brown

1.352 Consultant ID: US, VA, Linux and Web Professionals

```
PO Box 2763
Lynchburg, VA 24501

US
```

Consulting Specialties:

Custom Programming Firewalls and Internet Security Internet Server Development Internet (Web) Programming

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

1 (973) 283-4712

Company Fax:

1 (973) 283-4712

Support Phone Number:

1 (973) 283-4712

Contact Email Address:

support@linuxwebpro.com

Company URL:

http://www.linuxwebpro.com

Primary Contact:

Floyd Jennings

1.353 Consultant ID: US, VA, NetSource of Virginia, Inc.

```
7639 Hull Street Rd.   STE 202
Richmond, VA 23235-6438

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server

Development Netware Connectivity Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

Slackware and Red Ha

Specialty Distribution:

Other

Other Specialty Distribution:

none currently

Reseller Authorizations:

HP

Linux Certifications:

N/A

Company Phone Number:

1-804-674-6040

Company Fax:

1-804-674-6041

Support Phone Number:

N/A

Contact Email Address:

netsource@erols.com

Company URL:

http://www.erols.com/netsource

Primary Contact:

Jeff Davis

1.354 Consultant ID: US, VA, Noblestar Systems Corporation

```
12021 Sunset Hills Road
```

Reston, VA 20190

US

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Internet Server Development Internet (Web) Programming System Administration

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

703.464.4000

Company Fax:

703.464.4001

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://http://www.noblestar.com

Primary Contact:

Jerry Matus

1.355 Consultant ID: US, VA, Phaedo Consulting, Inc.

TOC

8000 Towers Crescent Drive, Suite 1350
Vienna, VA 22182

US

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

SparcLinux, AlphaLin

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

703-847-3639

Company Fax:

540-370-4185

Support Phone Number:

703-847-3639

Contact Email Address:

info@phaedo.com

Company URL:

http://www.phaedo.com

Primary Contact:

Ryan Quick

1.356 Consultant ID: US, VA, Signet Computers, Inc.

```
1901 N. Moore St. Suite 805
Arlington, VA 22209

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

Slackware

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

703 812 8585

Company Fax:

703 812 8547

Support Phone Number:

703 812 8585

Contact Email Address:

sci-info@sigcomp.com

Company URL:

http://www.sigcomp.com

Primary Contact:

Chuck Olczak

1.357 Consultant ID: US, VA, Spectrum Office Systems, Inc.

```
11320 Random Hills Road, Suite 630
Fairfax, VA 22030-6001

US
```

Consulting Specialties:

Firewalls and Internet Security Netware Connectivity Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

703-591-7400

Company Fax:

703-591-9780

Support Phone Number:

703-591-7400

Contact Email Address:

tech@spectrum-systems.com

Company URL:

http://www.spectrum-systems.com

Primary Contact:

Nancy Fallon

1.358 Consultant ID: US, VT, Everest Systems

```
Route 100 South
Morrisville, VT 05661

US
```

Consulting Specialties:

Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq IBM

Linux Certifications:

Not Applicable

Company Phone Number:

802-888-7857

Company Fax:

802-888-7857

Support Phone Number:

802-888-7857

Contact Email Address:

braman@world.std.com

Company URL:

http://world.std.com/~braman/

Primary Contact:

Balu Raman, MSEE.,CC Veda Aranyam

1.359 Consultant ID: US, WA, Cheek Consulting

```
620 5th Avve W, Suite 309
Seattle, WA 98119

US
```

Consulting Specialties:

CIFS (Samba) E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

206 282-2892

Company Fax:

N/A

Support Phone Number:

206 282-2892

Contact Email Address:

support@cheek.com

Company URL:

http://www.cheek.com

Primary Contact:

joseph cheek

1.360 Consultant ID: US, WA, Foresight Technologies

```
705 204th DR SE
Snohomish, WA 98290

US
```

Consulting Specialties:

N/A

Main Distribution:

Other

Other Distribution:

slakware

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

425-334-0933

Company Fax:

N/A

Support Phone Number:

425-334-0933

Contact Email Address:

dagnabit2@yahoo.com

Company URL:

http://N/A

Primary Contact:

Dan Thacker

1.361 Consultant ID: US, WA, K and D Development

```
2103 N 106th St. #C-402
Seattle, WA 98133

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

206 365 5585

Company Fax:

N/A

Support Phone Number:

206 365 5585

Contact Email Address:

info@kd-dev.com

Company URL:

http://www.kd-dev.com

Primary Contact:

Derek Simkowiak

1.362 Consultant ID: US, WA, PeakServ

```
104 Prospect Street
Bellingham, WA 98225

US
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

800-453-5322

Company Fax:

360-733-7818

Support Phone Number:

800-453-5322

Contact Email Address:

info@peakserv.com

Company URL:

http://www.peakserv.com

Primary Contact:

John Blanford

1.363 Consultant ID: US, WA, Tacoma and Puget Sound Company

```
4419 Harbor Ridge Road NE
Tacoma, WA 98422

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet

Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

Slackware

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(253)927-4051

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

spj@tapsco.com

Company URL:

http://N/A

Primary Contact:

Steven Johnson

1.364 Consultant ID: US, WA, Veriteam, Inc.

```
3039 173rd Ct NE
Redmond, WA 98052-6228
```

US

Consulting Specialties:

Custom Programming Internet Server Development Internet (Web) Programming

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

425.882.0938

Company Fax:

425.882.0938

Support Phone Number:

425.882.0938

Contact Email Address:

support@veriteam.com

Company URL:

http://www.veriteam.com

Primary Contact:

Bernie Thompson

1.365 Consultant ID: US, WA, buzzwerks, Inc.

1000 Second Ave, Suite 4060

Seattle, WA 98104-1094

US

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

UltraPenguin

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+001 206 625 0135

Company Fax:

+001 206 625 0304

Support Phone Number:

+001 206 625 0135

Contact Email Address:

support@buzzwerks.com

Company URL:

http://www.buzzwerks.com

Primary Contact:

Mitch Pirtle

1.366 Consultant ID: US, WI, Midnight Software

TOC

N11546 Old 41 Road
Daggett, (Michigan), WI 49821

US

Consulting Specialties:

Custom Programming Custom Systems

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

906-639-2882

Company Fax:

906-639-2558

Support Phone Number:

N/A

Contact Email Address:

midnight@alphacomm.net

Company URL:

http://N/A

Primary Contact:

Aaron Tuinstra

TOC

1.367 Consultant ID: US, WI, Network Integration Services

S29W30139 S. Bethesda Cir.
Waukesha, WI 53188

US

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

414-256-9617

Company Fax:

414-968-3044

Support Phone Number:

414-256-9617

Contact Email Address:

nis@pobox.com

Company URL:

http://N/A

Primary Contact:

Michael Bloxham

1.368 Consultant ID: US, WI, Perry Systems

PO Box 207
Milwaukee, WI 53201-0207

US

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

414/803-4797

Company Fax:

NA

Support Phone Number:

NA

Contact Email Address:

help@perrysystems.com

Company URL:

http://www.perrysystems.com

Primary Contact:

James Offutt

1.369 Consultant ID: US, WI, Rindy Consulting

```
125 North Executive Drive; Suite 380
Brookfield, WI 53005

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq HP

Linux Certifications:

Not Applicable

Company Phone Number:

414-252-4273

Company Fax:

414-252-4274

Support Phone Number:

414-252-4273

Contact Email Address:

support@rindy.com

Company URL:

http://www.rindy.com

Primary Contact:

Troy Rindy

1.370 Consultant ID: US, WI, WhitePine Consulting

```
606 N. Main St
Fall River, WI 53932

US
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Internet (Web) Programming

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

920.484.6005

Company Fax:

N/A

Support Phone Number:

920.484.6005

Contact Email Address:

mjmc@whitepine.com

Company URL:

http://www.whitepine.com

Primary Contact:

N/A N/A

1.371 Consultant ID: United Arab Emirates, JM Infotech Solutions STPI

```
P.O.Box 2687
Dubai, NON United States 2687

United Arab Emirates
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

971 50 6326822

Company Fax:

971 4 359295

Support Phone Number:

971 50 6326822

Contact Email Address:

mammen@emirates.net.ae

Company URL:

http://www.emirates.net.ae

Primary Contact:

John Mammen

1.372 Consultant ID: United Kingdom, CiteCall Ltd.

```
Claremont, Coppice Lane
Chorley, Lancashire, NON United States PR6 9DA

United Kingdom
```

Consulting Specialties:

CIFS (Samba) Custom Programming Internet (Web) Programming Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

Suse

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+44 1257 230 638

Company Fax:

+44 870 164 0846

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Kevin Mulholland

1.373 Consultant ID: United Kingdom, Enterprise Management Consulting

```
Barnaby House, Charles Dickens Terrace, Maple Rd.
London, NON United States SE20 8RE

United Kingdom
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

RedHat

Company Phone Number:

44 181 659 2000

Company Fax:

44 181 778 0101

Support Phone Number:

44 181 659 2000

Contact Email Address:

emcweb@btinternet.com

Company URL:

http://www.emcuk.com

Primary Contact:

Malcolm Macsween

1.374 Consultant ID: United Kingdom, FGM Technology LTD EuroDev Cybernetics LTD

```
1 South Terrace
Thirsk, NON United States YO7 1RH

United Kingdom
```

Consulting Specialties:

Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

EuroDev Linux and ARM

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

0441 197 869 / 01845

Company Fax:

NA

Support Phone Number:

ONLY GIVEN TO CLIENT

Contact Email Address:

ONLY GIVEN TO CLIENTS

Company URL:

http://N/A

Primary Contact:

N/A N/A

1.375 Consultant ID: United Kingdom, J.S Unix Consultants Ltd

```
1073 Manchester Road, Castleton
Rochdale, NON United States OL11 2XJ

United Kingdom
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Internet (Web) Programming Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

Suse

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

(+44) 958 402172

Company Fax:

(+44) 970 0548678

Support Phone Number:

(+44) 958 402172

Contact Email Address:

support@santoroj.demon.co.uk

Company URL:

http://www.netspear.co.uk/js_unix

Primary Contact:

Joe Santoro

1.376 Consultant ID: United Kingdom, Jason Clifford

```
264 Ripon Road
Stevenage, NON United States SG1 4NF

United Kingdom
```

TOC

Consulting Specialties:

CIFS (Samba) Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

Other

Other Distribution:

All distributions

Specialty Distribution:

Other

Other Specialty Distribution:

All

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+44 7957 356660

Company Fax:

+44 1438 223433

Support Phone Number:

Not Available

Contact Email Address:

support@dlsl.demon.co.uk

Company URL:

http://www.dlsl.demon.co.uk

Primary Contact:

Jason Clifford

1.377 Consultant ID: United Kingdom, Michiel Dethmers

Quipu, 91 Brick Lane
London, NON United States E1 6QL

United Kingdom

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

44 171 3776601

Company Fax:

44 171 3776601

Support Phone Number:

N/A

Contact Email Address:

support@quipu.co.uk

Company URL:

http://www.quipu.co.uk

Primary Contact:

Michiel Dethmers

1.378 Consultant ID: United Kingdom, Philip Hanc Computing Ltd


```
Unit 1, Cherry Close, Caversham
Reading, NON United States RG4 8UP

United Kingdom
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development System Security

Main Distribution:

Debian

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

N/A

Linux Certifications:

N/A

Company Phone Number:

+44 118 9545656

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

N/A

Company URL:

http://N/A

Primary Contact:

Philip Hands

1.379 Consultant ID: United Kingdom, RedAmazon

```
N/A
London, NON United States N/A

United Kingdom
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxRouter

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

N/A

Company Fax:

N/A

Support Phone Number:

Not Available

Contact Email Address:

support@redamazon.com

Company URL:

http://www.redamazon.com

Primary Contact:

Jim Thoreaux

1.380 Consultant ID: United Kingdom, Share Net International Ltd

```
Belle Vue, Crowborough
East Sussex, NON United States TN6 1ST

United Kingdom
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security Virtual Private Networking

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+044 01892 661 044

Company Fax:

+044 01892 665 731

Support Phone Number:

+044 01892 661 044

Contact Email Address:

support@share-net.co.uk

Company URL:

http://www.share-net.co.uk

Primary Contact:

Grace Nunn

1.381 Consultant ID: United Kingdom, iCode Systems Ltd

```
Consort Court, High Street
Fareham, NON United States PO16 7AL

United Kingdom
```

Consulting Specialties:

Custom Programming E-Commerce Firewalls and Internet Security Internet Server Development Internet (Web) Programming Network Administration System Administration

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq

Linux Certifications:

N/A

Company Phone Number:

+44 1329 829898

Company Fax:

+44 1329 829899

Support Phone Number:

+44 1329 829898

Contact Email Address:

linux@icode.co.uk

Company URL:

http://www.icode.co.uk

Primary Contact:

Darren Durbin

1.382 Consultant ID: Venezuela, Ernesto Hernández-Novich

```
Av. Ppal. La Bonita Res. La Montaña Apto. 7-F
Caracas, NON United States 1080

Venezuela
```

Consulting Specialties:

CIFS (Samba) Custom Programming Custom Systems Firewalls and Internet Security Internet Server Development Internet (Web) Programming Netware Connectivity Network Administration System Administration System Security

Main Distribution:

Other

Other Distribution:

All of them

Specialty Distribution:

Other

Other Specialty Distribution:

None

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

58-2-9414198

Company Fax:

N/A

Support Phone Number:

58-14-9254779

Contact Email Address:

emhn@telcel.net.ve

Company URL:

http://www.telcel.net.ve

Primary Contact:

Ernesto Hernández-Novich

1.383 Consultant ID: Venezuela, Grupo INABOX,C.A.

```
Torre Capriles, Planta Baja, Local 13
Caracas, NON United States N/A

Venezuela
```

Consulting Specialties:

System Administration

Main Distribution:

Caldera

Other Distribution:

NA

Specialty Distribution:

LinuxPPC

Other Specialty Distribution:

NA

Reseller Authorizations:

Not Applicable

Linux Certifications:

Caldera

Company Phone Number:

005827934692

Company Fax:

005827934692

Support Phone Number:

005827934692

Contact Email Address:

inaboxca@yahoo.com

Company URL:

http://www.yahoo.com

Primary Contact:

Gilberto Hernandez Meza

1.384 Consultant ID: Venezuela, Linux Sistemas, C.A.

```
Paseo Los Ilustres, Ed. Don Paco, Local 1, Los Cha
Caracas, NON United States 1041

Venezuela
```

Consulting Specialties:

CIFS (Samba) Firewalls and Internet Security Internet Server Development Netware Connectivity Network Administration System Administration System Security

Main Distribution:

RedHat

Other Distribution:

NA

Specialty Distribution:

Other

Other Specialty Distribution:

NA

Reseller Authorizations:

Compaq IBM

Linux Certifications:

Not Applicable

Company Phone Number:

+58 2 6934233

Company Fax:

+58 2 6623580

Support Phone Number:

+58 2 6934233

Contact Email Address:

linuxsistemas@mercared.com

Company URL:

http://http://www.mercared.com/linuxsistemas

Primary Contact:

Jose Rey

1.385 Consultant ID: Yugoslavia, Berislav Todorovic

```
Elektrotehnicki fakultet, Bulevar Revolucije 73
Belgrade SERBIA, NON United States 11000

Yugoslavia
```

Consulting Specialties:

CIFS (Samba) Custom Systems Firewalls and Internet Security Internet Server Development Network Administration System Administration System Security Virtual Private Networking

TOC

Main Distribution:

RedHat

Other Distribution:

N/A

Specialty Distribution:

Other

Other Specialty Distribution:

Slackware

Reseller Authorizations:

Not Applicable

Linux Certifications:

Not Applicable

Company Phone Number:

+381-11-3221-419

Company Fax:

+381-11-3248-681

Support Phone Number:

N/A

Contact Email Address:

beri@etf.bg.ac.yu

Company URL:

http://ubbg.etf.bg.ac.yu/~beri/

Primary Contact:

Berislav Todorovic

The Linux Cyrillic HOWTO

Alexander L. Belikoff, (abel@bfr.co.il), Berger Financial Research Ltd.

v4.0, 23 January 1998

This document describes how to set up your Linux box to typeset, view and print the documents in the Russian language.

1. Administrativia

- 1.1 Introduction
- 1.2 Availability and feedback
- 1.3 Acknowledgments and copyrights

2. Theoretical background

- 2.1 Characters and codesets

3. Preparing your environment

- 3.1 Text mode setup
- 3.2 The X Window System
- 3.3 First steps - Cyrillic in shells
- 3.4 bash
- 3.5 csh/tcsh
- 3.6 ksh
- 3.7 less
- 3.8 mc (The Midnight Commander)
- 3.9 rlogin
- 3.10 zsh

TOC

4. Editing text

- 4.1 Emacs and XEmacs
- 4.2 Using vi
- 4.3 Editing text with joe
- 4.4 Spell-checking Russian

5. Using Cyrillic with mail and news

- 5.1 Setting up Mail User Agents
- 5.2 Configuring your MTA

6. Browsing the Cyrillic Web

- 6.1 lynx
- 6.2 Netscape navigator

7. Cyrillic wordprocessing

- 7.1 TeX-based environments
- 7.2 The StarOffice suite

8. Printing and PostScript

- 8.1 Text to PostScript conversion
- 8.2 Text to TeX conversion

9. Cyrillic in PostScript

- 9.1 Adding Cyrillic fonts to Ghostscript

10. Print setup

- 10.1 Pre-loading Cyrillic fonts into a non-PostScript printer
- 10.2 Printing with different fonts

11. Localization and Internationalization

- 11.1 Locale
- 11.2 Internationalization

12. Staying compatible

- 12.1 MIME-based data compatibility
- 12.2 Explicit character set conversion
- 12.3 Cyrillic in the DOS emulator

13. Bibliography

14. Summary of the various useful resources

---

1. Administrativia

1.1 Introduction

This document covers the things you need to successfully work with information containing cyrillic text (mostly Russian) under Linux. Although this document assumes your using Linux as an operating system, most of information presented is equally applicable to many other Unix flavors. I shall try to keep the distinction as visible as possible.

There are a number of popular Linux distributions. As an example system I describe the RedHat 4.1 Linux (Vanderbildt) - the one I am personally using. Nevertheless, I shall try to highlight the differences, if they exist, in other popular distributions, such as Debian GNU/Linux and Slackware Linux.

Since such setup directly modifies and extends the Operating System, you should understand, what you are doing. Even though I tried to keep things as easy as possible, having some experience with a given piece of software is an advantage. I am not going to describe what the X Window System is or how to typeset the documents with TeX and LaTeX, or how to install printer in Linux. Those issues are covered in other documents.

For the same reason, in most cases I describe a system-wide setup, by default requiring *root* privileges. Still, if there is a possibility for user-level setup, I'll try to mention it.

NOTE: The X Window System, TeX and other Linux components are complex systems with a sofisticated configuration. If you do something wrong, you can not only fail with Russian setup, but to break the component as well, if not the entire system. This is not to scare you off, but merely to make you understand the seriousness of the process and be careful. Preliminary backup of the config files is **highly** recommended. Having a guru around is also advantageous.

1.2 Availability and feedback

This document is available at sunsite.unc.edu or tsx-11.mit.edu as a part of the *Linux Document Project*. Also, it may be available at various FTP sites containing Linux. Moreover, it may be included as a part of Linux distribution.

If you have any suggestions or corrections regarding this document, please, don't hesitate to contact me as abel@bfr.co.il. Any new and useful information about Cyrillic support in various Unices is *highly appreciated*. Remember, it will help the others.

1.3 Acknowledgments and copyrights

TOC

Many people helped me (and not only me) with valuable information and suggestions. Even more people contributed software to the public community. I am sorry if I forgot to mention somebody.

So, here they go:

- Bas V. de Bakker
- David Daves
- Serge Vakulenko
- Sergei O. Naoumov
- Winfried Truemper
- Ilya K. Orehov
- Michael Van Canneyt
- Alex Bogdanov
- ...and the countless helpful people from the relcom.fido.ru.unix and relcom.fido.ru.linux Usenet newsgroups.

This document is Copyright (C) 1995,1997 by Alexander L. Belikoff. It may be used and distributed under the usual Linux HOWTO terms described below.

The following is a Linux HOWTO copyright notice:

> *Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.*
>
> *All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.*
>
> *In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.*

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu. You may finger this address for phone number and additional contact information.

Unix is a technology trademark of the X/Open Ltd.; MS-DOS, Windows, Windows 95, and Windows NT are trademarks of the Microsoft Corp.; The X Window System is a trademark of The X Consortium Inc. Other trademarks belong to the appropriate holders.

---

2. Theoretical background

2.1 Characters and codesets

In order to understand and print characters of various languages, the system and software should be able to distinguish them from other characters. That is, each unique character must have a unique representation inside the operating system, or the particular software package. Such collection of all unique characters, that the system is able to represent at once, is called a *codeset*.

At the time of the most operating system's creation, nobody cared about software being multilingual. Therefore, the most popular codeset was (and actually is) an *ASCII* (American Standard Code for Information Interchange).

The *standard ASCII* (aka 7-bit ASCII) comprises 128 unique codes. Some of them ASCII defines as real printable characters, and some are so-called *control characters*, which had special meanings in the old communication protocols. Each element of the set is identified by an integer *character code* (0-127). The subset of printable characters represents those found on the typewriter's keyboard with some minor additions. Each character occupies 7 least significant bits of a byte, whereas the most significant one was used for control purposes (say, transmission control in old communication packages).

The 7-bit ASCII concept was extended by 8-bit ASCII (aka *extended ASCII*). In this codeset, the characters' codes' range is 0-255. The lower half (0-127) is pure ASCII, whereas the upper one contains 127 more characters. Since this codeset is backward compatible with the ASCII (character still occupies 8 bit, the codes correspond the old ASCII), this codeset gained wide popularity.

The 8-bit ASCII doesn't define the contents of the upper half of the codeset. Therefore the ISO organization took the responsibility of defining a family of standards known as *ISO 8859-X* family. It is a collection of 8-bit codesets, where the lower half of each codeset (characters with codes 0-127) matches the ASCII and the upper parts define characters for various languages. For example, the following codesets are defined:

- `8859-1` - Europe, Latin America (also known as *Latin 1*)
- `8859-2` - Eastern Europe
- `8859-5` - Cyrillic
- `8859-8` - Hebrew

In Latin 1, the upper half of the table defines various characters which are not part of the English alphabet, but are present in various european languages (german umlauts, french accentes etc).

Another popular extended ASCII implementation is so-called *IBM codepage* (named after some computer company, that developed this codeset for it's infamous personal computers). This one contains pseudo-graphic characters in the upper half.

Software, that doesn't make any assumptions about the 8-th bit of the ASCII data is called *8-bit clean*.

Some older programs, designed with 7-bit ASCII in mind are not 8-bit clean and may work incorrectly with your extended ASCII data. Most of packages, however, are able to deal with the extended ASCII by default, or require some very basic setup. **NOTE:** before posting the question *"I did all setup right, but I cannot enter/view Cyrillic characters!"*, please consult the section shells for the notes on the program, you are using.

For information about making your software 8-bit clean, see section locale-programming .

Since on most systems character occupies 8 bits, there is no way to extend ASCII more and more. The way to implement new symbols in ASCII-based codesets is creation of other extended ASCII implementations. This is the way, the Cyrillic ASCII set is implemented.

We already mentioned *ISO 8859-5* standard as the one defining the Cyrillic codeset. But as it often happens to the standards, this one was developed without taking into account the real practices in the former USSR. Therefore, one thing that standard really achieved was another degree of confusion. I wouldn't say that *ISO 8859-5* is widely used anywhere.

Other standards for Cyrillic include the so-called *Alt* codeset and *Microsoft CP1251* codepage. The former one was developed by (who?) for MS-DOS quite a while ago. Back then, there was not very buzz yet about internetworking, so the intention was to make it as compatible as possible with the IBM standard. Therefore the Alt codeset is effectively the same IBM codepage, where all specific European characters in the upper half were replaced with the Cyrillic ones, leaving the pseudographic ones. Therefore, it didn't screw the text windowing facilities and provided Cyrillic characters as well. The *Alt* standard is still alive and extremely popular in MS-DOS.

Microsoft CP1251 codepage is just an attempt of Microsoft to come up with the new standard for Cyrillic codeset in Windows. As far as I know, it is not compatible with anything else (not very surprizing, huh?)

And finally there is *KOI8-R*. This one is also quite old, but it was designed wisely and nowadays the design points of it look really useful.

Again, it is compatible with ASCII, and the Cyrillic characters are located in the upper half. But the main design point of *KOI8-R* is that the Cyrillic characters' positions must correspond to the English characters with the same phonetics. Namely, if we set the eighth bit of the English character `'a'`, we'll get the Cyrillic `'a'`. This means that, given the Cyrillic text written in KOI8-R, we can strip the eighth bit of each character *and we still get a readable text, although written with English characters!* This is very important now, since there are many mailers on the Internet, that just strip the eighth bit silently, being sure that every single soul on the face of the Earth speaks English.

Not surprisingly, *KOI8-R* quickly became a de-facto standard for Cyrillic on the Internet. Andrew A. Chernov did a tremendous amount of work to make a standard in this area. He is an author of RFC 1489 (*"Registration of a Cyrillic Character Set"*).

These two standards differ only in positions of the cyrillic characters in the table (that is in cyrillic character codes).

The principal difference is that the Alt codeset is used by MS-DOS users only, whereas KOI8-R is used in Unix, as well as in MS-DOS (though in the latter KOI8-R is much less popular). Since we are doing

the right thing (namely working in the Unix operating system), we shall focuse mostly on KOI8-R.

As for the ISO standard, it is more popular in Europe and the US as a standard for Cyrillic. The leader in Russia is definitely KOI8-R.

There are other standards, which are different from ASCII and much more flexible. *Unicode* is most known. However, they are not implemented as good as the basic ones in Unix in general and Linux in particular. Therefore, I am not describing them here.

---

3. Preparing your environment

Before we start customizing various parts of the system functionality, we have to set up a couple basic things. Most of tools described below assume that there are Cyrillic fonts available and a user is able to input Cyrillic characters. To make it true we have to configure the environment to provide both fonts and input facility for Cyrillic.

There are effectively two interface models supported by Linux. One is the text mode, and the other one is the graphic mode, provided by the X Window System. Both require different setup, which will be described below.

3.1 Text mode setup

Generally, the text mode setup is the easiest way to show and input Cyrillic characters. There is one significant complication, however: the text mode fonts and keyboard layout manipulations depend on terminal driver implementation. Therefore, there is no portable way to achieve the goal across different systems.

Right now, I describe the way to deal with the Linux console driver. Thus, if you have another system, don't expect it to work for you. Instead, consult your terminal driver manual. Nevertheless, send me any information you find, so I'll be able to include it in further versions of this document.

Linux Console

The Linux console driver is quite a flexible piece of software. It is capable of changing fonts as well as keyboard layouts. To achieve it, you'll need the kbd package. Both RedHat and Slackware install kbd as part of a system.

The kbd package contains keyboard control utilities as well as a big collection of fonts and keyboard layouts.

Cyrillic setup with **kbd** usually involves two things:

1. Screen font setup. This is performed by the `setfont` program. The fonts files are located in `/usr/lib/kbd/consolefonts`. **NOTE:** Never run the `setfont` program under X because it will hang your system. This is because it works with low-level video card calls which X doesn't like.
2. Load the appropriate keyboard layout with the `loadkeys` program.

NOTE: In RedHat 3.0.3, `/usr/bin/loadkeys` has too restrictive access permissions, namely 700 (`rwx------`). There are no reasons for that, since everyone may compile his own copy and execute it (the appropriate system calls are not root-only). Thus, just ask your sysadmin to set more reasonable permissions for it (for example, 755).

The following is an excerpt from my `cyrload` script, which sets up the Cyrillic mode for Linux console:

```
if [ notset.$DISPLAY != notset. ]; then
    echo "`basename $0`:  cannot run under X"
    exit
fi

loadkeys /usr/lib/kbd/keytables/ru.map
setfont /usr/lib/kbd/consolefonts/Cyr_a8x16
mapscrn /usr/lib/kbd/consoletrans/koi2alt
echo -ne "\033(K"              # the magic sequence
echo "Use the right Ctrl key to switch the mode..."
```

Let me explain it a bit. You load the appropriate keyboard mapping. Then you load a font corresponding to the *Alt* codeset. Then, in order to be able to display text in *KOI8-R* correctly, you load a *screen translation table*. What it does is a translation of *some* characters from the upper half of the codeset to the *Alt* encoding. The word 'some' is crucial here - not all characters get translated, therefore some of them, like IBM pseudographic characters get unmodified to the screen and display correctly, since they are compatible with the *Alt* codeset, as opposed to *KOI8-R*. To ensure this, run **mc** and pretend you are back to MS-DOS 3.3...

Finally, the magic sequence is important but I have no idea what on the Earth it does. I stole/borrowed/learned it from German HOWTO back in 1994, when it was like the only national language oriented HOWTO. *If you have any idea about this magic sequence, please tell me*.

Finally, for those purists, who don't wont to give the *Alt* codeset a chance, I'm attaching yet another version of the script above, using native *KOI8-R* fonts.

```
if [ notset.$DISPLAY != notset. ]; then
    echo "`basename $0`:  cannot run under X"
    exit
fi

loadkeys /usr/lib/kbd/keytables/ru.map
setfont /usr/lib/kbd/consolefonts/koi-8x16
echo "Use the right Ctrl key to switch the mode..."
```

However, don't expect nice borders in your text mode-based windowing applications.

Now you probably want to test it. Do the appropriate bash or tcsh setup, rerun it, then press the right `Control` key and make sure you are getting the cyrillic characters right. The '`q`' key must produce russian "`short i`" character, '`w`' generates "`ts`", etc.

If you've screwed something up, the very best thing to do is to reset to the original (that is, US) settings. Execute the following commands:

```
loadkeys /usr/lib/kbd/keytables/defkeymap.map
setfont /usr/lib/kbd/consolefonts/default8x16
```

NOTE: unfortunately enough, the console driver is not able to preserve it's state (at least easily enough),

while running the X Window System. Therefore, after you leave the X (or switch from it to a console), have to reload the console russian font.

TOC

FreeBSD Console

I am not using FreeBSD so I couldn't test the following information. All data in this section should be treated as just pointers to begin with. The FreeBSD project homepage may have some information on the subject. Another good source is the relcom.fido.ru.unix newsgroup. Also, check the resources listed in section resources .

Anyway, this is what Ilya K. Orehov suggests to do in order to make FreeBSD console speak Russian:

1. In `/etc/sysconfig` add:

   ```
   keymap=ru.koi8-r
   keyrate=fast
   # NOTE: '^[' below is a single control character
   keychange="61 ^[[K"
   cursor=destructive
   scrnmap=koi8-r2cp866
   font8x16=cp866b-8x16
   font8x14=cp866-8x14
   font8x8=cp866-8x8
   ```
2. In `/etc/csh.login`:

   ```
   setenv ENABLE_STARTUP_LOCALE
   setenv LANG ru_SU.KOI8-R
   setenv LESSCHARSET latin1
   ```
3. Make analogous changes in `/etc/profile`

3.2 The X Window System

Like the console mode, the X environment also requires some setup. This involves setting up the input mode and the X fonts. Both are being discussed below.

The X fonts.

First of all, you have to obtain the fonts having the Cyrillic glyphs at the appropriate positions.

If you are using the most recent X (or XFree86) distribution, chances are, that you already have such fonts. In the late 1995, the X Window System incorporated a set of Cyrillic fonts, created by Cronyx. Ask your system administrator, or, if *you* are the one, check your system, namely:

1. Run '`xlsfonts | grep koi8`'. If there are fonts listed, your X server is already aware about the fonts.

TOC

2. Otherwise, run

   ```
   find -name crox\*.pcf\*
   ```

   to find the location of the Cyrillic fonts in the system. You'll have to `enable` those fonts to the X server, as I explain below.

If you haven't found such fonts installed, you'll have to do it yourself.

There is some ambiguity with the fonts. XFree86 docs claim that the russian fonts collection included in the distribution is developed by Cronyx. Nevertheless, you may find another set of Cronyx Cyrillic fonts on the net (eg. on ftp.kiae.su), known as the **xrus** package (don't confuse it with the `xrus` program, which is used to setup a Cyrillic keyboard layout. Hopefully, tha letter one was renamed to **xruskb** recently). **Xrus** has fewer fonts than the collection in Xfree86 (38 vs 68), but the latter one didn't go along with my Netscape setup - it gave me some really huge font in the menubar. The **xrus** package doesn't have this problem.

I would suggest you to download and try both of them. Pick up the one which you'll like more. Also, I'm going to creat RPM packages soon for both collections and download them to ftp.redhat.com.

There are also older stuff, for example the **vakufonts** package, created by Serge Vakulenko, which was the base for the one in the X distribution. There are also a number of others. The important point is that the fonts' names in the old collection were not strictly conforming to the standard. The latter is fine in general, but sometimes it may cause various weird errors. For example, I had a bad experience with Maple V for Linux, which crashed mysteriously with the **vakufonts** package, but ran smoothly with the "standard" ones.

So, let's start with the fonts:

1. Download the appropriate fonts collection. The package for XFree86 may be found at any FTP site, containing the X distribution, for example, directly from the XFree86 FTP site. The **xrus** package may be found on ftp.kiae.su
2. Now when you have the fonts, you create some directory for them. It is generally a bad idea to put new fonts to the already existing font directory. So, place them, to, say, `/usr/lib/X11/fonts/cyrillic` for a system-wide setup, or just create a private directory for personal use.
3. If the new fonts are in BDF format (`*.bdf` files), you have to compile them. For each font do:

   ```
   bdftopcf -o <font>.pcf <font>.bdf
   ```

   If your server supports compressed fonts, do it, using the *compress* program:

   ```
   compress *.pcf
   ```

   Also, if you do want to put the new fonts to an already existing font directory. you have to concatenate the old and the new files named `fonts.alias` in the case both of them exist.
4. Each font directory in the X must contain a list of fonts in it. This list is stored in the file `fonts.dir`. You don't have to create this list manually. Instead, do:

   ```
   cd <new font directory>
   mkfontdir .
   ```

TOC

5. Now you have to make this font directory known to the X server. Here, you have a number of options:
   - System-wide setup for XFree86. If you are running this version of X, then append the new directory to the list of directories in the file `XF86Config`. To find the location of this file, see output of `startx`. Also, see **XF86Config(4/5)** for details.
   - System-wide setup through `xinit`. Add the new directory to the `xinit` startup file. See **xinit(1x)** and the next option for details.
   - Personal setup. You have a special start-up file for the X - `~/.xinitrc` (or `~/.Xclients`, or `~/.xsession` for the RedHat users). Add the following commands to it:

   ```
   xset +fp <new font directory>
   xset fp rehash
   ```

   It is important to note that '`+fp`' means that the new fonts will be added to the head of the font path list. That is, if an application requests say a `fixed` font, it'll be given the one with Cyrillic characters, which is definitely what we are trying to achieve. There are problems, though. The `fixed` font in the cyrillic fonts distribution doesn't have it's bold and italic counterparts. My font of choice is `6x13`, so, since it also lacks bold and italic typefaces, I cannot use Emacs/XEmacs faces in their full glory. Hopefully somebody will ultimately create those fonts and the situation will change.
6. Now restart your X. If you have done everything right, the tests in the beginning of the section will be successful. Also, play with **xfontsel(1x)** to make sure you are able to select the cyrillic fonts.

In order to make the X clients use the Cyrillic fonts, you have to set up the appropriate X resources. For example, I make the russian font the default one in my `~/.Xdefaults`:

```
*font:          6x13
```

Since my cyrillic fonts are first in the font path (see output of '`xset q`'), the font above is taken from the "cyrillic" directory.

This just a simple case. If you want to set the appropriate part of the X client to a cyrillic font, you have to figure out the name of the resource (eg. using **editres(1x)**) and to specify it either in the resource database, or in the command line. Here go some examples:

```
$ xterm -font '-cronyx-*-bold-*-*-*-19-*-*-*-*-*-*-*'
```

...will run xterm with some ugly font; and

```
$ xfontsel -xrm '*quitButton.font: -*-times-*-*-*-*-13-*-*-*-*-*-koi8-*'
```

...will set a Cyrillic Times font for the **Quit** button in `xfontsel`.

The input translation

In the newest X releases (X11R61 and higher) there are two "standard" input methods: the original one, working through the **xmodmap** utility, and the new one called *Xkb* (X KeyBoard). The very first thing you have to do is **to disable the Xkb method!** Don't get charmed by it's ability to set up a "russian keyboard".

TOC

It looks like this method is using the Cyrillic keysyms defined in `keysymdef.h`. This file defines keysyms for many languages. The only problem is that those definitions have nothing to do with the extended ASCII codeset - the one most programs are only able to operate with! I hardly know any programs being able to grok the `keysymdef.h` keysyms, different from 8-bit ASCII. However our goal is to get the KOI8-R support to work.

To disable the `Xkb` support, browse through the `Keyboard` section of your `XF86Config` file and comment all lines starting with *Xkb* (case doesn't matter). Instead, put the following line:

```
XkbDisable
```

The `xmodmap` program.allows customization of codes emitted by various characters and their combinations. It sets the things up based on the file containing the translation table.

In the previous versions of this document I used to describe the `xmodmap`-based setup in a great detail. This proved to be almost useless. The `Xmodmap`-based input translation method is well known as being it is non-portable, inflexible, and incomplete. Your configuration may work with one XFree version and fail with a different one. Even worse, sometimes things differ accross different servers in the same distribution.

I strongly suggest you not to play with this `xmodmap`, at least for now. Apart from headache and disappointment you'll gain nothing. Instead, I recommend installing the xruskb package, which allows you to configure most of the input translation parameters without having to know about `xmodmap`. Again, the RedHat Linux users are free to download and install an RPM package.

3.3 First steps - Cyrillic in shells

3.4 bash

Three variables should be set on order to make `bash` understand the 8-bit characters. The best place is `~/.inputrc` file. The following should be set:

```
set meta-flag on
set convert-meta off
set output-meta on
```

3.5 csh/tcsh

The following should be set in `.cshrc`:

```
setenv LC_CTYPE iso_8859_5
stty pass8
```

If you don't have the POSIX `stty` (impossible for Linux), then replace the last call to the following:

```
stty -istrip cs8
```

TOC

3.6 ksh

As for the public domain `ksh` implementation - `pdksh 5.1.3`, you can input 8 bit characters only in `vi` input mode. Use:

```
set -o vi
```

3.7 less

So far, `less` doesn't support the KOI8-R character set, but the following environment variable will do the job:

```
LESSCHARSET=latin1
```

3.8 mc (The Midnight Commander)

To display Cyrillic text correctly, select the *full 8 bits* item in the **Options/Display** menu.

If your problem is the ugly windows' borders, consult the linux-console section.

As an off-topic, if you want to make **mc** use color in an `Xterm` window, set the variable `COLORTERM`:

```
COLORTERM= ; export COLORTERM
```

3.9 rlogin

Make sure that the shell on the destination site is properly set up. Then, if your `rlogin` doesn't work by default, use '`rlogin -8`'.

3.10 zsh

Use the same way as with `csh` (see section csh). The startup files in this case are `.zshrc` or `/etc/zshrc`.

---

4. Editing text

In this section I'll describe how to customize various text editors to work with Cyrillic text. This doesn't cover the *word processors*, which will be described later (see section <@@ref>word-processingword-processing).

4.1 Emacs and XEmacs

There are two version of the Emacs editor - **GNU Emacs** and **XEmacs**. While they provide more or less same functionality, some implementation details are significantly different. Cyrillic setup requires some low-level (in Emacs Lisp sense) tweaking, and it differs a bit for those two versions.

NOTE: Apart from the setup described here, there is an alternative way to configure both versions of emacs - use **MULE** (MULtilanguage Emacs support). The latter way is fairly complicated and (to the best of my knowledge) rarely used, so I don't discuss it here.

The minimal cyrillic support in **GNU emacs** (you don't have to do it for the **XEmacs**) is done by adding the following calls to one's `.emacs` (provided that the Cyrillic character set support is installed for console or X respectively):

```
(standard-display-european t)

(set-input-mode (car (current-input-mode))
   (nth 1 (current-input-mode))
   0)
```

This allows the user to view and input documents in Russian.

However, it isn't enough. Emacs doesn't know yet, that Cyrililic characters may constitute a word, let alon the upper/lower case conversion rules. In order to teach Emacs doing that, you have to modify the syntax and case tables of emacs:

```
(require 'case-table)

(let* ((ruc
"\341\342\367\347\344\345\263\366\372\351\352\353\354\355\356\357\360\362\363\364\365\346\350\343\376\373\375\370\371\377\374\340\361")
       (rlc
"\301\302\327\307\304\305\243\326\332\311\312\313\314\315\316\317\320\322\323\324\325\306\310\303\336\333\335\330\331\337\334\300\321")
       (i 0)
       (len (length ruc)))
  (while (< i len)
    (modify-syntax-entry (elt ruc i) "w  ")
    (modify-syntax-entry (elt rlc i) "w  ")
    (set-case-syntax-pair (elt ruc i) (elt rlc i) (standard-case-table))
    (setq i (+ i 1))))
```

For this purpose I created a `rusup.el` file which does this, as well as a couple handy functions. You have to load it in your `~/.emacs`.

Finally, the russian.el package by Valery Alexeev (`valery@math.uga.edu`) allows the user to switch between cyrillic and regular input mode and to translate the contents of a buffer from one Cyrillic coding standard to another (which is especially useful while reading the texts imported from MS-DOS or Windows).

4.2 Using vi

The **vi** editor (at least it's clone **vim**, available in most Linux distributions) is aware of 8-bit characters. It will allow you to enter cyrillic characters and will be able to recognize the word boundaries correctly. I don't know about the upper-/lower-case conversion rules, since I don't use **vi** much. *If you know something about it, please inform me*.

4.3 Editing text with joe

Joe requires a special `-asis` option to recognize 8-bit characters. You may either specify this option at the command line, or to put it in `~/.joerc` file (for personal use, or in `/usr/lib/joerc` for system-wide setup.

If your program doesn't understand `-asis` option, you have to upgrade to the newer version.

However, **joe** doesn't seem to understand the cyrillic words' boundaries correctly. I assume, that it applies both to the case conversion rules.

4.4 Spell-checking Russian

The program I use to spell-check text is the **GNU ispell**. It is very flexible and extensible, so it is possible to use it to spell-check text in languages, other than English, by adding new *spell dictionaries*.

Constantine Knizhnik has created a very good Russian dictionary for **ispell**. You may find it at his homepage. The distribution includes a handy incremental spelling script for **emacs**.

Ideally, if you already have an **ispell** properly installed, you have to just step into the newly-created directory and generate the dictionary, using the commands provided in the `Makefile`. However, chances are quite high, that you'll see a lot of complaints about the **ispell**'s unawareness of the 8-bit data. This is because in most distributions, **ispell** is compiled without 8-bit data support. In this case, you cannot avoid recompiling the **ispell** package.

Again, RedHat users will be delighted to know that I've rebuilt the **ispell** package with both Russian and German dictionaries. As usual, you may grab it from the RedHat FTP site.

Once you have everything installed, you may invoke Russian spell-check, by supplying `'-d russian'` option to **ispell**.

Now, if you use **Emacs**, you may want to add a menu item for a russian dictionary. I sent a proposed menu entry to the `ispell.el` maintainer and he kindly agreed to include it in the the next public release of the file. Meanwhile, you may do it by adding the following code in your `~/.emacs` (or in `/usr/share/emacs/site-lisp/site-start.el` for a system-wide setup):

```
(setq ispell-dictionary-alist
  (append ispell-dictionary-alist
          '(("russian"

"[\341\342\367\347\344\345\263\366\372\351\352\353\354\355\356\357\360\362\363\364\365\346\350\343\376\373\375\370\371\377\374\340\361\301\302\327\307\304\305\243\326\332\311\312\313\314\315\316\317\320\322\323\324\325\306\310\303\336\333\335\330\331\337\334\300\321]"

"[^\341\342\367\347\344\345\263\366\372\351\352\353\354\355\356\357\360\362\363\364\365\346\350\343\376\373\375\370\371\377\374\340\361\301\302\327\307\304\305\243\326\332\311\312\313\314\315\316\317\320\322\323\324\325\306\310\303\336\333\335\330\331\337\334\300\321]"
              "']" t ("-C" "-d" "russian") "~latin1"))))

(define-key-after ispell-menu-map [ispell-select-russian]
  '("Select Russian (KOI-8)" . (lambda ()
                                 (interactive)
                                 (ispell-change-dictionary "russian")))
  'british)
```

Unfortunately, it won't work for the **XEmacs**. I'll try to solve this problem later.

5. Using Cyrillic with mail and news

Setting up your mail and news software to recognize Cyrillic text is not very difficult, although you have to possess some knowledge of principles, mail and news work by.

Internet electronic mail software generally consists of two parts: **MUA** (Mail User Agent) and **MTA** (Mail Transfer Agent). MUA is the program you use to read, compose, and send mail. However, MUA doesn't transfer mail messages by itself. Instead, it calls the MTA, which is reponsible to send message using an appropriate protocol to the appropriate direction. For example, your MUA may be **Pine** and MTA - **qmail**.

Until quite recently, both MTA and MUA weren't 8-bit clean by default. Therefore, whenever you sent your message from say America to Russia, you were never sure, that some intermediate MTA won't strip the 8th bit from each character of your message. Therefore, a set of protocols was developed, which allowed encoding various kinds of data using only printable characters from 7-bit ASCII. This family of protocols is called **MIME** (MultimedIa Mail Encoding).

Since MIME is usually pre-configured to reasonable defaults, we won't describe it here. We will talk more about MIME when we provide a backward compatibility with other Cyrillic encodings (section mime).

Meanwhile, we start MUA setup, because it is usually up to an end-user. Then, we will describe the basic priciples of the MTA configuration for Cyrillic.

5.1 Setting up Mail User Agents

Emacs-based mail readers

Basically, you don't need any special setup for Emacs-based readers, geivedn, that you've already configured the emacs itself (see section emacs).

pine

Set the following directive in `~/.pinerc` for personal configuration, or in `/usr/lib/pine.conf` for a global one:

```
character-set=ISO-8859-5
```

5.2 Configuring your MTA

TOC

There are a number of MTAs available now. These include **sendmail**, **qmail**, **smail**, **exim**, and others.

sendmail

So far, **sendmail** is much more popular than other MTAs, because it's long history and widespread use. Personally, I hate this program - it is a perfect example of a completely moronic design and even it's "improvements" with the passion of time show, that this approach is not going to cease. Any system administrator shudders, when he hears the ominous "`sendmail.cf`" name...

As of now, **sendmail** doesn't strip the 8th bit anymore. However, it may *encode* the 8-bit data using a special *base64* encoding. Although most MUAs are supposed to recognize it and decode it back to a regular data, you may want to start with sending raw 8-bit text to make sure everything works.

As of version 8, **sendmail** handles 8-bit data correctly by default. If it doesn't do it for you, check the `EightBitMode` option and option `7` given to mailers in your `/etc/sendmail.cf`. See *"Sendmail. Operation and Installation Guide"* for details.

Other MTAs

I don't know much about other MTAs. If you know something, which may be important for Cyrillic setup, please inform me.

---

6. Browsing the Cyrillic Web

Unlike e-mail and news, there is no definitive standard for Cyrillic encoding for the Web. This is primarily because Microsoft offers Web authoring tools, which only allow *cp1251* codeset for Cyrillic, completely ignoring the fact that any other standards may already exist.

The setup described here is very basic. It will allow you to view pages in the *KOI8-R* codeset. If the situation improves, I'll add more information.

6.1 lynx

As of version 2.6, you may select the appropriate encoding for the `display Character set` option.

6.2 Netscape navigator

Make sure you are using `Netscape` version higher than 3. If your `Netscape` is older, download a new one from www.netscape.com.

Basic setup

To be able to see Cyrillic text in most parts of the HTML document, do the following:

- In menu **Options/Document Encoding** select **Cyrillic(KOI-8)**.
- In menu **Options/General Preferences/Fonts** select **Cyrillic (KOI-8)** encoding, **Times(Cronyx)** as a proportional font and **Courier(Cronyx)** as a fixed one.
- save options.

NOTE: This setup will work with most parts of the document. However, you won't be able to display Cyrillic text in the window header, menus and some controls. Attempts to fix it follows.

Cyrillic text in frames and input areas

To fix this, it is usually enough to:

1. Copy the Netscape properties database (usually `Netscape.ad`) to `~/Netscape`.
2. In the latter file, set the following property:

```
*documentFonts.charset*iso8859-1:          koi8-r
```

This will force all frame and input elements to use the fonts with *koi8-r* encoding instead of the default ones, therefore you have to make sure you have installed such fonts (see section xfonts).

The bad news about the trick above is that if you load a document which is supposed to be displayed in `iso-8859-1` fonts, it will be displayed using the `koi8` fonts instead. Sometimes such documents will look worse.

Advanced setup

Andrew A. Chernov is the one, who knows more than others about KOI-8 in general and netscape in particular. Visit his excellent KOI-8 page and download a patch for Netscape resource file, making Netscape speak Russian as much as it is able to.

---

7. Cyrillic wordprocessing

7.1 TeX-based environments

In this section I'll describe several ways to make TeX and LaTeX typeset Cyrillic texts. There are several ways, which differ in setup sophistication and usage convenience. For example, one possibility is to start without any preliminary setup and use the *Washington AMSTeX Cyrillic fonts*. On the other hand, you may install a LaTeX package, providing a very high degree of Cyrillic setup. I have an experience with two such packages. One is the `cmcyralt` package by Vadim V. Zhytnikov (`vvzhy@phy.ncu.edu.tw`) and Alexander Harin (`harin@lourie.und.ac.za`), and the other one is the `LH` package by the *CyrTUG* group with styles and hyphenation for LaTeX2e by Sergei O. Naoumov (`serge@astro.unc.edu`). I'll describe both.

Note, that there are two versions of LaTeX available - 2.09 is the old one, while 2e is a new pre-3.0 release. If you are using LaTeX 2.09, then switch quickly to the 2e. The latter retains compatibility with the old one, but has much more features. Hopefully, version 3 will be released soon. I describe a LaTeX 2e setup.

Also, both of these packages require the Cyrillic text to be typeset using the *Alt* codeset, not *KOI8-R*! This is caused by historical reasons, since the creators of these packages used to work with `EmTeX` - the MS-DOG version of TeX (they didn't know about Linux yet :-). Switching to the *KOI8-R* requires some effort and is being expected to be done soon. So far, use some utility to convert your russian text from *KOI8-R* to *Alt*. See section <@@ref>user-toolsuser-tools .

Using the Washington Cyrillic

This package was created for the American Mathematic Society to provide documents with Russian references. Therefore, the authors were not very careful and the fonts look quite clumsy. This package is usually referred to as a `"really bad cyrillic package for TeX"`.

Nevertheless, we'll discuss it, because it is very easy to use and doesn't require any setup - this collection is supplied with most of TeX distributions.

Of course, you won't be able to use such luxury as automatic hyphenation, but anyway...

1. Prepend your document with the following directives:

```
\input cyracc.def
\font\tencyr=wncyr10
\def\cyr{\tencyr\cyracc}
```

2. Now to type a cyrillic letter, you enter

```
\cyr
```

and use a corresponding latin letter or a TeX command. Thus, the lower case of the Russian alphabet is expressed by the following codes:

```
a b v g d e \"e zh z i {\u i} k l m n o p r s t u f kh c ch sh shch
{\cprime} y {\cdprime} \`e yu ya
```

It is extremely inconvenient to convert your Russian texts to such encoding, but you can automate the process. The translit program (section <@@ref>user-toolsuser-tools) supports a TeX output option.

KOI-8 package for teTeX

There is some new teTeX-rus package. It is reported to support KOI-8 character set and have all basic stuff required for TeX and LaTeX. I personally haven't tried it yes, although I heard about it's successfull usage.

NOTE: This package requires you to reconfigure and rebuild some parts of your **teTeX** package (for example the precompiled LaTeX macros). **Unless you know what you are doing, you shouldn't try it without necessary care. Otherwise, you may be better off by borrowing the precompiled parts fron somebody on the net**

Using the cmcyralt package for LaTeX

The `cmcyralt` package can be found on any CTAN (Comprehensive TeX Archive Network) site like `ftp.dante.de`. You should obtain two pieces: the fonts collection from `fonts/cmcyralt` and the styles and hyphenation rules from `macros/latex/contrib/others/cmcyralt`.

Note: Make sure you have the `Sauter` package installed, since `cmcyralt` requires some fonts from it. You can get this package from CTAN site as well.

Now you should do the following:

1. Put the new fonts to the TeX fonts tree. On my system (Slackware 2.2) I created a `cmcyralt` directory in the `/usr/lib/texmf/fonts/cm/`. Create the `src`, `tfm`, and `vf` subdirectories in it. Put there `.mf`, `.tfm`, and `vf` files respectively.
2. Put the font driver files (`*.fd`) from the styles archive to the appropriate place (in my case it was `/usr/lib/texmf/tex/latex/fd`).
3. Put the style files (`*.sty`) to the appropriate LaTeX styles directory (in my case `/usr/lib/texmf/tex/latex/sty`).

Now the hyphenation setup. This requires to remake the LaTeX base file.

1. The file `hyphen.cfg` contains the directives for both English and Russian hyphenation. Extract the one for Russian and place it to the LaTeX hyphenation config file `lthyphen.ltx`. In my case, that file was in `/usr/lib/texmf/tex/latex/latex-base`.
2. Put the `rhyphen.tex` to the same directory. It is needed for making the new base file. Later, you can remove it.

3. Do 'make' in that directory. Don't for get to make a link from Makefile to Makefile.ur
   During the make process check the output. There should be a message:

   ```
   Loading hyphenation patterns for Russian.
   ```

   If everything goes OK, you will get the new latex.fmt in that directory. Put it to the appropriate place, where the previous one was (like /usr/lib/texmf/ini/). **Don't forget to save the previous one!**.

This is it. The installation is complete. Try processing the examples found in the styles archive. If you are to create the PostScript files without any problems, then everything is OK. Now, to use Cyrillic in LaTeX, prepend your document with the following directive:

```
\usepackage{cmcyralt}
```

For more details, see the README file in the cmcyralt styles archive.

Note: if you do have problems with the examples, provided you have installed the things right, then probably your TeX system hasn't been installed correctly. For example, during my first try, every attempt to create the .pk files for the russian fonts failed (MakeTeXPK stage). A substantial investigation discovered some implicit conflict between the *localfont* and *ljfour* METAFONT configurations. It used to work before, but kept crashing after the cmcyralt installation. Contact your local TeX guru - TeX is very (sometimes too much) complicated to reconfigure it without any prior knowledge.

Using the CyrTUG package

You can obtain the CyrTUG package from the SunSite archive. Get the files CyrTUGfonts.tar.gz, CyrTUGmacro.tar.gz, and hyphen.tar.Z.

The process of installation doesn't differ from too much the previous one.

7.2 The StarOffice suite

Youri Kovalenko (http://www.inp.nsk.su/~kovalenko) has compiled a concise summary on StarOffice russification. It is located at ftp://sky.inp.nsk.su/archives_src/linux/StarOffice/russification.txt. I never had a chance to try it, so I cannot say anything about it's correctness.

Another source of information on the subject is compiled by Eugene Demidov (mailto:jack@gpi.ru) and is located at ftp://ftp.kapella.gpi.ru/pub/cyrillic/psfonts/README.

---

8. Printing and PostScript

8.1 Text to PostScript conversion

Sometimes you have just a plain ASCII KOI8-R text and you want to print it just to get it on the paper. One of the easiest ways to achieve that is to use special programs converting text to PostScript.

There are a number of programs doing such conversion. I personally prefer a2ps. Originally developed as a simple text-to-PostScript converter it became a big and highly configurable program with many options and allows you to manage various page layouts, syntax highlighting etc. Another tool (now available as a part of the *GNU* project) is enscript.

An a2ps converter

A text to PostScript converter has been around for a while and is one of the most versatile printing tools. The author proved to be very open to suggestions, so since the release 4.9.8 **a2ps** supports Cyrillic right off-the-shelf. All you need is a PostScript printer.

The command I use is:

```
a2ps -X koi8r --print-anyway  <file>
```

The GNU enscript

The GNU **enscript** program is also designed for converting text to PostScript and it also has a non-ASCII codeset support. It doesn't have Cyrillic PostScript fonts, but it is very easy to get them, as will be explained below (thanks to Michael Van Canneyt):

1. Install the newest **enscript**. As of now, the most recent release is 1.5. You may either get the one from the GNU FTP archive, or take an RPM package from the Redhat site.
2. Now, if you are a lucky RedHat Linux user, download and install Cyrillic Textbook font.
3. If you don't use RPM, download a file `textbook.tar.gz` from the Cyrillic Software collection on sunsite.unc.edu. Extract it to a directory, where **enscript** fonts are located (usually `/usr/share/enscript`). Now change to that directory and run the following command:

   ```
   mkafmmap *.afm
   ```
4. The setup is finished. Try to print some text in KOI8-R Cyrillic with the following command:

   ```
   enscript --font=Textbook8 --encoding=koi8 some.file
   ```

If you want a really quick and dirty solution and you don't care about the output quality and all you need

is just Cyrillic on the paper, try the rtxt2ps package. It is a very simple no-frills text-to-PostScript conversion program. The output quality is not very good (or, to be honest, just *bad*) but it does it's job.

8.2 Text to TeX conversion

If all you need is just to print an ASCII text without any additional word processing, you may try to use some programs, which would convert your Cyrillic text to a ready-to-process TeX file. One of the best programs for such purposes is **translit** (see section conversion). In this case, you don't even have to bother about installing the Cyrillic fonts for TeX, since **translit** uses a *Washington Cyrillic* package, which is included in most TeX distributions (or am I wrong?)

---

9. Cyrillic in PostScript

Experts say PostScript is easy. I cannot judge - I've got too many things to learn to spare some time to learn PostScript. So I'll try to use my sad experience with it. **I'll appreciate any feedback from you guys who know more on the subject than I do** (approx. 99% of the Earth population).

Basically, in order to print a Cyrillic text using PostScript, you have to make sure about the following things:

- Cyrillic font is *loaded* or included in the document.
- Cyrillic text is included in the document.
- Cyrillic text uses the appropriate character codes which correspond to the font's requirements.
- An appropriate font is *selected* in order to print Cyrillic text.

There is no solution general enough to be recommended as an ultimate treatment. I'll try to outline various ways to cope with different problems related to the subject.

One way to address Cyrillic setup problems generally enough is to use Ghostscript. **Ghostscript** (or just **gs** in the newspeak) is a free (well quasi-free) PostScript interpreter. It has many advantages; among them:

- Ability to run on many platforms (various Unices, Windows etc)
- Support for a wide number of non-PostScript printers
- Good degree of configurability

What is important in our particular case, is that once **Ghostscript** is set up, we can do all printing through it, thus eliminating extra setup for other PostScript devices (for example *HP LaserJet IV*)

9.1 Adding Cyrillic fonts to Ghostscript

This is important, since you probably don't want to put a responsibility to other programs to insert Cyrillic fonts in the PostScript output. Instead, you add them to **gs** and just make the programs generate Cyrillic output compatible with the fonts.

To add a new font (in `pfa` or `pfb` form) in **gs**, you have to:

1. Put it in the **gs** fonts directory (ie. `/usr/lib/ghostscript/fonts`).
2. Add the appropriate names and aliases for the font in the `Fontmap` file in the **gs** directory.

Recently a decent set of Cyrillic fonts for **GhostScript** appeared. It is located in ftp.kapella.gpi.ru. This one even has a necessary part to add to the `Fontmap` file. You have to download the contents of the `/pub/cyrillic/psfonts` directory. The `README` file describes the necessary details.

10. Print setup

Printing is always tricky. There are different printers from different vendors with different facilities. Even for a native printing there is no uniform solution (this applies not only to UNIX, but to other operating systems as well.

Printers have different control languages and often they have very different views on foreign language support. The good news is that on control language seems to be recognized as a de-facto standard for print job description - it is a PostScript language developed by Adobe Corporation.

Another problem is a variety of requirements to the print services. For example, sometimes you want just to print a piece if C program, containing comments in Russian, so you don't need any pretty-printing - just a raw ASCII output in a single font. Another time, when you design a postcard for your girlfriend, you'll probably need to typeset some document with different fonts etc. This will definitely require more effort to setup Cyrillic support.

To accomplish the former task you just have to make your printer understand *one* Cyrillic font and (maybe) install some filter program to generate data in appropriate format. To accomplish the latter one, you have to teach your printer different fonts and have a special software.

There is also something in the middle, when you get a program which knows how to generate both the fonts and the appropriate printer input, so you can say do some aource code pretty-printing without sophisticated word processing systems.

All these options will be more or less covered below.

10.1 Pre-loading Cyrillic fonts into a non-PostScript printer

If you have a good old dot matrix printer and all you need is to print a raw KOI8-R text, try the following:

1. Find a proper KOI8-R font for your printer. Check out the MS-DOSish stuff on the Internet (for example the SimTel archive).
2. Learn from the manual, how to load such font into your printer and, probably, write a simple program doing that.
3. Run this program from the appropriate `rc` file at a boot time.

Thus, having Cyrillic characters in the upper part of the printer's character set will allow you to print you texts in Russian without any hussle.

Alternatively to the *KOI8-R* fonts you may try to use the *Alt* font. There are two reasons for that:

- It may be probably much easier to find an *Alt* font, since those were very widespread in the

MS-DOS culture.

- Having a proper *Alt* font will allow you to print pseudo-graphic characters as well.

However in this case, you'll have to convert your texts from *KOI8-R* to *Alt* before sending them to a printer. This is quite easy, since there are a lot of programs doing that (see <@@ref>user-toolstranslit for example), so you just have to call such program properly in the `if` field in `/etc/printcap` file. For example, with the **translit** program you may specify:

```
if=/usr/bin/translit -t koi8-alt.rus
```

See **printcap(5)** for details.

10.2 Printing with different fonts

One great way to cope with different printers and fonts is to use **TeX** (see section tex). TeX drivers handle all details, so once you make TeX understand Cyrillic fonts, you are done.

Another possibility is to use *PostScript*. I decided to devote an entire chapter postscript to the subject, since it is not simple.

Finally, there are other word processors, which have printer drivers. I never tried anything apart from TeX, so I cannot suggest anything.

---

11. Localization and Internationalization

So far, I described how to make various programs understand Cyrillic text. Basically, each program required it's own method, very different from the others. Moreover, some programs had incomplete support of languages other than English. Not to mention their inability to interact using user's mother tongue instead of English.

The problems outlined above are very pressing, since software is rarely developed for home market only. Therefore, rewriting substantial parts of software each time the new international market is approached is very ineffective; and making each program implement it's own proprietary solution for handling different languages is not a great idea in a long term either.

Therefore, a need for standardization arises. And the standard shows up.

Everything related to the problems above is divided by two basic concepts: *localization* and *internationalization*. By localization we mean making programs able to handle different language conventions for different countries. Let me give an example. The way date is printed in the United States is MM/DD/YY. In Russia however, the most popular format is DD.MM.YY. Another issues include time representation, printing numbers and currency representation format. Apart from it, one of the most important aspect of localization is defining the appropriate character classes, that is, defining which characters in the character set are language units (letters) and how they are ordered. On the other hand, localization doesn't deal with fonts.

Internationalization (or *i18n* for brevity) is supposed to solve the problems related to the ability of the program interact with the user in his native language.

Both of the concepts above had to be implemented in a standard, giving programmers a consistent way of making the programs aware of national environments.

Althogh the standard hasn't been finished yet, many parts actually have; so they can be used without much of a problem.

I am going to outline the general scheme of making the programs use the features above in a standard way. Since this deserves a separate document, I'll just try to give a very basic description and pointers to more thorough sources.

11.1 Locale

One of the main concept of the localization is a *locale*. By locale is meant a set of conventions specific to a certain language in a certain country. It is usually wrong to say that locale is just country-specific. For example, in Canada two locales can be defined - Canada/English language and Canada/French language. Moreover, Canada/English is not equivalent to UK/English or US/English, just as Canada/French is not equivalent to France/French or Switzerland/French.

TOC

How to use locale

Each locale is a special database, defining at least the following rules:

1. character classification and conversion
2. monetary values representation
3. number representation (ie. the decimal character)
4. date/time formatting

In RedHat 4.1, which I am using there are actually *two* locale databases: one for the C library (`libc`) and one for the `X` libraries. In the ideal case there should be only one locale database for everything.

To change your default locale, it is usually enough to set the `LANG` environment variable. For example, in **sh**:

```
LANG=ru_RU
export LANG
```

Sometimes, you may want to change only one aspect of the locale without affecting the others. For example, you may decide (God knows why) to stick with `ru_RU` locale, but print numbers according to the standard POSIX one. For such cases, there is a set of environment variables, which you can you to configure specific parts for the current locale. In the last exaple it would be:

```
LANG=ru_RU
LC_NUMERIC=POSIX
export LANG LC_NUMERIC
```

For the full description of those variables, see **locale(7)**.

Now let's be more Linux-specific. Unfortunately, Linux `libc` version 5.3.12, supplied with RedHat 4.1, doesn't have a russian locale. In this case one must be downloaded from the Internet (I don't know the exact address, however).

To check, locale for which languages you have, run '`locale -a`'. It will list all locale databases, available to libc.

Fortunately, Linux community is rapidly moving to the new GNU libc (`glibc` version 2, which is much more POSIX-compliant and has a proper russian locale. Next "stable" RedHat system will already use `glibc`.

As for the `X` libraries, they have their own locale database. In the version I am using (`XFree86 3.3`), there already is a russian locale database. I am not sure about the previous versions. In any case, you may check it by looking into usr/lib/X11/locale/ (on most systems). In my case, there already are subdirectories named `koi8-r` and even `iso8859-5`.

TOC

Locale-aware programming

With locale, program don't have to implement explicitly various character conversion and comparison rules, described above. Instead, they use special API which make use of the rules defined by locale. Also, it is not necessary for program to use the same locale for all rules - it is possible to handle different rules using different locales (although such technique should be strongly discouraged).

From the **setlocale(3)** manual page:

> A program may be made portable to all locales by calling `setlocale(LC_ALL, "" )` after program initialization, by using the values returned from a `localeconv()` call for locale - dependent information and by using `strcoll()` or `strxfrm()` to compare strings.

SunSoft, for example, defines 5 levels of program localization:

1. *8-bit clean* software. That is, the program calls `setlocale()`, it doesn't make any assumptions about the 8th bit of each character, it users functions from `ctype.h` and limits from `limits.h`, and it takes care about `signed/unsigned` issues. It is very important *not* to do any assumption about the character set nature and ordering. The following programming practices must be avoided:

   ```
   if (c >= 'A' && c <= 'Z') {
       ...
   ```

   Instead, macros from the `ctype.h` header file are locale-aware and should be used in all such occasions.
2. Formats, sorting methods, paper sizes. The program uses `strcoll()` and `strxfrm()` instead of `strcmp()` for strings, it uses `time()`, `localtime()`, and strftime()/ for time services, and finally, it uses `localeconv()` for a proper numbers and currency representation.
3. Visible text in message catalogs. The program must isolate all visible text in special *message catalogs*. Those map strings in English to their translation to other languages. Selection of messages in an appropriate for a particular environment language is done in a way which is completely transparent for both the program and it's user. To make use of those facilities, the program must call `gettext()` (Sun/POSIX standard), or `catgets()` (X/Open standard). For more information on that see section i18n .
4. EUC/Unicode support. At this level, the program doesn't use the `char` type. Instead it uses `wchar_t`, which defines entities big enough to contain Unicode characters. ANSI C defines this data type and an appropriate API.

For a more detaled explanation of locale, see, for example (Voropay1) or (SingleUnix).

11.2 Internationalization

While localization describes, how to adapt a program to a foreign environment, *internationalization* (or *i18n* for brevity) details the ways to make program communicate with a non-English speaking user.

Before, that was done by developing some abstraction of the messages to output from the program's code. Now, such mechanism is (more or less) standardized. And, of course, there are free implementations of it!

The GNU project has finally adopted the way of making the internationalized applications. Ulrich Drepper (`drepper@ipd.info.uni-karlsruhe.de`) developed a package `gettext`. This package is available at all GNU sites like prep.ai.mit.edu. It allows you to develop programs in the way that you can easily make them support more languages. I don't intend to describe the programming techniques, especially because the `gettext` package is delivered with excellent manual.

Request for collaboration: If you want to learn the `gettext` package and to contribute to the GNU project simultaneously; or even if you just want to contribute, then you can do it! GNU goes international, so all the utilities are being made locale-aware. The problem is to translate the messages from English to Russian (and other languages if you'd like). Basically, what one has to do is to get the special `.po` file consisting of the English messages for a certain utility and to append each message with it's equivalent in Russian. Ultimately, this will make the system speak Russian if the user wants it to! For more details and further directions contact Ulrich Drepper (drepper@ipd.info.uni-karlsruhe.de).

---

12. Staying compatible

Being standard is not the only issue. To be really nice, one has to provide the backward compatibility. In our case, this means that the configuration should be tolerant to the data created using non-standard character sets - that is the *Alt (cp866)* and *cp1251* ones. Also, we should be able to run Cyrillic programs for MS-DOS.

In most cases (except for HTTP), it is enough to provide a timely conversion of data to *KOI8-R*. When we talk about raw unstructured data, it is quite trivial - see section <@@ref>user-toolsConversion Utilities.

Another issue is the structured data. This case is more tricky. I'll try to outline the basic roadmap of fixing it.

12.1 MIME-based data compatibility

MIME is a standard for architecture-independent data representation. Originally developed for mail messages, it has now many more applications. MIME defines format, which is open to extensions and allows architecture-specific handling of data. For example, if I receive a mail message, containing a *MIME object* of the `video/mpeg` type (an encoded MPEG file), my mail reader will automatically decode it and start an MPEG player.

Most UNIX programs, offering MIME capabilities, are based on the **metamail** package, which contains a set of utilities and data files to work with MIME objects. Several configuration files (`/etc/mailcap` for global usage and `~/.mailcap` for personal setup) define rules for handling MIME object of various types.

Thus, if you receive a proper MIME data stream, containing text in one of the obsolete character sets, you may define a MIME rule to convert such text to KOI8.

Below a number of MIME rules are shown, which are supposed to handle plain text and richtext objects, using both of the obsolete codesets, discussed above. You may incorporate these rules into one of the MIME configuration files.

Note, that these rules use the **translit** package to perform the actual conversion. For more information on that program and the conversion in general see section <@@ref>user-toolsConversion Utilities.

```
text/plain; translit -t cp1251-koi8.rus < %s; test=test \
    "`echo %{charset} | tr '[A-Z]' '[a-z]'`"  = cp1251; copiousoutput

text/richtext; translit -t cp1251-koi8.rus < %s; test=test \
    "`echo %{charset} | tr '[A-Z]' '[a-z]'`"  = cp1251; copiousoutput
```

```
text/plain; translit -t alt-koi8.rus < %s; test=test \
    "`echo %{charset} | tr '[A-Z]' '[a-z]'`"  = cp866; copiousoutpu

text/richtext; translit -t alt-koi8.rus < %s; test=test \
    "`echo %{charset} | tr '[A-Z]' '[a-z]'`"  = cp866; copiousoutput

text/plain; translit -t alt-koi8.rus < %s; test=test \
    "`echo %{charset} | tr '[A-Z]' '[a-z]'`"  = alt; copiousoutput

text/richtext; translit -t alt-koi8.rus < %s; test=test \
    "`echo %{charset} | tr '[A-Z]' '[a-z]'`"  = alt; copiousoutput
```

Obviously enough, this will work for plain text data only. Binary files are supposed to handle the codeset issues themselves (at least their "parent" applications are). Therefore, if you receive a Microsoft Word document in the *cp1251* character set, the duty of providing appropriate conversion capabilities lays upon an application you use to read that document (for example Microsoft Word, or Applix Words).

Unfortunately, the real situation is not that ideal. Many application have their own idea on how to use MIME. Until recently Microsoft Mail software had a broken MIME engine. Also, the Netscape Navigator/Communicator mail client is notorious because of it's sending of mail messages, encoded in *cp1251* with the *charset=koi8-r* field in the message header and vice versa.

12.2 Explicit character set conversion

There are a lot of conversion routines for Cyrillic on the Internet. Each of them has it's own quirks and it's own degree of Cyrillic support.

In my opinion tools must be standard. In this particular case the "standard" conversion tool is **GNU recode**. Unfortunately, the version, found on the official GNU site (3.4) doesn't support Cyrillic yet (only *ISO-8859-5*). I developed a set of conversion tables for *KOI8-R*, *Alt*, and *cp1251* for **recode** and submitted them to the **recode** maintainer. He promised to provide Cyrillic support in the upcoming release. Once it happens, I'll rewrite this section to recommend **GNU recode** as the standard conversion engine for Cyrillic.

Meanwhile, I would recommend a translit package. It supports many popular codesets and is even able to produce a *TeX files (see section tex) from text in Russian. Also, RedHat users will enjoy an RPM package for translit.

For other conversion routines, Look at SovInformBureau or ftp.funet.fi. You can even use the special mode for `emacs` (see section Emacs).

12.3 Cyrillic in the DOS emulator

This seems to be the only application, which may require `Alt` Cyrillic character set. The reason is that `Alt` is native to DOS and most of DOS programs dealing with Cyrillic are `Alt`-oriented.

For the console version (`dos`) you just have to load a keyboard and screen driver. Most of DOS drivers will work fine. I personally use the **rk** driver by A. Strakhov, which works for both console and X versions of **dosemu**. Another choice is the `r` driver by V. Kurland (sorry for possible misspelling). It is perfectly customizable and supports many codesets, `Alt` and `KOI8` among them. However it won't work for the X window (at least version 1.14 I'm using).

Both drivers can be found on most Russian Internet sites, for example Kurchatov Institute FTP server.

For the X version of **dosemu** you have to provide an appropriate X font as well. Alex Bogdanov sent me such font by e-mail. It is an original `vga` font from the **dosemu** distribution, modified for the `Alt` codeset. Unfortunately I don't know who is the creator of this font and where the official site is.

To setup the font for `dosemu` you should

- Introduce this font to the X. This is described in X fonts setup.
- Introduce this font to `dosemu`. If the font just replaces the original `vga` font, then it will be recognized by default. Otherwise, you have to describe it in `/etc/dosemu.conf`:

```
# Font to use (without filename extensions). For example:
X { updatefreq 8 title "MS DOS" icon_name "xdos" font "vga-alt"}
```

Finally, you have to load a keyboard driver. Note, the you don't need a screen driver for the X window. Therefore, not all drivers will work. At least two will: `rk` by A. Strakhov, and `cyrkeyb` by Pete Kvitek.

---

13. Bibliography

1. Andrey Chernov. KOI-8. KOI-8 information and setup.
2. Ulrich Drepper. Internationalization in the GNU project. Very thorough description of a GNU approach to i18n.
3. Michael Karl Gschwind. Internationalization. Various resources on i18n.
4. Sergei Naumov. Information on Cyrillic Software. Cyrillic setup information.
5. The Open Group Single UNIX specification.
6. RFC 1489 RFC 1489
7. Alec Voropay. Localization as it is. General locale usage in Russian.

14. Summary of the various useful resources

a2ps homepage

General Linux Information

Collection of Cyrillic resources

Cyrillic resources at KIAE

Cyrillic resources at RELCOM

Cyrillic resources at FUNET

Cronyx - the creators of Cyrillic fonts for the X Window System.

Cyrillic fonts for Ghostscript and StarOffice

Cyrillic fonts for X

Ghostscript

GNU enscript

relcom.fido.ru.linux newsgoup.

relcom.fido.ru.unix newsgoup.

Russian dictionary for GNU ispell

SovInformBureau

teTeX russification package

The kbd package for Linux

The remap package for Emacs

The rtxt2ps package

The russian.el package for emacs

The translit package

The xruskb package

Useful Cyrillic packages

TOC

X fonts collections

XFree86 FTP site

DNS HOWTO

Nicolai Langfeldt `janl@math.uio.no`

v2.2, 11 February 1999

HOWTO become a totally small time DNS admin.

1. Preamble

- 1.1 Legal stuff
- 1.2 Credits and request for help.
- 1.3 Dedication

2. Introduction.

3. A caching only name server.

- 3.1 Starting named
- 3.2 Making it even better
- 3.3 Congratulations

4. A *simple* domain.

- 4.1 But first some dry theory
- 4.2 Our own domain
- 4.3 The reverse zone
- 4.4 Words of caution
- 4.5 Why reverse lookups don't work.

TOC

5. A real domain example

- 5.1 /etc/named.conf (or /var/named/named.conf)
- 5.2 /var/named/root.hints
- 5.3 /var/named/zone/127.0.0
- 5.4 /var/named/zone/land-5.com
- 5.5 /var/named/zone/206.6.177

6. Maintenance

7. Converting from version 4 to version 8

8. Questions and Answers

9. How to become a bigger time DNS admin.

---

1. Preamble

Keywords: DNS, bind, bind-4, bind-8, named, dialup, ppp, slip, isdn, Internet, domain, name, hosts, resolving, caching.

This document is part of the Linux Documentation Project.

1.1 Legal stuff

(C)opyright 1995-1999 Nicolai Langfeldt. Do not modify without amending copyright, distribute freely but retain copyright message.

1.2 Credits and request for help.

I want to thank Arnt Gulbrandsen whom I cause to suffer through the drafts to this work and whom provided many useful suggestions. I also want to thank the numerous people that have e-mailed suggestions and notes.

This will never be a finished document, please send me mail about your problems and successes, it can make this a better HOWTO. So please send comments and/or questions or money to janl@math.uio.no. If you send e-mail and want an answer please show the simple courtesy of *making sure* that the return address is correct and working. Also, **please** read the QnA section before mailing me. Another thing, I can only understand Norwegian and English.

If you want to translate this HOWTO please notify me so I can keep track of what languages it has been published in, and also I can notify you when the HOWTO has been updated.

1.3 Dedication

This HOWTO is dedicated to Anne Line Norheim Langfeldt. Though she will probably never read it since she's not that kind of girl.

2. Introduction.

What this is and isn't.

DNS is is the Domain Name System. DNS converts machine names to the IP addresses that all machines on the net have. It maps from name to address and from address to name, and some other things. This HOWTO documents how to define such mappings using a Linux system. A mapping is simply a association between two things, in this case a machine name, like `ftp.linux.org`, and the machines IP number (or address) `199.249.150.4`.

DNS is, to the uninitiated (you ;-), one of the more opaque areas of network administration. This HOWTO will try to make a few things clearer. It describes how to set up a *simple* DNS name server. Starting with a caching only server and going on to setting up a primary DNS server for a domain. For more complex setups you can check the QnA section of this document. If it's not described there you will need to *read* the Real Documentation. I'll get back to what this Real Documentation consists of in the last chapter.

Before you start on this you should configure your machine so that you can telnet in and out of it, and successfully make all kinds of connections to the net, and you should especially be able to do `telnet 127.0.0.1` and get your own machine (test it now!). You also need a good `/etc/nsswitch.conf` (or `/etc/host.conf`), `/etc/resolv.conf` and `/etc/hosts` files as a starting point, since I will not explain their function here. If you don't already have all this set up and working the NET-3-HOWTO and/or the PPP-HOWTO explains how to set it up. Read them.

When I say `your machine' I mean the machine you are trying to set up DNS on. Not any other machine you might have that's involved in your networking effort.

I assume you're not behind any kind of firewall that blocks name queries. If you are you will need a special configuration, see the section on QnA.

Name serving on Unix is done by a program called `named`. This is a part of the ``bind'' package which is coordinated by Paul Vixie for The Internet Software Consortium. `Named` is included in most Linux distributions and is usually installed as `/usr/sbin/named`. If you have a named you can probably use it; if you don't have one you can get a binary off a Linux ftp site, or get the latest and greatest source from ftp.isc.org:/isc/bind/src/cur/bind-8/. This HOWTO is about bind version 8. The old version of the HOWTO, about bind 4 is still available at http://www.math.uio.no/~janl/DNS/ in case you use bind 4. If the named man page talks about (at the very end, the FILES section) `named.conf` you have bind 8, if it talks about `named.boot` you have bind 4. If you have 4 and are security conscious you really ought to upgrade to a recent 8.

DNS is a net-wide database. Take care about what you put into it. If you put junk into it, you, and others will get junk out of it. Keep your DNS tidy and consistent and you will get good service from it. Learn to use it, admin it, debug it and you will be another good admin keeping the net from falling to it's knees by

mismanagement.

In this document I state flatly a couple of things that are not completely true (they are at least half truths though). All in the interest of simplification. Things will (probably ;-) work if you believe what I say.

Tip: Make backup copies of all the files I instruct you to change if you already have them, so if after going through this nothing works you can get it back to your old, working state.

---

3. A caching only name server.

A first stab at DNS config, very useful for dialup users.

A caching only name server will find the answer to name queries and remember the answer the next time you need it. This will shorten the waiting time the next time significantly, especially if you're on a slow connection.

First you need a file called `/etc/named.conf`. This is read when named starts. For now it should simply contain:

```
// Config file for caching only name server

options {
        directory "/var/named";

        // Uncommenting this might help if you have to go through a
        // firewall and things are not working out:

        // query-source port 53;
};

zone "." {
        type hint;
        file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "pz/127.0.0";
};
```

The `directory' line tells named where to look for files. All files named subsequently will be relative to this. Thus `pz` is a directory under `/var/named`, i.e., `/var/named/pz`. `/var/named` is the right directory according to the *Linux File system Standard*.

The file named `/var/named/root.hints` is named in this. `/var/named/root.hints` should contain this:

```
;
; There might be opening comments here if you already have this file.
; If not don't worry.
;
.                       6D IN NS        G.ROOT-SERVERS.NET.
.                       6D IN NS        J.ROOT-SERVERS.NET.
.                       6D IN NS        K.ROOT-SERVERS.NET.
.                       6D IN NS        L.ROOT-SERVERS.NET.
.                       6D IN NS        M.ROOT-SERVERS.NET.
.                       6D IN NS        A.ROOT-SERVERS.NET.
```

TOC

```
.                       6D IN NS        H.ROOT-SERVERS.NET.
.                       6D IN NS        B.ROOT-SERVERS.NET.
.                       6D IN NS        C.ROOT-SERVERS.NET.
.                       6D IN NS        D.ROOT-SERVERS.NET.
.                       6D IN NS        E.ROOT-SERVERS.NET.
.                       6D IN NS        I.ROOT-SERVERS.NET.
.                       6D IN NS        F.ROOT-SERVERS.NET.

G.ROOT-SERVERS.NET.     5w6d16h IN A    192.112.36.4
J.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.10
K.ROOT-SERVERS.NET.     5w6d16h IN A    193.0.14.129
L.ROOT-SERVERS.NET.     5w6d16h IN A    198.32.64.12
M.ROOT-SERVERS.NET.     5w6d16h IN A    202.12.27.33
A.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.4
H.ROOT-SERVERS.NET.     5w6d16h IN A    128.63.2.53
B.ROOT-SERVERS.NET.     5w6d16h IN A    128.9.0.107
C.ROOT-SERVERS.NET.     5w6d16h IN A    192.33.4.12
D.ROOT-SERVERS.NET.     5w6d16h IN A    128.8.10.90
E.ROOT-SERVERS.NET.     5w6d16h IN A    192.203.230.10
I.ROOT-SERVERS.NET.     5w6d16h IN A    192.36.148.17
F.ROOT-SERVERS.NET.     5w6d16h IN A    192.5.5.241
```

The file describes the root name servers in the world. This changes over time and must be maintained. See the maintenance section for how to keep it up to date.

The next section in `named.conf` is the last `zone`. I will explain its use in a later chapter, for now just make this a file named `127.0.0` in the subdirectory `pz`:

```
@               IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                                1       ; Serial
                                8H      ; Refresh
                                2H      ; Retry
                                1W      ; Expire
                                1D)     ; Minimum TTL
                        NS      ns.linux.bogus.
1                       PTR     localhost.
```

Next, you need a `/etc/resolv.conf` looking something like this:

```
search subdomain.your-domain.edu your-domain.edu
nameserver 127.0.0.1
```

The `search' line specifies what domains should be searched for any host names you want to connect to. The `nameserver' line specifies the address of your nameserver, in this case your own machine since that is where your named runs (127.0.0.1 is right, no matter if your machine has an other address too). If you want to list several name servers put in one `nameserver' line for each. (Note: Named never reads this file, the resolver that uses named does.)

To illustrate what this file does: If a client tries to look up `foo`, then `foo.subdomain.your-domain.edu` is tried first, then `foo.your-fomain.edu`, finally `foo`. If a client tries to look up `sunsite.unc.edu`,

TOC

`sunsite.unc.edu.subdomain.your-domain.edu` is tried first (yes, it's silly, but that's the way it works) `sunsite.unc.edu.your-domain.edu`, and finally `sunsite.unc.edu`. You may not want to put in too m domains in the search line, it takes time to search them all.

The example assumes you belong in the domain `subdomain.your-domain.edu`, your machine then, is probably called `your-machine.subdomain.your-domain.edu`. The search line should not contain your TLD (Top Level Domain, `edu' in this case). If you frequently need to connect to hosts in another domain you can add that domain to the search line like this:

```
search subdomain.your-domain.edu your-domain.edu other-domain.com
```

and so on. Obviously you need to put real domain names in instead. Please note the lack of periods at the end of the domain names. This is important, please note the lack of periods at the end of the domain names.

Next, depending on your libc version you either need to fix `/etc/nsswitch.conf` or `/etc/host.conf`. If you already have `nsswitch.conf` that's what we'll fix, if not, we'll fix `host.conf`.

/etc/nsswitch.conf

This is a long file specifying where to get different kinds of data types, from what file or database. It usually contains helpful comments at the top, which you should consider reading. After that find the line starting with `hosts:', it should read

```
hosts:      files dns
```

If there is no line starting with `hosts:' then put in the one above. It says that programs should first look in the `/etc/hosts` file, then check DNS according to `resolv.conf`.

/etc/host.conf

It probably contains several lines, one should start with `order` and it should look like this:

```
order hosts,bind
```

If there is no `order' line you should add one. It tells the host name resolving routines to first look in `/etc/hosts`, then ask the name server (which you in `resolv.conf` said is at 127.0.0.1).

3.1 Starting named

After all this it's time to start named. If you're using a dialup connection connect first. Type `ndc start', and press return, no options. If that does not work try `/usr/sbin/ndc start' instead. If that back-fires see the QnA section. If you view your syslog message file (usually called `/var/adm/messages`, but another directory to look in is `/var/log` and another file to look in is `syslog`) while starting named (do `tail -f /var/log/messages`) you should see something like:

(the lines ending in \ continue on the next line)

```
Feb 15 01:26:17 roke named[6091]: starting.  named 8.1.1 Sat Feb 14 \
```

TOC

```
  00:18:20 MET 1998 ^Ijanl@roke.uio.no:/var/tmp/bind-8.1.1/src/bin/named
Feb 15 01:26:17 roke named[6091]: cache zone "" (IN) loaded (serial 0)
Feb 15 01:26:17 roke named[6091]: master zone "0.0.127.in-addr.arpa" \
  (IN) loaded (serial 1)
Feb 15 01:26:17 roke named[6091]: listening [127.0.0.1].53 (lo)
Feb 15 01:26:17 roke named[6091]: listening [129.240.230.92].53 (ippp0)
Feb 15 01:26:17 roke named[6091]: Forwarding source address is [0.0.0.0].1040
Feb 15 01:26:17 roke named[6092]: Ready to answer queries.
```

If there are any messages about errors then there is a mistake. Named will name the file it is in (one of named.conf and root.hints I hope :-) Kill named and go back and check the file.

Now you can test your setup. Start nslookup to examine your work.

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1

>
```

If that's what you get it's working. We hope. Anything else, go back and check everything. Each time you change the `named.conf` file you need to restart named using the `ndc restart` command.

Now you can enter a query. Try looking up some machine close to you. `pat.uio.no` is close to me, at the University of Oslo:

```
> pat.uio.no
Server:  localhost
Address:  127.0.0.1

Name:    pat.uio.no
Address:  129.240.130.16
```

nslookup now asked your named to look for the machine `pat.uio.no`. It then contacted one of the name server machines named in your `root.hints` file, and asked its way from there. It might take tiny while before you get the result as it may need to search all the domains you named in `/etc/resolv.conf`.

If you ask the same again you get this:

```
> pat.uio.no
Server:  localhost
Address:  127.0.0.1

Non-authoritative answer:
Name:    pat.uio.no
Address:  129.240.2.50
```

Note the ``Non-authoritative answer:" line we got this time around. That means that named did not go out on the network to ask this time, the information is in the cache now. But the cached information *might* be out of date (stale). So you are informed of this (very slight) possibility by it saying `Non-authorative answer:'. When `nslookup` says this the second time you ask for a host it's a sure sign that named caches the information and that it's working. You exit `nslookup` by giving the command `exit'.

3.2 Making it even better

In large, well organized, academic or ISP (Internet Service Provider) networks you will sometimes find that the network people has set up a forwarder hierarchy of DNS servers which helps lighten the internal network load and on the outside servers as well. It's not easy to know if you're inside such a network or not. It is however not important and by using the DNS server of your network provider as a ``forwarder'' you can make the responses to queries faster and less of a load on your network. If you use a modem this can be quite a win. For the sake of this example we assume that your network provider has two name servers they want you to use, with IP numbers `10.0.0.1` and `10.1.0.1`. Then, in your `named.conf` file, inside the opening section called ``options'' insert these lines:

```
forward first;
forwarders {
     10.0.0.1;
     10.1.0.1;
 };
```

Restart your nameserver and test it with nslookup. Should work fine.

3.3 Congratulations

Now you know how to set up a caching named. Take a beer, milk, or whatever you prefer to celebrate it.

4. A *simple* domain.

How to set up your own domain.

4.1 But first some dry theory

Before we *really* start this section I'm going to serve you some theory on and an example of how DNS works. And you're going to read it because it's good for you. If you don't want to you should at least skim it very quickly. Stop skimming when you get to what should go in your `named.conf` file.

DNS is a hierarchical, tree structured, system. The top is written `.' and pronounced `root'. Under `.` there are a number of Top Level Domains (TLDs), the best known ones are `ORG`, `COM`, `EDU` and `NET`, but there are many more. Just like a tree it has a root and it branches out. If you have any computer science background you will recognize DNS as a search tree, and you will be able to find nodes, leaf nodes and edges.

When looking for a machine the query proceeds recursively into the hierarchy starting at the top. If you want to find out the address of `prep.ai.mit.edu` your name server has to find a name server that serves `edu`. It asks a `.` server (it already knows the `.` servers, that's what the `root.hints` file is for), the `.` server gives a list of `edu` servers:

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1
```

Start asking a root server:

```
> server c.root-servers.net.
Default Server:  c.root-servers.net
Address:  192.33.4.12
```

Set the Query type to NS (name server records):

```
> set q=ns
```

Ask about `edu`:

```
> edu.
```

The trailing . here is significant, it tells `nslookup` we're asking that `edu` is right under `.` (and not under any of our `search` domains, it speeds the search).

```
edu     nameserver = A.ROOT-SERVERS.NET
edu     nameserver = H.ROOT-SERVERS.NET
edu     nameserver = B.ROOT-SERVERS.NET
edu     nameserver = C.ROOT-SERVERS.NET
edu     nameserver = D.ROOT-SERVERS.NET
edu     nameserver = E.ROOT-SERVERS.NET
edu     nameserver = I.ROOT-SERVERS.NET
edu     nameserver = F.ROOT-SERVERS.NET
edu     nameserver = G.ROOT-SERVERS.NET
A.ROOT-SERVERS.NET      internet address = 198.41.0.4
```

TOC

```
H.ROOT-SERVERS.NET      internet address = 128.63.2.53
B.ROOT-SERVERS.NET      internet address = 128.9.0.107
C.ROOT-SERVERS.NET      internet address = 192.33.4.12
D.ROOT-SERVERS.NET      internet address = 128.8.10.90
E.ROOT-SERVERS.NET      internet address = 192.203.230.10
I.ROOT-SERVERS.NET      internet address = 192.36.148.17
F.ROOT-SERVERS.NET      internet address = 192.5.5.241
G.ROOT-SERVERS.NET      internet address = 192.112.36.4
```

This tells us that all `ROOT-SERVERS.NET` servers serves `EDU.`, so we can go on asking any of them. We'll continue asking `C`. Now we want to know who serves the next level of the domain name: `mit.edu.`:

```
> mit.edu.
Server:  c.root-servers.net
Address:  192.33.4.12

Non-authoritative answer:
mit.edu nameserver = W20NS.mit.edu
mit.edu nameserver = BITSY.mit.edu
mit.edu nameserver = STRAWB.mit.edu

Authoritative answers can be found from:
W20NS.mit.edu   internet address = 18.70.0.160
BITSY.mit.edu   internet address = 18.72.0.3
STRAWB.mit.edu  internet address = 18.71.0.151
```

`steawb`, `w20ns` and `bitsy` all serves `mit.edu`, we select one and inquire about the name one more level up: `ai.mit.edu`:

```
> server W20NS.mit.edu.
```

Host names are not case sensitive, but I use my mouse to cut and paste so it gets copied as-is from the screen.

```
Server:  W20NS.mit.edu
Address:  18.70.0.160

> ai.mit.edu.
Server:  W20NS.mit.edu
Address:  18.70.0.160

Non-authoritative answer:
ai.mit.edu      nameserver = ALPHA-BITS.AI.MIT.EDU
ai.mit.edu      nameserver = GRAPE-NUTS.AI.MIT.EDU
ai.mit.edu      nameserver = TRIX.AI.MIT.EDU
ai.mit.edu      nameserver = MUESLI.AI.MIT.EDU
ai.mit.edu      nameserver = LIFE.AI.MIT.EDU
ai.mit.edu      nameserver = BEET-CHEX.AI.MIT.EDU
ai.mit.edu      nameserver = MINI-WHEATS.AI.MIT.EDU
ai.mit.edu      nameserver = COUNT-CHOCULA.AI.MIT.EDU
ai.mit.edu      nameserver = MINTAKA.LCS.MIT.EDU

Authoritative answers can be found from:
AI.MIT.EDU      nameserver = ALPHA-BITS.AI.MIT.EDU
AI.MIT.EDU      nameserver = GRAPE-NUTS.AI.MIT.EDU
```

TOC

```
AI.MIT.EDU      nameserver = TRIX.AI.MIT.EDU
AI.MIT.EDU      nameserver = MUESLI.AI.MIT.EDU
AI.MIT.EDU      nameserver = LIFE.AI.MIT.EDU
AI.MIT.EDU      nameserver = BEET-CHEX.AI.MIT.EDU
AI.MIT.EDU      nameserver = MINI-WHEATS.AI.MIT.EDU
AI.MIT.EDU      nameserver = COUNT-CHOCULA.AI.MIT.EDU
AI.MIT.EDU      nameserver = MINTAKA.LCS.MIT.EDU
ALPHA-BITS.AI.MIT.EDU   internet address = 128.52.32.5
GRAPE-NUTS.AI.MIT.EDU   internet address = 128.52.36.4
TRIX.AI.MIT.EDU internet address = 128.52.37.6
MUESLI.AI.MIT.EDU       internet address = 128.52.39.7
LIFE.AI.MIT.EDU internet address = 128.52.32.80
BEET-CHEX.AI.MIT.EDU    internet address = 128.52.32.22
MINI-WHEATS.AI.MIT.EDU  internet address = 128.52.54.11
COUNT-CHOCULA.AI.MIT.EDU        internet address = 128.52.38.22
MINTAKA.LCS.MIT.EDU     internet address = 18.26.0.36
```

So `museli.ai.mit.edu` is a nameserver for `ai.mit.edu`:

```
> server MUESLI.AI.MIT.EDU
Default Server:  MUESLI.AI.MIT.EDU
Address:  128.52.39.7
```

Now I change query type, we've found the name server so now we're going to ask about everything `wheaties` knows about `prep.ai.mit.edu`.

```
> set q=any
> prep.ai.mit.edu.
Server:  MUESLI.AI.MIT.EDU
Address:  128.52.39.7

prep.ai.mit.edu CPU = dec/decstation-5000.25    OS = unix
prep.ai.mit.edu
        inet address = 18.159.0.42, protocol = tcp
          ftp  telnet  smtp  finger
prep.ai.mit.edu preference = 1, mail exchanger = gnu-life.ai.mit.edu
prep.ai.mit.edu internet address = 18.159.0.42
ai.mit.edu      nameserver = beet-chex.ai.mit.edu
ai.mit.edu      nameserver = alpha-bits.ai.mit.edu
ai.mit.edu      nameserver = mini-wheats.ai.mit.edu
ai.mit.edu      nameserver = trix.ai.mit.edu
ai.mit.edu      nameserver = muesli.ai.mit.edu
ai.mit.edu      nameserver = count-chocula.ai.mit.edu
ai.mit.edu      nameserver = mintaka.lcs.mit.edu
ai.mit.edu      nameserver = life.ai.mit.edu
gnu-life.ai.mit.edu     internet address = 128.52.32.60
beet-chex.ai.mit.edu    internet address = 128.52.32.22
alpha-bits.ai.mit.edu   internet address = 128.52.32.5
mini-wheats.ai.mit.edu  internet address = 128.52.54.11
trix.ai.mit.edu internet address = 128.52.37.6
muesli.ai.mit.edu       internet address = 128.52.39.7
count-chocula.ai.mit.edu        internet address = 128.52.38.22
mintaka.lcs.mit.edu     internet address = 18.26.0.36
```

```
life.ai.mit.edu internet address = 128.52.32.80
```

So starting at `.` we found the successive name servers for the each level in the domain name. If you had used your own DNS server instead of using all those other servers, your named would of-course cache all the information it found while digging this out for you, and it would not have to ask again for a while.

In the tree analogue each ``.'' in the name is a branching point. And each part between the ``.''s are the names of individual branches in the tree.

We climb the tree by taking the name we want (`prep.ai.mit.edu`) first finding the root (`.`) and then looking for the next branch to climb, in this case `edu`. Once we have found it we climb it by switching to the server that knows about that part of the name. Next we look for the `mit` branch over the `edu` branch (the combined name is `mit.edu`) and climb it by switching to a server that knows about `mit.edu`. Again we look for the next branch, it's `ai.mit.edu` and again we switch to the server that knows about it. Now we have arrived at the right server, at the right branching point. The last part is finding `prep.ai.mit.edu`, which is simple. In computer science we usually call `prep` a *leaf* on the tree.

A much less talked about, but just as important domain is `in-addr.arpa`. It too is nested like the `normal' domains. `in-addr.arpa` allows us to get the hosts name when we have its address. A important thing here is to note that ip addresses are written in reverse order in the `in-addr.arpa` domain. If you have the address of a machine: `192.128.52.43` named proceeds just like for the `prep.ai.mit.edu` example: find `arpa.` servers. Find `in-addr.arpa.` servers, find `192.in-addr.arpa.` servers, find `128.192.in-addr.arpa.` servers, `find` `52.128.192.in-addr.arpa.` servers. Find needed records for `43.52.128.192.in-addr.arpa.` Clever huh? (Say `yes'.) The reversion of the numbers can be confusing for years though.

I have just told a lie. DNS does not work precisely the way I just told you. But it's close enough.

4.2 Our own domain

Now to define our own domain. We're going to make the domain `linux.bogus` and define machines in it. I use a totally bogus domain name to make sure we disturb no-one Out There.

One more thing before we start: Not all characters are allowed in host names. We're restricted to the characters of the English alphabet: a-z, and numbers: 0-9 and the character '-' (dash). Keep to those characters. Upper and lower-case characters are the same for DNS, so `pat.uio.no` is identical to `Pat.UiO.No`.

We've already started this part with this line in `named.conf`:

```
zone "0.0.127.in-addr.arpa" {
        type master;
        file "pz/127.0.0";
};
```

Please note the lack of `.' at the end of the domain names in this file. This says that now we will define the zone `0.0.127.in-addr.arpa`, that we're the master server for it and that it is stored in a file called `pz/127.0.0`. We've already set up this file, it reads:

```
@               IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                                1       ; Serial
                                8H      ; Refresh
                                2H      ; Retry
                                1W      ; Expire
                                1D)     ; Minimum TTL
```

```
                        NS      ns.linux.bogus.
1                       PTR     localhost.
```

TOC

Please note the `.' at the end of all the full domain names in this file, in contrast to the `named.conf` file above. Some people like to start each zone file with a `$ORIGIN` directive, but this is superfluous. The origin (where in the DNS hierarchy it belongs) of a zone file is specified in the zone section of the `named.conf` file, in this case it's `0.0.127.in-addr.arpa`.

This `zone file' contains 3 `resource records' (RRs): A SOA RR. A NS RR and a PTR RR. SOA is short for Start Of Authority. The `@' is a special notation meaning the origin, and since the `domain' column for this file says 0.0.127.in-addr.arpa the first line really means

```
0.0.127.in-addr.arpa.   IN      SOA ...
```

NS is the Name Server RR. There is no '@' at the start of this line, it is implicit since the last line started with a '@'. Saves some typing that. So the NS line could also be written

```
0.0.127.in-addr.arpa.   IN      NS      ns.linux.bogus
```

It tells DNS what machine is the name server of the domain `0.0.127.in-addr.arpa`, it is `ns.linux.bogus`. 'ns' is a customary name for name-servers, but as with web servers who are customarily named `www.`*something* the name may be anything.

And finally the PTR record says that the host at address 1 in the subnet `0.0.127.in-addr.arpa`, i.e., 127.0.0.1 is named `localhost`.

The SOA record is the preamble to *all* zone files, and there should be exactly one in each zone file. It describes the zone, where it comes from (a machine called `ns.linux.bogus`), who is responsible for its contents (`hostmaster@linux.bogus`, you should insert your e-mail address here), what version of the zone file this is (serial: 1), and other things having to do with caching and secondary DNS servers. For the rest of the fields (refresh, retry, expire and minimum) use the numbers used in this HOWTO and you should be safe.

Now restart your named (the command is `ndc restart`) and use nslookup to examine what you've done:

```
$ nslookup

Default Server:  localhost
Address:  127.0.0.1

> 127.0.0.1
Server:  localhost
Address:  127.0.0.1

Name:    localhost
Address:  127.0.0.1
```

so it manages to get `localhost` from 127.0.0.1, good. Now for our main task, the `linux.bogus` domain, insert a new 'zone' section in `named.conf`:

```
zone "linux.bogus" {
        notify no;
        type master;
        file "pz/linux.bogus";
```

```
};
```

TOC

Note again the lack of ending `.' on the domain name in the `named.conf` file.

In the `linux.bogus` zone file we'll put some totally bogus data:

```
;
; Zone file for linux.bogus
;
; The full zone file
;
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151       ; serial, todays date + todays serial #
                        8H              ; refresh, seconds
                        2H              ; retry, seconds
                        1W              ; expire, seconds
                        1D )            ; minimum, seconds
;
                NS      ns              ; Inet Address of name server
                MX      10 mail.linux.bogus     ; Primary Mail Exchanger
                MX      20 mail.friend.bogus.   ; Secondary Mail Exchanger
;
localhost       A       127.0.0.1
ns              A       192.168.196.2
mail            A       192.168.196.4
```

Two things must be noted about the SOA record. `ns.linux.bogus` *must* be a actual machine with a A record. It is not legal to have a CNAME record for he machine mentioned in the SOA record. It's name need not be `ns', it could be any legal host name. Next, hostmaster.linux.bogus should be read as hostmaster@linux.bogus, this should be a mail alias, or a mailbox, where the person(s) maintaining DNS should read mail frequently. Any mail regarding the domain will be sent to the address listed here. The name need not be `hostmaster', it can be your normal e-mail address, but the e-mail address `hostmaster' is often expected to work as well.

There is one new RR type in this file, the MX, or Mail eXchanger RR. It tells mail systems where to send mail that is addressed to `someone@linux.bogus`, namely too `mail.linux.bogus` or `mail.friend.bogus`. The number before each machine name is that MX RRs priority. The RR with the lowest number (10) is the one mail should be sent to if possible. If that fails the mail can be sent to one with a higher number, a secondary mail handler, i.e., `mail.friend.bogus` which has priority 20 here.

Restart named by running `ndc restart`. Examine the results with nslookup:

```
$ nslookup
> set q=any
> linux.bogus
Server:  localhost
Address:  127.0.0.1

linux.bogus
        origin = ns.linux.bogus
        mail addr = hostmaster.linux.bogus
        serial = 199802151
        refresh = 28800 (8 hours)
```

TOC

```
            retry   = 7200 (2 hours)
            expire  = 604800 (7 days)
            minimum ttl = 86400 (1 day)
    linux.bogus     nameserver = ns.linux.bogus
    linux.bogus     preference = 10, mail exchanger = mail.linux.bogus.linux.bogus
    linux.bogus     preference = 20, mail exchanger = mail.friend.bogus
    linux.bogus     nameserver = ns.linux.bogus
    ns.linux.bogus  internet address = 192.168.196.2
    mail.linux.bogus        internet address = 192.168.196.4
```

Upon careful examination you will discover a bug. The line

```
    linux.bogus     preference = 10, mail exchanger = mail.linux.bogus.linux.bogus
```

is all wrong. It should be

```
    linux.bogus     preference = 10, mail exchanger = mail.linux.bogus
```

I deliberately made a mistake so you could learn from it :-) Looking in the zone file we find that the line

```
                    MX      10 mail.linux.bogus     ; Primary Mail Exchanger
```

is missing a period. Or has a 'linux.bogus' too many. If a machine name does not end in a period in a zone file the origin is added to its end causing the double `linux.bogus.linux.bogus`. So either

---

```
                MX      10 mail.linux.bogus.    ; Primary Mail Exchanger
```

---

or

```
                MX      10 mail                 ; Primary Mail Exchanger
```

---

is correct. I prefer the latter form, it's less to type. There are some bind experts that disagree, and some that agree with this. In a zone file the domain should either be written out and ended with a `.' or it should not be included at all, in which case it defaults to the origin.

I must stress that in the named.conf file there should *not* be `.'s after the domain names. You have no idea how many times a `.' too many or few have fouled up things and confused the h*ll out of people.

So having made my point here is the new zone file, with some extra information in it as well:

```
;
; Zone file for linux.bogus
;
; The full zone file
;
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151       ; serial, todays date + todays serial #
                        8H              ; refresh, seconds
                        2H              ; retry, seconds
```

TOC

```
                        1W              ; expire, seconds
                        1D )            ; minimum, seconds
;
                TXT     "Linux.Bogus, your DNS consultants"
                NS      ns              ; Inet Address of name server
                NS      ns.friend.bogus.
                MX      10 mail         ; Primary Mail Exchanger
                MX      20 mail.friend.bogus. ; Secondary Mail Exchanger

localhost       A       127.0.0.1

gw              A       192.168.196.1
                HINFO   "Cisco" "IOS"
                TXT     "The router"

ns              A       192.168.196.2
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "Pentium" "Linux 2.0"
www             CNAME   ns

donald          A       192.168.196.3
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "i486"  "Linux 2.0"
                TXT     "DEK"

mail            A       192.168.196.4
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "386sx" "Linux 1.2"

ftp             A       192.168.196.5
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "P6" "Linux 2.1.86"
```

There are a number of new RRs here: HINFO (Host INFOrmation) has two parts, it's a good habit to quote each. The first part is the hardware or CPU on the machine, and the second part the software or OS on the machine. The machine called 'ns' has a Pentium CPU and runs Linux 2.0. CNAME (Canonical NAME) is a way to give each machine several names. So www is an alias for ns.

CNAME record usage is a bit controversial. But it's safe to follow the rule that a MX, CNAME or SOA record should *never* refer to a CNAME record, they should only refer to something with a A record, so it is inadvisable to have

---

```
foobar          CNAME   www                     ; NO!
```

---

but correct to have

```
foobar          CNAME   ns                      ; Yes!
```

TOC

It's also safe to assume that a CNAME is not a legal host name for a e-mail address: `webmaster@www.linux.bogus` is an illegal e-mail address given the setup above. You can expect quite a few mail admins Out There to enforce this rule even if it works for you. The way to avoid this is to use A records (and perhaps some others too, like a MX record) instead:

```
www             A       192.168.196.2
```

A number of the arch-bind-wizards, recommend *not* using CNAME at all. But the discussion of why or why not is beyond this HOWTO.

But as you see, this HOWTO and many sites does not follow this rule.

Load the new database by running `ndc reload`, this causes named to read its files again.

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1

> ls -d linux.bogus
```

This means that all records should be listed. It results in this:

```
[localhost]
$ORIGIN linux.bogus.
@                       1D IN SOA       ns hostmaster (
                                        199802151       ; serial
                                        8H              ; refresh
                                        2H              ; retry
                                        1W              ; expiry
                                        1D )            ; minimum

                        1D IN NS        ns
                        1D IN NS        ns.friend.bogus.
                        1D IN TXT       "Linux.Bogus, your DNS consultants"
                        1D IN MX        10 mail
                        1D IN MX        20 mail.friend.bogus.
gw                      1D IN A         192.168.196.1
                        1D IN HINFO     "Cisco" "IOS"
                        1D IN TXT       "The router"
mail                    1D IN A         192.168.196.4
                        1D IN MX        10 mail
                        1D IN MX        20 mail.friend.bogus.
                        1D IN HINFO     "386sx" "Linux 1.0.9"
localhost               1D IN A         127.0.0.1
www                     1D IN CNAME     ns
donald                  1D IN A         192.168.196.3
                        1D IN MX        10 mail
                        1D IN MX        20 mail.friend.bogus.
                        1D IN HINFO     "i486" "Linux 1.2"
                        1D IN TXT       "DEK"
ftp                     1D IN A         192.168.196.5
```

```
                                1D IN MX        10 mail
                                1D IN MX        20 mail.friend.bogus.
                                1D IN HINFO     "P6" "Linux 1.3.59"
    ns                          1D IN A         192.168.196.2
                                1D IN MX        10 mail
                                1D IN MX        20 mail.friend.bogus.
                                1D IN HINFO     "Pentium" "Linux 1.2"
```

That's good. As you see it looks a lot like the zone file itself. Let's check what it says for `www` alone:

```
    > set q=any
    > www.linux.bogus.
    Server:  localhost
    Address:  127.0.0.1

    www.linux.bogus canonical name = ns.linux.bogus
    linux.bogus     nameserver = ns.linux.bogus
    linux.bogus     nameserver = ns.friend.bogus
    ns.linux.bogus  internet address = 192.168.196.2
```

In other words, the real name of `www.linux.bogus` is `ns.linux.bogus`, and it gives you some of the information it has about ns as well, enough to connect to it if you were a program.

Now we're halfway.

4.3 The reverse zone

Now programs can convert the names in linux.bogus to addresses which they can connect to. But also required is a reverse zone, one making DNS able to convert from an address to a name. This name is used buy a lot of servers of different kinds (FTP, IRC, WWW and others) to decide if they want to talk to you or not, and if so, maybe even how much priority you should be given. For full access to all services on the Internet a reverse zone is required.

Put this in `named.conf`:

```
zone "196.168.192.in-addr.arpa" {
        notify no;
        type master;
        file "pz/192.168.196";
};
```

This is exactly as with the `0.0.127.in-addr.arpa`, and the contents are similar:

```
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151 ; Serial, todays date + todays serial
                        8H      ; Refresh
                        2H      ; Retry
                        1W      ; Expire
                        1D)     ; Minimum TTL
                NS      ns.linux.bogus.
```

```
1                PTR     gw.linux.bogus.
2                PTR     ns.linux.bogus.
3                PTR     donald.linux.bogus.
4                PTR     mail.linux.bogus.
5                PTR     ftp.linux.bogus.
```

Now you restart your named (`ndc restart`) and examine your work with nslookup again:

```
> 192.168.196.4
Server:  localhost
Address:  127.0.0.1

Name:    mail.linux.bogus
Address:  192.168.196.4
```

---

so, it looks OK, dump the whole thing to examine that too:

```
> ls -d 196.168.192.in-addr.arpa
[localhost]
$ORIGIN 196.168.192.in-addr.arpa.
@                       1D IN SOA       ns.linux.bogus. hostmaster.linux.bogus. (
                                        199802151       ; serial
                                        8H              ; refresh
                                        2H              ; retry
                                        1W              ; expiry
                                        1D )            ; minimum

                        1D IN NS        ns.linux.bogus.
1                       1D IN PTR       gw.linux.bogus.
2                       1D IN PTR       ns.linux.bogus.
3                       1D IN PTR       donald.linux.bogus.
4                       1D IN PTR       mail.linux.bogus.
5                       1D IN PTR       ftp.linux.bogus.
@                       1D IN SOA       ns.linux.bogus. hostmaster.linux.bogus. (
                                        199802151       ; serial
                                        8H              ; refresh
                                        2H              ; retry
                                        1W              ; expiry
                                        1D )            ; minimum
```

Looks good! If your output didn't look like that look for error-messages in your syslog, I explained how to do that at the very beginning of this chapter.

4.4 Words of caution

There are some things I should add here. The IP numbers used in the examples above are taken from one of the blocks of 'private nets', i.e., they are not allowed to be used publicly on the internet. So they are safe to use in an example in a HOWTO. The second thing is the `notify no;` line. It tells named not to notify its secondary (slave) servers when it has

TOC

gotten a update to one of its zone files. In bind-8 the named can notify the other servers listed in NS records in the zo when a zone is updated. This is handy for ordinary use, but for private experiments with zones this feature should be we don't want the experiment to pollute the Internet do we?

And, of course, this domain is highly bogus, and so are all the addresses in it. For a real example of a real-life domain see the next main-section.

4.5 Why reverse lookups don't work.

There are a couple of ``gotchas'' that normally are avoided with name lookups that are often seen when setting up reverse zones. Before you go on you need reverse lookups of your machines working on your own nameserver. If it isn't go back and fix it before continuing.

I will discuss two failures of reverse lookups as seen from outside your network:

The reverse zone isn't delegated.

When you ask a service provider for a network-address range and a domain name the domain name is normally delegated as a matter of course. A delegation is the glue NS record that helps you get from one nameserver to another as explained in the dry theory section above. You read that, right? If your reverse zone dosn't work go back and read it. Now.

The reverse zone also needs to be delegated. If you got the `192.168.196` net with the `linux.bogus` domain from your provider they need to put `NS` records in for your reverse zone as well as for your forward zone. If you follow the chain from `in-addr.arpa` and up to your net you will probably find a break in the chain. Most probably at your service provider. Having found the break in the chain contact your service-provider and ask them to correct the error.

You've got a classless subnet

This is a somewhat advanced topic, but classless subnets are very common these days and you probably have one unless you're a medium sized company.

A classless subnet is what keeps the Internet going these days. Some years ago there was much ado about the shortage of ip numbers. The smart people in IETF (the Internet Engineering Task Force, they keep the Internet working) stuck their heads together and solved the problem. At a price. The price is that you'll get less than a ``C'' subnet and some things may break. Please see Ask Mr. DNS at http://www.acmebw.com/askmrdns/00007.htm for an good explanation of this and how to handle it.

Did you read it? I'm not going to explain it so please read it.

The first part of the problem is that your ISP must understand the technique described by Mr. DNS. Not all small ISPs have a working understanding of this. If so you might have to explain to them and be persistent. But be sure you understand it first ;-). They will then set up a nice reverse zone at their server which you can examine for correctness with nslookup.

The second and last part of the problem is that you must understand the technique. If you're unsure go back and read about it again. Then you can set up your own classless reverse zone as described by Mr. DNS.

There is another trap lurking here. Old resolvers will *not* be able to follow the `CNAME` trick in the resolving chain and will fail to reverse-resolve your machine. This can result in the service assigning it an incorrect access class, deny access or something along those lines. If you stumble into such a service the only solution (that I know of) is for your ISP to insert your PTR record directly into their trick classless zone file instead of the trick CNAME record.

Some ISPs will offer other ways to handle this, like Web based forms for you to input your reverse-mappings in or other automagical systems.

---

5. A real domain example

Where we list some *real* zone files

Users have suggested that I include a real example of a working domain as well as the tutorial example.

I use this example with permission from David Bullock of LAND-5. These files were current 24th of September 1996, and were then edited to fit bind 8 restrictions and use extensions by me. So, what you see here differs a bit from what you find if you query LAND-5's name servers now.

5.1 /etc/named.conf (or /var/named/named.conf)

Here we find master zone sections for the two reverse zones needed: the 127.0.0 net, as well as LAND-5's `206.6.177` subnet. And a primary line for land-5's forward zone `land-5.com`. Also note that instead of stuffing the files in a directory called `pz`, as I do in this HOWTO, he puts them in a directory called `zone`.

```
// Boot file for LAND-5 name server

options {
        directory "/var/named";
};

zone "." {
        type hint;
        file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};

zone "land-5.com" {
        type master;
        file "zone/land-5.com";
};

zone "177.6.206.in-addr.arpa" {
        type master;
        file "zone/206.6.177";
};
```

If you put this in your named.conf file to play with **PLEASE** put ``notify no;" in the zone sections for the two `land-5` zones so as to avoid accidents.

5.2 /var/named/root.hints

TOC

Keep in mind that this file is dynamic, and the one listed here is old. You're better off using one produced now, with dig, as explained earlier.

```
; <<>> DiG 8.1 <<>> @A.ROOT-SERVERS.NET.
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10
;; flags: qr aa rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13
;; QUERY SECTION:
;;      ., type = NS, class = IN

;; ANSWER SECTION:
.                       6D IN NS        G.ROOT-SERVERS.NET.
.                       6D IN NS        J.ROOT-SERVERS.NET.
.                       6D IN NS        K.ROOT-SERVERS.NET.
.                       6D IN NS        L.ROOT-SERVERS.NET.
.                       6D IN NS        M.ROOT-SERVERS.NET.
.                       6D IN NS        A.ROOT-SERVERS.NET.
.                       6D IN NS        H.ROOT-SERVERS.NET.
.                       6D IN NS        B.ROOT-SERVERS.NET.
.                       6D IN NS        C.ROOT-SERVERS.NET.
.                       6D IN NS        D.ROOT-SERVERS.NET.
.                       6D IN NS        E.ROOT-SERVERS.NET.
.                       6D IN NS        I.ROOT-SERVERS.NET.
.                       6D IN NS        F.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
G.ROOT-SERVERS.NET.     5w6d16h IN A    192.112.36.4
J.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.10
K.ROOT-SERVERS.NET.     5w6d16h IN A    193.0.14.129
L.ROOT-SERVERS.NET.     5w6d16h IN A    198.32.64.12
M.ROOT-SERVERS.NET.     5w6d16h IN A    202.12.27.33
A.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.4
H.ROOT-SERVERS.NET.     5w6d16h IN A    128.63.2.53
B.ROOT-SERVERS.NET.     5w6d16h IN A    128.9.0.107
C.ROOT-SERVERS.NET.     5w6d16h IN A    192.33.4.12
D.ROOT-SERVERS.NET.     5w6d16h IN A    128.8.10.90
E.ROOT-SERVERS.NET.     5w6d16h IN A    192.203.230.10
I.ROOT-SERVERS.NET.     5w6d16h IN A    192.36.148.17
F.ROOT-SERVERS.NET.     5w6d16h IN A    192.5.5.241

;; Total query time: 215 msec
;; FROM: roke.uio.no to SERVER: A.ROOT-SERVERS.NET.  198.41.0.4
;; WHEN: Sun Feb 15 01:22:51 1998
;; MSG SIZE  sent: 17  rcvd: 436
```

5.3 /var/named/zone/127.0.0

Just the basics, the obligatory SOA record, and a record that maps 127.0.0.1 to `localhost`. Both are required. No more should be in this file. It will probably never need to be updated, unless your nameserver or hostmaster address changes.

```
@                IN      SOA     land-5.com. root.land-5.com. (
                                 199609203       ; Serial
                                 28800   ; Refresh
                                 7200    ; Retry
                                 604800  ; Expire
                                 86400)  ; Minimum TTL
                         NS      land-5.com.

1                        PTR     localhost.
```

5.4 /var/named/zone/land-5.com

Here we see the mandatory SOA record, the needed NS records. We can see that he has a secondary name server at `ns2.psi.net`. This is as it should be, *always* have a off site secondary server as backup. We can also see that he has a master host called `land-5` which takes care of many of the different Internet services, and that he's done it with CNAMEs (a alternative is using A records).

As you see from the SOA record, the zone file originates at `land-5.com`, the contact person is `root@land-5.com`. `hostmaster` is another oft used address for the contact person. The serial number is in the customary yyyymmdd format with todays serial number appended; this is probably the sixth version of zone file on the 20th of September 1996. Remember that the serial number *must* increase monotonically, here there is only *one* digit for todays serial#, so after 9 edits he has to wait until tomorrow before he can edit the file again. Consider using two digits.

```
@       IN      SOA     land-5.com. root.land-5.com. (
                        199609206       ; serial, todays date + todays serial #
                        8H              ; refresh, seconds
                        2H              ; retry, seconds
                        1W              ; expire, seconds
                        1D )            ; minimum, seconds
                NS      land-5.com.
                NS      ns2.psi.net.
                MX      10 land-5.com.  ; Primary Mail Exchanger
                TXT     "LAND-5 Corporation"

localhost       A       127.0.0.1

router          A       206.6.177.1

land-5.com.     A       206.6.177.2
```

TOC

```
ns              A       206.6.177.3
www             A       207.159.141.192

ftp             CNAME   land-5.com.
mail            CNAME   land-5.com.
news            CNAME   land-5.com.

funn            A       206.6.177.2

;
;       Workstations
;
ws-177200       A       206.6.177.200
                MX      10 land-5.com.   ; Primary Mail Host
ws-177201       A       206.6.177.201
                MX      10 land-5.com.   ; Primary Mail Host
ws-177202       A       206.6.177.202
                MX      10 land-5.com.   ; Primary Mail Host
ws-177203       A       206.6.177.203
                MX      10 land-5.com.   ; Primary Mail Host
ws-177204       A       206.6.177.204
                MX      10 land-5.com.   ; Primary Mail Host
ws-177205       A       206.6.177.205
                MX      10 land-5.com.   ; Primary Mail Host
; {Many repetitive definitions deleted - SNIP}
ws-177250       A       206.6.177.250
                MX      10 land-5.com.   ; Primary Mail Host
ws-177251       A       206.6.177.251
                MX      10 land-5.com.   ; Primary Mail Host
ws-177252       A       206.6.177.252
                MX      10 land-5.com.   ; Primary Mail Host
ws-177253       A       206.6.177.253
                MX      10 land-5.com.   ; Primary Mail Host
ws-177254       A       206.6.177.254
                MX      10 land-5.com.   ; Primary Mail Host
```

If you examine land-5s nameserver you will find that the host names are of the form `ws_`*number*. As of late bind 4 versions named started enforcing the restrictions on what characters may be used in host names. So that does not work with bind-8 at all, and I substituted '-' (dash) for '_' (underline) for use in this HOWTO.

Another thing to note is that the workstations don't have individual names, but rather a prefix followed by the two last parts of the IP numbers. Using such a convention can simplify maintenance significantly, but can be a bit impersonal, and, in fact, be a source of irritation among your customers.

We also see that `funn.land-5.com` is an alias for `land-5.com`, but using an A record, not a CNAME record. This is a good policy as noted earlier.

5.5 /var/named/zone/206.6.177

TOC

I'll comment on this file below

```
@               IN      SOA     land-5.com. root.land-5.com. (
                                199609206       ; Serial
                                28800   ; Refresh
                                7200    ; Retry
                                604800  ; Expire
                                86400)  ; Minimum TTL
                        NS      land-5.com.
                        NS      ns2.psi.net.
;
;       Servers
;
1       PTR     router.land-5.com.
2       PTR     land-5.com.
2       PTR     funn.land-5.com.
;
;       Workstations
;
200     PTR     ws-177200.land-5.com.
201     PTR     ws-177201.land-5.com.
202     PTR     ws-177202.land-5.com.
203     PTR     ws-177203.land-5.com.
204     PTR     ws-177204.land-5.com.
205     PTR     ws-177205.land-5.com.
; {Many repetitive definitions deleted - SNIP}
250     PTR     ws-177250.land-5.com.
251     PTR     ws-177251.land-5.com.
252     PTR     ws-177252.land-5.com.
253     PTR     ws-177253.land-5.com.
254     PTR     ws-177254.land-5.com.
```

The reverse zone is the bit of the setup that seems to cause the most grief. It is used to find the host name if you have the IP number of a machine. Example: you are an IRC server and accept connections from IRC clients. However you are a Norwegian IRC server and so you only want to accept connections from clients in Norway and other Scandinavian countries. When you get a connection from a client the C library is able to tell you the IP number of the connecting machine because the IP number of the client is contained in all the packets that are passed over the network. Now you can call a function called gethostbyaddr that looks up the name of a host given the IP number. Gethostbyaddr will ask a DNS server, which will then traverse the DNS looking for the machine. Supposing the client connection is from ws-177200.land-5.com. The IP number the C library provides to the IRC server is 206.6.177.200. To find out the name of that machine we need to find `200.177.6.206.in-addr.arpa`. The DNS server will first find the `arpa.` servers, then find `in-addr.arpa.` servers, following the reverse trail through 206, then 6 and at last finding the server for the `177.6.206.in-addr.arpa` zone at LAND-5. From which it will finally get the answer that for `200.177.6.206.in-addr.arpa` we have a ``PTR ws-177200.land-5.com'' record, meaning that the name that goes with `206.6.177.200` is `ws-177200.land-5.com`. As with the explanation of how `prep.ai.mit.edu` is looked up, this is slightly fictitious.

Getting back to the IRC server example. The IRC server only accepts connections from the Scandinavian countries, i.e., `*.no`, `*.se`, `*.dk`, the name `ws-177200.land-5.com` clearly does not match any of those, and the server will deny the connection. If there was *no* reverse mapping of `206.2.177.200` through the `in-addr.arpa` zone, the server would have been unable to find the name at all and would have to settle to comparing `206.2.177.200` with `*.no`, `*.se` and `*.dk`, none of which will match.

Some people will tell you that reverse lookup mappings are only important for servers, or not important at all. Not so: Many ftp, news, IRC and even some http (WWW) servers will *not* accept connections from machines of which they are not able to find the name. So reverse mappings for machines are in fact *mandatory*.

---

6. Maintenance

Keeping it working.

There is one maintenance task you have to do on nameds, other than keeping them running. That's keeping the `root.hints` file updated. The easiest way is using dig, first run dig with no arguments, you will get the `root.hints` according to your own server. Then ask one of the listed root servers with `dig @rootserver`. You will note that the output looks terribly like a `root.hints` file. Save it to a file (`dig @e.root-servers.net . ns >root.hints.new`) and replace the old `root.hints` with it.

Remember to reload named after replacing the cache file.

Al Longyear sent me this script, that can be run automatically to update `root.hints`, install a crontab entry to run it once a month and forget it. The script assumes you have mail working and that the mail-alias `hostmaster' is defined. You must hack it to suit your setup.

```
#!/bin/sh
#
# Update the nameserver cache information file once per month.
# This is run automatically by a cron entry.
#
# Original by Al Longyear
# Updated for bind 8 by Nicolai Langfeldt
# Miscelanious error-conditions reported by David A. Ranch
# Ping test suggested by Martin Foster
#
(
 echo "To: hostmaster <hostmaster>"
 echo "From: system <root>"
 echo "Subject: Automatic update of the root.hints file"
 echo

 PATH=/sbin:/usr/sbin:/bin:/usr/bin:
 export PATH
 cd /var/named

 # Are we online?  Ping a server at your ISP
 case `ping -qnc some.machine.net` in
   *'100% packet loss'*)
        echo "The network is DOWN. root.hints NOT updated"
        echo
        exit 0
        ;;
 esac

 dig @rs.internic.net . ns >root.hints.new 2>&1
```

TOC

```
 case `cat root.hints.new` in
   *NOERROR*)
        # It worked
        ;;;
   *)
        echo "The root.hints file update has FAILED."
        echo "This is the dig output reported:"
        echo
        cat root.hints.new
        exit 0
        ;;
 esac

 echo "The root.hints file has been updated to contain the following
information:"
 echo
 cat root.hints.new

 chown root.root root.hints.new
 chmod 444 root.hints.new
 rm -f root.hints.old
 mv root.hints root.hints.old
 mv root.hints.new root.hints
 ndc restart
 echo
 echo "The nameserver has been restarted to ensure that the update is complete."
 echo "The previous root.hints file is now called
/var/named/root.hints.old."
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

Some of you might have picked up that the `root.hints` file is also available by ftp from Internic. Please don't use ftp to update `root.hints`, the above method is much more friendly to the net, and Internic.

---

7. Converting from version 4 to version 8

This was originally a section on using bind 8 written by David E. Smith (dave@bureau42.ml.org). I have edited it some to fit the new section name.

There's not much to it. Except for using named.conf instead of named.boot, everything is identical. And bind8 comes with a perl script that converts old-style files to new. Example named.boot (old style) for a cache-only name server:

```
directory /var/named
cache   .                                       root.hints
primary 0.0.127.IN-ADDR.ARPA                    127.0.0.zone
primary localhost                               localhost.zone
```

On the command line, in the bind8/src/bin/named directory (*this assumes you got a source distribution. If you got a binary package the script is probably around, I'm not sure where it would be though. -ed.*), type:

```
./named-bootconf.pl < named.boot > named.conf
```

Which creates named.conf:

```
// generated by named-bootconf.pl

options {
        directory "/var/named";
};

zone "." {
        type hint;
        file "root.hints";
};

zone "0.0.127.IN-ADDR.ARPA" {
        type master;
        file "127.0.0.zone";
};

zone "localhost" {
```

```
        type master;
        file "localhost.zone";
};
```

It works for everything that can go into a `named.boot` file, although it doesn't add all of the new enhancements and configuration options that bind8 allows. Here's a more complete `named.conf` that does the same things, but a little more efficiently.

```
// This is a configuration file for named (from BIND 8.1 or later).
// It would normally be installed as /etc/named.conf.
// The only change made from the `stock' named.conf (aside from this
// comment :) is that the directory line was uncommented, since I
// already had the zone files in /var/named.

options {
        directory "/var/named";
        datasize 20M;
};

zone "localhost" IN {
        type master;
        file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "127.0.0.zone";
};

zone "." IN {
        type hint;
        file "root.hints";
};
```

In the bind 8 distributions directory bind8/src/bin/named/test you find this, and copies of the zone files, that many people can just drop in and use instantly.

The formats for zone files and `root.hints` files are identical, as are the commands for updating them.

---

8. Questions and Answers

Please read this section before mailing me.

1. My named wants a named.boot file

   You are reading the wrong HOWTO. Please see the old version of this HOWTO, which covers bind 4, at http://www.math.uio.no/~janl/DNS/

2. How do use DNS from inside a firewall?

   A hint: `forward only;`, You will probably also need

   ```
   query-source port 53;
   ```

   inside the ``options'' part of the `named.conf` file as suggested in the example caching section.

3. How do I make DNS rotate through the available addresses for a service, say www.busy.site to obtain a load balancing effect, or similar?

   Make several **A** records for www.busy.site and use bind 4.9.3 or later. Then bind will round-robin the answers. It will *not* work with earlier versions of bind.

4. I want to set up DNS on a (closed) intranet. What do I do?

   You drop the root.hints file and just do zone files. That also means you don't have to get new hint files all the time.

5. How do I set up a secondary (slave) name server?

   If the primary/master server has address 127.0.0.1 you put a line like this in the named.conf file of your secondary:

   ```
   zone "linux.bogus" {
         type slave;
         file "sz/linux.bogus";
         masters { 127.0.0.1; };
   };
   ```

   You may list several alternate master servers the zone can be copied from inside the `masters` list, separated by ';' (semicolon).

6. I want bind running when I'm disconnected from the net.

   There are three items regarding this:

   - I have received this mail from Ian Clark <ic@deakin.edu.au> where he explains his way of doing this:

     ```
     I run named on my 'Masquerading' machine here. I have
     two root.hints files, one called root.hints.real which contains
     the real root server names and the other called root.hints.fake
     which contains...
     ```

```
----
; root.hints.fake
; this file contains no information
----

When I go off line I copy the root.hints.fake file to root.hints and
restart named.

When I go online I copy root.hints.real to root.hints and restart
named.

This is done from ip-down & ip-up respectively.

The first time I do a query off line on a domain name named doesn't
have details for it puts an entry like this in messages..

Jan 28 20:10:11 hazchem named[10147]: No root nameserver for class IN

which I can live with.

It certainly seems to work for me. I can use the nameserver for
local machines while off the 'net without the timeout delay for
external domain names and I while on the 'net queries for external
domains work normally
```

- I have also received information about how bind interacts with NFS and the portmapper on a mostly offline machine from Karl-Max Wanger:

```
I use to run my own named on all my machines which are only
occasionally connected to the Internet by modem. The nameserver only
acts as a cache, it has no area of authority and asks back for
everything at the name servers in the root.cache file. As is usual with
Slackware, it is started before nfsd and mountd.

With one of my machines (a Libretto 30 notebook) I had the problem
that sometimes I could mount it from another system connected to my
local LAN, but most of the time it didn't work.  I had the same effect
regardless of using PLIP, a PCMCIA ethernet card or PPP over a serial
interface.

After some time of guessing and experimenting I found out that
apparently named messed with the process of registration nfsd and
mountd have to carry out with the portmapper upon startup (I start
these daemons at boot time as usual). Starting named after nfsd and
mountd eliminated this problem completely.

As there are no disadvantages to expect from such a modified boot
sequence I'd advise everybody to do it that way to prevent potential
trouble.
```

- Finally, there is HOWTO information about this at Ask Mr. DNS at http://www.acmebw.com/askmrdns/#linux-ns. It is about bind 4 though, so you have to adapt what he says to bind 8.

7. Where does the caching name server store its cache? Is there any way I can control the size of the cache?

The cache is completely stored in memory, it is *not* written to disk at any time. Every time you kill named the ca lost. The cache is *not* controllable in any way. named manages it according to some simple rules and that is it. Y cannot control the cache or the cache size in any way for any reason. If you want to you can ``fix'' this by hacking named. This is however not recommended.

8. Does named save the cache between restarts? Can I make it save it?

   No, named does *not* save the cache when it dies. That means that the cache must be built anew each time you kill and restart named. There is *no* way to make named save the cache in a file. If you want you can ``fix'' this by hacking named. This is however not recommended.

9. How can I get a domian? I want to set up my own domain called (for example) `linux-rules.net`. How can I get the domain I want assigned to me?

   Please contact your network service provider. They will be able to help you with this. Please note that in most parts of the world you need to pay money to get a domain.

---

9. How to become a bigger time DNS admin.

Documentation and tools.

Real Documentation exists. Online and in print. The reading of several of these is required to make the step from small time DNS admin to a big time one. In print the standard book is *DNS and BIND* by C. Liu and P. Albitz from O'Reilly & Associates, Sebastopol, CA, ISBN 0-937175-82-X. I read this, it's excellent, though based on bind 4, this is not a real problem though. There is also a section in on DNS in *TCP/IP Network Administration*, by Craig Hunt from O'Reilly..., ISBN 0-937175-82-X. Another must for Good DNS administration (or good anything for that matter) is *Zen and the Art of Motorcycle Maintenance* by Robert M. Pirsig :-) Available as ISBN 0688052304 and others.

Online you will find stuff on http://www.dns.net/dnsrd/ (DNS Resources Directory), http://www.isc.org/bind.html; A FAQ, a reference manual (BOG; Bind Operations Guide) as well as papers and protocol definitions and DNS hacks (these, and most, if not all, of the RFCs mentioned below, are also contained in the bind distribution). I have not read most of these, but then I'm not a big-time DNS admin either. Arnt Gulbrandsen on the other hand has read BOG and he's ecstatic about it :-). The newsgroup comp.protocols.tcp-ip.domains is about DNS. In addition there are a number of RFCs about DNS, the most important are probably these:

RFC 2052

A. Gulbrandsen, P. Vixie, *A DNS RR for specifying the location of services (DNS SRV)*, October 1996

RFC 1918

Y. Rekhter, R. Moskowitz, D. Karrenberg, G. de Groot, E. Lear, *Address Allocation for Private Internets*, 02/29/1996.

RFC 1912

D. Barr, *Common DNS Operational and Configuration Errors*, 02/28/1996.

RFC 1912 Errors

B. Barr *Errors in RFC 1912*, this is available at http://www.cis.ohio-state.edu/~barr/rfc1912-errors.html

RFC 1713

A. Romao, *Tools for DNS debugging*, 11/03/1994.

RFC 1712

C. Farrell, M. Schulze, S. Pleitner, D. Baldoni, *DNS Encoding of Geographical Location*, 11/01/1994.

RFC 1183

R. Ullmann, P. Mockapetris, L. Mamakos, C. Everhart, *New DNS RR Definitions*, 10/08/1990.

RFC 1035

P. Mockapetris, *Domain names - implementation and specification*, 11/01/1987.

RFC 1034

P. Mockapetris, *Domain names - concepts and facilities*, 11/01/1987.

RFC 1033

M. Lottor, *Domain administrators operations guide*, 11/01/1987.

RFC 1032

M. Stahl, *Domain administrators guide*, 11/01/1987.

RFC 974

C. Partridge, *Mail routing and the domain system*, 01/01/1986.

From DOS/Windows to Linux HOWTO

By Guido Gonzato, REMOVE_MEguido@ibogeo.df.unibo.it (Remove ``REMOVE_ME'')

v1.3.2, 22 February 1999

This HOWTO is dedicated to all the (soon to be former?) DOS and Windows users who have decided to switch to Linux, the free UNIX clone. The purpose of this document is to help the reader translate his or her knowledge of DOS and Windows into the Linux environment, as well as providing hints on exchanging files and resources between the two OSes.

1. Introduction

- 1.1 Is Linux Right for You?
- 1.2 It Is. Tell Me More
- 1.3 Conventions

2. For the Impatient

3. Meet bash

4. Files and Programs

- 4.1 Files: Preliminary Notions
- 4.2 Symbolic Links
- 4.3 Permissions and Ownership
- 4.4 Files: Translating Commands
- 4.5 Running Programs: Multitasking and Sessions
- 4.6 Running Programs on Remote Computers

TOC

5. Using Directories

- 5.1 Directories: Preliminary Notions
- 5.2 Directories Permissions
- 5.3 Directories: Translating Commands

6. Floppies, Hard Disks, and the Like

- 6.1 Managing Devices the DOS Way
- 6.2 Managing Devices the UNIX Way
- 6.3 Backing Up

7. What About Windows?

8. Tailoring the System

- 8.1 System Initialisation Files
- 8.2 Program Initialisation Files

9. Networking: Concepts

10. A Bit of Programming

- 10.1 Shell Scripts: .BAT Files on Steroids
- 10.2 C for Yourself

11. The Remaining 1%

- 11.1 Using tar and gzip
- 11.2 Installing Applications
- 11.3 Tips You Can't Do Without
- 11.4 Where to Find Applications
- 11.5 A Few Things You Couldn't Do
- 11.6 Practicing UNIX under DOS/Windows
- 11.7 Common Extensions and Related Programs

- 11.8 Converting Files
- 11.9 Free Office Suites

TOC

12. The End, for Now

- 12.1 Copyright
- 12.2 Disclaimer

---

1. Introduction

1.1 Is Linux Right for You?

You want to switch from the DOS world (this definition includes Windows) to Linux? Good idea: Linux is technically superior to DOS, Windows 9x and even Windows NT. But beware: it might not be useful for you. These are the main differences between DOS/Windows and Linux:

- Windows runs Microsoft Office and lots of games; is perceived to be easy to install and configure; is notoriously unstable; performs poorly; crashes are frequent.
- Linux runs StarOffice, scores of technical software and fewer games; can be difficult to install and configure; is rock solid; performs impeccably; crashes are extremely rare.

It's up to you to decide what you need. Furthermore, Linux gives you power, but it takes some time to learn how to harness it. Thus, if mostly need commercial sw, or if you don't feel like learning new commands and concepts, you had better look elsewhere. Be aware that many newcomers give up because of initial difficulties.

Work is underway to make Linux simpler to use, but *don't expect to be proficient with it unless you read a lot of documentation and use it at least for a few months*. Linux won't give you instant results. In spite of these warnings, I'm 100% confident that if you are the right user type you'll find in Linux your computer Nirvana. By the way, Linux + DOS/Win can coexist happily on the same machine.

Prerequisites for this howto: I'll assume that

- you know the basic DOS commands and concepts;
- Linux, possibly with X Window System (X11 for short), is properly installed on your PC;
- your shell (the equivalent of `COMMAND.COM`) is `bash`.

Unless specified, all information in this work is aimed at bad ol' DOS. There is information about Windows here and there, but bear in mind that Windows and Linux are totally different, unlike DOS that is sort of a UNIX poor relation.

Please also note that this work is neither a complete primer nor a configuration guide!

The latest version of this document is available in several formats on ftp://sunsite.unc.edu:/pub/Linux/docs/howto.

1.2 It Is. Tell Me More

You installed Linux and the programs you needed on the PC. You gave yourself an account (if not, type `adduser yourname` *now!*) and Linux is running. You've just entered your name and password, and now you are looking at the screen thinking: ``Well, now what?''

TOC

Now, don't despair. You're almost ready to do the same things you used to do with DOS/Win, and m more. If you were running DOS/Win instead of Linux, you would be doing some of the following ta

- running programs and creating, copying, viewing, deleting, printing, renaming files;
- CD'ing, MD'ing, RD'ing, and DIR'ring your directories;
- formatting floppies and copying files from/to them;
- tailoring the system;
- surfing the Internet;
- writing .BAT files and programs in your favourite language;
- the remaining 1%.

You'll be glad to know that these tasks can be accomplished under Linux in a fashion similar to DOS. Under DOS, the average user uses very few of the 100+ commands available: the same, up to a point, applies to Linux.

Introductory Concepts

The best way to learn something new is to get your feet wet. You are strongly encouraged to experiment and play with Linux: unless you login as ``root'', you can't damage the system that way. A few points:

- first of all, how to quit Linux safely. If you see a text mode screen, press <CTRL-ALT-DEL>, wait for the system to reboot, then switch off the PC. If you are working under X Window System, press <CTRL-ALT-BACKSPACE> first, then <CTRL-ALT-DEL>. *Never* switch off or reset the PC directly: this could damage the file system;
- unlike DOS or Windows, Linux has built-in security mechanisms. Files and directories have permissions associated to them; as a result, some cannot be accessed by the normal user; (see Section Permissions and Ownership). DOS and Windows, on the contrary, will let you wipe out the entire contents of your hard disk;
- there's a special user called ``root'': the system administrator, with full power of life and death on the machine. If you work on your own PC, you'll be root as well. Working as root is *dangerous*: any mistake can seriously damage or destroy the system just like with DOS/Win. Don't work as root unless absolutely necessary;
- much of the complexity of Linux comes from its extreme configurability: virtually every feature and every application can be tailored through one or more configuration files. Complexity is the price to pay for power;
- redirection and piping are a side DOS feature, a very inportant one and much more powerful under Linux. Simple commands can be strung together to accomplish complex tasks. I strongly suggest that you learn how to use them.

Getting Help

There are many ways to get help with Linux. The most important are:

- *reading the documentation*---I mean it. Although the HOWTO you are reading may serve as an introduction to Linux, there are several books that you really should read: Matt Welsh's ``Linux

Installation and Getting Started" (http://sunsite.unc.edu/mdw/LDP/gs/gs.html), Larry Greenf ``Linux User Guide" (ftp://sunsite.unc.edu/pub/Linux/docs/linux-doc-project/users-guide), and the Linux FAQ (http://sunsite.unc.edu/mdw/FAQ/Linux-FAQ.html). Feel a guilty conscience until you have read at least one of them;

- the documentation of the packages installed on the machine is often found in subdirectories under /usr/doc/;
- to get some help about the ``internal commands" of the shell, type `help` or, better, `man bash` or `info bash`;
- to get help about a command, type `man command` that invokes the manual (``man") page of `command`. Alternatively, type `info command` that invokes, if available, the info page pertinent of `command`; info is a hypertext-based documentation system, perhaps not intuitive to use at first. Finally, you may try `apropos command` or `whatis command`. With all of these commands, press `q' to exit.
- finally, on the Internet: the right place for getting help is Usenet, like news:comp.os.linux.setup. Please don't email me for help, because I'm quite overloaded.

1.3 Conventions

Throughout this work, examples will often follow the following format: `<...>` is a required argument, while `[...]` an optional one. Example:

```
$ tar -tf <file.tar> [> redir_file]
```

`file.tar` must be indicated, but redirection to `redir_file` is optional.

``RMP" means ``please Read the Man Pages for further information". I can't stress enough how important reading the documentation is. ``DOSWin" stands for ``DOS/Windows".

When the prompt of a command example is #, the command can only be performed by root.

---

2. For the Impatient

Want to strike out? Have a look at this table:

DOS	Linux	Notes
ATTRIB (+-)attr file	chmod <mode> file	completely different
BACKUP	tar -Mcvf device dir/	ditto
CD dirname\	cd dirname/	almost the same syntax
COPY file1 file2	cp file1 file2	ditto
DEL file	rm file	beware - no undelete
DELTREE dirname	rm -R dirname/	ditto
DIR	ls	not exactly the same syntax
DIR file /S	find . -name file	completely different
EDIT file	vi file	I think you won't like it
	jstar file	feels like dos' edit
EDLIN file	ed file	forget it
FORMAT	fdformat,	
	mount, umount	quite different syntax
HELP command	man command,	same philosophy
	info command	
MD dirname	mkdir dirname/	almost the same syntax
MORE < file	less file	much better
MOVE file1 file2	mv file1 file2	ditto
NUL	/dev/null	ditto
PRINT file	lpr file	ditto
PRN	/dev/lp0,	
	/dev/lp1	ditto
RD dirname	rmdir dirname/	almost the same syntax
REN file1 file2	mv file1 file2	not for multiple files
RESTORE	tar -Mxpvf device	different syntax
TYPE file	less file	much better
WIN	startx	poles apart!

If you need more than a table of commands, please refer to the following sections.

3. Meet bash

Good news: with Linux you type much less at the prompt, because the `bash` shell types for you whenever possible, and features cool line editing capabilities. To begin with, the arrow-up key recalls previous command lines; but there's more. Pressing <TAB> completes file and directory names, so typing

```
$ ls /uTABloTABbTAB
```

is like typing

```
$ ls /usr/local/bin
```

If there were ambiguities, as typing

```
$ ls /uTABloTABiTAB
```

`bash` stops because it doesn't know if you mean /usr/local/info or /usr/local/include. Supply more characters then press <TAB> again.

Other useful key presses are <ESC-BACKSPACE> that deletes a word to the left, while <ESC-D> deletes a word to the right; <ESC-F> moves the cursor one word to the right, <ESC-B> to the left; <CTRL-A> moves to the beginning of the line, <CTRL-E> to the end. The <ALT> key is equivalent to <ESC>.

Enough for now. Once you get used to these shortcuts, you'll find the DOS prompt very annoying...

4. Files and Programs

4.1 Files: Preliminary Notions

Linux has a structure of directories and files very similar to that of DOSWin. Files have filenames that obey special rules, are stored in directories, some are executable, and among these most have command switches. Moreover, you can use wildcard characters, redirection, and piping. There are only a few minor differences:

- under DOS, file names are in the so-called 8.3 form; e.g. `NOTENOUG.TXT`. Under Linux we can do better. If you installed Linux using a file system like ext2 or umsdos, you can use longer filenames (up to 255 characters), and with more than one dot: for example, `This_is.a.VERY_long.filename`. Please note that I used both upper and lower case characters: in fact...
- upper and lower case characters in file names or commands are different. Therefore, `FILENAME.tar.gz` and `filename.tar.gz` are two different files. `ls` is a command, `LS` is a mistake;
- Windows users, beware when using long file names under Linux. If a file name contains spaces (not recommended but possible), you must enclose the file name in double quotes whenever you refer to it. For example:

  ```
  $ # the following command makes a directory called "My old files"
  $ mkdir "My old files"
  $ ls
  My old files    bin     tmp
  ```

  Further, some characters shouldn't be used: some are `!*$&#`.
- there are no compulsory extensions like .COM and .EXE for programs, or .BAT for batch files. Executable files are marked by an asterisk `*' at the end of their name when you issue the `ls -F` command. For example:

  ```
  $ ls -F
  I_am_a_dir/   cindy.jpg    cjpg*   letter_to_Joe    my_1st_script*  old~
  ```

  The files `cjpg*` and `my_1st_script*` are executables, that is ``programs". Under DOS, backup files end in .BAK, while under Linux they end with a tilde `~'. Further, a file whose name starts with a dot is considered as hidden. Example: the file `.I.am.a.hidden.file` won't show up after the `ls` command;
- DOS program switches are obtained with `/switch`, Linux switches with `-switch` or `--switch`. Example: `dir /s` becomes `ls -R`. Note that many DOS programs, like `PKZIP` or `ARJ`, use UNIX-style switches.

You can now jump to Section Translating Commands from DOS to Linux, but if I were you I'd read on.

4.2 Symbolic Links

UNIX has a type of file that doesn't exist under DOS: the symbolic link. This can be thought of as a pointer to a file or to a directory, and can be used instead of the file or directory it points to; it's similar to Windows shortcuts. Examples of symbolic links are `/usr/X11`, which points to `/usr/X11R6`; `/dev/modem`, which points to either `/dev/ttyS0` or `/dev/ttyS1`.

To make a symbolic link:

```
$ ln -s <file_or_dir> <linkname>
```

Example:

```
$ ln -s /usr/doc/g77/DOC g77manual.txt
```

Now you can refer to `g77manual.txt` instead of `/usr/doc/g77/DOC`. Links appear like this in directory listings:

```
$ ls -F
g77manual.txt@
$ ls -l
(several things...)           g77manual.txt -> /usr/doc/g77/DOC
```

4.3 Permissions and Ownership

DOS files and directories have the following attributes: A (archive), H (hidden), R (read-only), and S (system). Only H and R make sense under Linux: hidden files start with a dot, and for the R attribute, read on.

Under UNIX a file has ``permissions'' and an owner, who in turn belongs to a ``group''. Look at this example:

```
$ ls -l /bin/ls
-rwxr-xr-x  1  root  bin  27281 Aug 15 1995 /bin/ls*
```

The first field contains the permissions of the file `/bin/ls`, which belongs to root, group bin. Leaving the remaining information aside, remember that `-rwxr-xr-x` means, from left to right:

`-` is the file type (`-` = ordinary file, `d` = directory, `l` = link, etc); `rwx` are the permissions for the file owner (read, write, execute); `r-x` are the permissions for the group of the file owner (read, execute); (I won't cover the concept of group, you can survive without it as long as you're a beginner ;-) `r-x` are the permissions for all other users (read, execute).

The directory `/bin` has permissions, too: see Section Directories Permissions for further details. This is why you can't delete the file `/bin/ls` unless you are root: you don't have the permission to do so. To change a file's permissions, the command is:

```
$ chmod <whoXperm> <file>
```

where who is `u` (user, that is owner), `g` (group), `o` (other), X is either `+` or `-`, perm is `r` (read), `w` (write), or `x` (execute). Common examples of `chmod` use are the following:

```
$ chmod +x file
```

this sets the execute permission for the file.

```
$ chmod go-rw file
```

this removes read and write permission for everyone but the owner.

```
$ chmod ugo+rwx file
```

this gives everyone read, write, and execute permission.

```
# chmod +s file
```

this makes a so-called ``setuid'' or ``suid'' file---a file that everyone can execute with its owner's privileges. Typically, you'll come across root suid files; these are often important system files, like the X server.

A shorter way to refer to permissions is with digits: `rwxr-xr-x` can be expressed as 755 (every letter corresponds to a

bit: `---` is 0, `--x` is 1, `-w-` is 2, `-wx` is 3...). It looks difficult, but with a bit of practice you'll understand the conc
root, being the superuser, can change everyone's file permissions. RMP.

TOC

4.4 Files: Translating Commands

On the left, the DOS commands; on the right, their Linux counterpart.

```
ATTRIB:         chmod
COPY:           cp
DEL:            rm
MOVE:           mv
REN:            mv
TYPE:           more, less, cat
```

Redirection and plumbing operators: `< > >> |`

Wildcards: `* ?`

`nul: /dev/null`

`prn, lpt1: /dev/lp0 or /dev/lp1; lpr`

Examples

```
DOS                                         Linux
-----------------------------------------------------------------------

C:\GUIDO>ATTRIB +R FILE.TXT                 $ chmod 400 file.txt
C:\GUIDO>COPY JOE.TXT JOE.DOC               $ cp joe.txt joe.doc
C:\GUIDO>COPY *.* TOTAL                     $ cat * > total
C:\GUIDO>COPY FRACTALS.DOC PRN              $ lpr fractals.doc
C:\GUIDO>DEL TEMP                           $ rm temp
C:\GUIDO>DEL *.BAK                          $ rm *~
C:\GUIDO>MOVE PAPER.TXT TMP\                $ mv paper.txt tmp/
C:\GUIDO>REN PAPER.TXT PAPER.ASC            $ mv paper.txt paper.asc
C:\GUIDO>PRINT LETTER.TXT                   $ lpr letter.txt
C:\GUIDO>TYPE LETTER.TXT                    $ more letter.txt
C:\GUIDO>TYPE LETTER.TXT                    $ less letter.txt
C:\GUIDO>TYPE LETTER.TXT > NUL              $ cat letter.txt > /dev/null
        n/a                                 $ more *.txt *.asc
        n/a                                 $ cat section*.txt | less
```

Notes:

- `*` is smarter under Linux: `*` matches all files except the hidden ones; `.*` matches all hidden files (but also the current directory `.' and parent directory `..': beware!); `*.*` matches only those that have a `.' in the middle or that end with a dot; `p*r` matches both `peter' and `piper'; `*c*` matches both `picked' and `peck';
- when using `more`, press <SPACE> to read through the file, `q' to exit. `less` is more intuitive and lets you use the arrow keys;
- there is no `UNDELETE`, so *think twice* before deleting anything;
- in addition to DOS' `< > >>`, Linux has `2>` to redirect error messages (stderr); moreover, `2>&1` redirects stderr to stdout, while `1>&2` redirects stdout to stderr;

- Linux has another wildcard: the `[]`. Usage: `[abc]*` matches files starting with a, b, c; `*[I-N1-3]` matches ending with I, J, K, L, M, N, 1, 2, 3;
- `lpr` <file> prints a file in background. To check the status of the print queue, use `lpq`; to remove a file from the print queue, use `lprm`;
- there is no DOS-like `RENAME`; that is, `mv *.xxx *.yyy` won't work. A REN-like command is available on ftp://sunsite.unc.edu/pub/Linux/utils/file;
- use `cp -i` and `mv -i` to be warned when a file is going to be overwritten.

4.5 Running Programs: Multitasking and Sessions

To run a program, type its name as you would do under DOS. If the directory (Section Using Directories) where the program is stored is included in the PATH (Section System Initialisation Files), the program will start. Exception: unlike DOS, under Linux a program located in the current directory won't run unless the directory is included in the PATH. Escamotage: being `prog` your program, type `./prog`.

This is what the typical command line looks like:

```
$ command [-s1 [-s2] ... [-sn]] [par1 [par2] ... [parn]] [< input] [> output]
```

where `-s1`, ..., `-sn` are the program switches, `par1`, ..., `parn` are the program parameters. You can issue several commands on the command line:

```
$ command1 ; command2 ; ... ; commandn
```

That's all about running programs, but it's easy to go a step beyond. One of the main reasons for using Linux is that it is a multitasking os---it can run several programs (from now on, processes) at the same time. You can launch processes in background and continue working straight away. Moreover, Linux lets you have several sessions: it's like having many computers to work on at once!

- To switch to session 1..6 on the virtual consoles, press <ALT-F1> ... <ALT-F6>
- To start a new session in the same v.c. without leaving the current one, type `su - <loginname>`. Example: `su - root`. This is useful, for instance, when you need to perform a task that only root can do.
- To end a session, type `exit`. If there are stopped jobs (see later), you'll be warned.
- To launch a process in background, add an ampersand '&' at the end of the command line:

  ```
  $ progname [-switches] [parameters] [< input] [> output] &
  [1] 123
  ```

  the shell identifies the process with a job number (e.g. `[1]`; see below), and with a PID (Process Identification Number; 123 in our example).
- To see how many processes there are, type `ps ax`. This will output a list of currently running processes.
- To kill (terminate) a process, type `kill <PID>`. You may need to kill a process when you don't know how to quit it the right way.... Unless you're root, you can't kill other people's processes. Sometimes, a process will only be killed by `kill -SIGKILL <PID>`. In addition, the shell allows you to stop or temporarily suspend a process, send a process to background, and bring a process from background to foreground. In this context, processes are called ``jobs''.
- To see how many jobs there are, type `jobs`. Here the jobs are identified by their job number, not by their PID.
- To stop a process running in foreground, press <CTRL-C> (it won't always work).
- To suspend a process running in foreground, press <CTRL-Z> (ditto).
- To send a suspended process into background, type `bg <%job>` (it becomes a job).

- To bring a job to foreground, type `fg <%job>`. To bring to foreground the last job sent to background, simp type `fg`.
- To kill a job, type `kill <%job>` where <job> may be 1, 2, 3,...

Using these commands you can format a disk, zip a bunch of files, compile a program, and unzip an archive all at the same time, and still have the prompt at your disposal. Try this with Windows, just to see the difference in performance (if it doesn't crash, of course).

4.6 Running Programs on Remote Computers

To run a program on a remote machine whose name is `remote.machine.edu`:

```
$ telnet remote.machine.edu
```

After logging in, start your favourite program. Needless to say, you must have a shell account on the remote machine.

If you have X11, you can even run an X application on a remote computer, displaying it on your X screen. Let `remote.machine.edu` be the remote X computer and let `local.linux.box` be your Linux machine. To run from `local.linux.box` an X program that resides on `remote.machine.edu`, do the following:

- fire up X11, start an `xterm` or equivalent terminal emulator, then type:

  ```
  $ xhost +remote.machine.edu
  $ telnet remote.machine.edu
  ```

- after logging in, type:

  ```
  remote:$ DISPLAY=local.linux.box:0.0
  remote:$ progname &
  ```

  (instead of `DISPLAY...`, you may have to write: `setenv DISPLAY local.linux.box:0.0`. It depends on the remote shell.)

Et voila! Now `progname` will start on `remote.machine.edu` and will be displayed on your machine. Don't try this over the modem though, for it's too slow to be usable. Moreover, this is a crude and insecure method: please read the ``Remote X Apps mini-HOWTO'' at http://sunsite.unc.edu/LDP/HOWTO/mini/Remote-X-Apps.

---

5. Using Directories

5.1 Directories: Preliminary Notions

We have seen the differences between files under DOSWin and Linux. As for directories, under DOSWin the root directory is `\`, under Linux it is `/`. Similarly, nested directories are separated by `\` under DOSWin, by `/` under Linux. Example of file paths:

```
DOS:    C:\PAPERS\GEOLOGY\MID_EOC.TEX
Linux:  /home/guido/papers/geology/middle_eocene.tex
```

As usual, `..` is the parent directory and `.` is the current directory. Remember that the system won't let you `cd`, `rd`, or `md` everywhere you want. Each user has his or her stuff in a directory called `home', given by the system administrator; for instance, on my PC my home dir is `/home/guido`.

5.2 Directories Permissions

Directories, too, have permissions. What we have seen in Section Permissions and Ownership applies to directories as well (user, group, and other). For a directory, `rx` means you can `cd` to that directory, and `w` means that you can delete a file in the directory (according to the file's permissions, of course), or the directory itself.

For example, to prevent other users from snooping in `/home/guido/text`:

```
$ chmod o-rwx /home/guido/text
```

5.3 Directories: Translating Commands

```
DIR:            ls, find, du
CD:             cd, pwd
MD:             mkdir
RD:             rmdir
DELTREE:        rm -rf
MOVE:           mv
```

Examples

```
DOS                                         Linux
-----------------------------------------------------------------------
```

TOC

`C:\GUIDO>DIR`	`$ ls`
`C:\GUIDO>DIR FILE.TXT`	`$ ls file.txt`
`C:\GUIDO>DIR *.H *.C`	`$ ls *.h *.c`
`C:\GUIDO>DIR/P`	`$ ls | more`
`C:\GUIDO>DIR/A`	`$ ls -l`
`C:\GUIDO>DIR *.TMP /S`	`$ find / -name "*.tmp"`
`C:\GUIDO>CD`	`$ pwd`
`n/a - see note`	`$ cd`
`ditto`	`$ cd ~`
`ditto`	`$ cd ~/temp`
`C:\GUIDO>CD \OTHER`	`$ cd /other`
`C:\GUIDO>CD ..\TEMP\TRASH`	`$ cd ../temp/trash`
`C:\GUIDO>MD NEWPROGS`	`$ mkdir newprogs`
`C:\GUIDO>MOVE PROG ..`	`$ mv prog ..`
`C:\GUIDO>MD \PROGS\TURBO`	`$ mkdir /progs/turbo`
`C:\GUIDO>DELTREE TEMP\TRASH`	`$ rm -rf temp/trash`
`C:\GUIDO>RD NEWPROGS`	`$ rmdir newprogs`
`C:\GUIDO>RD \PROGS\TURBO`	`$ rmdir /progs/turbo`

Notes:

- when using `rmdir`, the directory to remove must be empty. To delete a directory and all of its contents, use `rm -rf` (at your own risk).
- the character `~' is a shortcut for the name of your home directory. The commands `cd` or `cd ~` will take you to your home directory from wherever you are; the command `cd ~/tmp` will take you to `/home/your_home/tmp`.
- `cd -` ``undoes" the last `cd`.

---

6. Floppies, Hard Disks, and the Like

There are two ways to manage devices under Linux: the DOS way and the UNIX way. Take your pick.

6.1 Managing Devices the DOS Way

Most Linux distributions include the Mtools suite, a set of commands that are perfectly equivalent to their DOS counterpart, but start with an `m': i.e., `mformat`, `mdir`, `mdel`, `mmd`, and so on. They can even preserve long file names, but not file permissions. If you configure Mtools editing a file called /etc/mtools.conf (a sample is provided in the distribution), you can also access the DOS/Win partition, the CD--ROM, and the Zip drive. To format a fresh disk though, the `mformat` command won't do. As root, you'll have to issue this command beforehand: `fdformat /dev/fd0H1440`.

You can't access files on the floppy with a command like, say, `less a:file.txt`! This is the disadvantage of the DOS way of accessing disks.

6.2 Managing Devices the UNIX Way

UNIX has a different way to handle devices. There are no separate volumes like A: or C:; a disk, be it a floppy or whatever, becomes part of the local file system through an operation called ``mounting''. When you're done using the disk, before extracting it you must ``unmount'' it.

Physically formatting a disk is one thing, making a file system on it is another. The DOS command `FORMAT A:` does both things, but under Linux there are separate commands. To format a floppy, see above; to create a file system:

```
# mkfs -t ext2 -c /dev/fd0H1440
```

You can use `dos`, `vfat` (recommended) or other formats instead of `ext2`. Once the disk is prepared, mount it with the command

```
# mount -t ext2 /dev/fd0 /mnt
```

specifying the right file system if you don't use `ext2`. Now you can address the files in the floppy using `/mnt` instead of A: or B:. Examples:

```
DOS                                       Linux
-----------------------------------------------------------------------

C:\GUIDO>DIR A:                           $ ls /mnt
C:\GUIDO>COPY A:*.*                       $ cp /mnt/* .
C:\GUIDO>COPY *.ZIP A:                    $ cp *.zip /mnt
C:\GUIDO>EDIT A:FILE.TXT                  $ jstar /mnt/file.txt
C:\GUIDO>A:                               $ cd /mnt
A:> _                                     /mnt/$ _
```

When you've finished, before extracting the disk you *must* unmount it with the command

```
# umount /mnt
```

Obviously, you have to `fdformat` and `mkfs` only unformatted disks, not previously used ones. If you want to use the drive B:, refer to `fd1H1440` and `fd1` instead of `fd0H1440` and `fd0` in the examples above.

Needless to say, what applies to floppies also applies to other devices; for instance, you may want to mount another hard disk or a CD--ROM drive. Here's how to mount the CD--ROM:

```
# mount -t iso9660 /dev/cdrom /mnt
```

This was the ``official'' way to mount your disks, but there's a trick in store. Since it's a bit of a nuisance having to be root to mount a floppy or a CD--ROM, every user can be allowed to mount them this way:

- as root, do the following:

```
# mkdir /mnt/floppy ; mkdir /mnt/cdrom
# chmod 777 /mnt/floppy /mnt/cd*
# # make sure that the CD-ROM device is right
# chmod 666 /dev/hdb ; chmod 666 /dev/fd*
```

- add in /etc/fstab the following lines:

```
/dev/cdrom      /mnt/cdrom  iso9660 ro,user,noauto          0       0
/dev/fd0        /mnt/floppy vfat    user,noauto             0       0
```

Now, to mount a DOS floppy and a CD--ROM:

```
$ mount /mnt/floppy
$ mount /mnt/cdrom
```

/mnt/floppy and /mnt/cdrom can now be accessed by every user. Remember that allowing everyone to mount disks this way is a gaping security hole, if you care.

Two useful commands are `df`, which gives information on the mounted file systems, and `du dirname` which reports the disk space consumed by the directory.

6.3 Backing Up

There are several packages to help you, but the very least you can do for a multi-volume backup is (as root):

```
# tar -M -cvf /dev/fd0H1440 dir_to_backup/
```

Make sure to have a formatted floppy in the drive, and several more ready. To restore your stuff, insert the first floppy in the drive and do:

```
# tar -M -xpvf /dev/fd0H1440
```

7. What About Windows?

The ``equivalent'' of Windows is the graphic system X Window System. Unlike Windows or the Mac, X11 wasn't designed for ease of use or to look good, but just to provide graphic facilities to UNIX workstations. These are the main differences:

- while Windows looks and feels the same all over the world, X11 does not: it's much more configurable. X11's overall look is given by a key component called ``window manager'', of which you have a wide choice: `fvwm`, basic but nice and memory efficient, `fvwm2-95`, `Afterstep`, `WindowMaker`, `Enlightenment`, and many more. The w.m. is usually invoked from `.xinitrc`;
- your w.m. can be configured so as a window acts as under, er, Windows: you click on it and it comes to foreground. Another possibility is that it comes to foreground when the mouse moves over it (``focus''). Also, the placement of windows on the screen can be automatic or interactive: if a strange frame appears instead of your program, left click where you want it to appear;
- most features can be tailored editing one or more configuration files. Read the docs of your w.m.: the configuration file can be `.fvwmrc`, `.fvwm2rc95`, `.steprc`, etc. A sample configuration file is typically found in /etc/X11/window-manager-name/system.window-manager-name;
- X11 applications are written using special libraries (``widget sets''); as several are available, applications look different. The most basic ones are those that use the Athena widgets (2--D look; `xdvi`, `xman`, `xcalc`); others use Motif (`netscape`), others still use Tcl/Tk, XForms, Qt, Gtk, and what have you. Nearly all of these libraries provide roughly the same look and feel as Windows;
- the feel, unfortunately, can be incoherent. For instance, if you select a line of text using the mouse and press <BACKSPACE>, you'd expect the line to disappear, right? This doesn't work with Athena--based apps, but it does with other widget sets;
- how scrollbars and resizing work depends on the w.m. and the widget set. Tip: if you find that the scrollbars don't behave as you would expect, try using the central button or the two buttons together to move them;
- applications don't have an icon by default, but they can have many. Most w.m. feature a menu you recall by clicking on the desktop (``root window''); needless to say, the menu can be tailored. To change the root window appearance, use `xsetroot` or `xloadimage`;
- the clipboard can only contain text, and behaves strange. Once you've selected text, it's already copied to the clipboard: move elsewhere and press the central button to paste it. There's an application, `xclipboard`, that provides for multiple clipboard buffers;
- drag and drop is an option, and is only available if you use X11 applications and/or w.m. that support it.

To save memory, one should use applications that use the same libraries, but this is difficult to do in practice.

There are projects that aim at making X11 look and behave as coherently as Windows. Gnome, http://www.gnome.org and KDE, http://www.kde.org, are awesome. Give them a try: you won't regr your Windows desktop anymore.

8. Tailoring the System

8.1 System Initialisation Files

Two important files under DOS are `AUTOEXEC.BAT` and `CONFIG.SYS`, which are used at boot time to initialise the system, set some environment variables like PATH and FILES, and possibly launch a program or batch file. Under Linux there are lots of initialisation files, some of which you had better not tamper with until you know exactly what you are doing. I'll tell you what the most important are, anyway:

```
FILES                                   NOTES

/etc/inittab                            don't touch for now!
/etc/rc.d/*                             ditto
```

If all you need is setting the PATH and other environment variables, or you want to change the login messages or automatically launch a program after the login, have a look at the following files:

```
FILES                                   NOTES

/etc/issue                              sets pre-login message
/etc/motd                               sets post-login message
/etc/profile                            sets $PATH and other variables, etc.
/etc/bashrc                             sets aliases and functions, etc.
/home/your_home/.bashrc                 sets your aliases + functions
/home/your_home/.bash_profile   or
/home/your_home/.profile                sets environment + starts your progs
```

If the latter file exists (note that it is a hidden file), it will be read after the login, and the commands therein will be executed.

Example---look at this `.bash_profile`:

```
# I am a comment
echo Environment:
printenv | less   # equivalent of command SET under DOS
alias d='ls -l'   # easy to understand what an alias is
alias up='cd ..'
echo "I remind you that the path is "$PATH
echo "Today is `date`"  # use the output of the command 'date'
echo "Have a good day, "$LOGNAME
# The following is a "shell function"
ctgz() # List the contents of a .tar.gz archive.
{
  for file in $*
  do
    gzip -dc ${file} | tar tf -
  done
```

```
}
# end of .profile
```

$PATH and $LOGNAME, you guessed right, are environment variables. There are many others to play with; for instance, RMP for apps like less or bash.

Putting this line in your /etc/profile will provide the rough equivalent of PROMPT PG:

```
export PS1="\w\\$ "
```

8.2 Program Initialisation Files

Under Linux, virtually everything can be tailored to your needs. Most programs have one or more initialisation files you can fiddle with, often as a .prognamerc in your home dir. The first ones you'll want to modify are:

- .inputrc: used by bash to define key bindings;
- .xinitrc: used by startx to initialise X Window System;
- .fvwmrc: used by the window manager fvwm.
- .joerc, .jstarrc: used by the editor joe;
- .jedrc: used by the editor jed;
- .pinerc: used by the mail reader pine;
- .Xdefault: used by many X programs.

For all of these and the others you'll come across sooner or later, RMP. Perhaps I could interest you in the Configuration HOWTO, http://sunsite.unc.edu/mdw/HOWTO/Config-HOWTO.html?

---

9. Networking: Concepts

Not only is ``Dialup Networking'' available under Linux, it's also more stable and quicker. The name of the game is ``PPP'', the protocol employed for connecting to the Internet using modems. You need a kernel that supports PPP and a tool that dials and makes the connection.

To retrieve your mail from the ISP's server you need a tool called ``email fetcher'' that uses the POP protocol; when the mail is fetched it will appear as though it had been directly delivered to your Linux box. You'll then use a MUA (Mail User Agent) like `pine`, `mutt`, `elm` or many others to manage it.

While under Windows the dialer is automatically invoked when you launch an Internet application, under Linux the path is the other way round: you dial first, then launch the application. A thing called `diald` provides the usual behaviour. Installing and configuring dialup networking used to be one of the most difficult things to do under Linux, but not anymore: please consult the Configuration HOWTO.

Finally, a word about ``Network neighborhood'': you can make your Linux workstation appear as Windows NT/9x in a local network of Windows machines! The magic word is Samba: not the lively Brazilian dance, but an implementation of the SMB protocol for Linux. Go to http://samba.anu.edu.au/samba.

10. A Bit of Programming

10.1 Shell Scripts: .BAT Files on Steroids

If you used .BAT files to create shortcuts of long command lines (I did a lot), this goal can be attained by inserting appropriate alias lines (see example above) in `profile` or `.bash_profile`. But if your .BATs were more complicated, then you'll love the scripting language made available by the shell: it's as powerful as good ol' QBasic, if not more. It has variables, structures like while, for, case, if... then... else, and lots of other features: it can be a good alternative to a ``real'' programming language.

To write a script---the equivalent of a .BAT file under DOS---all you have to do is write a standard ASCII file containing the instructions, save it, then make it executable with the command `chmod +x <scriptfile>`. To execute it, type its name.

A word of warning. The system editor is called `vi`, and in my experience most new users find it very difficult to use. I'm not going to explain how to use it; please consult Matt Welsh's book or search for a tutorial on the net. Suffice it here to say that:

- to insert some text, type `i` then your text;
- to delete characters, type <ESC> then `x`;
- to quit `vi` whithout saving, type <ESC> then `:q!`
- to save and quit, type <ESC> then `:wq`.

A good beginner editor is `joe`: invoking it by typing `jstar` you'll get the same key bindings as the DOSWin editor. `jed` in WordStar or IDE mode is even better. Please consult Section Where to Find Applications to see where to get these editors.

Writing scripts under `bash` is such a vast subject it would require a book by itself, and I will not delve into the topic any further. I'll just give you an example of shell script, from which you can extract some basic rules:

```
#!/bin/sh
# sample.sh
# I am a comment
# don't change the first line, it must be there
echo "This system is: `uname -a`" # use the output of the command
echo "My name is $0" # built-in variables
echo "You gave me the following $# parameters: "$*
echo "The first parameter is: "$1
echo -n "What's your name? " ; read your_name
echo notice the difference: "hi $your_name" # quoting with "
```

TOC

```
echo notice the difference: 'hi $your_name' # quoting with '
DIRS=0 ; FILES=0
for file in `ls .` ; do
  if [ -d ${file} ] ; then # if file is a directory
    DIRS=`expr $DIRS + 1`  # DIRS = DIRS + 1
  elif [ -f ${file} ] ; then
    FILES=`expr $FILES + 1`
  fi
  case ${file} in
    *.gif|*jpg) echo "${file}: graphic file" ;;
    *.txt|*.tex) echo "${file}: text file" ;;
    *.c|*.f|*.for) echo "${file}: source file" ;;
    *) echo "${file}: generic file" ;;
  esac
done
echo "there are ${DIRS} directories and ${FILES} files"
ls | grep "ZxY--!!!WKW"
if [ $? != 0 ] ; then # exit code of last command
  echo "ZxY--!!!WKW not found"
fi
echo "enough... type 'man bash' if you want more info."
```

10.2 C for Yourself

Under UNIX, the system language is C, love it or hate it. Scores of other languages (Java, FORTRAN, Pascal, Lisp, Basic, Perl, awk...) are also available.

Taken for granted that you know C, here are a couple of guidelines for those of you who have been spoilt by Turbo C++ or one of its DOS kin. Linux's C compiler is called `gcc` and lacks all the bells and whistles that usually accompany its DOS counterparts: no IDE, on-line help, integrated debugger, etc. It's just a rough command-line compiler, very powerful and efficient. To compile your standard `hello.c` you'll do:

```
$ gcc hello.c
```

which will create an executable file called `a.out`. To give the executable a different name, do

```
$ gcc -o hola hello.c
```

To link a library against a program, add the switch -l<libname>. For example, to link in the math library:

```
$ gcc -o mathprog mathprog.c -lm
```

(The `-l<libname>` switch forces `gcc` to link the library `/usr/lib/lib<libname>.so`; so `-lm` links `/usr/lib/libm.so`).

So far, so good. But when your prog is made of several source files, you'll need to use the utility `make`. Let's suppose you have written an expression parser: its source file is called `parser.c` and #includes two header files, `parser.h` and `xy.h`. Then you want to use the routines in `parser.c` in a program, say, `calc.c`, which in turn #includes `parser.h`. What a mess! What do you have to do to compile `calc.c`?

You'll have to write a so-called `Makefile`, which teaches the compiler the dependencies between sources and objects files. In our example:

```
# This is Makefile, used to compile calc.c
# Press the <TAB> key where indicated!

calc: calc.o parser.o
<TAB>gcc -o calc calc.o parser.o -lm
# calc depends on two object files: calc.o and parser.o

calc.o: calc.c parser.h
<TAB>gcc -c calc.c
# calc.o depends on two source files

parser.o:  parser.c parser.h xy.h
<TAB>gcc -c parser.c
# parser.o depends on three source files

# end of Makefile.
```

Save this file as `Makefile` and type `make` to compile your program; alternatively, save it as `calc.mak` and type `make -f calc.mak`, and of course RMP. You can invoke some help about the C functions, that are covered by man pages, section 3; for example,

```
$ man 3 printf
```

To debug your programs, use `gdb`. `info gdb` to learn how to use it.

There are lots of libraries available; among the first you'll want to use are `ncurses` (textmode effects), and `svgalib` (console graphics). If you feel brave enough to tackle X11 programming (it's not that difficult), there are several libraries that make writing X11 programs a breeze. Have a look at http://www.xnet.com/~blatura/linapp6.html, bearing in mind that Gtk is becoming a Linux standard.

Many editors can act as an IDE; `emacs` and `jed`, for instance, also feature syntax highlighting, automatic indent, and so on. Alternatively, get the package `rhide` from ftp://sunsite.unc.edu:/pub/Linux/devel/debuggers/. It's a Borland IDE clone, and chances are that you'll like it.

---

11. The Remaining 1%

Much more than 1%, actually...

11.1 Using tar and gzip

Under UNIX there are some widely used applications to archive and compress files. `tar` is used to make archives---it's like `PKZIP` or `Winzip`but it doesn't compress, it only archives. To make a new archive:

```
$ tar -cvf <archive_name.tar> <file> [file...]
```

To extract files from an archive:

```
$ tar -xpvf <archive_name.tar> [file...]
```

To list the contents of an archive:

```
$ tar -tf <archive_name.tar> | less
```

You can compress files using `compress`, which is obsolete and shouldn't be used any more, or `gzip`:

```
$ compress <file>
$ gzip <file>
```

that creates a compressed file with extension `.Z` (`compress`) or `.gz` (`gzip`). These programs can compress only one file at a time. To decompress:

```
$ compress -d <file.Z>
$ gzip -d <file.gz>
```

RMP.

There are also the `unarj`, `zip` and `unzip` (PK??ZIP compatible) utilities. Files with extension `.tar.gz` or `.tgz` (archived with `tar`, then compressed with `gzip`) are as common in the UNIX world as .ZIP files are under DOS. Here's how to list the contents of a `.tar.gz` archive:

```
$ tar -ztf <file.tar.gz> | less
```

11.2 Installing Applications

First of all: installing packages is root's work. Most Linux applications are distributed as a `.tar.gz` archive, which typically will contain a directory aptly named containing files and/or subdirectories. A good rule is to install these packages from `/usr/local` with the command

```
# tar -zxf <archive.tar.gz>
```

TOC

reading then the README or INSTALL file. In most cases, the application is distributed in source, which you'll have to compile; often, typing `make` then `make install` will suffice. If the archive contains a `configure` script, run it first. Obviously, you'll need the `gcc` or `g++` compiler.

Other archives have to be unpacked from /; this is the case with Slackware's `.tgz` archives. Other archives contain the files but not a subdirectory - careful not to mess things up!. Always list the contents of the archive before installing it.

Debian and Red Hat have their own archive format; respectively, `.deb` and `.rpm`. The latter is gaining wide acceptance; to install an `rpm` package, type

```
# rpm -i package.rpm
```

11.3 Tips You Can't Do Without

Backscrolling: pressing <SHIFT + PAG UP> (the grey key) allows you to backscroll a few pages, depending on how much video memory you have.

Resetting the screen: if you happen to `more` or `cat` a binary file, your screen may end up full of garbage. To fix it, blind type `reset` or this sequence of characters: `echo CTRL-V ESC c RETURN`.

Pasting text: in console, see below; in X, click and drag to select the text in an `xterm` window, then click the middle button (or the two buttons together if you have a two-button mouse) to paste. There is also `xclipboard` (alas, only for text); don't get confused by its very slow response.

Using the mouse: if you installed `gpm`, a mouse driver for the console, you can click and drag to select text, then right click to paste the selected text. It works across different VCs.

Messages from the kernel: have a look at `/var/adm/messages` or `/var/log/messages` as root to see what the kernel has to tell you, including bootup messages. The command `dmesg` is also handy.

11.4 Where to Find Applications

If you're wondering whether you can replace your old and trusted DOS/Win application with a Linux one, I suggest that you browse the main Linux software repositories: ftp://sunsite.unc.edu/pub/Linux, ftp://tsx-11.mit.edu/pub/linux, and ftp://ftp.funet.fi/pub/Linux. Other good starting places are the ``Linux Applications and Utilities Page'' http://www.xnet.com/~blatura/linapps.shtml, and the ``official'' Linux page http://www.linux.org.

11.5 A Few Things You Couldn't Do

Linux can do an awful lot of things that were cumbersome, difficult or impossible do to with DOS/Windows. Here's a short list that may whet your appetite:

- `at` allows you to run programs at a specified time;
- `awk` is a simple yet powerful language to manipulate data files (and not only). For example, being `data.dat` your multi field data file,

  ```
  $ awk '$2 ~ "abc" {print $1, "\t", $4}' data.dat
  ```

  prints out fields 1 and 4 of every line in `data.dat` whose second field contains ``abc''.

TOC

- `cron` is useful to perform tasks periodically, at specified date and time. Type `man 5 crontab`.
- `file <filename>` tells you what `filename` is (ASCII text, executable, archive, etc.);
- `find` (see also Section Directories: Translating Commands) is one of the most powerful and useful commands. It's used to find files that match several characteristics and perform actions on them. General use of `find` is:

  ```
  $ find <directory> <expression>
  ```

  where <expression> includes search criteria and actions. Examples:

  ```
  $ find . -type l -exec ls -l {} \;
  ```

  finds all the files that are symbolic links and shows what they point to.

  ```
  $ find / -name "*.old" -ok rm {} \;
  ```

  finds all the files matching the pattern and deletes them, asking for your permission first.

  ```
  $ find . -perm +111
  ```

  finds all the files whose permissions match 111 (executable).

  ```
  $ find . -user root
  ```

  finds all the files that belong to root. Lots of possibilities here---RMP.
- `grep` finds text patterns in files. For example,

  ```
  $ grep -l "geology" *.tex
  ```

  lists the files *.tex that contain the word ``geology''. The variant `zgrep` works on gzipped files. RMP;
- **regular expressions** are a complex but darn powerful way of performing search operations on text. For example, `^a[^a-m]X{4,}txt$` matches a line that starts with `a', followed by any character except those in the interval a-m, followed by 4 or more `X', and ends in `txt'. You use regular expressions with advanced editors, `less`, and many other applications. `man grep` for an introduction.
- `script <script_file>` dumps the screen contents on `script_file` until you issue the command `exit`. Useful for debugging;
- `sudo` allows users to perform some of root's tasks (e.g. formatting and mounting disks; RMP);
- `uname -a` gives you info about your system;
- `zcat` and `zless` are useful for browsing and piping gzipped files without decompressing them. For example:

  ```
  $ zless textfile.gz
  $ zcat textfile.gz | lpr
  ```

- The following commands often come in handy: `bc, cal, chsh, cmp, cut, fmt, head, hexdump, nl, passwd, printf, sort, split, strings, tac, tail, tee, touch, uniq, w, wall, wc, whereis, write, xargs, znew`. RMP.

11.6 Practicing UNIX under DOS/Windows

Believe it or not, there are fine tools that provide a UNIX-like environment under DOS/Windows! One is the Djgpp suite (http://www.delorie.com/djgpp) for DOS, while Cygnus (http://www.cygnus.com) is a more complex port for

TOC

Win32. Both include the same GNU development tools and utilities as Linux; you won't get the same stability a performance, though.

If you'd like to have a taste of Linux, try out Djgpp. Download and install the following files (as of this writing, the latest version is 2.02): `djdev202.zip, bnu281b.zip, bsh1147b.zip, fil316b.zip, find41b.zip, grep22b.zip, gwk303b.zip, lss332b.zip, shl112b.zip.`. Installation instructions are provided, and you can find assistance on news:comp.os.msdos.djgpp.

In particular, using `bash` under DOSWin is a whiff of fresh air. To configure it properly, edit the supplied file `BOOT.BAT` to reflect your installation, then put these files in your home directory (in the Windows partition) instead of those provided:

```
# this is _bashrc

LS_OPTIONS="-F -s --color=yes"
alias cp='cp -i'
alias d='ls -l'
alias l=less
alias ls="ls $LS_OPTIONS"
alias mv='mv -i'
alias rm='rm -i'
alias u='cd ..'

# this is _bprof
if [ -f ~/_bashrc ]; then
  . ~/_bashrc
fi
PS1='\w\$ '
PS2='> '
CDPATH="$CDPATH:~"
# stuff for less(1)
LESS="-M-Q"                     # long prompt, silent
LESSEDIT="%E ?lt+%lt. %f"       # edit top line
VISUAL="jed"                    # editor
LESSCHARSET=latin1              # visualise accented letters
export PS1 PS2 CDPATH LS_OPTIONS LESS LESSEDIT LESSOPEN VISUAL LESSCHARSET
```

11.7 Common Extensions and Related Programs

You may come across scores of file extensions. Excluding the more exotic ones (i.e. fonts, etc.), here's a list of who's what:

- `1 ... 8`: man pages. Read them with `groff -Tascii -man <file.1>`.
- `arj`: archive made with `arj`.
- `dvi`: output file produced by TeX (see below). `xdvi` to visualise it; `dvips` to turn it into a PostScript `.ps` file.
- `gz`: archive made with `gzip`.
- `info`: info file (sort of alternative to man pages). Get `info`.
- `lsm`: Linux Software Map file. It's a plain ASCII file containing the description of a package.

TOC

- `ps`: PostScript file. To visualise or print it get `gs` and, optionally, `ghostview` or `gv`.
- `rpm`: Red Hat package. You can install it on any system using the package manager `rpm`.
- `taz, tar.Z`: archive made with `tar` and compressed with `compress`.
- `tgz, tar.gz`: archive made with `tar` and compressed with `gzip`.
- `tex`: text file to submit to TeX, a powerful typesetting system. Get the package `tex`, available in many distributions.
- `texi`: texinfo file, can produce both TeX and info files (cp. `info`). Get `texinfo`.
- `xbm, xpm, xwd`: graphic file.
- `Z`: archive made with `compress`.

11.8 Converting Files

If you need to exchange text files between DOS/Win and Linux, be aware of the ``end of line'' problem. Under DOS, each line of text ends with CR/LF (that is, ASCII 13 + ASCII 10), with LF under Linux. If you edit a DOS text file under Linux, each line will likely end with a strange--looking `M' character; a Linux text file under DOS will appear as a kilometric single line with no paragraphs. There are a couple of tools, `dos2unix` and `unix2dos`, to convert the files.

If your text--only files contain accented characters, make sure they are made under Windows (with Notepad) and not under plain DOS; otherwise, all accented characters will be screwed up.

To convert Word or WordPerfect files to plain text, the matter is a bit trickier but possible. You'll need one of the tools that can be found on the CTAN sites; one is ftp://ftp.tex.ac.uk. Get the program `word2x` from the directory /pub/tex/tools/, or try one the programs available in directory /pub/tex/support/. In particular, `word2x` converts Word 6 files, while for Word 97 files you'll need `mswordview` (http://www.csn.ul.ie/~caolan/docs/MSWordView.html) that turns them to HTML.

11.9 Free Office Suites

If converting files is not enough, you have the choice of sinking your teeth into a (free!) Microsoft Office--like package.

The StarOffice suite is free for private use. It's big, somewhat slow, but very good anyway: it offers a lot of functionality not found in Microsoft Office. It can also read and write Word and Excel files, although the conversion isn't always perfect. Home page: http://www.stardivision.com.

Another good package is Corel WordPerfect, a free edition of which is available for download. Should I say more? Go fetch it: http://www.corel.com.

---

12. The End, for Now

Congratulations! You have now grasped a little bit of UNIX and are ready to start working. Remember that your knowledge of the system is still limited, and that you are expected to do more practice with Linux to use it comfortably. But if all you had to do was get a bunch of applications and start working with them, what I included here is enough.

I'm sure you'll enjoy using Linux and will keep learning more about it---everybody does. I bet, too, that you'll never go back to DOSWin! I hope I made myself understood and did a good service to my 3 or 4 readers.

12.1 Copyright

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu via email.

12.2 Disclaimer

``From DOS to Linux HOWTO'' was written by Guido Gonzato, REMOVE_MEguido@ibogeo.df.unibo.it. (Remove ``REMOVE_ME''.) Many thanks to Matt Welsh, the author of ``Linux Installation and Getting Started'', to Ian Jackson, the author of ``Linux frequently asked questions with answers'', to Giuseppe Zanetti, the author of ``Linux'', to all the folks who emailed me suggestions, and especially to Linus Torvalds and GNU who gave us Linux.

This document is provided ``as is''. I put great effort into writing it as accurately as I could, but you use the information contained in it at your own risk. In no event shall I be liable for any damages resulting from the use of this work.

Feedback is welcome. For any requests, suggestions, flames, etc., feel free to contact me.

Enjoy Linux and life,

Guido =8-)

The dosemu HOWTO

by Mike Deisher.

Updated by Uwe Bonnes for dosemu-0.64.4, 15 March 1997

Updated by David Hodges, **davidhodges@altavista.net** For dosemu-0.98.8, 24 April 1999

---

*This is the `Frequently Asked Questions' (FAQ) / HOWTO document for dosemu. The most up-to-date version of the dosemu-HOWTO may be found at **http://www.dosemu.org**.*

---

1. The preliminaries

- 1.1 What is dosemu, anyway?
- 1.2 What operating systems does dosemu work under ?
- 1.3 What processors does dosemu work on ?
- 1.4 What if I have an Alpha/Sparc/Motorola/other non-Intel CPU ?
- 1.5 Do I need MS-DOS to use dosemu ?
- 1.6 Can I run Microsoft Windows programs under dosemu ?
- 1.7 I want to run something that won't run under DOSemu or Wine
- 1.8 Names and numbers
- 1.9 What version of Dosemu should I use?
- 1.10 What's the newest version of dosemu and where can I get it?
- 1.11 Where can I ask questions?
- 1.12 Where can I report bugs and ask questions?
- 1.13 What documentation is available for dosemu?
- 1.14 I have a program that fails, not listed in EMUfailure.txt
- 1.15 How do I submit changes or additions to the HOWTO?
- 1.16 Message from Greg...

TOC

2. Compiling and installing dosemu

- 2.1 Where are the installation instructions?
- 2.2 Top ten problems while compiling and installing dosemu.
- 2.3 How do I make aout binaries?
- 2.4 How do I compile dosemu on a machine with low memory?
- 2.5 Compilation fails with some strange error regarding "slang"
- 2.6 What configurable options are available ?
- 2.7 Do I need to run dosemu as root?
- 2.8 How do I patch dosemu ?
- 2.9 What versions of DOS are known to run with dosemu ?

3. Hard disk setup

- 3.1 How do I use my hard disk with dosemu?
- 3.2 How can I access the hdimage from Linux?
- 3.3 Can I use my stacked/double-spaced/super-stored disk?
- 3.4 Creating your own hdimage file.

4. Parallel ports, serial ports and mice

- 4.1 Port access worked with older version, but doesn't work now!
- 4.2 Port access was faster with older versions!
- 4.3 Where are the (microsoft compatible) mouse drivers?
- 4.4 Why doesn't the mouse driver work?
- 4.5 Why does dosemu clobber COM4?
- 4.6 How do I use dosemu over the serial ports?
- 4.7 How can I switch between dosemu and a shell over the serial line?
- 4.8 How can I get the parallel ports to work?

5. Multiple users and Non-interactive sessions

- 5.1 Can I use dosemu on a multi-user system?
- 5.2 How can I run dos commands non-interactively?

TOC

6. dosemu and Netware

- 6.1 How do I get Netware access from dosemu?

7. dosemu and X-windows(97/2/9).

- 7.1 Can I run dosemu in console mode while running X?
- 7.2 Is it possible to run dosemu in a window in X-windows?
- 7.3 Xdos dosen't work on a remote X-display!
- 7.4 Xdos dosen't find the VGA font
- 7.5 The vga font is very small on my high resolution display
- 7.6 Dosemu compilation fails with some strange error regarding X!
- 7.7 Does ansi emulation work properly?

8. dosemu and MS-Windows 3.1

- 8.1 Is it possible to run MS-Windows 3.1 under dosemu?
- 8.2 Can I run 32bit stuff with winemu
- 8.3 Errormessage: Cannot find the vdtapi.386 file when starting dosemu
- 8.4 Windows 3.x in xdos:
- 8.5 Can I install windows from within dosemu?
- 8.6 Notes for the mouse under win31-in-xdos:
- 8.7 Why did my Icon disappear from the Program manager?

9. Video and sound

- 9.1 Can I run 32-bit video games under dosemu?
- 9.2 Exiting from dosemu gives me a screen full of garbage.
- 9.3 Why doesn't my soundcard software work with dosemu?

10. Games

- 10.1 Duke3d dosen't work

11. Problems and fixes

- 11.1 Security issues
- 11.2 dosemu says "ERROR: general protection" and terminates when I run some program
- 11.3 Dosemu dies when booting. I have Win95 installed.
- 11.4 Dosemu hangs! How can I kill it?
- 11.5 Dosemu crashed and now I can't type anything.
- 11.6 I've enabled EMS memory in dosemu.conf but it does not help.
- 11.7 How do I get rid of all those annoying "disk change" messages?
- 11.8 Why won't dosemu run a second time after exiting in console mode?
- 11.9 Why will dosemu run in a term but not in the console?
- 11.10 How can I speed up dosemu?
- 11.11 My CDROM drive has problems reading some files under dosemu.
- 11.12 How do I see debugging output?
- 11.13 Why are my keystrokes echoed ttwwiiccee??
- 11.14 Dosemu scrambles my screen?
- 11.15 MS FoxPro 2.6 won't run

12. Contributing to the dosemu project

- 12.1 Who is responsible for dosemu?
- 12.2 I want to help. Who should I contact?

---

1. The preliminaries

1.1 What is dosemu, anyway?

To quote the manual, "dosemu" is a user-level program which uses certain special features of the Linux kernel and the 80386 processor to run MS-DOS/FreeDOS/DR-DOS in what we in the biz call a `DOS box.' The DOS box, a combination of hardware and software trickery, has these capabilities:

o the ability to virtualize all input/output and processor control instructions

o the ability to support the word size and addressing modes of the iAPX86 processor family's "real mode," while still running within the full protected mode environment

o the ability to trap all DOS and BIOS system calls and emulate such calls as are necessary for proper operation and good performance

o the ability to simulate a hardware environment over which DOS programs are accustomed to having control.

o the ability to provide DOS services through native Linux services; for example, dosemu can provide a virtual hard disk drive which is actually a Linux directory hierarchy."

1.2 What operating systems does dosemu work under ?

Dosemu is primarily written for Linux. It also works on NetBSD and possibly FreeBSD, however the graphics emulation is unlikely to work on NetBSD, which means that all DOS programs that use graphics mode (most DOS programs) will not work under NetBSD.

1.3 What processors does dosemu work on ?

Dosemu only works on Intel 80x86 processors, e.g. 80386, 80486, Pentium etc.

1.4 What if I have an Alpha/Sparc/Motorola/other non-Intel CPU ?

Dosemu only works on Intel 80x86 processors but there are alternatives: Bochs (**http://www.bochs.com**) is an open source shareware PC emulator that runs on most Unixes (as well as MS-Windows), and there are other (non-free) alternatives - see the comp.emulators.misc FAQ, available

via usenet or **ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/comp/emulators/misc/**

TOC

1.5 Do I need MS-DOS to use dosemu ?

No. You need some version of DOS but not necessarily MS-DOS. See the section "What versions of DOS are known to run with dosemu ?"

1.6 Can I run Microsoft Windows programs under dosemu ?

Not reliably. You would be better to use the Windows emulator Wine (**http://www.winehq.com**). If you insist on running Windows under dosemu, see section 8 - dosemu and MS-Windows 3.1.

1.7 I want to run something that won't run under DOSemu or Wine

See section 1.4.

1.8 Names and numbers

```
(xx/yy/zz) means day zz in month yy in year xx.
winemu mean WinOS/2 running in dosemu.
```

1.9 What version of Dosemu should I use?

Dosemu uses the same numbering scheme as the kernel. Uneven second numbers are for possible unstable developer releases, even second numbers are for releases considered stable. At the time of writing, **0.98.8** is the latest stable release, while **0.99.10** is the latest developer's release. So if you want to use dosemu, get the latest **stable** release.

1.10 What's the newest version of dosemu and where can I get it?

The newest version of dosemu as of 99/04/09 is **dosemu0.98.8** and can be ftp'ed from:

```
ftp://ftp.dosemu.org/pub/dosemu/
```

Remember that this is ALPHA code, however: there may be serious bugs and very little documentation for new features. The development version is particularly likely to have bugs. Please use it only if you like to do active development. Don't report bugs in the development version, fix them instead.

TOC

1.11 Where can I ask questions?

If you have problems regarding installing and running dosemu after reading the documentation, first try to help yourself: Your question has probably been asked and perhaps answered before. Try some search engine on the internet to retrieve that information. E.g. you can ask

`http://www.dejanews.com`

to find all newsnet articles containing the keywords of your question. Helping yourself will probably be faster than asking a well known question. It also frees up the time of developers from answering trivial question and so helps the further development of dosemu.

1.12 Where can I report bugs and ask questions?

If you want to ask questions and report bugs regarding dosemu, you should consider subscribing to the linux-msdos mailing list. To subscribe, send mail to **Majordomo@vger.rutgers.edu** with the following command in the body of your email message:

```
subscribe linux-msdos your_username@your.email.address
```

If you ever want to remove yourself from the mailing list, you can send mail to **Majordomo@vger.rutgers.edu** with the following command in the body of your email message:

```
unsubscribe linux-msdos your_username@your.email.address
```

(95/8/11). When you are subscribed to linux-msdos, you can send your report as mail to linux-msdos@vger.rutgers.edu. There is a gate that send mails to linux-msdos@vger.rutgers.edu as postings to the newsgroup named **linux.dev.msdos**. If your News provider doesn't carry that group, ask her(him) to add that group. Before you ask a question, you should carefully read all of the documentation, including this HOWTO and check one of the mailing list archives listed at **http://www.dosemu.org/mailinglist.html** to see whether your question has already been asked and answered.

1.13 What documentation is available for dosemu?

Dosemu comes with documentation. The main documentation files README.txt and README-tech.txt cover virtually all aspects of dosemu and may be more up-to-date than this HOWTO.

The "dosemu Novice's Altering Guide" or DANG is a road map to the inner workings of dosemu. It is designed for the adventurous, those who wish to modify the source code themselves. The DANG is maintained by Alistair MacDonald (**alistair@slitesys.demon.co.uk**) and is found in the doc directory of the dosemu source tree.

The EMU failure list (EMUfailure.txt) is a partial list of programs known **not** to work under dosemu.

And then, of course, there is the dosemu FAQ/HOWTO. But you already know about that, don't you. It is

TOC

also posted once in a while to the mailing list and found in the doc-directory. The most recent version can be found at **http://www.dosemu.org**.

1.14 I have a program that fails, not listed in EMUfailure.txt

First check, if the failure of your program is not caused by some of the fundamental incapabilities of dosemu, listed in EMUfailure.txt. If you think you have something new, please report to linux-msdos@vger.rutgers.edu. Perhaps it can be made going with the help others. Give detailed information about your setup, tell the version of kernel, dosemu etc and name the observed errors. You can use xdos to cut and paste the error message into your report. But keep your report in a readable form. We know the content of ../etc/config.dist. So only send the active lines from your dosemu.conf. Try running dosemu with some or all debug output turned on and scan through your debug output and at first only send those parts you think are relevant. Few people are willing to decode some long attachment to a mail, to do debugging for others. But keep your logs at hand, in case others ask detailed questions.

1.15 How do I submit changes or additions to the HOWTO?

The preferred method is to edit the file **dosemu-HOWTO-xx.x.sgml** to incorporate the changes, create a diff file by typing something like

```
diff -uw original-file new-file
```

and send it to **davidhodges@altavista.net**. If you do not know SGML, that's ok. Changes or new information in any form will be accepted. Creating the diff file just makes it easier on the HOWTO maintainer. :-)

1.16 Message from Greg...

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Greg Hankins, the Linux HOWTO coordinator, at gregh@cc.gatech.edu(95/8/11).

2. Compiling and installing dosemu

2.1 Where are the installation instructions?

The installation instructions are in the file, "QuickStart", included in the distribution.

2.2 Top ten problems while compiling and installing dosemu.

1. Forgetting to read the **QuickStart** Guide.

2. Try to compile some old version of dosemu.

3. Try to compile with a kernel older than 2.0.28 or 2.1.15.

4. Having the wrong linux kernel source sitting in **/usr/src/linux** or missing **/usr/src/linux/include/version.h**

5. Use dosemu with a kernel that does not have **IPC** compiled in.

6. Compile with gcc older than 2.7.2 or libc older than 5.x.x.

7. Forget to edit your **/etc/dosemu.conf**, **/etc/dosemu.users** and **/var/lib/dosemu/global.conf** files

8. Run DOSEMU with partition access while they are already mounted.

9. Don't install dosemu with sufficient privileges (i.e., root).

10. Try to run programs that use DPMI without enabling dpmi in /etc/dosemu.conf

2.3 How do I make aout binaries?

Starting with version 0.64.4 there is no a.out support any more. If you absolutely need it, you must use version 0.64.3.1. The configure script then should take care for this, if you setup is a standard setup.

2.4 How do I compile dosemu on a machine with low memory?

Marty Leisner (**leisner@sdsp.mc.xerox.com**) reported (95/4/8) that

If you have problems with running out of swap space you may want to add `CFLAGS+=-fno-inline` after `CFLAGS` is defined in **dpmi/Makefile**. Be careful before you do this and check for the existence of

TOC

swap space. I found Linux crashes at times when it has no swap space.

2.5 Compilation fails with some strange error regarding "slang"

You probably have

```
slangforce off
```

in your compiletime-settings file. You need to change it to

```
slangforce on
```

and recompile.

2.6 What configurable options are available ?

The compiletime-settings.help file describes the options you can change at compile time. The README.txt file describes the options you can change at run time.

2.7 Do I need to run dosemu as root?

No. Although dosemu drops root privilege wherever possible, it is still safer to not run dosemu as root, especially if you run DPMI programs under dosemu. Most normal DOS applications don't need dosemu to run as root, especially if you run dosemu under X. Thus you should not allow users to run a suid root copy of dosemu, wherever possible, but only a non-suid copy. You can configure this on a per-user basis using the /etc/dosemu.users file.

Programs that require direct access to hardware ports (e.g. if the program needs to talk to some unusual hardware that is not supported by Linux and dosemu) require dosemu to run as root but printers, serial ports, mice and video cards (and to a small extent sound - hopefully more soon) are emulated by dosemu so you don't need direct hardware access and don't need to run as root to use these.

2.8 How do I patch dosemu ?

If you do patch dosemu from one version to another, do "make pristine;./configure; make". If you don't make pristine, at least the version of the new executable will be wrong, if the whole thing compiles at all(97/2/9).

2.9 What versions of DOS are known to run with dosemu ?

All versions of DOS should work with dosemu, with the following caveats:

FreeDOS (**http://www.freedos.org**) does not (yet) have a working redirector so you will not be able to

TOC

access your Linux filesystem or Novell networking under FreeDOS. Note also that FreeDOS is unreliable beta software and may crash unexpectedly, trash its filesystem, etc. etc.

DOS 4.01 had problems by itself, so it won't work reliably with dosemu either.

With **MsDos-7** aka **Win95** you must not start the graphic shell at bootup. If you make the hdimage bootable with the so called "Rescue Disk" you are offered to make during the Windows installation, you get the right settings. If you use your normal Win95 installation to transfer the system files, have a look at the msdos.sys written on the hdimage and change the settings under the section [Options] to have entries like [Options] Logo=0 BootGUI=0

---

3. Hard disk setup

3.1 How do I use my hard disk with dosemu?

First, mount your dos hard disk partition as a Linux subdirectory. For example, you could create a directory in Linux such as /dos (mkdir -m 755 /dos) and add a line like

```
/dev/hda1        /dos     msdos   umask=022
```

to your /etc/fstab. (In this example, the partition is mounted read-only. You may want to mount it read/write by replacing "022" with "000" and using the -m 777 option with mkdir). Now **mount /dos**. The README.txt says:

```
you just can have a Linux directory containing all what you
want to have under your DOS C:. Copy your IO.SYS, MSDOS.SYS or what
ever to that directory (e.g. /var/lib/dosemu/bootdir), put

        $_hdimage = "bootdir"


into your /etc/dosemu.conf, and up it goes. DOSEMU makes a lredir'ed
drive out of it and can boot from it. You can edit the config.sys and
the autoexec.bat within this directory before you start dosemu.  Fur-
ther more, you may have a more sohisticated setup. Given you want to
run the same DOS drive as you normal have when booting into native
DOS, then you just mount you DOS partition under Linux (say to /dos)
and put links to its subdirectories into the boot dir. This way you
can decide which files/directories have to be visible under DOSEMU and
which have to be different. Here's a small and incomplete example
bootdir setup:


       config.sys
       autoexec.bat
       command.com -> /dos/command.com
       io.sys -> /dos/io.sys
       msdos.sys -> /dos/msdos.sys
       dos -> /dos/dos
       bc -> /dos/bc
       windows -> /dos/windows
```

```
There is, however, one drawback, you can't use the DosC kernel
(FreeDos) for it, because it hasn't yet a working redirector (will
hopefully be available some time in the future).
```

TOC

3.2 How can I access the hdimage from Linux?

Use mtools. With a line in **/etc/mtools.conf** like

```
drive n:  file="/var/lib/dosemu/hdimage" MTOOLS_SKIP_CHECK=1 \
MTOOLS_LOWER_CASE=1 MTOOLS_NO_VFAT=1 partition=1 offset=128
```

you can use the mtools on the hdimage, like "mdir n:". "mcopy n:/config.emu /tmp" copies the config.emu file from the hdimage to /tmp/config.emu. You can edit it there and copy it back. Use a drive letter you find sensible. "N:" is only an example.

3.3 Can I use my stacked/double-spaced/super-stored disk?

At this time, compressed drives cannot be accessed via the redirector (lredir or emufs) on a standard kernel. There is a patch for the kernel to mount compressed files under the name "dmsdosfs". Find it on sunsite.unc.edu and its mirrors

http://sunsite.unc.edu:/pub/Linux/system/filesystems/dosfs/

A good idea is also to look in http://sunsite.unc.edu:/pub/Linux/Incoming for a newer version. The "wholedisk" option in older versions of dosemu is no longer allowed in recent versions, however, a line like

```
$_hdimage = "/dev/hda1"
```

may work, at the risk that you could lose all data in that partition on a dosemu crash.

If your dos partition is already mounted with write access and you try to run dosemu with partition access, dosemu will print a warning message and abort. This prevents DOS and Linux from making independent writes to your disk and trashing the data on your dos partition(95/8/11).

If LILO is installed, the above will not work. However...

Thomas Mockridge (**thomas@aztec.co.za**) reported (94/8/5) that

To boot dosemu with LILO and Stacker 4.0 I did a little work around...

1. **dd** the MBR to a file. (or norton utility, etc., first 512 bytes)

2. Boot dos (from full boot not emu), do a fdisk /mbr, make your dos partition active with (dos) fdisk.

3. Copy the new MBR to a file.

4. Replace the original MBR

5. Copy the second MBR to /var/lib/dosemu/partition.hda? (Whichever is your dos partition)

6. Set dosemu.conf

```
disk {partition "/dev/hda? ?"}
```

7. Start dosemu and and voila! No LILO.

For recent versions of dosemu you need to change the

```
disk {partition "/dev/hda? ?"}
```

line to something like

```
$_hdimage = "/dev/hda1"
```

Holger Schemel (**q99492@pbhrzx.uni-paderborn.de**) reported (94/2/10) that

Works even fine under dosemu with MS-DOS 6.0. If you have problems, then you have to edit the file 'DBLSPACE.INI' manually and change the disk letter to the letter your drive gets under dosemu.

Darren J Moffat (**moffatd@dcs.gla.ac.uk**) also reported (94/3/27)

"...use 6.2 if you can get it!! Just make sure you have a LILO boot disk on hand since dos 6{.2} will change the MBR of the boot HZ."

3.4 Creating your own hdimage file.

The easy way is to use mkdexe - see README.txt for details. The old-fashioned way (I don't know why anyone would want to do it that way any more but here it is, just in case) is as follows:

There is an extra util program called mkfatimage16 which allows for creating a hdimage file headers. The full information is in the manpage (man/mkfatimage16.1) included in your distribution.

To create a hard disk image file with a geometry corresponding to that of a real hard disk of 32 megabytes run:

```
mkfatimage16 -k 32768 > hdimage
```

This is probably too large for most needs; if you need this much space, consider using the disk redirector."

Usually it is a good idea to format the drive after it.

---

4. Parallel ports, serial ports and mice

4.1 Port access worked with older version, but doesn't work now!

Read **../doc/README.txt** and the port-section in **../etc/config.dist**

4.2 Port access was faster with older versions!

To have a chance to log port access, by default every port access produces an exception out of vm86-mode. This takes some time. If you don't want to log port access, use the keyword **"fast"** in the appropriate port statement.

4.3 Where are the (microsoft compatible) mouse drivers?

Tom Kimball (**tk@pssparc2.oc.com**) reported (93/11/24) that

Several people said to use a different mouse driver and suggested some. I found a couple that seem to work fine.

```
oak.oakland.edu:/pub/msdos/mouse/mouse701.zip   (mscmouse)
oak.oakland.edu:/pub/msdos/mouse/gmous102.zip   (gmouse)
```

Normally you can use dosemu's internaldriver, so you don't need any additional mousedriver in dosemu outside winemu(97/2/10).

4.4 Why doesn't the mouse driver work?

Mark Rejhon (**mdrejhon@magi.com**) reported (95/4/7) that

If you start the mouse driver and it just hangs (it might actually take 30-60s), but if you are waiting longer than a minute for the mouse driver to start, try specifying the COM port that the mouse is on, at the mouse driver command line.

TOC

4.5 Why does dosemu clobber COM4?

Rob Janssen (**rob@pe1chl.ampr.org**) reported (94/3/24) that

According to **jmorriso@bogomips.ee.ubc.ca**, "dosemu still clobbers COM4 (0x2e8, IRQ 5). 0x2e8 isn't in ports{} in config. I have to run setserial /dev/cua3 irq 5 on it after dosemu exits."

This is caused by your VGA BIOS. I have found that by enabling the IO port trace and seeing where it was clobbered.

Disable the "`allowvideoportaccess on`" line in config and it will work fine. When you then have problems with the video, try to enable more selective ranges of IO addresses (e.g., 40-43).

4.6 How do I use dosemu over the serial ports?

4.7 How can I switch between dosemu and a shell over the serial line?

John Taylor (**taylor@pollux.cs.uga.edu**) reported (94/5/25) that

I am running Linux 1.1.13 and want to point out a great feature that should be protected and not taken out (IMHO). With the 52 version, I can run the program, "screen." From screen, i can invoke dos -D-a. What is really great (IMHO) is the screen commands (the CTRL-A cmds) still work. This means I can do a CTRL-A C and add another unix shell, and switch between the two (DOS / UNIX). This allows me to use dosemu over the serial line really well, because switching is made easy.

4.8 How can I get the parallel ports to work?

The dosemu.conf has lines at the end to redirect printers to either lpr or a file. If you want direct access to the bare metal, comment out these emulation lines, and add the line

```
$_ports { device /dev/lp0 fast range 0x3bc 0x3bf } # lpt0
```

for the "monitor card" printer port (corresponds to /dev/lp0), or

```
$_ports { device /dev/lp1 fast range 0x378 0x37f } # lpt1
```

or

```
$_ports { device /dev/lp1 fast range 0x278 0x27f } # lpt2
```

for LPT1 (/dev/lp1) or LPT2 (/dev/lp2) respectively(97/2/9).

Hans Lermen (lermen@dosemu.org) writes:

But NOTE: these lines should not be _added_ simply, the string should be concatenated such as:

$_ports = "...." $_ports = $_ports, " device /dev/lp0 fast range 0x3bc 0x3bf" = This blank is important

---

5. Multiple users and Non-interactive sessions

5.1 Can I use dosemu on a multi-user system?

Yes, you can configure dosemu on a per-user basis. See README.txt for details.

5.2 How can I run dos commands non-interactively?

You can do any of the following: 1. redirect a file to dosemu standard input 2. use the keystroke configuration option and the -I command line option as described in the README.txt file to specify keystrokes on the command line that will act exactly as if you had typed them within dosemu 3. use the mkdexe program to create a DEXE file - a small DOS filesystem image containing just the program you want to run - see the README.txt file for details.

Here are some additional alternatives:

Daniel T. Schwager (**danny@dragon.s.bawue.de**) reported (94/7/2) that

You can use different dosemu.conf files (and different hd-boot-images with different autoexec.bat's) and call dosemu like

```
$ dos -F my_quicken_q_exe_dosemu.conf
```

Dietmar Braun (**braun@math20.mathematik.uni-bielefeld.de**) reported (94/7/4) that

This is no problem at all when you use the redirector of dosemu. It is possible to redirect a drive letter to a linux path given by an environment variable.

So I have a shell script named "**DOS**" which does something like

```
mkdir /tmp/dos.$$
DOSTMP=/tmp/dos.$$; export DOSTMP
```

and then a little trick to get "**echo $* > $DOSTMP/startup.bat**" really working (actually a small C Program which turns '/' in '\' and terminates lines correctly for messy dos with cr/lf pairs and adds **^Z** at the end of the file), creates startup files, links and so on in this directory, and then starts dosemu. Within "**autoexec.bat**" drive c: is redirected from hdimage to this tmp-directory, which has links for **$HOME** and **$PWD**.

So if I want to see my filenames shortened to 8.3 I can type "**DOS dir**" and I get my current directory

listing. So I have full DOS multi user (I don't have any DOS partition and redirecting to Linux prese user permissions) and multi tasking. (dosemu sessions are completely independent). I did this once able to use a dos driver for my printer. My printcap df is actually a DOS program. So you can even make DOS executables act as lpr filters.

6. dosemu and Netware

6.1 How do I get Netware access from dosemu?

As always, access through the Linux filesystem is preferred. Mount your Netware drives with Caldera's Netware utilities or Volker Lendecke's free ncpfs utility (**ftp://ftp.gwdg.de:/pub/linux/misc/ncpfs**). If you need real IPX access, e.g. to run Novell's "syscon", read **../doc/NOVELL-HOWTO.txt**. You probably can not currently access Netware from FreeDOS.

7. dosemu and X-windows(97/2/9).

7.1 Can I run dosemu in console mode while running X?

Ronald Schalk (**R.Schalk@uci.kun.nl**) reported (94/1/17) that

Yes, no problem. Just remember to use ctrl-alt-<Fn> to go to a Virtual Console (VC), and you can run any Linux application (dosemu is a linux-application). I've got almost always WP5.1 in a dos session.

[Note: Use ctrl-alt-F7 to switch back to X from dosemu, if X runs on VC7.]

7.2 Is it possible to run dosemu in a window in X-windows?

If you have X installed and you have successfully compiled dosemu and run it successfully outside X-windows, you should be able to run "`xdos`" or "`dos -X`" right away to bring up a dosemu window. If this does not work, make sure:

```
1. Dosemu has X support compiled in. This is default, however
   if you you have compiled dosemu with "x off" in the
   compiletime-settings file you don't have X support. So changing
   "x off" to "x on" in the compiletime-settings file, followed by
   "make pristine; make; make install" should build
   you a dosemu-executable with X support, if you have the
   X-libraries installed in /usr/X11R6.
2. Set up your X key-mappings.  In an xterm, type

   xmodmap -e "keycode 22 = 0xff08"
   xmodmap -e "keycode 107 = 0xffff"

   These lines fix the backspace and delete keys respectively.
3. Configure the X-related configuration options in your
   /etc/dosemu.conf file.
```

Alternatively, you can run dosemu inside a color xterm, which is not recommended because many color xterms have buggy support for the complex text display capabilities of dosemu. This does not require `X_SUPPORT` to be compiled into dosemu. However, if you really want to do this, do the following steps:

```
1. Install ansi_xterm.  The recommended package is available as:
   tsx-11.mit.edu:/pub/linux/ALPHA/dosemu/Development/ansi-xterm-R6.tar.gz

2. Set up your X key-mappings.  In an xterm, type
```

```
    xmodmap -e "keycode 22 = 0xff08"
    xmodmap -e "keycode 107 = 0xffff"

    These lines fix the backspace and delete keys, respectively.

  3. Configure the terminal-related (not X-related) settings in
     /etc/dosemu.conf
```

Marty Leisner (**leisner@sdsp.mc.xerox.com**) reported (95/3/31) that

I have xrdb log the following resources

```
    dosxterm*Font: vga dosxterm*geometry: 80x25 dosxterm*saveLines: 25
```

or I alias "`dosxterm`" to "`term -fn vga -title dosxterm -geometry 80x25 -sl 25`"

If you use the xrdb method, all you have to do is run "`xterm -name dosxterm`"

7.3 Xdos dosen't work on a remote X-display!

At present, dosemu is set up to use the MIT shared memory extensions. This extension only works on a local display. If you want to run xdos on a remote display, compile dosemu with "mitshm off" in the compiletime-settings file after a **"make pristine"** or on the clean source tree(97/2/9).

7.4 Xdos dosen't find the VGA font

Check that the vga fonts you installed are listed in the font.dir of the directory you installed the fonts in:

```
    hertz:~> grep misc /usr/X11R6/lib/X11/XF86Config
        FontPath    "/usr/X11R6/lib/X11/fonts/misc/"
    hertz:~> grep vga /usr/X11R6/lib/X11/fonts/misc/fonts.dir
    vga.pcf vga
    vga11x19.pcf vga11x19
    hertz:~> ls /usr/X11R6/lib/X11/fonts/misc/vga*
    /usr/X11R6/lib/X11/fonts/misc/vga.pcf
    /usr/X11R6/lib/X11/fonts/misc/vga11x19.bdf
    /usr/X11R6/lib/X11/fonts/misc/vga11x19.pcf
```

If you installed some X-fonts, like you did when you installed dosemu with X-Support for the first time, "mkfontdir" and then "xset fp rehash" needs to be run. The dosemu install should take care for "mkfontdir" and tells you about "xset fp rehash". Tell us if it doesn't work for you. (97/2/13)

7.5 The vga font is very small on my high resolution display

Look for the vga11x19 font. (97/2/13)

7.6 Dosemu compilation fails with some strange error regarding X!

TOC

As stated above, dosemu uses the MIT shared memory extensions by default. Under XFree86 they are only available with Version 3.1.2 and above. If you have an older version, consider to upgrade, or configure dosemu to not use this extension (see last section)(97/2/9).

7.7 Does ansi emulation work properly?

Marty Leisner (**leisner@sdsp.mc.xerox.com**) reported (95/3/31) that

Yes. I use `nnansi.com` under X windows. I find 25, 43 and 50 line mode work properly, however 50 line mode is difficult to use on a 1024x768 screen (unless smaller fonts are used or you use a bigger screen. 43 line mode will resize the `xterm` window to use 43 lines.

---

8. dosemu and MS-Windows 3.1

8.1 Is it possible to run MS-Windows 3.1 under dosemu?

The **../doc/README.Windows** file says:

```
*****************************************************************
*    WARNING!!! WARNING!!! WARNING!!! WARNING!!! WARNING!!!    *
*                                                               *
*  Danger Will Robinson!!!  This is not yet fully supported     *
*  and there are many known bugs!  Large programs will almost  *
*  certainly NOT WORK!!!  BE PREPARED FOR SYSTEM CRASHES IF     *
*  YOU TRY THIS!!!                                              *
*                                                               *
*    WARNING!!! WARNING!!! WARNING!!! WARNING!!! WARNING!!!    *
*****************************************************************
```

Okay, it is possible to boot WINOS2 (the modified version of Windows 3.1 that OS/2 uses) under DOSEMU. Many kudos to Lutz & Dong!

However, YOU NEED BOTH LICENSES, for WINDOWS-3.1 as well OS/2 !!!

There are many known problems. Windows is prone to crash, could take data with it, large programs will not load, etc. etc. etc. In other words, it is NOT ready for daily use. Many video cards are known to have problems (you may see a nice white screen, however, look below for win31-in-xdos). Your program groups are all likely to disappear. ... Basically, it's a pain.

for more details see the README.txt file.

8.2 Can I run 32bit stuff with winemu

Sorry, no you can't. Win32s needs stuff running in ring 0. That's a "nono" with linux. (97/04/27)

8.3 Errormessage: Cannot find the vdtapi.386 file when starting dosemu

As explained above, winemu can't handle 32 bit stuff. All drivers ending in 386 are 32bit. So to run winemu, you have to delete all references to .386 drivers in your win.ini. (97/04/27)

TOC

8.4 Windows 3.x in xdos:

As of version 0.64.3 DOSEMU is able to run Windows in xdos. Of course, this is not recommended at all, but if you really want to try, it is safer then starting windows-31 on the console, because _when_ it crashes, it doesn't block your keyboard or freeze your screen.

Hints:

```
1. Get Dosemu & Linux source.
2. Unpack dosemu.
3. Run "./configure" to configure Dosemu (it will enable vm86plus as a
   default).
4. Type "make" to compile.
5. Get a Trident SVGA drivers for Windows. The files are tvgaw31a.zip
   and/or tvgaw31b.zip. They are available at garbo.uwasa.fi in
   /windows/drivers (any mirrors?).
6. Unpack the Trident drivers.
7. In Windows setup, install the Trident "800x600 256 color for 512K
   boards" driver.
8. Do the things described above to get and install OS2WIN31.
10. Start xdos.
11. In Dosemu, go to windows directory and start winemu.
12. Cross your fingers.
```

8.5 Can I install windows from within dosemu?

No you can't. Dos will tell you something like

```
The XMS driver you have on your system is not compatible with
Windows...
```

You need to install windows from Dos. You can copy the windows tree to somewhere on your Linux Filesystem and use lredir to mount it on the same place as it is in dos. Example:

```
You have windows in d:\windows You have d:\ mounted as /dosc
in Linux You copy the windows tree to Linux, e.g. "cp -a
/dosd/windows /usr/share Inside dosemu you redirect the
copied tree like lredir d: linux\fs\dosd
```

Now dosemu can't mess around in you windows directory, but changes in the windows directory aren't seen by dosemu too. If you want to do the same with windows on drive c:> look in this FAQ how to redirect c:.

But you can use "setup" from inside the windows directory to install drivers and change some settings. (97/2/1497)

8.6 Notes for the mouse under win31-in-xdos:

TOC

1. Use the mouse driver "mouse.drv" from WinOS2

2. In order to let the mouse properly work you need the following in your win.ini file:

```
[windows]
MouseThreshold1=0
MouseThreshold2=0
MouseSpeed=0
```

3. The mouse cursor gets not painted by X, but by windows itself, so it depends on the refresh rate how often it gets updated, though the mouse coordinates movement itself will not get delayed. (In fact you have 2 cursors, but the X-cursor is given an 'invisible' cursor shape while within the DOS-Box.)

4. Because the coordinates passed to windows are interpreted relatively, we need to calibrate the cursor. This is done automatically whenever you enter the DOS-Box window: The cursor gets forced to 0,0 and then back to its right coordinates. Hence, if you want to re-calibrate the cursor, just move the cursor outside and then inside the DOS-Box again. (97/2/10)

8.7 Why did my Icon disappear from the Program manager?

MS-Windows and WinOS2 handle the program manager group differently. While MS-Win store the setup in progman.ini, WinOS2 wants the contents of progman.ini in system.ini. Here's a tip from **(friest@acm.org (Todd T. Fries))**:

```
cat progman.ini >> system.ini
```

Be sure to use >> :-)

---

9. Video and sound

9.1 Can I run 32-bit video games under dosemu?

Mark Rejhon (**mdrejhon@magi.com**) reported (95/4/8) that

With the recent DPMI improvements that has gone into 0.60, you can now run some 32-bit video games in dosemu. If the game is compatible in an OS/2 DOS box, there are chances that it will work in dosemu. (Example 32-bit games include Descent, Dark Forces, Mortal Kombat 2, Rise of The Triad, which have all successfully been tested in recent dosemu releases).

Before you attempt to run a video game, you must have the keyboard configured in raw keyboard mode and enabled VGA graphics modes, in the **/etc/dosemu.conf** file.

Note, however, you will have to turn off the sound in the game. (Someone will have to program in sound board emulation before we can avoid this). Note that game timers can be a little bit slow, due to Linux multitasking and lack of high-frequency timer support. So the games may run from anywhere from 5 to 100 percent speed. Typically, the speed is approximately 50 percent in recent dosemu releases and is expected to improve eventually.

Who knows, it might even work. If you can't get it to work, check EMUfailure.txt if the program is listed there, or falls in a category of programs that at present don't or probably never work with dosemu. If you think, it should be listed in EMUfailure.txt, report to **linux-msdos@vger.rutgers.edu**

There is a security hole when having enabled DPMI and having dosemu suid root (especially when using dos4gw-based games), the client is able to access the whole user space, hence also can modify the dosemu code itself. Use of the 'secure' option in /etc/dosemu.conf, e.g.

```
$_secure = "ngd" # secure for: n (normal users), g (guest), d
(dexe)
```

disables this. It is better to run dosemu as a normal user under X rather than root anyway.

9.2 Exiting from dosemu gives me a screen full of garbage.

(95/4/8)

The problem is that the font information for the VGA text screen is not being saved. Get a copy of the svgalib package. The current source is in

```
sunsite.unc.edu:/pub/Linux/libs/graphics/svgalib-1.3.1.tar.gz
```

It may also be available as a pre-compiled package in your favorite Linux distribution (e.g., Slackware,

TOC

etc.). Use `savetextmode` to save the current text mode and font to a file in **/tmp** before running dosemu. Then run `textmode` upon exiting dosemu to restore it.

Addition from (**lermen@elserv.ffm.fgan.de**(97/2/11)):

Have a look also at src/arch/linux/debugger/README.recover and README.dosdebug, dosdebug can aid you recovering.

9.3 Why doesn't my soundcard software work with dosemu?

Dosemu includes a rudimentary soundblaster emulator called SBemu - the documentation for it is currently at http://www.slitesys.demon.co.uk/a.macdonald/dosemu/sound/ This is a work in progress, not completed yet so much or most DOS software that uses sound will not work correctly with it yet.

---

10. Games

10.1 Duke3d dosen't work

Hans Lermen <**lermen@elserv.ffm.fgan.de**> said (97/2/16): duke3d must be 'configured' via a setup, within this setup you have to choose 'keyboard + mouse', else it won't work.

11. Problems and fixes

11.1 Security issues

A full featured Dosemu needs to be suid root, e.g to access ports. Most dos programs don't need this however so when running under X you usually do not need to run dosemu as root. Dosemu runs as suid "root" only where it is needed, and releases this right thereafter. But with DPMI, the Dos client programm can access the whole user space, hence also can modify the dosemu code itself. Use of the 'secure' option in /etc/dosemu.conf disables this. There are several other important security considerations discussed in the Security section of the README.txt file.

11.2 dosemu says "ERROR: general protection" and terminates when I run some program

This will happen if your program uses DPMI and you do not have DPMI enabled. Try changing the line

```
$_dpmi = (off)
```

in /etc/dosemu.conf to

```
$_dpmi = (nnnn)
```

where nnnn is the number of kilobytes of memory you wish to give to the DOS program (e.g. Doom requires about 4000 kilobytes). If you do this, you should not run dosemu as root (see the Security Issues section) - it is better not to run dosemu as root whenever possible, in any case.

Another likely cause is that your program uses VCPI (see EMUfailure.txt for more details) or some other means of switching to protected mode (other than DPMI), in which case it will never work under dosemu.

If none of these is the case, see section 1.14 I have a program that fails, not listed in EMUfailure.txt.

11.3 Dosemu dies when booting. I have Win95 installed.

Dosemu relies that the Dos-Version on the hdimage and the Drive you map to contain command.com are the same. If not, dosemu will crash sooner or later. With the dual boot option Win95 offers when pressing the F4,F5 and F8 Keys with the "Starting Win95" text, versions on the Win95 drive may swap. Take special care for command.com. Let your shell variable in config.emu point to the correct static version of command.com, e.g.: shell=c:\win95\command.com c:\ /P /E:1024 (97/02/28)

TOC

11.4 Dosemu hangs! How can I kill it?

Switch to another console and type dosdebug, then type kill (it may take a while but will work eventually).

11.5 Dosemu crashed and now I can't type anything.

Daniel Barlow(jo95004@sable.ox.ac.uk) reported (95/4/8) that

If you have no terminal or network access that you can use to log in, you may have to press the reset button. If you can still get a usable shell somehow, run "`kbd_mode -a`" to switch the keyboard out of raw mode, and/or "`stty sane`" on the console so that you can see what you're typing.

A useful thing to do is to use a script to run dosemu, and run "`kbd_mode -a`" automatically right after dosemu. When dosemu crashes, the script usually will resume running, and execute the "`kbd_mode -a`" command.

11.6 I've enabled EMS memory in dosemu.conf but it does not help.

Rob Janssen (rob@pe1chl.ampr.org) reported (94/7/11)

Don't forget to load the provided ems.sys from the config.sys file.

11.7 How do I get rid of all those annoying "disk change" messages?

(94/8/11)

Grab and install `klogd`. Try

```
sunsite.unc.edu:/pub/Linux/system/Daemons/sysklogd1.2.tgz
```

11.8 Why won't dosemu run a second time after exiting in console mode?

Aldy Hernandez (**aldy@sauron.cc.andrews.edu**) reported (94/7/8) that

You should disable your video and/or BIOS caching.

11.9 Why will dosemu run in a term but not in the console?

JyiJiin Luo (**jjluo@casbah.acns.nwu.edu**) reported (94/4/19) that

I experienced exactly the same problem before. I figured out all the video shadow in my AMI BIOS must be disabled. Now dosemu runs fine on my system.

11.10 How can I speed up dosemu?

In some cases it is useful to play with the value of the `HogThreshold` variable in your `dosemu.conf` file.

Daniel Barlow(**jo95004@sable.ox.ac.uk**) reported (95/4/8) that

HogThreshold should now be set to approximately half of the BogoMips value that the system reports on boot.

11.11 My CDROM drive has problems reading some files under dosemu.

Vinod G Kulkarni (**vinod@cse.iitb.ernet.in**) reported (94/4/7) that

When a CDROM is mounted from linux and used from within dosemu (mapped drive), there could be some problems. The CD-ROM driver (iso9660) in the kernel tries to find out the type of the file (i.e. binary or text). If it can't find, it tries to guess the type of the file using a heuristic. This heuristic fails under some circumstances when a (almost) text file is to be treated as binary. (I do not know if it is a bug or feature.)

The result of this is that if you copy such a file from CD-ROM (from linux itself, and not necessarily dosemu), the resulting file will be usually bigger than original file. (Blanks get added before `^J`,`^M`.) So a program running in dosemu gives an error or hangs, which may be mistaken as problem of dosemu.

Rob Janssen (**pe1chl@rabo.nl**) reported (94/8/10) that

The way to solve this is to turn off conversion altogether. Pass the option "**-o conv=binary**" to the mount command mounting the CD-ROM, or use the following in **/etc/fstab**:

```
/dev/cdrom      /cdrom          iso9660 conv=binary,ro
```

No patches to the kernel are necessary.

11.12 How do I see debugging output?

Daniel Barlow(**jo95004@sable.ox.ac.uk**) reported (95/4/8) that

As of dosemu 0.60, debugging output is redirected to a file specified on the command line. Use "**dos -D+a -o /tmp/debug**" to log all debug output to **/tmp/debug**. There should no longer be any need to redirect `stderr`.

11.13 Why are my keystrokes echoed ttwwiiccee??

Nick Holloway (**alfie@dcs.warwick.ac.uk**) reported (94/2/22) that

After running dos after playing with some stty settings, I was getting doubled key presses. I can now reveal what the reason is!

It only happens when dos is run on the console with 'istrip' set. This is (I think) because the raw scancodes are mutilated by the 'istrip', so that key release events look like key press events.

So, the input processing needs to be turned off when using the scan codes on a console (it wouldn't be a good idea to do it for tty lines).

11.14 Dosemu scrambles my screen?

For those graphics cards not fully supported in dosemu, with allowed console graphics a dosemu crash may leave your console in a scrambles and nearly unusable way. To prepare for that situation, Spudgun <spudgun@earthlight.co.nz> posted the following solution. First save your registers when running on the console ~> cat /usr/bin/savetextmode ~> restoretextmode -w /etc/textregs ~> restorefont -w /etc/fontdata Then, when a crash happened, run following script: restoretextmode -r /etc/textregs restorefont -r /etc/fontdata restorepalette

If it doesn't fix it nothing will I also found having an X server running sometimes put my Vid card's registers into a strange state where this script made things worse I think since changing X servers and/or running savetextmode on a vt while X was running helped. (97/04/08)

11.15 MS FoxPro 2.6 won't run

FoxPro 2.6 doesn't run on network drives. Alexey Naidyonov <growler@growler.tsu.tula.ru> states on that problem: And I guess your FoxPro files are on lredir'ed disk, yeah? The matter is that FoxPro doesn't run on such disk, but when I said disk { partition ... } in /etc/dosemu/conf, it runs.

12. Contributing to the dosemu project

12.1 Who is responsible for dosemu?

Dosemu is built upon the work of Matthias Lautner and Robert Sanders. Hans Lermen (lermen@dosemu.org) is responsible for organizing the latest releases of dosemu.

History of dosemu

Version	Date	Person
0.1	September 3, 1992	Matthias Lautner
0.2	September 13, 1992	Matthias Lautner
0.3	???	Matthias Lautner
0.4	November 26, 1992	Matthias Lautner
0.47	January 27, 1993	Robert Sanders
0.47.7	February 5, 1993	Robert Sanders
0.48	February 16, 1993	Robert Sanders
0.48pl1	February 18, 1993	Robert Sanders
0.49	May 20, 1993	Robert Sanders
0.49pl2	November 18, 1993	James MacLean
0.49pl3	November 30, 1993	James MacLean
0.49pl3.3	December 3, 1993	James MacLean
0.50	March 4, 1994	James MacLean
0.50pl1	March 18, 1994	James MacLean
0.52	June 16, 1994	James MacLean
0.60	April 9, 1995	James MacLean
0.64.4	February 9,1997	Hans Lermen
0.66.3	April 20, 1997	Hans Lermen
0.98.1	December 9, 1998	Hans Lermen
0.98.6	March 21, 1999	Hans Lermen

12.2 I want to help. Who should I contact?

The dosemu project is a team effort. If you wish to contribute, see the DPR (dosemu Project Registry). A current copy may be found in **doc/DANG**.

The Linux Danish/International HOWTO

Niels Kristian Bech Jensen, nkbj@sslug.dk

v2.5, 20 February 1999

This document describes how to configure Linux and various Linux applications for Danish locale standards such as keyboard, font, paper-size etc. It is hoped that Linux users from other places in Western Europe will find this document useful too.

1. Introduction

2. Keyboard setup

- 2.1 Loading a keytable
- 2.2 Getting the AltGr key to work under X11
- 2.3 Dead keys and accented characters
- 2.4 Making $ (the dollar sign), ø (oslash) and Ø (Oslash) work

3. Display and application setup

- 3.1 Loading the ISO-8859-1 font on the console
- 3.2 The Euro symbol
- 3.3 Characters you can display under Linux
- 3.4 International character sets in specific applications

4. Miscellaneous problems

- 4.1 Time zone
- 4.2 A4 papersize
- 4.3 Text file formats for other platforms

TOC

5. Locale support in libc 5.4.x and higher

6. Programming tips for X11

7. Getting X11 applications to speak Danish

8. Information resources

- 8.1 Other documents of relevance
- 8.2 FTP and Web sites

9. Credits and legal stuff

- 9.1 Legal stuff

---

1. Introduction

All European users of almost any operating system have two problems: The first is to tell the computer that you have a non-American keyboard, and the second is to get the computer to display the special characters. To make matters worse some applications will also consider you an exception if you are not an American and require special options or the setting of environment variables.

Under Linux you change the way your computer interprets the keyboard with the commands `loadkeys` and `xmodmap`. `loadkeys` will modify the keyboard for plain Linux while `xmodmap` makes the modifications necessary when the handshaking between X11 and Linux is imperfect.

To display the characters you need to tell your applications that you use the ISO-8859-1 (a.k.a. Latin-1) international set of glyphs. This is not always necessary, but a number of key applications need special attention.

This HOWTO is intended to tell Danish users how to do this. If you continue to have problems after reading this you can try the German HOWTO, the Linux Keyboard and Console HOWTO or the ISO 8859-1 National Character Set FAQ. Many of the hints contained herein are cribbed from there. See section Other documents of relevance for pointers to these documents. You should also send me a mail describing your problems.

A final problem is that error-messages, menus and documentation of the applications are mostly in English. There is a GNU project under way to address this problem. You can see what it is all about by downloading the file `ABOUT-NLS` or the package `gettext-0.10.tar.gz` (or any later version) from your favourite mirror of the GNU archive. This project needs volunteers for the translations. Send a mail to `da-request@li.org` with the body ``subscribe'' if you want to contribute to the Danish part of the project. The documentation in the `gettext` package describes how to use such translations in your own programs.

2. Keyboard setup

2.1 Loading a keytable

You have two tools for configuring your keyboard. Under plain Linux you have `loadkeys` and under X11 you have `xmodmap`.

To try out `loadkeys` type one of these two commands:

```
loadkeys /usr/lib/kbd/keytables/dk.map
```

or

```
loadkeys /usr/lib/kbd/keytables/dk-latin1.map
```

The difference between the two keymaps is that `dk-latin1.map` enables `dead' keys while `dk.map` does not. Dead keys are explained in section Dead keys and accented characters. The program `loadkeys` and the keymaps are part of the package `kbd-0.??.tar.gz` which (with differing version numbers ??) is available with all Linux distributions.

Usually `loadkeys` is executed at boot-time from one of the scripts under the directory `/etc/rc.d/`. Details vary between distributions.

(Note for non-Danish readers: Support for other languages is enabled in a similar manner. Use `es.map` for Spanish keyboards etc.)

Versions of XFree86 up to and including v3.1.2 will normally follow the keymap used by plain Linux, but you can modify keyboard behavior under X11 with `xmodmap`. Usually the X11 initialization process will run this command automatically if you have a file called `.Xmodmap` in your home directory.

In XFree86 v3.2 and higher you should have the following `Keyboard` section in your `/etc/XF86Config` file (it is made automatically by the program `XF86Setup` if you choose a Danish keytable):

```
Section "Keyboard"
   Protocol        "Standard"
   XkbRules        "xfree86"
   XkbModel        "pc101"
   XkbLayout       "dk"
   XkbVariant      "nodeadkeys"
EndSection
```

The only keyboard variant available at the moment is `"nodeadkeys"`, but dead keys can still be made

to work. See section Dead keys and accented characters for more information on this.

2.2 Getting the AltGr key to work under X11

For versions of XFree86 up to and including v3.1.2 you should edit the file `/etc/X11/XF86Config` (or possibly `/etc/XF86Config`) and make sure the line

```
RightAlt    ModeShift
```

appears in the `Keyboard` section. Usually you can do this by uncommenting the appropriate line. In XFree86 v3.1.2 you can use `AltGr` as an alias for `RightAlt`.

The `AltGr` key should work as expected in XFree86 v3.2 and higher if you choose Danish keyboard support.

Making {, [,] and } work under Metro-X

You can't input the characters ``{" (`<AltGr><7>`), ``[" (`<AltGr><8>`), ``]" (`<AltGr><9>`) and ``}" (`<AltGr><0>`) under the Metro-X server. This bug has been observed under versions 3.1.5 and 3.1.8 of the server.

To correct this bug you have to edit the file `/usr/X11R6/lib/X11/xkb/symbols/dk` and change the lines

```
key <AE07> {    [               7,           slash      ]       };
key <AE08> {    [               8,       parenleft      ]       };
key <AE09> {    [               9,      parenright      ]       };
key <AE10> {    [               0,           equal      ]       };
```

to

```
key <AE07> {    [               7,           slash      ],
                [       braceleft,        NoSymbol      ]       };
key <AE08> {    [               8,       parenleft      ],
                [     bracketleft,        NoSymbol      ]       };
key <AE09> {    [               9,      parenright      ],
                [    bracketright,        NoSymbol      ]       };
key <AE10> {    [               0,           equal      ],
                [      braceright,        NoSymbol      ]       };
```

2.3 Dead keys and accented characters

Dead keys are those that do not type anything until you hit another key. Tildes and umlauts are like this by default under plain Linux if you use the `dk-latin1.map` keymap. This is the default behaviour for these keys under Microsoft Windows as well.

Removing dead key functionality

- Removing dead key functionality under plain Linux and XFree86 v3.1.2 Under plain Linux type

  ```
  loadkeys dk.map
  ```

- Removing dead key functionality under XFree86 v3.2 and higher Put the following line in the `Keyboard` section of your `/etc/XF86Config` file:

  ```
  XkbVariant      "nodeadkeys"
  ```

Invoking dead key functionality

- Invoking dead key functionality under plain Linux Under plain Linux type

  ```
  loadkeys dk-latin1.map
  ```

- Invoking dead key functionality under X11R6 sessions First you must make sure you are running XFree86 v3.1.2 or higher. Download and install everything related to the newest release if you have a lower version number. Neither compose nor dead keys will work in X11R6 applications unless these are compiled with support for accented (8-bit) character input. An example of such an application is `GNU emacs` version 19.30 (or higher.) Some X11 applications still do not support this input method. Eventually this situation might improve, but until that happens you can either hack your applications or submit polite bug reports to the program authors. The latter approach is often the most efficient. See section Programming tips for X11 for some advice on what needs to be done. Next you will have to map a key to `Multi_key` (`Compose`.) The `Scroll Lock` key is most likely already mapped as such if you use XFree86 v3.1.2 (you can verify this with the program `xev`,) and it is easy to map the right `Control` key by uncommenting the appropriate line in the `Keyboard` section of the XFree86 configuration file (often `/etc/XF86Config`.) If you wish to use some other key, or if you are using XFree86 v3.2 or higher and want to change the default, you should put something like

  ```
  keycode 78 = Multi_key
  ```

  in your `~/.Xmodmap` file. The statement in the example defines `Scroll Lock` as the `Compose` key. The default `Compose` key in XFree86 v3.2 and higher is `<Shift><AltGr>`. XFree86 v3.2 and higher comes without support for the dead keys on the standard Danish keyboard. To get this support you have to change a few lines in the `xkb_symbols "basic"` section of the file `/usr/X11R6/lib/X11/xkb/symbols/dk`. The lines

  ```
  key <AE12> {    [           acute,           grave      ],
                  [             bar,     dead_ogonek      ]       };
  key <AD12> {    [       diaeresis,     asciicircum      ],
                  [      asciitilde,     dead_macron      ]       };
  ```

  should be changed to

TOC

```
key <AE12> {    [      dead_acute,      dead_grave      ],
                [             bar,     dead_ogonek      ]
key <AD12> {    [  dead_diaeresis, dead_circumflex      ],
                [      dead_tilde,     dead_macron      ]      };
```

After these changes you can get support for dead keys by removing the line

```
XkbVariant      "nodeadkeys"
```

from the `Keyboard` section of your `/etc/XF86Config` file. (Note for non-Danish readers: There are files for many local keyboard maps in `/usr/X11R6/lib/X11/xkb/symbols`.) The available keystroke combinations are listed in `/usr/X11R6/lib/X11/locale/iso8859-1/Compose`. There are some bugs in that file you will want to fix:

- The line reading

  ```
  <dead_tilde> <space>                    : "~"   tilde
  ```

  should be changed to

  ```
  <dead_tilde> <space>                    : "~"   asciitilde
  ```

- In several places `asciicircum` is misspelled as `asciicirum`

Finally make sure your shells and/or applications are set up for ISO-8859-1 compatibility as described in section International character sets in specific applications and you should be all set.

2.4 Making $ (the dollar sign), ø (oslash) and Ø (Oslash) work

$ (the dollar sign)

There is a bug in the Danish keymaps causing the dollar sign to be accessed with `<Shift><4>` instead of `<AltGr><4>` by default. If this is a problem for you, determine what keymap you load at boot-time. You can find it by looking around in the directory `/etc/rc.d/` or simply by paying attention to what happens at boot-time. On my computer the relevant keymap is called `/usr/lib/kbd/keytables/dk-latin1.map`. You can fix the problem by changing the line

```
keycode   5 = four              dollar          dollar
```

in the keymap file to

```
keycode   5 = four              currency        dollar
```

and then (re-)loading the keytable as described in section Loading a keytable. Currency (dansk: ``soltegn'') is the default `<Shift><4>` character on a Danish keyboard.

This should fix the problem for both X11 and plain Linux.

TOC

ø (oslash) and Ø (Oslash)

In some older distributions ``ø'' and ``Ø'' appear as cent and yen. Find the line for keycode 40 in the keymap file and change it from

```
keycode  40 = cent                yen
```

to

```
keycode  40 = +oslash             +Ooblique
```

This bug appears to have been fixed in `kbd-0.88.tar.gz` and newer versions.

The plus signs are necessary to get `Caps Lock` working properly. ``Oslash'' can be used as an alias for ``Ooblique'' in `kbd-0.90.tar.gz` and newer versions.

You can read more about keyboard configuration at this site.

---

3. Display and application setup

Most applications need to be compiled as ``8-bit-clean'' to work well with European characters. Some need a few extra hints to get it right.

3.1 Loading the ISO-8859-1 font on the console

Execute the following commands from your shell prompt:

```
setfont lat1-16.psf
mapscrn trivial
echo -ne '\033(K'
```

(Note: Change the last line to `echo -n '\033(K'` if you use the `tcsh` shell.)

You could also choose to load a unicode font to ensure that line graphics is displayed correctly in programs such as `mc` and `workbone`. Execute the following command to do that:

```
setfont lat1u-16.psf
```

In Red Hat Linux 5.2 you can do this by putting these lines in `/etc/sysconfig/i18n`:

```
LANG=da
LINGUAS=da
LC_ALL=da_DK
SYSFONT=lat1u-16.psf
SYSTERM=linux
```

Due to a bug in the `ncurses` package on Red Hat Linux 5.2, you also have to change ``linux-lat'' to ``linux'' in `/etc/profile.d/lang.sh`.

3.2 The Euro symbol

A new symbol has been added to the Danish character set: The symbol for the Euro (the new currency of the European Monetary Union.) A new character set called ISO-8859-15 a.k.a. latin0 (or latin9) has been created to replace ISO-8859-1 (latin1.) You must use the EURO package to get support for latin0. The package includes both fonts and keymaps.

3.3 Characters you can display under Linux

TOC

Type `dumpkeys -l | less` at the prompt to find out which characters that are readily available. You can map them to your keyboard via the keymap files mentioned in section Loading a keytable.

3.4 International character sets in specific applications

A number of applications demand special attention. This section describes how to set up configuration files for them.

`bash`:

Put the following in your `~/.inputrc` file:

```
set meta-flag on
set convert-meta off
set output-meta on
```

`elm`:

Put the following definitions in your `~/.elm/elmrc` file:

```
charset = iso-8859-1
displaycharset = iso-8859-1
textencoding = 8bit
```

This may not work on some versions of `elm`. You can get partial **MIME** support in `elm` if you use `metamail`.

`emacs`:

Put the following in your `~/.emacs` or the the system-wide initialization file (probably `/usr/lib/emacs/site-lisp/default.el` or `/usr/share/emacs/site-lisp/default.el`):

```
(standard-display-european t)
(set-input-mode (car (current-input-mode))
        (nth 1 (current-input-mode))
        0)
```

You can leave out the first two of the lines above if you have installed locale support, and your `LC_CTYPE` environment variable includes one of the strings `8859-1` or `88591`. See section Locale support in libc 5.4.x for some information on locales.

Dead keys should work under GNU emacs provided you use GNU emacs v19.30 or higher and XFree86 v3.1.2 or higher (it works for me anyway,) so do not start researching available elisp

TOC

packages implementing ``electric keys'' or anything like that. If you want to implement European keyboard conventions in emacs without upgrading, the best choice is probably the `remap` package available from SunSite DK. There are also two packages called `iso-acc.elc` and `iso-trans.elc` included with emacs that have similar functionality, but they are not nearly as powerful.

`groff`:

Issue the command as

```
groff -Tlatin1 <your_groff_input_file>
```

Remember to change this in `/etc/man.config` to get latin1 characters working in `man` (don't remove the `-mandoc` switch.)

`ispell` --- Spell checking in Danish:

First make sure that you install version 3.1.20 instead of version 4.0 of `ispell`. The latter is obsolete and multiple brain-damaged. You can download the sources for `ispell` at the GNU archive and you can get a Danish dictionary from SSLUG. Follow the compilation instructions and you should have no trouble (One caveat: When defining the variables necessary for compilation you must tell ispell that Linux is a SysV type OS by defining the variable `USG`.)

When you have installed the Danish dictionary for `ispell` you can check the spelling of a Danish language file by executing the command:

```
ispell -d danish -T latin1 -w "æøåÆØÅ" <your_danish_text_file>
```

(Note for non-Danish readers: You can find dictionaries for most Western languages by reading the file `Where` included with the sources for `ispell`.)

`joe`:

Issue the command as

```
joe -asis
```

or put the following in your `~/.joerc` file:

```
-asis
```

The hyphen character *must* be in the first column.

`kermit`:

This is as close as I can get, but not completely satisfying yet. Put the following in your `~/.kermrc` file:

```
set terminal bytesize 8
set command bytesize 8
set file bytesize 8
```

```
set language danish
set file character-set latin1-iso
set transfer character-set latin1-iso
set terminal character-set latin1-iso
```

I think there are more variables to set, but they are hiding. You would have to modify these settings if the remote system is DOS or OS/2 based.

less:

Set the following environment variable:

```
LESSCHARSET=latin1
```

ls:

Issue the command as

```
ls -N
```

or possibly

```
ls --8bit
```

lynx:

Put the following definition in your `~/.lynxrc` file:

```
character_set=ISO Latin 1
```

This can also be set via the `Options` menu in `lynx`. Type `o' and set the relevant option.

man:

See entry for `groff` in this section.

metamail:

Set the following environment variable:

```
MM_CHARSET=ISO-8859-1
```

nn:

Put the following in your `~/.nn/init` file:

```
set data-bits 8
```

pine:

Put the following definition in your `~/.pinerc` file:

```
character-set=ISO-8859-1
```

This can also be set via the `Setup`, `Config` menu option in `pine`. It won't hurt to enable `enable-8bit-esmtp-negotiation` and `enable-8bit-nntp-posting` (for news) in that menu too.

`rlogin`:

Issue the command as

```
rlogin -8 foo.bar.dk
```

`sendmail`:

Put (or uncomment) the following in your `/etc/sendmail.cf` file:

```
O SevenBitInput=False
O EightBitMode=pass8
O DefaultCharSet=iso-8859-1
```

`tcsh`:

Put the following in your `/etc/csh.login` or `~/.tcshrc` file:

```
setenv LANG C
```

Actually you just have to define one of the environment variables `LANG` or `LC_CTYPE`. The value does not matter. Read the `tcsh` man page for more information.

`telnet`:

Put one line of the following type in your `~/.telnetrc` file for **each** host you want to log on to using `telnet`:

```
<hostname> set outbinary true
```

Example:

```
localhost set outbinary true
foo.bar.dk set outbinary true
```

TeX/LaTeX:

There are several problems with TeX/LaTeX: You want LaTeX to understand the special characters and you do not want LaTeX to put in English words like ``Chapter'' at the beginning of every chapter or use English typesetting conventions.

Under LaTeX2e the header of your input file should look something like this:

```
\documentclass[a4paper]{article}

\usepackage[latin1]{inputenc}
```

```
\usepackage{t1enc}
\usepackage[danish]{babel}
```

The first `usepackage` statement ensures that LaTeX will interpret European characters correctly, so you do not have to use escape codes for European characters. The second one is not strictly necessary, but it is recommended including it to use the new EC fonts (previously called DC fonts.) The third `usepackage` statement defines a range of standards for typesetting texts in Danish.

All the major Linux distributions now includes the **teTeX** package. To set up teTeX you must run the script `texconfig`. Here you can choose Danish hyphenation (dansk: ``orddeling''), A4 papersize for `dvips` and `xdvi` etc.

All new Linux distributions include LaTeX2e, but on older systems you might come across LaTeX 2.09. If that happens you can use

```
\documentstyle[a4,isolatin]{article}
```

to include support for ISO-8859-1 characters and European paper sizes. A better thing to do would be to ask your system administrator to upgrade to LaTeX2e.

`isolatin.sty` is available from all CTAN servers.

Some people prefer to use emacs in a special mode which translates ``special'' letters into TeX escape codes, but this method is obsolete.

`tin`:

Put the following definitions in your `~/.tin/headers` file:

```
Mime-Version: 1.0
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 8bit
```

Now you can post messages with the proper Danish characters in the message body.

---

4. Miscellaneous problems

4.1 Time zone

Denmark is placed in the Central European Time zone (CET or MET,) which (in the winter) is equivalent to Greenwich Mean Time plus 1 (GMT+1.) You set the time zone on a Linux system by making a symbolic link between `/usr/lib/zoneinfo/localtime` and the file in `/usr/lib/zoneinfo/` with a name corresponding to your zone or country. Danes will want to execute one of the commands

```
ln -sf /usr/lib/zoneinfo/MET /etc/localtime
```

or

```
ln -sf /usr/lib/zoneinfo/Europe/Copenhagen /etc/localtime
```

This automatically sets Daylight Saving Time (GMT+2) in the summer.

You synchronize the system time with the CMOS clock by issuing the command `clock` as root. If your CMOS clock is set to GMT (a.k.a. UTC --- the standard on proper Unix systems) use

```
clock -u -s
```

or if your CMOS clock is set to local time use

```
clock -s
```

4.2 A4 papersize

- `ghostscript`: Add the command line option `-sPAPERSIZE=a4`.
- `ghostview`: Define the following Xresource:

  ```
  Ghostview.pageMedia:  A4
  ```
- `TeX/LaTeX`, `dvips`, `xdvi`: See the entry for TeX/LaTeX in section International character sets in specific applications.

4.3 Text file formats for other platforms

TOC

You can translate files between an ISO-8859-1 formatted text file and e.g. a DOS text file using codepage 850 with the `recode` package. A DOS file called `foo.txt` would be translated into a proper Unix file with the command

```
recode cp850:latin1 foo.txt
```

`recode` is available as `recode-3.4.tar.gz` from all mirrors of the GNU archive.

---

5. Locale support in libc 5.4.x and higher

The locale support has been updated in `libc 5.4.x`. You can avoid many of the individual program setups described in section International character sets in specific applications if the programs on your system is prepared for locale support. The Debian distribution comes with this support if you install the `wg15-locale` package. Systems with `GNU libc 2` (`libc 6.x`) support locales by default (see remarks about Red Hat Linux release 5.0 later in this section.)

If you use a system without locale support, you can add such support using the following method:

1. Make sure you have the latest `libc 5.4.x` library. You can get this from Yggdrasil Computing.
2. Make sure you have the `localedef` program installed. It should come with the library.
3. Get the locale sources. You can get them from DKUUG. You need to get both locale and charmap sources.
4. Put the locale sources in `/usr/share/i18n/locales/` and the charmap sources in `/usr/share/i18n/charmaps/`.
5. Execute the `localedef` program to build the locale data files:

   ```
   localedef -ci da_DK -f ISO_8859-1:1987 da_DK
   ```

   (Note for non-Danish readers: You can build locale data files for other locales in the same way. All locale and charmap sources are at the DKUUG site.)

To enable support for the Danish locale on a system with locale support you just have to set one of the following environment variables:

```
LANG=da_DK
```

or

```
LC_ALL=da_DK
```

Try `da_DK.ISO_8859-1` if `da_DK` does not work.

Both environment variables set all the individual locale catgories. You can also set a single locale category by using the name of the category as an environment variable. The locale catogories are:

```
Locale category          Application
---------------          -----------
LC_COLLATE               Collation of strings (sort order.)
LC_CTYPE                 Classification and conversion of characters.
LC_MESSAGES              Translations of yes and no.
LC_MONETARY              Format of monetary values.
```

```
LC_NUMERIC                  Format of non-monetary numeric values.
LC_TIME                     Date and time formats.
LC_ALL                      Sets all of the above (overrides all of them.)
LANG                        Sets all the categories, but can be overridden
                            by the individual locale categories.
```

A few programs such as `bash` and `GNU emacs` still need specific setup as described in section International character sets in specific applications, but most should work without further attention. Programs such as `nvi` which did not work with 8 bit characters before should work now.

Locale support should be more common as distributions based on the new `GNU libc 2` become available. Beware that although Red Hat Linux release 5.0 comes with `GNU libc 2`, the locale support is not working. You have to build the locale data files by executing `localedef` yourself. You can build the Danish locale data files with the following command:

```
localedef -c -i da_DK -f ISO-8859-1 da_DK
```

As of `glibc-2.0.7-4.i386.rpm` the locale data files are included with the libraries and this is no longer necessary.

---

6. Programming tips for X11

Displaying 8-bit charaters is easy. You can use them just as you would use 7-bit ASCII. Getting applications to accept input of special characters is an entirely different matter.

If you are using e.g. the Xt toolkit and a widget set like Motif you need only add one line to your program. As your **first** call to Xt use `XtSetLanguageProc`. Like this:

```
int main (int argc, char** argv)
{
    ...
    XtSetLanguageProc (NULL, NULL, NULL);
    top = XtAppInitialize ( ... );
    ...
}
```

Now your program will automagically look up the `LC_CTYPE` variable and interpret dead keys etc. according to the `Compose` tables in `/usr/X11R6/lib/X11/locale/`. This should work for all Western European keyboard layouts and is entirely portable. As XFree86 multilanguage support gets better your program will also be useful in Eastern Europe and the Middle East.

This method of input is supported by Xt, Xlib and Motif v1.2 (and higher.) According to the information I have available it is only partially supported by Xaw. If you have further information on this subject I would like to hear from you.

This section was adapted from a more extensive discussion in Michael Gschwind's Programming for Internationalization. See section Other documents of relevance for a pointer to that document.

7. Getting X11 applications to speak Danish

To get Danish texts on menus, buttons, etc. in a well behaved X11 application, you just have to translate the resource strings defining the texts. Jacob Nordfalk has done such translations for a lot of applications including `Netscape` and `Ghostview`. The translations and a description of how to install them can be found at this site.

8. Information resources

8.1 Other documents of relevance

The HOWTOs are available from all mirrors of `metalab.unc.edu` (the former `sunsite.unc.edu.`) There is a Danish mirror at SunSite DK.

The German HOWTO (in German) by Winfried Trümper. A lot of other national HOWTOs such as Finnish, Spanish and Polish are also available in the native languages.

The Linux Keyboard and Console HOWTO by Andries Brouwer.

The ISO 8859-1 National Character Set FAQ and Programming for Internationalization (plus much more) by Michael Gschwind is available from this site.

8.2 FTP and Web sites

SSLUG (Skåne Sjælland Linux User Group) is a Swedish/Danish Linux user group. Their mailing list is a good place to get help with Linux in Danish (or Swedish.) They are also hosts for this document.

AUC in Ålborg is the home of SunSite DK which has the Debian and Red Hat distributions, the latest kernels, a mirror of the Linux Documentation Project and mirrors of metalab.unc.edu and the GNU archive. There is also a mirror of the CTAN archive with everything you need to get TeX and LaTeX running.

9. Credits and legal stuff

Thanks to Peter Dalgaard, Anders Majland, Jon Haugsand, Jacob Nordfalk, the authors of the German HOWTO, Michael Gschwind and numerous others for suggestions and help with several questions. And a big thanks to the people at Aalborg University Center for writing and making available several of the packages described in this document. A special Thank You to Thomas Petersen; the original author of this document.

9.1 Legal stuff

Trademarks are owned by their owners.

Although the information given in this document is believed to be correct, the author will accept no liability for the content of this document. Use the tips and examples given herein at your own risk.

Copyright © 1996 by Thomas Petersen. Copyright © 1997, 1998, 1999 by Niels Kristian Bech Jensen. This document may be distributed only subject to the terms and conditions set forth in the LDP license.

Diskless Nodes HOW-TO document for Linux

Robert Nemkin buci@math.klte.hu , Al Dev (Alavoor Vasudevan) - Maintainer of this HOWTO alavoor@yahoo.com , Markus Gutschke markus+etherboot@gutschke.com , Ken Yap ken.yap@acm.org , Zurück zu Gero gero@gkminix.han.de

v1.0, 13 May 1999

This document describes how to set up a diskless Linux box. As technology is advancing rapidly, networkcards are becoming cheaper and much faster - 100 MBits ethernet is standard now and in about 1 to 2 years 1000 MBits i.e. 1GigBits ethernet cards will become a industry standard. With high-speed network cards, remote access will become as fast as the local disk access which will make diskless nodes a viable alternative to workstations in local LAN. Also diskless nodes eliminates the cost of software upgrades and system administration costs like backup, recovery which will be centralized on the server side. Diskless nodes also enable "sharing/optimization" of centralised server CPU, memory, hard-disk, tape and cdrom resources. Diskless nodes provides mobility for the users i.e., users can log on from any one of diskless nodes and are not tied to one workstation. Diskless Linux box completely eliminates the need for local floppy disk, cdrom drive, tape drive and hard-disk. Diskless nodes JUST has a network card, 8MB RAM, a low-end cpu and a very simple mother-board which does not have any interface sockets/slots for harddisks, modem, cdrom, floppy etc.. With Diskless linux nodes you can run programs on remote Linux 64 CPU SMP box or even on Linux super-computer! Diskless nodes lowers the "Total Cost of Ownership" of the computer system. This document is copy-righted by Robert Nemkin and other authors as listed above. Copyright policy is GPL. Thanks to Bela Kis bkis@cartan.math.klte.hu for translating this initial document v0.0.3 (which was a mini-howto) to English.

1. Other Formats of this Document

2. Introduction to Network Booting and Etherboot

- 2.1 What is Network booting?
- 2.2 How does it work
- 2.3 Netbooting in Practice
- 2.4 Uses of Network booting
- 2.5 For more information

3. Begin setup

- 3.1 Document Changes
- 3.2 How to set up a diskless Linux box
- 3.3 Related documents
- 3.4 How does it work?
- 3.5 Hardware

4. Fundamental ideas

- 4.1 Setting up the PC
- 4.2 Setting up a bootpd on the server
- 4.3 Configure the bootpd on the server
- 4.4 Understanding tftp
- 4.5 Kernel Image
- 4.6 Setting up a minimal Linux configuration on the remote server
- 4.7 Configuring the tftp server
- 4.8 Final work
- 4.9 Net Loader
- 4.10 RH5 configuration
- 4.11 Gotchas and caveats
- 4.12 X-terminal

TOC

5. Trouble shooting

- 5.1 Memory and diskspace requirements; speed
- 5.2 Possible errors
- 5.3 Errors and possible further expansions of this document

6. EEPROM Burners and Memory chips

- 6.1 List of EEPROM Burner manufacturers
- 6.2 Non-Volatile Memory chips

7. Etherboot

- 7.1 Introduction
- 7.2 Unpacking, compiling and testing the package
- 7.3 Setting up a diskless boot
- 7.4 Testing the network booting
- 7.5 Booting DOS
- 7.6 Making an Etherboot EPROM or EEPROM
- 7.7 Troubleshooting tips
- 7.8 Acknowledgements
- 7.9 Copyright

8. Netboot

- 8.1 Introduction
- 8.2 README
- 8.3 Download Netboot
- 8.4 Netboot useful links
- 8.5 RFC 951
- 8.6 RFC 1533
- 8.7 RFC 1350
- 8.8 Install Instructions
- 8.9 Troubleshoot Problems

---

1. Other Formats of this Document

This document is published in 10 different formats namely - DVI, Postscript, Latex, LyX, GNU-info, HTML, RTF(Rich Text Format), Plain-text, Unix man pages and SGML.

- You can get this HOWTO document as a single file tar ball in HTML, DVI, Postscript or SGML formats from - ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/other-formats/
- Plain text format is in: ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO
- Translations to other languages like French, German, Spanish, Chinese, Japanese are in ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO Any help from you to translate to other languages is welcome.

The document is written using a tool called "SGML tool" which can be got from - http://www.xs4all.nl/~cg/sgmltools/ Compiling the source you will get the following commands like

- sgml2html databasehowto.sgml (to generate html file)
- sgml2rtf databasehowto.sgml (to generate RTF file)
- sgml2latex databasehowto.sgml (to generate latex file)

This document is located at -

- http://sunsite.unc.edu/LDP/HOWTO/Diskless-HOWTO.html

Also you can find this document at the following mirrors sites -

- http://www.caldera.com/LDP/HOWTO/Diskless-HOWTO.html
- http://www.WGS.com/LDP/HOWTO/Diskless-HOWTO.html
- http://www.cc.gatech.edu/linux/LDP/HOWTO/Diskless-HOWTO.html
- http://www.redhat.com/linux-info/ldp/HOWTO/Diskless-HOWTO.html
- Other mirror sites near you (network-address-wise) can be found at http://sunsite.unc.edu/LDP/hmirrors.html select a site and go to directory /LDP/HOWTO/Diskless-HOWTO.html

In order to view the document in dvi format, use the xdvi program. The xdvi program is located in tetex-xdvi*.rpm package in Redhat Linux which can be located through ControlPanel | Applications | Publishing | TeX menu buttons.

```
To read dvi document give the command -
        xdvi -geometry 80x90 howto.dvi
And resize the window with mouse. See man page on xdvi.
To navigate use Arrow keys, Page Up, Page Down keys, also
you can use 'f', 'd', 'u', 'c', 'l', 'r', 'p', 'n' letter
keys to move up, down, center, next page, previous page etc.
To turn off expert menu press 'x'.
```

You can read postscript file using the program 'gv' (ghostview) or 'ghostscript'. The ghostscript program in ghostscript*.rpm package and gv program is in gv*.rpm package in Redhat Linux which can be located through ControlPanel | Applications | Graphics menu buttons. The gv program is much more user friendly than ghostscript. Also ghostscript and gv are available on other platforms like OS/2, Windows 95 and NT, you view this document even on those platforms.

```
To read postscript document give the command -
        gv howto.ps

To use ghostscript give -
        ghostscript howto.ps
```

---

2. Introduction to Network Booting and Etherboot

This chapter is written by Ken Yap and explains how to bootstrap your computer from a program stored in non-volatile memory without accessing your hard disk. It is an ideal technique for maintaining and configure a farm of linux boxes

2.1 What is Network booting?

Network booting is an old idea. The central idea is that the computer has some bootstrap code in non-volatile memory, e.g. a ROM chip, that will allow it to contact a server and obtain system files over a network link. One goal is to avoid the use of a hard disk for booting. There are various reasons for doing this. One is to reduce the cost of maintaining software on many different machines. With network booting the files are held at a central server and can be updated at one location. Another goal is to use computers in locations where hard disks are not robust enough. For example this might be a computer on a factory floor where a hard disk might be too fragile. Finally another goal is to have a system that can be switched between different operating systems without having to reload the software. Network booting often co-exists with disk booting. For example, a system could run Windows from disk but sometimes boot Linux from the network. There are some interesting applications of this technique. For example: a friend of mine uses this to reload Windows over the network. When a Windows installation becomes corrupted, as it often does, the sysadmin can reload a fresh installation by booting Linux over the network and letting the automatic script format the disk and copy a fresh Windows installation onto it.

2.2 How does it work

In order to boot over the network, the computer must get 1. an identity, 2. an operating system image and 3. usually, a working filesystem.

Consider a diskless computer (DC) that has a network boot ROM. It may be one of several identical DCs. How can we distinguish this computer from others? There is one piece of information that is unique to that computer (actually its network adapter) and that is its Ethernet address. Every Ethernet adapter in the world has a unique 48 bit Ethernet address because every Ethernet hardware manufacturer has been assigned blocks of addresses. By convention these addresses are written as hex digits with colons separating each group of two digits, for example: 00:60:08:C7:A3:D8.

The protocols used for obtaining an IP address, given an Ethernet address, are called Boot Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP). DHCP is an evolution of BOOTP. In our discussion, unless otherwise stated, anything that applies to BOOTP also applies to DHCP. (Actually it's a small lie that BOOTP and DHCP only translate Ethernet addresses. In their foresight, the designers made provision for BOOTP and DHCP to work with any kind of hardware address. But Ethernet is what most people will be using.)

An example of a BOOTP exchange goes like this:

TOC

DC: Hello, my hardware address is 00:60:08:C7:A3:D8, please give me my IP address.

BOOTP server: (Looks up address in database.) Your name is aldebaran, your IP address is 192.168.1.100, your server is 192.168.1.1, the file you are supposed to boot from is /tftpboot/vmlinux.nb (and a few other pieces of information).

You may wonder how the DC found the address of the BOOTP server in the first place. The answer is that it didn't. The BOOTP request was broadcast on the local network and any BOOTP server that can answer the request will.

After obtaining an IP address, the DC must download an operating system image and execute it. Another Internet protocol is used here, called Trivial File Transfer Protocol (TFTP). TFTP is like a cut-down version of FTP---there is no authentication, and it runs over User Datagram Protocol (UDP) instead of Transmission Control Protocol (TCP). UDP was chosen instead of TCP for simplicity. The implementation of UDP on the DC can be small so the code is easy to fit on a ROM. Because UDP is a block oriented, as opposed to a stream oriented, protocol, the transfer goes block by block, like this:

```
DC: Give me block 1 of /tftpboot/vmlinux.nb.
TFTP server: Here it is.
DC: Give me block 2.
```

and so on, until the whole file is transferred. Handshaking is a simple acknowledge each block scheme, and packet loss is handled by retransmit on timeout. When all blocks have been received, the network boot ROM hands control to the operating system image at the entry point.

Finally, in order to run an operating system, a root filesystem must be provided. The protocol used by Linux and other Unixes is normally Network File System (NFS), although other choices are possible. In this case the code does not have to reside in the ROM but can be part of the operating system we just downloaded. However the operating system must be capable of running with a root filesystem that is a NFS, instead of a real disk. Linux has the required configuration variables to build a version that can do so.

2.3 Netbooting in Practice

Besides commercial boot ROMs, there are two sources for free packages for network booting. They are Etherboot and Netboot. Both can be found through the Etherboot home page. First you have to ascertain that your network card is supported by Etherboot or Netboot. Eventually you have to find a person who is willing to put the code on an EPROM (Erasable Programmable Read Only Memory) for you but in the beginning you can do network booting from a floppy.

To create a boot floppy, a special boot block is provided in the distribution. This small 512 byte program loads the disk blocks following it on the floppy into memory and starts execution. Thus to make a boot floppy, one has only to concatenate the boot block with the Etherboot binary containing the driver for one's network card like this:

cat floppyload.bin 3c509.lzrom > /dev/fd0

TOC

Before you put in the network boot floppy, you have to set up three services on Linux: BOOTP (or DHCP), TFTP and NFS. You don't have to set up all three at once, you can do them step by step, making sure each step works before going on to the next.

I assume you have installed the bootpd server from a distribution or by compiling from source. You then have to ensure that this server is waiting for bootp requests. There are two ways to do this: one is to start bootpd as a network service that is always listening when the computer is up, and the other is to start it from inetd. For the latter, /etc/inetd.conf should contain a line like this:

bootps dgram udp wait root /usr/sbin/tcpd bootpd

If you had to modify /etc/inetd.conf, then you need to restart inetd by sending the process a HUP signal.

Next, you need to give bootp a database to map Ethernet addresses to IP addresses. This database is in /etc/bootptab. It contains lines of the following form:

aldebaran.foo.com:ha=006008C7A3D8:ip=192.168.1.100:bf=/tftpboot/vmlinuz.nb

Other information can be specified but we will start simple.

Now boot the DC with the floppy and it should detect your Ethernet card and broadcast a BOOTP request. If all goes well, the server should respond to the DC with the information required. Since /tftpboot/vmlinux.nb doesn't exist yet, it will fail when it tries to load the file. Now you need to compile a special kernel, one that has the option for mounting the root filesystem from NFS turned on. You also need to enable the option to get the IP address of the kernel from the original BOOTP reply. You also need to compile the Linux driver for your network adapter into the kernel instead of loading it as a module. It is possible to download an initial ramdisk so that module loading works but this is something you can do later.

You cannot install the zImage resulting from the kernel compilation directly. It has to be turned into a tagged image. A tagged image is a normal kernel image with a special header that tells the network bootloader where the bytes go in memory and at what address to start the program. You use a program called mknbi-linux to create this tagged image. This utility can be found in the Etherboot distribution. After you have generated the image, put it in the /tftpboot directory under the name specified in /etc/bootptab. Make sure to make this file world readable because the tftp server does not have special privileges.

For TFTP, I assume that you installed tftpd from a distribution or by compiling from source. Tftpd is normally started up from inetd with a line like this in /etc/inetd.conf.

tftp dgram udp wait root /usr/sbin/tcpd in.tftpd -s /tftpboot

Again, restart inetd with a HUP signal and you can retry the boot and this time it should download the kernel image and start it. You will find that the boot will continue until the point where it tries to mount a root filesystem. At this point you must configure and export NFS partitions to proceed.

For various reasons, it's not a good idea to use the root filesystem of the server as the root filesystem of the DCs. One is simply that there are various configuration files there and the DC will get the wrong information that way. Another is security. It's dangerous to allow write access (and write access is needed for the root filesystem, for various reasons) to your server's root. However the good news is that a root

filesystem for the DC is not very large, only about 30 MB and a lot of this can be shared between multiple DCs.

Ideally, to construct a root filesystem, you have to know what files your operating system distribution is expecting to see there. Critical to booting are device files, files in /sbin and /etc. You can bypass a lot of the hard work by making a copy of an existing root filesystem and modifying some files for the DC. In the Etherboot distribution, there is a tutorial and links to a couple of shell scripts that will create such a DC root filesystem from an existing server root filesystem. There are also troubleshooting tips in the Etherboot documentation as this is often the trickiest part of the setup.

The customised Linux kernel for the DC expects to see the root filesystem at /tftpboot/(IP address of the DC), for example: /tftpboot/192.168.1.100 in the case above. This can be changed when configuring the kernel, if desired.

Now create or edit /etc/exports on the server and put in a line of the following form:

/tftpboot/192.168.1.100 aldebaran.foo.com(rw,no_root_squash)

The rw access is needed for various system services. The no_root_squash attribute prevents the NFS system from mapping root's ID to another one. If this is not specified, then various daemons and loggers will be unhappy.

Start or restart the NFS services (rpc.portmap and rpc.mountd) and retry the diskless boot. If you are successful, the kernel should be able to mount a root filesystem and boot all the way to a login prompt. Most likely, you will find several things misconfigured. Most Linux distributions are oriented towards disked operation and require a little modification to suit diskless booting. The most common failing is reliance on files under /usr during the boot process, which is normally imported from a server late in the boot process. Two possible solutions are: 1. Provide the few required files under a small /usr directory on the root filesystem, which will then be overlaid when /usr is imported, and 2. Modify the paths to look for the files in the root filesystem. The files to edit are under /tftpboot/192.168.1.100 (remember, this is the root directory of the DC).

You may wish to mount other directories from the server, such as /usr (which can be exported read-only).

When you are satisfied that you can boot over the network without any problems, you may wish to put the code on an EPROM. An EPROM programmer starts at around $100 US, and is not a cost effective investment for a hobbyist who uses it sporadically. Occasionally one will appear on the used market at a bargain price, the main caveat being to ensure that the software to drive it is available. A proficient electronics hobbyist could build one using one of the several free designs published on the Internet, but for the majority of readers, the best solution is to make the acquaintance of someone who has access to one, perhaps someone in an electronics hobbyist group or working in the electronics industry.

A short note on EPROM technology: The bits of an EPROM are programmed by injecting electrons with an elevated voltage into the floating gate of a field-effect transistor where a 0 bit is desired. The electrons trapped there cause that transistor to conduct, reading as 0. To erase the EPROM, the trapped electrons are given enough energy to escape the floating gate by bombarding the chip with ultraviolet radiation through the quartz window. To prevent slow erasure over a period of years from sunlight and fluorescent lights, this quartz window is covered with an opaque label in normal use.

There is another technology, called EEPROM or Electrically Erasable PROM, sometimes called Flash PROM. Here the bits are cleared by an electrical signal. This obviates the need for an ultraviolet eraser, the EPROM is to be reused, but requires additional circuitry to support the erase phase. If one is handy with electronics, there is a contributed circuit design and driver software for an EEPROM board in the Etherboot distribution. The board is plugged into any spare ISA bus slot on a PC and boots a network card plugged into some other slot.

2.4 Uses of Network booting

X-terminals are one natural use of network booting. The lack of a disk in the terminal makes it quieter and contributes to a pleasant working environment. The machine should ideally have 16MB of memory or more and the best video card you can find for it. This is an ideal use for a high-end 486 or low-end Pentium that has been obsoleted by hardware advances. Other people have used network booting for clusters of machines where the usage is light on the DC and does not warrant a disk, e.g. a cluster of classroom machines.

2.5 For more information

Your first stop should be the Etherboot home page: http://www.slug.org.au/etherboot/

There you will find links to other resources, including a mailing list you can subscribe to, where problems and solutions are discussed.

---

3. Begin setup

3.1 Document Changes

Following are the history of changes to this document:

- v0.0.3 12 Sep 1996: Some minor error fixes
- v1.0.0 13 May 1999: Added some useful URLs. Changed this HOWTO to main HOWTO from mini-howto. Also rewrote this HOWTO using SGML tool. Added more chapters written by Ken Yap, Robert Nemkin.

3.2 How to set up a diskless Linux box

This document is about setting up a diskless Linux box. Sometimes it might be necessary to run Linux on PC's which have neither hard disks nor floppy drives. If a network, another Unix system with bootp, tftp, an NFS server, and an eprom burner is available then it is possible to set up and operate Linux without hard/floppy disks.

3.3 Related documents

- NFS-root Mini Howto
- Linux NET-2/3-HOWTO by Terry Dawson, 94004531@postoffice.csu.edu.au
- /usr/src/linux/README about configuring and compiling new kernels

3.4 How does it work?

- 1.Diskless computer (DC) broadcasts MAC address with bootp: Who am I?
- 2.Bootp or DHCP server on S looks up DB: Your IP address is X.X.X.X, your server is S, your boot file is vmlinuz.myname, etc.
- 3.DC asks to load file from TFTP server on S: Please give me vmlinuz.myname
- 4.S: Here you are (/tftpboot/vmlinuz.myname)

DC thinks a while (booting Linux).

- 5.DC: Please let me mount / with NFS
- 6.S: Here is your root FS (/tftpboot/IPnumber). (In 2.2 kernels, /tftpboot/domainname.)
- 7.DC: Please let me mount other NFSes (/usr, /home/, etc)
- 8.S: Here you are
- 9.DC: Runs intended application

Network boot ROM contains code to do 1 and 3.

3.5 Hardware

Whatever is described here was checked on the following configuration

- Sun-OS 4.1.3 as boot server
- Slackware 2.3 + Linux 1.2.8 + wd 8013 ethercard.
- Working Ethernet lan

---

4. Fundamental ideas

The fundamental idea is as follows: the PC will get its IP address from the boot server via the bootp protocol, using 0.0.0.0 as the initial IP address and its kernel via the tftp protocol. (-- Booting across segments (via router) not a simple question, so either put both the server and the diskless boxes on the same lan segment or configure an UDP helper address in your router to the address of the server. Refer to your router product manual for further info.--)

For this follow the steps below.

4.1 Setting up the PC

Get the nfsboot package (the package is available from your favourite linux mirror site in the /pub/Linux/system/Linux-boot directory). It contains a booteprom image for the wd8013 card which can be directly burned in.

There are alternative ways to prepare the PC:

- If your machine is not quite diskless, then you may use the little DOS program, or
- The binary floppy image contained in the same package. If you choose the latter option you must write the image onto a floppy by the dd command.

These images contain a bootp and tftp client. You need to prepare a linux kernel too, which contains the nfs-root option.

- If you are using the latest stable kernel, linux-1.2.13, then you need to patch the kernel with the patchfile included in the nfsboot package (-- Refer to patch(1)--)
- If you try to use the latest, but unstable kernel from the linux-1.3.x series, then you have to configure in the nfs-root option.

You may or may not configure block device (floppy or hard disk) support, but you must configure tcp/ip support, wd ethernet card support, nfs filesystem support. Then recompile the kernel as usual.

4.2 Setting up a bootpd on the server

It can be found in package bootpd-2.4.tar.gz (which can be found on your favourite linux mirror site in the /pub/Linux/system/Network/boot.net directory). Get the package, compile and install it. If your other Unix box happens to be a Slackware Linux then you may skip this step for the standard distributions contain a bootpd. The daemon can be run either directly by issuing command

```
       bootpd -s
```

or by using inetd. In this case you need to edit:

```
·  /etc/inetd.conf to remove the hashmark from the start of these
lines:
```

```
# tftp   dgram   udp     wait    root    /usr/sbin/in.tftpd     tftpd /export
# bootps dgram   udp     wait    root    /usr/sbin/in.bootpd    bootpd
```

```
·  insert or uncomment the following two lines in /etc/services:
```

```
bootps          67/tcp          # BOOTP server
tftp            69/udp          # TFTP server
```

- restart inetd by

```
kill -HUP <process id of inetd>.
```

4.3 Configure the bootpd on the server

First of all, bootpd have a config file called bootptab which usually resides in /etc. You must modify it by inserting the IP addresses of your gateway, dns server, and the ethernet address(es) of your diskless machine(s). An example /etc/bootptab:

```
global.prof:\
        :sm=255.255.255.0:\
        :ds=192.168.1.5:\
        :gw=192.168.1.19:\
        :ht=ethernet:\
        :bf=linux:
machine1:hd=/export/root/machine1:tc=global.prof:ha=0000c0863d7a:ip=192.168.1.140:
machine2:hd=/export/root/machine2:tc=global.prof:ha=0800110244e1:ip=192.168.1.141:
machine3:hd=/export/root/machine3:tc=global.prof:ha=0800110244de:ip=192.168.1.142:
```

global.prof is a general template for host entries, where

- sm field contains the subnet mask
- ds field contains the address of the Domain Name Server
- gw field contains the default gateway address
- ht field contains the lan media hardware type
- bf field contains the name of the boot file

After this, every machine must have a line:

- the first field contains the host name,
- hd field contains the directory of the bootfile,
- the global template can be included with the tc field,
- ha field contains the hardvare address of the ethernet card,
- ip field contains the assigned ip address.

4.4 Understanding tftp

TFTP (Trivial File Transfer Protocol) is a file transfer protocol, such as ftp, but it's much simpler to help coding it in EPROMs. TFTP can be used in two ways:

- simple tftp: means that the client can acces to your whole file system. It's simpler but it's a big security hole (anyone can get your password file via tftp).
- secure tftp: the tftp server uses a chroot.2 system call to change it's own root directory. Anything outside the new root directory will be completelly inaccessible. Because of the chroot dir becomes the new root dir, the hd filed in the bootptab must reflect the new situation. For example: when using insecure tftp, the hd field contains the full path to the boot directory: /export/root/machine1. When using secure tftp whith /export as root dir, then /export becomes / and the hd field must be /root/machine1.

Almost every Unix implementation contains tfpt server, probably you don't need to install your own one.

Install tftpd, make sure it's active in /etc/inetd.conf, typical line

tftp dgram udp wait root /usr/sbin/tcpd in.tftpd /tftpboot

TOC

4.5 Kernel Image

You must compile a kernel for the DC that includes NFS support and NIC driver compiled in (not modules). Answer yes to Root file system on NFS? and BOOTP support?

After building the kernel, run mknbi-linux from the Etherboot distribution on it. Install this tagged image as /tftpboot/(bf attribute in bootptab).

4.6 Setting up a minimal Linux configuration on the remote server

This may contain packages a, ap, n, and x of the Slackware distribution. To install more is OK; however the above packages suffice for the purposes of a diskless X terminal. For the installation you need a working Linux system. Find some disk space on the remote machine and export it read-write. Mount the exported directory onto somewhere (e.g. /mnt) on the file system of the Linux box. Start Linux setup and change the root option in the setup from / to /mnt. Then setup the above packages as usual. If you want to run no more than one diskless Linux then no changes are needed. On the other hand, if you plan to use more than one diskless machine then the above setup will not work because some files and directories must be private to the machines. The problem can be bypassed by moving the /usr (it contains no private data) and then create a separate subdir for each diskless machine. For example, if /export/linux/machine1 were mounted to /mnt then the directory structure after the initial setup will look like

```
/export/linux/machine1/bin
/export/linux/machine1/sbin
/export/linux/machine1/lib
/export/linux/machine1/etc
/export/linux/machine1/var
/export/linux/machine1/usr
```

After the changes you will have

```
/export/linux/machine1/bin
/export/linux/machine1/sbin
/export/linux/machine1/lib
/export/linux/machine1/etc
/export/linux/machine1/var
/export/linux/usr
```

Now create the subdirectories for the other machines. Assume for now that your diskless machines are called machine1, machine2, machine3, etc.; then you may use the following bash script to setup the other directories

```
cd /export/linux
for x in machine2 machine3 ; do
        mkdir $x; cd $x
        (cd ../machine1; tar cf - *) | tar xvf -
done
```

Then do the following export:

- /export/linux/usr readonly for everyone.
- /export/liunx/machine1 only to machine1 with rw,root rights.
- /export/liunx/machine2 only to machine2 with rw,root rights.

- /export/liunx/machine3 only to machine3 with rw,root rights.

as follows (-- the format of this example follows the SunOs 4.1.3 exports file syntax--) :

```
# This file is /etc/export
# for remote linux X terminals by Buci
# this line is only once
/export/root/usr             -access=linuxnet
# these lines once for every host
/export/root/machine1       rw=machine1,root=machine1
/export/root/machine2       rw=machine2,root=machine2
/export/root/machine3       rw=machine3,root=machine3
```

Don't forget to run exportfs -a.

4.7 Configuring the tftp server

Now it is time to configure the tftp server. If you do not need secure tftp then everything is quite simple for your clients can be booted from the /export directory.

If a secure tftp is used then you can either make a full /export/linux directory structure under /tftpboot (with a single real kernel and symbolic links for the other machines), or let the /export directory be the boot directory of the secure tftpd. Or, if you have a separate tftpboot directory then, similarly, you need only the original directory structure with a single kernel and symbolic links for the others. You can achieve this setup by typing the following:

```
mkdir -p /tftpboot/export/linux/machine1
cd /tftpboot/export/linux/machine1
cp /export/linux/machine1/<name of the kernel> .
```

Then type the following:

```
mkdir -p /tftpboot/export/linux/machine2
cd ../machine2
ln -s ../machine2/<name of the kernel>
```

4.8 Final work

Finally, you must insert

```
/sbin/mount nfs_server:/export/linux/usr /usr
```

as the first line of

```
/export/linux/<machinex>/etc/rc.d/rc.S
```

```
where <machinex> stands for machine1, machine2, etc.
```

4.9 Net Loader

A small program that runs as a BIOS extension, usually on an EPROM on the NIC. It handles the BOOTP query and TFTP loading and then transfers control to the loaded image.

It uses TCP/IP protocols but the loaded image doesn't have to be Linux. The loaded image can be anything, even DOG.

There are two free implementations of TCP/IP net loaders: Etherboot and Netboot : Etherboot uses built-in drivers while Netboot uses Packet drivers.

They can also be loaded from a floppy for testing and for temporary setups.

4.10 RH5 configuration

The DC requests to mount /tftpboot/(IP address of DC) (in 2.1 and above: /tftpboot/(name of DC in bootptab)) as its / by NFS from server. You must export this from the server (rw, no_root_squash) because the DC wants to write on it (log files, etc).

The / must contain /sbin, /bin, /lib, /etc, /var, /tmp, /root, /dev and /proc.

/sbin, /bin, /lib/ can be a copy of an existing RH5 system. They can be shared between all DCs. But hard links only. BTW, don't link to server originals.

/etc, /var and /dev should be non-sharable copies. Customise /etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-eth0, /etc/fstab, /etc/conf.modules, and others. Turn off all network services you don't need. Remove all stuff you don't need from /var, e.g. RPM db, lpd files.

/root and /proc should just exist. /tmp should exist and be mode 1777.

You probably want to create /usr and /home mount points. /usr can be mounted ro.

About 10 MB per DC plus about 15 MB of shared files should be sufficient. BTW: if your DCs are quite similar, the kernel image can also be shared.

Here is an illustrative script to create the first root filesystem.

```
#!/bin/sh
if [ $# != 1 ]
then
        echo Usage: $0 client-IP-addr
        exit 1
fi

cd /

umask 022

mkdir -p /tftpboot/$1

# just make these ones
for d in home mnt proc tmp usr
do
        mkdir /tftpboot/$1/$d
        done

        chmod 1777 /tftpboot/$1/tmp

        touch /tftpboot/$1/fastboot
        chattr +i /tftpboot/$1/fastboot
```

```
        # copy these ones
        cp -a bin lib sbin dev etc root var /tftpboot/$1

cat <<EOF
Now, in /tftpboot/$1/etc, edit

                sysconfig/network
                sysconfig/network-scripts/ifcfg-eth0
                fstab
                conf.modules

and configure

                rc.d/rc3.d
EOF
```

Here is an illustrative script to duplicate the root filesystem

```
#!/bin/sh
if [ $# != 2 ]
then
        echo Usage: $0 olddir newdir
        exit 1
fi

cd /tftpboot

if [ ! -d $1 ]
then
        echo $1 is not a directory
        exit 1
fi

umask 022

mkdir -p $2

# just make these ones
for d in home mnt proc tmp usr
do
        mkdir $2/$d
done

chmod 1777 $2/tmp

touch $2/fastboot
chattr +i $2/fastboot

# link these ones
for d in bin lib sbin
do
        (cd $1; find $d -print | cpio -pl ../$2)
done

# copy these ones
for d in dev etc root var
do
```

```
        cp -a $1/$d $2
done

cat <<EOF
Now, in /tftpboot/$2/etc, edit

        sysconfig/network
        sysconfig/network-scripts/ifcfg-eth0
        fstab (maybe)
        conf.modules (maybe)

and configure

        rc.d/rc3.d
EOF
```

4.11 Gotchas and caveats

RH5 wants to fsck the root FS. I stopped this with a /fastboot. But init script wants to delete it, so I did chattr +i /fastboot

/etc/localtime is a link to TZ file in /usr/share/... I made it a copy.

Turn off /etc/rc.d/rc6.d/K97network or it will disable the network before root FS is done with.

X server wants to write into /usr/X11R6/lib/X11/xkb/compiled. I made this a link to /etc/X11/kbd/compiled

Remember your DC will keep appending to log files so have logrotate or something deal with them at regular intervals.

4.12 X-terminal

On the server, make sure the DC is matched by a clause in /etc/X11/xdm/Xaccess and comment out the :0 in /etc/X11/xdm/Xservers. Then make sure that xdm is run from the init scripts.

On the client, run X -query server

You will get the xdm login box and then all your X clients will run on the server.

For other applications use - you could use diskless technique for netboot routers, print servers (but should not be spooling print server), standalone apps, etc.

---

5. Trouble shooting

5.1 Memory and diskspace requirements; speed

- I tested this for only Slackware 2.3; for other distributions/versions the following numbers may vary.
- Diskspace: 28MB + 6.5MB/machine
- RAM: I am using X on 8 MB. For only 4MB swap is needed, I guess, which can be created -- separately for each machine -- in /tmp. Do not forget to run mkswap.
- Speed: I had no problems on a 486 DX2/66 with 8 Megs.

5.2 Possible errors

- I found a strange error: in the /dev subdirectory SunOS corrupted the device entries so I needed to rerun MAKEDEV by mounting the subdirectory onto a disk based Linux box. (The reason was the differences between the linux nfs and the SunOs nfs: both use 32 bit for the Major and Minor device number, but linux uses 16 bit wide fields for both, SunOs uses 14 bit wide field for Major and 18 bit wide filed for Minor device number.)
- When the diskless linux gets booted, there is only one route included in the routing table to the tftp server, so you need to set up correct routing tables. You have two choices here:
- Configure every rc.S for every machine by hand
- Use a bootp client package and write a generalized setup script

5.3 Errors and possible further expansions of this document

- Correct citation of related documents.
- SunOs is BSD based. Need to include SVR4 (e.g. Solaris) based server configuration.
- Although Linux is quite similar to SunOs as bootp/tftp server, a linux based server example might be usefull.
- Update this document to the current etherboot package.
- Show the differences between the nfs root patched kernel version 1.2.13 and the newest 1.3.x kernel, which contains the nfs-root patch.
- Need to try other ethercards than wd8013
- Include configuration information for bootpc, a bootp client for linux to set up the correct rooting tables.

- Typos and other errors: please, report to buci@math.klte.hu Thank you.

6. EEPROM Burners and Memory chips

6.1 List of EEPROM Burner manufacturers

For a list of **EPROM burner manufacturers** visit the Yahoo site and go to economy->company->Hardware->Peripherals->Device programmers. Click on this site

- Yahoo URL for EPROMs at http://dir.yahoo.com/Business_and_Economy/Companies/Computers/Hardware/Peripherals/Device_Programmers/
- Advanced Research Technology B.V - development, production and sales of electronic programmer equipment; development of hard- and software.
- Advin Systems Inc. - PC-based device programmers that support the latest in package types and device technologies.
- Andromeda Research Labs - manufactures a portable eprom and device programming system.
- B and C Microsystems, Inc - offers test and duplication/programming equipment for PCMCIA (PC) Cards, ISA/PCI Cards, SIMMs, Memory Devices (including FLASH), PLDs.
- BP Microsystems - Device Programmers.
- Bytek - designs, develops, manufactures and markets micro-processor-based, modular electronic systems used to program and test semiconductor devices. Product line includes the ChipBurner.
- Concentrated Programming Ltd - offers a full range of device programming solutions.
- Dataman Programmmers Ltd. - manufacture of hand-help EPROM programmer/emulator. Also sell PC-based programmers, and Gang-Pro programmers.
- General Device Instruments - IC Device programmers. Universal and Gang programmers for Pld, Flash, microcontrollers, Proms, EEproms, Memory, Epld, Mach and many other ic devices.
- HI-LO System Research Co., Ltd. - manufacturer of universal and gang device programmers.
- ICE Technology - EPROM and universal device programmers which support memories, microcontrollers, and programmable logic devices.
- Iceprom - in-circuit erasable programmable read-only memory.
- Incept Ltd.
- International Microsystems Inc - High speed reliable gang programmer. (PROM, FLASH, Microcontroller, PCMCIA memory card).
- JED Microprocessors Pty. Ltd. - plugs into a PC printer port D25 connector, and programs any 28-pin or 32-pin EPROM and FLASH device.
- Logical Devices, Inc - device programming for PLDs, FPGAs, PROMs, microcontrollers. Producers of CUPL compiler for programmable logic and the ALLPRO and Chipmaster device programmer.
- MCL Systems - new method not only for programming but also for developing your new hardware with Integrated Controller Unit. And you don't need to be an expert.
- MQP Electronics - manufacturer of universal device programmers, gang programmers, production software, and package converters. High thoughput and reliability.
- Needham's Electronics - manufacturer of device programmers.
- NP Programming Services - provides programming for memory and logic parts.

- Program Automation, Inc. - independent service company specializing in high volume PROM programming, including flash I/Cs.
- Stag Programmers Inc - manufacturer of prom and logic programmers, production handling equipment and UV erasers.
- Sunrise Electronics - universal device programmers, gang and in-circuit programmers with life time support.
- System General Co. - Device Programmer, EPROM Writer and IC Tester
- Tribal Microsystems - universal and gang device programmers, 8051 and EPROM emulators, test and burn-in sockets and production sockets.
- Universal Device Programmers

6.2 Non-Volatile Memory chips

This chapter gives brief descriptions of memory chips.

- **PROM**: Pronounced prom, an acronym for programmable read-only memory. A PROM is a memory chip on which data can be written only once. Once a program has been written onto a PROM, it remains there forever. Unlike RAM, PROMs retain their contents when the computer is turned off. The difference between a PROM and a ROM (read-only memory) is that a PROM is manufactured as blank memory, whereas a ROM is programmed during the manufacturing process. To write data onto a PROM chip, you need a special device called a PROM programmer or PROM burner. The process of programming a PROM is sometimes called burning the PROM. An EPROM (erasable programmable read-only memory) is a special type of PROM that can be erased by exposing it to ultraviolet light. Once it is erased, it can be reprogrammed. An EEPROM is similar to a PROM, but requires only electricity to be erased.
- **EPROM**: Acronym for erasable programmable read-only memory, and pronounced ee-prom, EPROM is a special type of memory that retains its contents until it is exposed to ultraviolet light. The ultraviolet light clears its contents, making it possible to reprogram the memory. To write to and erase an EPROM, you need a special device called a PROM programmer or PROM burner. An EPROM differs from a PROM in that a PROM can be written to only once and cannot be erased. EPROMs are used widely in personal computers because they enable the manufacturer to change the contents of the PROM before the computer is actually shipped. This means that bugs can be removed and new versions installed shortly before delivery.
- **EEPROM**: Acronym for electrically erasable programmable read-only memory. Pronounced double-e-prom or e-e-prom, an EEPROM is a special type of PROM that can be erased by exposing it to an electrical charge. Like other types of PROM, EEPROM retains its contents even when the power is turned off. Also like other types of ROM, EEPROM is not as fast as RAM. EEPROM is similar to flash memory (sometimes called flash EEPROM). The principal difference is that EEPROM requires data to be written or erased one byte at a time whereas flash memory allows data to be written or erased in blocks. This makes flash memory faster.
- **FRAM**: Short for Ferroelectric Random Access Memory, a type of non-volatile memory developed by Ramtron International Corporation. FRAM combines the access speed of DRAM and SRAM with the non-volatility of ROM. Because of its high speed, it is replacing EEPROM in many devices. The term FRAM itself is a trademark of Ramtron.
- **NVRAM**: Abbreviation of Non-Volatile Random Access Memory, a type of memory that retains its contents when power is turned off. One type of NVRAM is SRAM that is made non-volatile by connecting it to a constant power source such as a battery. Another type of NVRAM uses EEPROM chips to save its contents when power is turned off. In this case, NVRAM is composed of a combination of SRAM and EEPROM chips.
- **Bubble Memory**: A type of non-volatile memory composed of a thin layer of material that can be easily magnetized in only one direction. When a magnetic field is applied to circular area of this substance that is not magnetized in the same direction, the area is reduced to a smaller circle, or bubble. It was once widely believed that bubble memory would become one of the leading memory technologies, but these promises have not been fulfilled. Other non-volatile memory types, such as EEPROM, are both faster and less expensive than bubble

memory.

- **Flash Memory**: A special type of EEPROM that can be erased and reprogrammed in blocks instead of one byte at a time. Many modern PCs have their BIOS stored on a flash memory chip so that it can easily be updated if necessary. Such a BIOS is sometimes called a flash BIOS. Flash memory is also popular in modems because it enables the modem manufacturer to support new protocols as they become standardized.

7. Etherboot

7.1 Introduction

This is the README file for the Etherboot package. This document explains how to install, configure and use the Etherboot package. The instructions here apply to version 4.0 of Etherboot.

Etherboot is a package for creating ROM images that can download code over the network to be executed on an x86 computer. Typically the computer is diskless and the code is Linux, but these are not the only possibilities. The code uses the bootp, tftp and NFS Internet Protocols.

What hardware is supported?

Etherboot supports the following network hardware (in no particular order): 3c503, 3c507, 3c509, 3c905b, NE1000, NE2000 (also the PCI cards, with the nepci driver), WD8003, WD8013, SMC8216/8416, Lance based cards such as the NE2100 and NI6510 (also the PCI Lance cards), Crystal CS89x0, Intel EtherExpress Pro, SMC 83c170 EPIC/100, SMC9000, Realtek 8139, NI5210, Schneider and Koch G16, and Tiara (Fujitsu Lancard). All Etherboot drivers are autoprobing, which means they attempt to detect the hardware addresses at which the card is installed. It's fairly easy to write a driver if you are familiar with Ethernet hardware interfacing. Please contact me for more information if you are interested in doing so or are interested in having a driver written.

Availability of this document

This document and related documents are also kept online at the Etherboot Home Page. This will in general have the latest distributions and documentation.

For a talk/tutorial type introduction to what Etherboot does and how to set it up, see my SLUG talk. You may wish to review this before reading further.

An older version of this README, which is now out of date, can be accessed via this link.

Getting help

There is a mailing list for all netbooting related issues. To subscribe follow the instructions on the Etherboot home page.

With the exception of the following section, the information on diskless booting is not specific to Etherboot but can be used for the Netboot package also.

7.2 Unpacking, compiling and testing the package

This section is Etherboot specific.

Unpacking the distribution

TOC

Unpack the distribution using gunzip and tar, using either of the following commands:

```
tar zxvf etherboot-4.0.tar.gz
gunzip < etherboot-4.0.tar.gz | tar xvf -
```

Compiling the ROM images

Precompiled ROM images are provided in the bin/ directory but you may wish to review the options compiled in and make your own versions. For the 32 bit version you need a recent release of gcc and the binutils tools. This package was compiled with the tools from a RedHat 5.2 distribution but it should work with any recent Linux distribution. For the 16 bit version you need the bcc tools from the Embedded Linux Kernel Subset (ELKS) project, for more details see the notes on 16 bit Etherboot. You need the 16 bit version only if you intend to run Etherboot on a 286 or 086/088 PC.

Assuming you have decided to make the 32 bit version, you only have to go to src-32/, edit the options in Config and say make. This will create all the ROM images available. The .lzrom images are the same as the .rom images. Since the .lzrom images are smaller and work exactly the same, there is no real reason to use .rom images any more, unless you are nervous about compression algorithm patents. We believe the algorithm used does not infringe patents, having been in public use for some time, but we do not know all the legal ramifications. See here for more details.

Here is a brief description of the options available:

```
Basic options:

-DDHCP_SUPPORT  - Use DHCP instead of BOOTP (default on)
-DIMAGE_MENU    - Allow to interactively chose between different
                  bootimages; read vendortags.html for further
                  information.
-DMOTD          - Display message of the day; read vendortags.html
                  for further information.
-DANSIESC       - evaluate a subset of common ANSI escape sequences
                  when displaying the message of the day; this
                  probably does not make sense unless you also
                  define -DMOTD or at least -DIMAGE_MENU.
                  Combining this option with -DSERIAL_CONSOLE
                  is a waste of EPROM space.
-DGFX           - support extensions to the ANSI escape sequences for
                  displaying graphics (icons or logos); this
                  requires -DANSIESC
-DASK_BOOT=n    - Ask "Boot from Network or from Local? " at startup,
                  timeout after n seconds (0 = no timeout); this
                  can be done in a more generic way by using the
                  IMAGE_MENU, but it requires that the "bootp"
                  server is accessible, even when booting locally.
-DANS_DEFAULT=ANS_NETWORK
                - Assume Network to previous question
```

TOC

```
                    (alternative: ANS_LOCAL) on timeout or Return key
                    See etherboot.h for prompt and answer strings.
-DEMERGENCYDISKBOOT
                  - if no BOOTP server can be found, then boot from
                    local disk. The accessibility of the TFTP server
                    has no effect, though! So configure your BOOTP
                    server properly.

Basic options only on Etherboot/32:

-DPASSWD          - enable password protection for boot images; this
                    requires -DIMAGE_MENU
-DUSRPARMS        - allow the user to interactively edit parameters
                    that are passed to the booted kernel; you should
                    probably enable -DPASSWD as well; this feature
                    requires -DIMAGE_MENU
-DFLOPPY          - boot from floppy/hd if bootimage matches the
                    pattern "/dev/[fh]d*"; if you do not have
                    enough space in the EPROM, then disable this
                    feature and use "mknbi-blkdev" for booting
                    from a local blockdevice.
-DCONFIG_PCI_DIRECT
                  - define this for PCI BIOSes that do not implement
                    BIOS32 or not correctly

These options should normally not need to be touched:

-DNOINT19H        - Take control as soon as BIOS detects the ROM
                    Normally hooks onto INT19H
-DMOVEROM         - if your motherboard does not cache adapter memory
                    space, then this option can speed up loading of
                    compressed BOOT-Prom images. It has no effect on
                    uncompressed images. Unless you are very tight on
                    free space, you will usually want to define this
                    option.
-DDELIMITERLINES - print a line of = characters at the start
                    and also just before starting an image.
-DSIZEINDICATOR - update a running total of the amount of code
                    loaded so far, in kilobytes
-DT509HACK        - send two bootp packets before waiting for a
                    reply to the first. Makes a 3c509 do bootp
                    quicker
-DT503_AUI        - Use AUI by default on 3c503 cards.

These options are only for a serial console for Etherboot/32:

-DSERIAL_CONSOLE- use a serial line for input and output
-DCOMPORT         - 0x0 for COM1, 0x1 for COM2 etc
-DCOMPARM         - configuration for COMPORT, save values:
                     0xe3 == 9600/8n1, 0xa3 == 2400/8n1
```

TOC

Testing the ROM images

You can test the image with a floppy before burning an EPROM. On Linux just put a blank floppy in fd0 and say make card.fd0 where card is the name of your network card and it will copy a bootable image onto the floppy. If you wish to do this by hand, it's easy, just prepend floppyload.bin to card.rom (or card.lzrom) and write this combined binary to the floppy raw, i.e. starting at the boot block. Like this:

```
cat floppyload.bin 3c509.lzrom > /dev/fd0
```

When you boot with this floppy it will load the Etherboot ROM image from floppy and execute it. It should be able to detect your card. To get the bootrom to acquire an IP address and load the intended code, you need to set up bootp, tftp and NFS services, which we will discuss next.

We suggest you continue to use floppy booting until you have completed the setup of the server and are satisfied that diskless booting works.

7.3 Setting up a diskless boot

In this section I assume you want to boot a Linux kernel. Booting a DOS kernel is similar, the main differences being in the way you set up the tagged image.

Making a tagged image

Etherboot expects to download a tagged image containing the code to be executed. Briefly explained, a tagged image is a wrapper around the pieces of code or data that need to be put in various places in the computer's memory. It contains a directory telling how large the pieces are and where they go in memory. It also says where to start execution.

A tagged image is created using a utility program. The utility program is specific to the kernel you want to load. The version for Linux is called mknbi-linux and that for DOS is mknbi-dos. These utilities are found in the netboot-<version> directory of the distribution.

Compiling a custom kernel

You will probably have to compile a custom kernel because the kernel needs to have the "Root file system on NFS" option compiled in. You should also select "BOOTP support". "RARP support" is not needed. In 2.2 kernels you have to enable the "Kernel Autoconfig" option to access the BOOTP support question. And unless you are using an initrd (initial ramdisk) you will probably have to compile in the driver for your network card too. For details, see the file /usr/src/linux/Documentation/nfsroot.txt in a Linux kernel source distribution.

After you have compiled the custom kernel, make the tagged image, typically like this:

```
mknbi -x -k zImage -o /tftpboot/vmlinuz.xterm
```

Then put the tagged image in where the tftp daemon expects to find it, typically /tftpboot. Make sure it is world-readable because typically the tftp daemon runs as an unprivileged user.

Setting up a bootp daemon

Now set up a bootp daemon. In RedHat 5.2 this means installing the bootp RPM package, making sure that the bootps service is active in /etc/inetd.conf, and editing /etc/bootptab. The essential pieces of information you need to put in bootptab are:

1. The domain name of the machine.
2. The Ethernet (MAC) address of the network card, which you generally obtain from a sticker on the card, a configuration program for the card, or in the last resort, from watching the output of Etherboot or from the packets sent from the card when trying to boot, using the debug option of bootpd.
3. The name of the tagged image file, relative to the tftpboot directory.
4. The IP address you intend to give it.
5. The IP addresses of various servers. You will need at least the tftp server's address.

Here is an example of a /etc/bootptab for the bootpd supplied with RedHat Linux 5.2 and probably many versions of Unix:

```
.default:\
        :ht=ethernet:\
        :hd=/tftpboot:bf=null:\
        :ds=nameserver:\
        :hn:to=36000:
xterm.ken.net.au:tc=.default:ha=08002BB7F380:ip=192.168.26.100:bf=vmlinuz.xterm
```

The first entry sets up some common defaults which applies to all succeeding entries which can be "included" using the tc=.default attribute. The first field is the domain name of the machine. The ha attribute is the Ethernet address. The ip attribute is self-explanatory. The bf field specifies the tagged image filename. For more details, consult the bootptab man page.

Please note that if you use the ef (extension file) attribute to be able to send more configuration data to the diskless machine, you must run bootpef everytime bootptab is modified.

If you are on a local network that is not directly connected to the Internet, you can use the "private" IP addresses 192.168.x.y (or in the other ranges mentioned in RFC1918). Otherwise please ask either your network administrator or your Internet service provider for your own IP address(es).

Setting up a DHCP daemon

As an alternative to bootp, you could set up a DHCP server which has the advantage of automating the handing out of IP addresses. However the kernel may still do a bootp request to find the IP address for mounting the NFS filesystem. You may then wish to investigate the option in mknbi-linux which tells the kernel to take the address from the initial DHCP reply.

More information about DHCP can be found at the DHCP FAQ Web Page.

Setting up a tftp daemon

Now set up a tftp daemon. In RedHat 5.2 this means installing the tftp RPM package and making sure that the tftp service is active in /etc/inetd.conf. You probably want to use the secure (-s) option so that files can only be fetched from /tftpboot or people may be able to fetch arbitrary files from your server.

7.4 Testing the network booting

TOC

Now when you start up Etherboot, it should obtain an IP address and print out what it received. If you do not get this to work, turn on debugging in bootpd and see if any query was received. If not, check your network hardware (cables, etc). If a query was received, check if bootpd was able to give an answer. If not, then the Ethernet address was not found in /etc/bootptab. If a reply was sent, then only faulty hardware or a bug in Etherboot would prevent it being received by Etherboot.

Assuming an IP address was received, the next thing Etherboot tries to do is load a file using tftp. Check your system logs to see if a tftp daemon was started up and a file requested. Generally if you run tftpd under tcpwrapper security, a log entry will be generated. If not, it could be a path problem or file permission problem (the file needs to be readable by tftpd). Fix the problem.

After the tagged image is loaded, Etherboot will jump to it. If it crashes here, check that the image is a tagged image. If it executes and stops at the point where it's trying to mount the NFS root, then you need to check that you have the "root on NFS" option compiled in and that you have compiled in the network card driver.

Setting up a NFS root filesystem

Now you need to set up a NFS root filesystem for the diskless computer. Typically this is under /tftpboot/<ip address of computer>. (In 2.1 and presumably future kernels, this should be /tftpboot/<name of computer in bootptab>). This needs to contain a complete root filesystem that will make the kernel boot happily. This means, for most kernels, it should contain /dev, /proc, /etc, /sbin, /bin, /tmp and /var. The details vary from distribution to distribution. Being lazy I just make a copy of the necessary files from an existing RedHat 5.2 filesystem and modify some key files appropriately. You can find a description in my tutorial and some shell scripts to copy the files. Since the amount of disk space needed is relatively small in these days of large disks, I don't bother to throw out things that may not be needed.

Warning: Do not attempt to reuse the root filesystem of your server, whether by exporting it directly or by making hard links (symbolic links will not work). First of all, the configuration files will contain information pertaining to the server, not the client, so your client will get the wrong information. Secondly, this is a security risk. NFS is already not totally safe, but this way you are directly exposing your server root to clients. Even if you make hard links, the clients could (maliciously or accidentally) overwrite key binaries, making the server unusable. Don't try to save a few megabytes of disk space this way. You can however share some directories between clients, typically /sbin, /bin and /lib. The sample scripts in the tutorial show you how.

The root filesystem should be exported rw and no_root_squash because the various processes need to be root and need to write to log files in the root partition. You may wish to export /usr and /home filesystems to the diskless computer also. These do not need no_root_squash permission. Be aware that the RedHat 5.2 distribution has a few "bugs" relating to symlinks and so forth for diskless booting. These are mentioned in the tutorial. Other distributions may have similar problems.

Swap over NFS

Swap over NFS can be arranged but you have to patch the kernel source. There is a patch for kernels in the contrib/ directory for 2.0.x kernels. Hopefully this will make it into future kernels. Be aware that opinions are divided on NFS swap. Some people think it's a bad thing because it just kills the network if you have lots of diskless computers and that you shouldn't be running into a swap regime on a diskless computer anyway. Some other people like having a bit of insurance.

There are patches in the contrib directory for NFS swap but for up to date patches, try here.

Also have a look at the NBD Network Block Device web page for swapping over that. This requires a 2.1 or 2.2 kernel.

7.5 Booting DOS

What about DOS? The deal with DOS is that one is loading a virtual floppy called A: into extended memory and then booting from this floppy. So you have to capture an image of a bootable DOS floppy first. Some more details can be found in the mknbi-dos directory.

I have booted DOS (both M$ and DR versions) diskless this way. Note that extended memory is used so that rules out 086/088 computers but 286s are ok. See this document for more details.

If you were thinking of booting a Windows machine via the network, it seems (I'm not masochistic enough to do this) the problem is not the network booting but the mounting of a file system over NetBIOS (Windows does not do remote mounts of root filesystems over NetBIOS on TCP). So that rules out a Samba server. It appears to be possible over a Netware server, for which Linux has workalikes. But then what do you do about the networking stack? This situation may change with with future Samba developments. But you will still have problems with pathnames and the usual Windows hassles. Do you really want to do this? In the Web page for Etherboot, there is a link to someone's Web page explaining how this was done with a commercial TCP/IP boot ROM.

7.6 Making an Etherboot EPROM or EEPROM

Assuming you have satisfactorily set up your server environment, you may now wish to put the Etherboot onto an EPROM or EEPROM. Naturally this assumes access to hardware to burn (and possibly erase) EPROMs. An alternative is to use an EEPROM card. There is a schematic and PCB artwork for such a card in the distribution. This EEPROM card plugs onto the ISA bus and can be reprogrammed with software.

Choosing the EPROM

Most network cards come with a blank (E)EPROM socket even though it is seldom used. When it is used, it is typically filled with a proprietary EPROM from the network card manufacturer. You can put an Etherboot EPROM there instead.

Enabling the EPROM

First you must discover how to enable the EPROM socket on your card. Typically the EPROM is not enabled at the factory and a jumper or a soft configuration is used to turn it on.

Size and speed of the EPROM

Secondly, you must discover what size and speed of EPROM is needed. This can be difficult as network card manufacturers often neglect to provide this information.

The smallest EPROM that is accepted by network cards is an 8K EPROM (2764). Some cards will even go up to 64KB EPROMs (27512). You want to use the smallest EPROM you can so that you don't take up more of the upper memory area than needed as other extensions BIOSes may need the space. However you also want to get a good price for the EPROM. Currently the 32KB and 64KB EPROMs (27256 and 27512) seem to be the cheapest per unit. Smaller EPROMs appear to be more expensive because they are out of mainstream production.

If you cannot find out from the documentation what capacity of EPROM your card takes, you could do it by trial and error. Take an ROM with some data on it (say a character generator ROM) and plug it into the socket. Be careful not

to use an extension BIOS for this test because it may be detected and activated and prevent you from booting your computer. Using the debug program under DOS, dump various regions of the memory space. Say you discover that you can see the data in a memory window from CC00:0 to CC00:3FFF (= 4000 hex = 16384 decimal locations). This indicates that a 16 KB EPROM is needed. However if you see an alias in parts of the memory space, say the region from CC00:0 to CC00:1FFF is duplicated in CC00:2000 to CC00:3FFF, then you have put an 8 KB EPROM into a 16 KB slot and you need to try a larger EPROM.

Note that because pinouts for 28 pin EPROMs are upward compatible after a fashion, you can probably use a larger capacity EPROM in a slot intended for a smaller one. The higher address lines will probably be held high so you will need to burn the image in the upper half or upper quarter of the larger EPROM, as the case may be. However you should double check the voltages on the pins armed with data sheet and a meter because CMOS EPROMs don't like floating pins.

The speed of the EPROM needed depends on how it is connected to the computer bus. If the EPROM is directly connected to the computer bus, as in the case of many cheap NE2000 clones, then you will probably have to get an EPROM that is at least as fast as the ROMs used for the main BIOS. This is typically 150 ns. Some network cards mediate access to the EPROM via an ASIC and this ASIC may insert wait states so that slower EPROMs can be used. Incidentally the slowness of the EPROM doesn't affect Etherboot execution speed much because Etherboot copies itself to RAM before executing. I'm told Netboot does the same thing.

7.7 Troubleshooting tips

- Floppy boot doesn't work. Have you copied the ROM image (with the floppyloader prepended) to the floppy raw? Is that size of floppy bootable by your computer? Are you trying to run a 32 bit Etherboot on a 16 bit machine (286, 086/088)? Have you selected too many compile time options? The real limit on Etherboot is not the size of the EPROM but the fact that it executes in the 32 KB region between 0x98000 and 0xA0000. If the sum of code, stack and bss is greater than 32 KB, then Etherboot might crash at unexpected places. You could increase the memory to 48 KB by lowering the RELOCADDR in the Makefile but this is outside the specifications. Definitely do not lower RELOCADDR below 0x94000 because various pieces of booting information are stored from 0x90000 upwards.
- Floppy version works but EPROM version doesn't work. There is a program called rom-scan (Linux and DOS versions) in the directory contrib/rom-scan which will help detect problems.
  - If the EPROM is not detected at all then the contents of the EPROM are not visible to the BIOS. Check that you have enabled the EPROM with any jumpers or soft configuration settings. Check that you do not have any conflicts in the memory address of the EPROM and any other hardware. Perhaps you have to prevent it from being mapped out by your BIOS settings. Or perhaps you have to shadow it with RAM. Maybe you put the code in the wrong half or wrong quarter of the EPROM. Maybe the access time of the EPROM is not low enough. You can also use the debug program under BIOS to examine the memory area in question.
  - If rom-scan says the EPROM is present but not active, then something prevented the BIOS from seeing it as a valid extension BIOS. This could be truncation of the EPROM window, maybe you have a larger EPROM in a slot meant for a smaller one. Maybe there is a checksum error in the EPROM due to some bits not properly programmed or the EPROM not being fast enough. In one case that we know of, the 3c503 network card, the ASIC actually returns 2 bytes of 80 80 in the end locations of the EPROM space. This apparently is a kind of signature. The makerom program in Etherboot compensates for this, but if the pattern is not 80 80, then the code needs to be modified.
  - If rom-scan says the EPROM is present and active, but BIOS does not see it, then perhaps the EPROM is located in an area that the BIOS does not scan. The range scanned is supposed to be 0xC0000 to 0xEF800 in increments of 2 KB but I have seen some BIOSes that continue the scan into the 0xF0000 page.

TOC

Note that rom-scan will also detect other extension BIOSes mounted on your computer, for example VGA BIOSes and SCSI adapter BIOSes. This is normal.

- Etherboot does not detect card. Are you using the right ROM image? Is the card properly seated in the computer? Can you see the card with other software? Are there any address conflicts with other hardware? Is the PCI id of the card one that is not known to Etherboot yet? In this case and where you think there is a bug in Etherboot, please contact the author with all details.
- Etherboot detects card but hangs computer after detection. Some cards are booby traps while they are enabled. The typical offenders are NE2000s which will hang the bus if any access is made to the reset addresses while interrupts are active. You may need to do a hard reset of the computer, i.e. power down and up again before running Etherboot.
- Etherboot detects card but does nothing after saying Searching for server. Check your network hardware. Did you select the right hardware interface (AUI, BNC, RJ45)? Is the cabling ok? If you have a Unix computer on the network and have root privileges, you could run tcpdump looking for broadcast packets on the bootps port. If the requests are getting sent out but no replies are getting back, check your bootpd setup. Also check if the server has a route to the client.
- Etherboot obtains IP address but fails to load file. Check the tftp server. Is the tagged image file installed? Is the file world readable? Did you put the right home directory and boot filename in bootptab?
- Etherboot loads file via tftp but Linux fails to boot. This is a large category. Here are some suggestions:
  - You do not have a private copy of the /, /etc, /var, ... directories
  - Your /dev directory is missing entries for /dev/zero and/or /dev/null or is sharing device entries from a server that uses different major and minor numbers (i.e. a server that is not running Linux).
  - Your /lib directory is missing libraries (most notably libc* and/or libm*) or does not have the loader files ld*.so*
  - You neglected to run ldconfig to update /etc/ldconfig.cache or you do not have a configuration file for ldconfig.
  - Your /etc/inittab and/or /etc/rc.d/* files have not been customized for the clients.
  - Your kernel is missing some crucial compile-time feature (such as NFS filesystem support, booting from the net, transname (optional), ELF file support, networking support, driver for your ethernet card).
  - Missing init executable (in one of the directories known by the kernel: /etc, /sbin, ?). Remember /sbin/init must be a real file, not a symlink.
  - Missing /etc/inittab
  - Missing /dev/tty?
  - Missing /bin/sh
  - System programs that insist on creating/writing to files outside of /var (mount and /etc/mtab* is the canonical example)

  The essence is that you must provide whatever is needed in the NFS root filesystem that your kernel needs to boot. This is somewhat distribution dependent. In the discussion above I solved the problem by making a copy of an existing root filesystem and modifying a few bits. Be aware that some, if not all, distributions contain bugs in their layout that hinder diskless mounting so you will have to fix any problems that arise. An example was a directory in /usr/X11R6/lib that needed to be writable, preventing /usr from being mounted ro.
- Etherboot works fine and kernel starts but network interface doesn't work. Check your network configuration in the OS. Etherboot uses polled I/O and not interrupt I/O to fetch the images. So the IRQ line doesn't get exercised. This manifested itself in one case with a NIC that could netboot but network programs run afterwards couldn't receive any packets. It seemed to be a combination of a weak NIC IRQ line and a fussy motherboard. But the same thing could happen if say the IRQ is not set to where the software is expecting it. The image will load fine with the wrong IRQ but the software will not run properly. This is more of an issue with say DOS packet drivers, where the user has to specify the IRQ line than with Linux, which autoprobes.

7.8 Acknowledgements

TOC

The following people have contributed substantially to Etherboot. If you feel your name has been left out, just let me know and I will fix it up.

Markus Gutschke

Co-author of Etherboot. He was the person who ported the Netboot suite from FreeBSD. He has enhanced Etherboot with many features, one new driver and has contributed various utilities and addons.

Gero Kuhlmann

The mknbi utilities used by Etherboot are from Netboot. He has also clarified the original specification by Jamie Honan.

Jamie Honan

Jamie started Netboot off by writing the first version that used code from a packet driver.

Martin Renters et. al

The original authors of Netboot on FreeBSD.

Bruce Evans

Created bcc compiler used by Etherboot/16.

Rob de Bath

Current maintainer of bcc and associated tools like as86.

Gerd Knorr

Contributed MASQ for making a boot floppy without DOS.

Adam Richter

Contributed comboot for making a boot floppy without DOS.

Claus-Justus Heine

Contributed patch for serial console and NFS swapping. See the contrib/nfs-swap directory for his Web page.

Dickon Reed

Contributed display of loading status and a hack for the 3c509 card.

David Munro

Contributed PCI detection code originally from Linux sources.

Charlie Brady

Donated NE2100 card so that a driver could be written, and helped test the LancePCI driver. Spotted bug with 4.1 header code.

Rogier Wolff

Created Intel EtherExpressPro 100 driver and binary to hex converter.

Vlag Lungu

Contributed patches to work with DHCP. Also contributed a fix to match the received XID against the transmitted one, important in a network with many requesters.

William Arbaugh

TOC

Patches for eepro to work with 3.2.

Jean Marc Lacroix

Contributed an improved bin2intelhex.

Jim Hague

Contributed fixes to 3c503 driver for PIO mode, fix to makerom for presetting EPROM bytes, and various endian fixes.

Andrew Coulthurst

Contributed patch for making Intel eepro work in 4.0.

Doug Ambrisko

Contributed patches to start32.S from FreeBSD version to make it boot Windoze after answering N to Boot from Network question.

Alex Harin

Contributed patches for prepended loaders and makerom to make bootrom PnP and PCI compatible.

Peter Dobcsanyi

Contributed vendor and device IDs for the Netvin NE2000/PCI clone.

adam@mudlist.eorbit.net

Contributed RARP code as alternative to BOOTP/DHCP. Activated by RARP_NOT_BOOTP define.

Daniel Engstrom

Contributed a SMC9000 driver.

Didier Poirot

Contributed an Etherpower II (EPIC 100) driver.

Martin Atkins

Contributed mntnbi for mounting DOS NBIs.

Atilla Bogar

Contributed a bug fix to the bootmenu code and a patch to main.c to remove looping menus on failure. Also code for ARP replies and TFTP retransmit (#ifdefed). Cleanup of tftp and tftpd.

Nathan R. Neulinger

Found bug due to tu_block being declared signed short in arpa/tftp.h on many platforms when it should be unsigned short.

David Sharp

Contributed a FreeBSD driver for Tulip based cards. Ken Yap ported it to Etherboot. Not tested because code needs to be written for all the variants of the Tulip and also because no hardware available to me.

Greg Beeley

Contributed a 3c905b driver. Be sure to read the release notes in 3c905b.txt before using.

Alex Nemirovsky

Contributed patches to use BIOS call to size memory otherwise Etherboot was trampling on top of 640kB area, which is used by some extended BIOSes. Also contributed patches to pci.c to implement PCI bus support on

TOC

BIOSes that do not implement BIOS32, or incorrectly.

Guenter Knauf

Suggested making the ASK_BOOT prompts more generic and clearer. (Why doesn't SGML-Tools accept &uumlaut;?) Also contributed a DOS utility for extracting the identifier string and PCI IDs, if any, out of the boot ROM.

Klaus Espenlaub

Contributed various cleanup patches to the code especially in the bootmenu area, as well as a completely revamped start32.S. Also introduced Rainer Bawidamann's code, see next paragraph.

Rainer Bawidamann

Contributed a Realtek 8139 driver.

Georg Baum

contributed a Schneider & Koch G16 driver.

jluke@deakin.edu.au

sent in a fix for the WD/SMC8013 which I finally verified.

7.9 Copyright

The boot code from FreeBSD is under the BSD license. The netboot code is under the GPL. This is not a problem really: the GPL states that mere aggregation of another work with a GPL'ed work on a storage medium, e.g. an archive file, does not put the other work under the GPL.

Send changes to this document to Ken Yap (ken.yap@acm.org).

---

8. Netboot

This chapter was written by Zurück zu Gero. The main site is at http://www.han.de/~gero/netboot.html.

8.1 Introduction

What do I need it for? The following list shows just a few examples of what Netboot can be used for:

- Printer spooler
- Terminal server
- X11 terminal
- Data logging system
- Network-Computer (NC)
- Some more ....

Basically Netboot can be used for any cases in which a computer should be utilized without having a hard disk. But also for computers with a hard disk Netboot can be used if you want to hold just one operating system on your disk and additionally want to be able to boot a different system over the network.

What operating systems are supported? Presently Netboot can boot Linux and MS-DOS. Since Windows-95 is also based on MS-DOS that can be used as well. In the future support is also planned for OS/2 and Free-BSD. However, Windows-NT will never be supported. The utility programs, which come with Netboot should run on an UNIX like system.

What do I need in order to work with netboot? Obviously first of all you need one (or more) computers which should get bootet. Those are called clients. Netboot itself should be able to run on almost any PC compatible system, even with the oldest 8088 and 80286 processors. However, it has been optimized to run on 386 and up. Of course this system also needs a network card. Since Netboot uses standard DOS packet drivers, all network cards can be used for which packet drivers exist, even PCI and 100MBps cards. In case no such driver has been delivered with a network card it can usually be downloaded from an FTP server of the manufacturer, such as 3Com, SMC or Intel. In addition there are packet drivers for many popular network cards available for free from the Crynwr packet driver collection. Next you need a system which acts as the server. Because Netboot uses the BOOTP and TFTP standard protocols this server has to be able to understand those. It doesn´t matter if the server runs a UNIX like operating system or Windows-NT.

What does it cost? Nothing! Netboot is given away for free and covered by the GNU General Public Licence.

How does it work? In order to boot a computer without a hard disk a bootrom is necessary, which gets plugged into a socket on the network card. Netboot contains the full source code and executable binaries for such a bootrom. After poweron this bootrom queries a BOOTP server for the system´s network name and boot image file name, which then gets loaded using the TFTP protocol. This boot image file contains the operating system which get executed by the bootrom after loading. Netboot contains all necessary utilities to build such a boot image file for Linux and MS-DOS.

8.2 README

Copyright Notice, Acknowledgements

You are allowed to modify and redistribute this code under the terms of the GNU General License. By making use of this software in any way you are automatically agreeing to these terms. My special thanks go to Jamie Honan for defining the initial bootrom specifications, to Markus Gutschke for enhancing mknbi-linux, and to Jens-Uwe Mager for explaining to me the basics of the NFS networking system and sometimes helping me out with his wealth of knowledge and experience.

TOC

Overview

Booting a computer without a hard disk usually requires a floppy disk drive or a network connection in order to get the operating system into the RAM and up running. This package allows a diskless PC to boot an operating system using an IP based ethernet network. This is done in the following steps:

- First the operating system has to be loaded into system memory. This can be done using a bootrom, which loads the operating system image over the network. The bootrom code found in this package can be burned into a real EPROM, or just copied onto a floppy and then started from it upon bootup. Please note that the term bootrom in all documentation texts in this package refers to the real EPROM version as well as the version copied onto and started from a floppy disk. For the bootrom to find the kernel image it uses the BOOTP protocol as defined in RFC 951 and RFC 1533 to get the necessary boot information, and then loads the actual image using the TFTP protocol as defined in RFC 1350 .
- When the operating system starts running, it has to setup it's memory layout using a special boot loader, which is included into the boot image. Then it has to mount it's root filesystem. This can either be done using NFS over the network, or using a ramdisk.

The exact specifications for this netboot process can be found here.

Features

Booting a diskless client either from an EPROM or from a floppy without additional utility tools The Bootrom code can use many standard DOS packet drivers and therefore supports almost any PC network card. Easy configuration of the bootrom under UNIX. The bootrom can load a BOOTP vendor extension file in order to support more space for tags according to RFC 1533 . Using the bootrom menu support it's possible to optionally boot different operating systems. Even if there is a hard disk installed in the client computer, the bootrom can be used as a boot manager, since it also allows booting from any partition. Cross-gateway booting Bootrom runs on any 80x86 processor but is optimized for the 386+. Able to boot different operating system. Currently supported are Linux and MS-DOS.

Installation

See the file Install Instructions for installation instructions. If you have any problems installing or running the programs see the file PROBLEMS .

Mailing list

There exists a mailing list devoted to network booting. To subscribe simply send a mail with the line

subscribe netboot

in it's body to majordomo@baghira.han.de

The subject in the mail header doesn't matter. This mailing list is intended to be a general discussion forum about any aspect of network booting. Any new versions of this Netboot package will also get announced on that mailing list. After subscribing to it, you can send messages into the list by writing a mail to netboot@baghira.han.de.

Disclaimer

The software in this package heavily interacts with hardware and software not only on the local system, but also on the boot server. Use it at your own risk. I cannot be held responsible for any damage done by this software.

8.3 Download Netboot

You can download netboot version 0.9 from http://www.han.de/~gero/netboot/files/netboot-0.9.0e.tar.gz and netboot version 0.8.1 from http://www.han.de/~gero/netboot/english/download.html

8.4 Netboot useful links

Netboot mailing list archive is at http://www.han.de/~gero/netboot/archive/maillist.html

- 3com drivers at http://support.3com.com/infodeli/tools/nic
- Accton drivers at here
- Artisoft
- CNET
- Compaq
- D-Link
- Microdyne
- Many NE2000 PCI cards are based on Realtek chipsets. Get drivers here
- Standard Microsystems Corp
- Surecom
- Thomas Conrad corp
- Winbond
- Xircom
- Webopaedia page on network cards
- Jargon's driver page with many drivers for older network cards.
- Etherboot This is a project similar to Netbot but based on the BSD bootrom code.
- How to make an X Window Terminal out of your old or outdated PC.
- List of jumper settings for various network cards. This page also contains many other good links.
- Freefire is the home page of the Freefire project, which lists many resources for network security issues.

8.5 RFC 951

```
Network Working Group                    Bill Croft (Stanford University)
Request for Comments: 951                 John Gilmore (Sun Microsystems)
                                                          September 1985

                        BOOTSTRAP PROTOCOL (BOOTP)


1. Status of this Memo

   This RFC suggests a proposed protocol for the ARPA-Internet
   community, and requests discussion and suggestions for improvements.
   Distribution of this memo is unlimited.

2. Overview
```

TOC

This RFC describes an IP/UDP bootstrap protocol (BOOTP) which allows a diskless client machine to discover its own IP address, the address of a server host, and the name of a file to be loaded into memory and executed. The bootstrap operation can be thought of as consisting of TWO PHASES. This RFC describes the first phase, which could be labeled 'address determination and bootfile selection'. After this address and filename information is obtained, control passes to the second phase of the bootstrap where a file transfer occurs. The file transfer will typically use the TFTP protocol [9], since it is intended that both phases reside in PROM on the client. However BOOTP could also work with other protocols such as SFTP [3] or FTP [6].

We suggest that the client's PROM software provide a way to do a complete bootstrap without 'user' interaction. This is the type of boot that would occur during an unattended power-up. A mechanism should be provided for the user to manually supply the necessary address and filename information to bypass the BOOTP protocol and enter the file transfer phase directly. If non-volatile storage is available, we suggest keeping default settings there and bypassing the BOOTP protocol unless these settings cause the file transfer phase to fail. If the cached information fails, the bootstrap should fall back to phase 1 and use BOOTP.

Here is a brief outline of the protocol:

1. A single packet exchange is performed. Timeouts are used to retransmit until a reply is received. The same packet field layout is used in both directions. Fixed length fields of maximum reasonable length are used to simplify structure definition and parsing.

2. An 'opcode' field exists with two values. The client broadcasts a 'bootrequest' packet. The server then answers with a 'bootreply' packet. The bootrequest contains the client's hardware address and its IP address, if known.

3. The request can optionally contain the name of the server the client wishes to respond. This is so the client can force the boot to occur from a specific host (e.g. if multiple versions of the same bootfile exist or if the server is in a far distant net/domain). The client does not have to deal with name / domain services; instead this function is pushed off to the BOOTP server.

4. The request can optionally contain the 'generic' filename to be

TOC

booted. For example 'unix' or 'ethertip'. When the server sends the bootreply, it replaces this field with the fully qualified path name of the appropriate boot file. In determining this name, the server may consult his own database correlating the client's address and filename request, with a particular boot file customized for that client. If the bootrequest filename is a null string, then the server returns a filename field indicating the 'default' file to be loaded for that client.

5. In the case of clients who do not know their IP addresses, the server must also have a database relating hardware address to IP address. This client IP address is then placed into a field in the bootreply.

6. Certain network topologies (such as Stanford's) may be such that a given physical cable does not have a TFTP server directly attached to it (e.g. all the gateways and hosts on a certain cable may be diskless). With the cooperation of neighboring gateways, BOOTP can allow clients to boot off of servers several hops away, through these gateways. See the section 'Booting Through Gateways' below. This part of the protocol requires no special action on the part of the client. Implementation is optional and requires a small amount of additional code in gateways and servers.

3. Packet Format

All numbers shown are decimal, unless indicated otherwise. The BOOTP packet is enclosed in a standard IP [8] UDP [7] datagram. For simplicity it is assumed that the BOOTP packet is never fragmented. Any numeric fields shown are packed in 'standard network byte order', i.e. high order bits are sent first.

In the IP header of a bootrequest, the client fills in its own IP source address if known, otherwise zero. When the server address is unknown, the IP destination address will be the 'broadcast address' 255.255.255.255. This address means 'broadcast on the local cable, (I don't know my net number)' [4].

The UDP header contains source and destination port numbers. The BOOTP protocol uses two reserved port numbers, 'BOOTP client' (68) and 'BOOTP server' (67). The client sends requests using 'BOOTP server' as the destination port; this is usually a broadcast. The server sends replies using 'BOOTP client' as the destination port; depending on the kernel or driver facilities in the server, this may

TOC

or may not be a broadcast (this is explained further in the section titled 'Chicken/Egg issues' below). The reason TWO reserved ports are used, is to avoid 'waking up' and scheduling the BOOTP server daemons, when a bootreply must be broadcast to a client. Since the server and other hosts won't be listening on the 'BOOTP client' port, any such incoming broadcasts will be filtered out at the kernel level. We could not simply allow the client to pick a 'random' port number for the UDP source port field; since the server reply may be broadcast, a randomly chosen port number could confuse other hosts that happened to be listening on that port.

The UDP length field is set to the length of the UDP plus BOOTP portions of the packet. The UDP checksum field can be set to zero by the client (or server) if desired, to avoid this extra overhead in a PROM implementation. In the 'Packet Processing' section below the phrase '[UDP checksum.]' is used whenever the checksum might be verified/computed.

```
   FIELD   BYTES   DESCRIPTION
   -----   -----   -----------

      op      1       packet op code / message type.
                      1 = BOOTREQUEST, 2 = BOOTREPLY

      htype   1       hardware address type,
                      see ARP section in "Assigned Numbers" RFC.
                      '1' = 10mb ethernet

      hlen    1       hardware address length
                      (eg '6' for 10mb ethernet).

      hops    1       client sets to zero,
                      optionally used by gateways
                      in cross-gateway booting.

      xid     4       transaction ID, a random number,
                      used to match this boot request with the
                      responses it generates.

      secs    2       filled in by client, seconds elapsed since
                      client started trying to boot.
```



```
      --      2       unused

      ciaddr  4       client IP address;
                      filled in by client in bootrequest if known.
```

```
          yiaddr  4       'your' (client) IP address;
                          filled by server if client doesn't
                          know its own address (ciaddr was 0).

          siaddr  4       server IP address;
                          returned in bootreply by server.

          giaddr  4       gateway IP address,
                          used in optional cross-gateway booting.

          chaddr  16      client hardware address,
                          filled in by client.

          sname   64      optional server host name,
                          null terminated string.

          file    128     boot file name, null terminated string;
                          'generic' name or null in bootrequest,
                          fully qualified directory-path
                          name in bootreply.

          vend    64      optional vendor-specific area,
                          e.g. could be hardware type/serial on request,
                          or 'capability' / remote file system handle
                          on reply.  This info may be set aside for use
                          by a third phase bootstrap or kernel.

4. Chicken / Egg Issues

   How can the server send an IP datagram to the client, if the client
   doesnt know its own IP address (yet)?  Whenever a bootreply is being
   sent, the transmitting machine performs the following operations:

      1. If the client knows its own IP address ('ciaddr' field is
      nonzero), then the IP can be sent 'as normal', since the client
      will respond to ARPs [5].

      2. If the client does not yet know its IP address (ciaddr zero),
      then the client cannot respond to ARPs sent by the transmitter of
      the bootreply.  There are two options:

         a. If the transmitter has the necessary kernel or driver hooks
```



```
         to 'manually' construct an ARP address cache entry, then it can
         fill in an entry using the 'chaddr' and 'yiaddr' fields.  Of
```

TOC

course, this entry should have a timeout on it, just like any other entry made by the normal ARP code itself. The transmitter of the bootreply can then simply send the bootreply to the client's IP address. UNIX (4.2 BSD) has this capability.

b. If the transmitter lacks these kernel hooks, it can simply send the bootreply to the IP broadcast address on the appropriate interface. This is only one additional broadcast over the previous case.

5. Client Use of ARP

The client PROM must contain a simple implementation of ARP, e.g. the address cache could be just one entry in size. This will allow a second-phase-only boot (TFTP) to be performed when the client knows the IP addresses and bootfile name.

Any time the client is expecting to receive a TFTP or BOOTP reply, it should be prepared to answer an ARP request for its own IP to hardware address mapping (if known).

Since the bootreply will contain (in the hardware encapsulation) the hardware source address of the server/gateway, the client MAY be able to avoid sending an ARP request for the server/gateway IP address to be used in the following TFTP phase. However this should be treated only as a special case, since it is desirable to still allow a second-phase-only boot as described above.

6. Comparison to RARP

An earlier protocol, Reverse Address Resolution Protocol (RARP) [1] was proposed to allow a client to determine its IP address, given that it knew its hardware address. However RARP had the disadvantage that it was a hardware link level protocol (not IP/UDP based). This means that RARP could only be implemented on hosts containing special kernel or driver modifications to access these 'raw' packets. Since there are many network kernels existent now, with each source maintained by different organizations, a boot protocol that does not require kernel modifications is a decided advantage.

BOOTP provides this hardware to IP address lookup function, in addition to the other useful features described in the sections above.

7. Packet Processing

7.1. Client Transmission

Before setting up the packet for the first time, it is a good idea to clear the entire packet buffer to all zeros; this will place all fields in their default state. The client then creates a packet with the following fields.

The IP destination address is set to 255.255.255.255. (the broadcast address) or to the server's IP address (if known). The IP source address and 'ciaddr' are set to the client's IP address if known, else 0. The UDP header is set with the proper length; source port = 'BOOTP client' port destination port = 'BOOTP server' port.

'op' is set to '1', BOOTREQUEST. 'htype' is set to the hardware address type as assigned in the ARP section of the "Assigned Numbers" RFC. 'hlen' is set to the length of the hardware address, e.g. '6' for 10mb ethernet.

'xid' is set to a 'random' transaction id. 'secs' is set to the number of seconds that have elapsed since the client has started booting. This will let the servers know how long a client has been trying. As the number gets larger, certain servers may feel more 'sympathetic' towards a client they don't normally service. If a client lacks a suitable clock, it could construct a rough estimate using a loop timer. Or it could choose to simply send this field as always a fixed value, say 100 seconds.

If the client knows its IP address, 'ciaddr' (and the IP source address) are set to this value. 'chaddr' is filled in with the client's hardware address.

If the client wishes to restrict booting to a particular server name, it may place a null-terminated string in 'sname'. The name used should be any of the allowable names or nicknames of the desired host.

The client has several options for filling the 'file' name field. If left null, the meaning is 'I want to boot the default file for my machine'. A null file name can also mean 'I am only interested in finding out client/server/gateway IP addresses, I dont care about file names'.

The field can also be a 'generic' name such as 'unix' or

'gateway'; this means 'boot the named program configured for my machine'. Finally the field can be a fully directory qualified path name.

The 'vend' field can be filled in by the client with vendor-specific strings or structures. For example the machine hardware type or serial number may be placed here. However the operation of the BOOTP server should not DEPEND on this information existing.

If the 'vend' field is used, it is recommended that a 4 byte 'magic number' be the first item within 'vend'. This lets a server determine what kind of information it is seeing in this field. Numbers can be assigned by the usual 'magic number' process --you pick one and it's magic. A different magic number could be used for bootreply's than bootrequest's to allow the client to take special action with the reply information.

[UDP checksum.]

7.2. Client Retransmission Strategy

If no reply is received for a certain length of time, the client should retransmit the request. The time interval must be chosen carefully so as not to flood the network. Consider the case of a cable containing 100 machines that are just coming up after a power failure. Simply retransmitting the request every four seconds will inundate the net.

As a possible strategy, you might consider backing off exponentially, similar to the way ethernet backs off on a collision. So for example if the first packet is at time 0:00, the second would be at :04, then :08, then :16, then :32, then :64. You should also randomize each time; this would be done similar to the ethernet specification by starting with a mask and 'and'ing that with with a random number to get the first backoff. On each succeeding backoff, the mask is increased in length by one bit. This doubles the average delay on each backoff.

After the 'average' backoff reaches about 60 seconds, it should be increased no further, but still randomized.

Before each retransmission, the client should update the 'secs' field. [UDP checksum.]

7.3. Server Receives BOOTREQUEST

[UDP checksum.] If the UDP destination port does not match the 'BOOTP server' port, discard the packet.

If the server name field (sname) is null (no particular server specified), or sname is specified and matches our name or nickname, then continue with packet processing.

If the sname field is specified, but does not match 'us', then there are several options:

1. You may choose to simply discard this packet.

2. If a name lookup on sname shows it to be on this same cable, discard the packet.

3. If sname is on a different net, you may choose to forward the packet to that address. If so, check the 'giaddr' (gateway address) field. If 'giaddr' is zero, fill it in with my address or the address of a gateway that can be used to get to that net. Then forward the packet.

If the client IP address (ciaddr) is zero, then the client does not know its own IP address. Attempt to lookup the client hardware address (chaddr, hlen, htype) in our database. If no match is found, discard the packet. Otherwise we now have an IP address for this client; fill it into the 'yiaddr' (your IP address) field.

We now check the boot file name field (file). The field will be null if the client is not interested in filenames, or wants the default bootfile. If the field is non-null, it is used as a lookup key in a database, along with the client's IP address. If there is a default file or generic file (possibly indexed by the client address) or a fully-specified path name that matches, then replace the 'file' field with the fully-specified path name of the selected boot file. If the field is non-null and no match was found, then the client is asking for a file we dont have; discard the packet, perhaps some other BOOTP server will have it.

The 'vend' vendor-specific data field should now be checked and if a recognized type of data is provided, client-specific actions should be taken, and a response placed in the 'vend' data field of the reply packet. For example, a workstation client could provide

an authentication key and receive from the server a capability for remote file access, or a set of configuration options, which can be passed to the operating system that will shortly be booted in.

Place my (server) IP address in the 'siaddr' field. Set the 'op' field to BOOTREPLY. The UDP destination port is set to 'BOOTP client'. If the client address 'ciaddr' is nonzero, send the packet there; else if the gateway address 'giaddr' is nonzero, set the UDP destination port to 'BOOTP server' and send the packet to 'giaddr'; else the client is on one of our cables but it doesnt know its own IP address yet --use a method described in the 'Egg' section above to send it to the client. If 'Egg' is used and we have multiple interfaces on this host, use the 'yiaddr' (your IP address) field to figure out which net (cable/interface) to send the packet to. [UDP checksum.]

7.4. Server/Gateway Receives BOOTREPLY

[UDP checksum.] If 'yiaddr' (your [the client's] IP address) refers to one of our cables, use one of the 'Egg' methods above to forward it to the client. Be sure to send it to the 'BOOTP client' UDP destination port.

7.5. Client Reception

Don't forget to process ARP requests for my own IP address (if I know it). [UDP checksum.] The client should discard incoming packets that: are not IP/UDPs addressed to the boot port; are not BOOTREPLYs; do not match my IP address (if I know it) or my hardware address; do not match my transaction id. Otherwise we have received a successful reply. 'yiaddr' will contain my IP address, if I didnt know it before. 'file' is the name of the file name to TFTP 'read request'. The server address is in 'siaddr'. If 'giaddr' (gateway address) is nonzero, then the packets should be forwarded there first, in order to get to the server.

8. Booting Through Gateways

This part of the protocol is optional and requires some additional code in cooperating gateways and servers, but it allows cross-gateway booting. This is mainly useful when gateways are diskless machines. Gateways containing disks (e.g. a UNIX machine acting as a gateway), might as well run their own BOOTP/TFTP servers.

Gateways listening to broadcast BOOTREQUESTs may decide to forward or rebroadcast these requests 'when appropriate'. For example, the

gateway could have, as part of his configuration tables, a list of other networks or hosts to receive a copy of any broadcast BOOTREQUESTs. Even though a 'hops' field exists, it is a poor idea to simply globally rebroadcast the requests, since broadcast loops will almost certainly occur.

The forwarding could begin immediately, or wait until the 'secs' (seconds client has been trying) field passes a certain threshold.

If a gateway does decide to forward the request, it should look at the 'giaddr' (gateway IP address) field. If zero, it should plug its own IP address (on the receiving cable) into this field. It may also use the 'hops' field to optionally control how far the packet is reforwarded. Hops should be incremented on each forwarding. For example, if hops passes '3', the packet should probably be discarded. [UDP checksum.]

Here we have recommended placing this special forwarding function in the gateways. But that does not have to be the case. As long as some 'BOOTP forwarding agent' exists on the net with the booting client, the agent can do the forwarding when appropriate. Thus this service may or may not be co-located with the gateway.

In the case of a forwarding agent not located in the gateway, the agent could save himself some work by plugging the broadcast address of the interface receiving the bootrequest into the 'giaddr' field. Thus the reply would get forwarded using normal gateways, not involving the forwarding agent. Of course the disadvantage here is that you lose the ability to use the 'Egg' non-broadcast method of sending the reply, causing extra overhead for every host on the client cable.

9. Sample BOOTP Server Database

As a suggestion, we show a sample text file database that the BOOTP server program might use. The database has two sections, delimited by a line containing an percent in column 1. The first section contains a 'default directory' and mappings from generic names to directory/pathnames. The first generic name in this section is the 'default file' you get when the bootrequest contains a null 'file' string.

The second section maps hardware addresstype/address into an ipaddress. Optionally you can also overide the default generic name by supplying a ipaddress specific genericname. A 'suffix' item is also an option; if supplied, any generic names specified by the

client will be accessed by first appending 'suffix' to the 'pathname'

appropriate to that generic name. If that file is not found, then the plain 'pathname' will be tried. This 'suffix' option allows a whole set of custom generics to be setup without a lot of effort. Below is shown the general format; fields are delimited by one or more spaces or tabs; trailing empty fields may be omitted; blank lines and lines beginning with '#' are ignored.

```
# comment line

homedirectory
genericname1    pathname1
genericname2    pathname2
...

% end of generic names, start of address mappings

hostname1 hardwaretype hardwareaddr1 ipaddr1 genericname suffix
hostname2 hardwaretype hardwareaddr2 ipaddr2 genericname suffix
...
```

Here is a specific example. Note the 'hardwaretype' number is the same as that shown in the ARP section of the 'Assigned Numbers' RFC. The 'hardwaretype' and 'ipaddr' numbers are in decimal; 'hardwareaddr' is in hex.

```
# last updated by smith

/usr/boot
vmunix          vmunix
tip             ethertip
watch           /usr/diag/etherwatch
gate            gate.

% end of generic names, start of address mappings

hamilton        1 02.60.8c.06.34.98     36.19.0.5
burr            1 02.60.8c.34.11.78     36.44.0.12
101-gateway     1 02.60.8c.23.ab.35     36.44.0.32      gate 101
mjh-gateway     1 02.60.8c.12.32.bc     36.42.0.64      gate mjh
welch-tipa      1 02.60.8c.22.65.32     36.47.0.14      tip
welch-tipb      1 02.60.8c.12.15.c8     36.46.0.12      tip
```

In the example above, if 'mjh-gateway' does a default boot, it will get the file '/usr/boot/gate.mjh'.

10. Acknowledgements

Ross Finlayson (et. al.) produced two earlier RFC's discussing TFTP bootstraping [2] using RARP [1].

We would also like to acknowledge the previous work and comments of Noel Chiappa, Bob Lyon, Jeff Mogul, Mark Lewis, and David Plummer.

REFERENCES

1. Ross Finlayson, Timothy Mann, Jeffrey Mogul, Marvin Theimer. A Reverse Address Resolution Protocol. RFC 903, NIC, June, 1984.

2. Ross Finlayson. Bootstrap Loading using TFTP. RFC 906, NIC, June, 1984.

3. Mark Lottor. Simple File Transfer Protocol. RFC 913, NIC, September, 1984.

4. Jeffrey Mogul. Broadcasting Internet Packets. RFC 919, NIC, October, 1984.

5. David Plummer. An Ethernet Address Resolution Protocol. RFC 826, NIC, September, 1982.

6. Jon Postel. File Transfer Protocol. RFC 765, NIC, June, 1980.

7. Jon Postel. User Datagram Protocol. RFC 768, NIC, August, 1980.

8. Jon Postel. Internet Protocol. RFC 791, NIC, September, 1981.

9. K. R. Sollins, Noel Chiappa. The TFTP Protocol. RFC 783, NIC, June, 1981.

8.6 RFC 1533

Obsoletes: 1497, 1395, 1084, 1048
Category: Standards Track

R. Droms
Bucknell University
October 1993

TOC

DHCP Options and BOOTP Vendor Extensions

Status of this Memo

This RFC specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

The Dynamic Host Configuration Protocol (DHCP) [1] provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the "options" field of the DHCP message. The data items themselves are also called "options."

This document specifies the current set of DHCP options. This document will be periodically updated as new options are defined. Each superseding document will include the entire current list of valid options.

All of the vendor information extensions defined in RFC 1497 [2] may be used as DHCP options. The definitions given in RFC 1497 are included in this document, which supersedes RFC 1497. All of the DHCP options defined in this document, except for those specific to DHCP as defined in section 9, may be used as BOOTP vendor information extensions.

Table of Contents

1. Introduction 2
2. BOOTP Extension/DHCP Option Field Format 2
3. RFC 1497 Vendor Extensions 3
4. IP Layer Parameters per Host 10
5. IP Layer Parameters per Interface 13
6. Link Layer Parameters per Interface 16
7. TCP Parameters 17
8. Application and Service Parameters 18

9. DHCP Extensions 23

10. Extensions 27
11. Acknowledgements 28
12. References 28
13. Security Considerations 19
14. Authors' Addresses 30

1. Introduction

This document specifies options for use with both the Dynamic Host Configuration Protocol and the Bootstrap Protocol.

The full description of DHCP packet formats may be found in the DHCP specification document [1], and the full description of BOOTP packet formats may be found in the BOOTP specification document [3]. This document defines the format of information in the last field of DHCP packets ('options') and of BOOTP packets ('vend'). The remainder of this section defines a generalized use of this area for giving information useful to a wide class of machines, operating systems and configurations. Sites with a single DHCP or BOOTP server that is shared among heterogeneous clients may choose to define other, site-specific formats for the use of the 'options' field.

Section 2 of this memo describes the formats of DHCP options and BOOTP vendor extensions. Section 3 describes options defined in previous documents for use with BOOTP (all may also be used with DHCP). Sections 4-8 define new options intended for use with both DHCP and BOOTP. Section 9 defines options used only in DHCP.

References further describing most of the options defined in sections 2-6 can be found in section 12. The use of the options defined in section 9 is described in the DHCP specification [1].

Information on registering new options is contained in section 10.

2. BOOTP Extension/DHCP Option Field Format

DHCP options have the same format as the BOOTP "vendor extensions" defined in RFC 1497 [2]. Options may be fixed length or variable length. All options begin with a tag octet, which uniquely identifies the option. Fixed-length options without data consist of only a tag octet. Only options 0 and 255 are fixed length. All other options are variable-length with a length octet following the tag octet. The value of the length octet does not include the two octets specifying the tag and length. The length octet is followed by "length" octets of data. In the case of some variable-length options the length field is a constant but must still be specified.

Any options defined subsequent to this document should contain a length octet even if the length is fixed or zero.

All multi-octet quantities are in network byte-order.

When used with BOOTP, the first four octets of the vendor information field have been assigned to the "magic cookie" (as suggested in RFC 951). This field identifies the mode in which the succeeding data is to be interpreted. The value of the magic cookie is the 4 octet dotted decimal 99.130.83.99 (or hexadecimal number 63.82.53.63) in network byte order.

All of the "vendor extensions" defined in RFC 1497 are also DHCP options.

Option codes 128 to 254 (decimal) are reserved for site-specific options.

Except for the options in section 9, all options may be used with either DHCP or BOOTP.

Many of these options have their default values specified in other documents. In particular, RFC 1122 [4] specifies default values for most IP and TCP configuration parameters.

3. RFC 1497 Vendor Extensions

This section lists the vendor extensions as defined in RFC 1497. They are defined here for completeness.

3.1. Pad Option

The pad option can be used to cause subsequent fields to align on word boundaries.

The code for the pad option is 0, and its length is 1 octet.

```
 Code
+-----+
|  0  |
+-----+
```

TOC

3.2. End Option

The end option marks the end of valid information in the vendor field. Subsequent octets should be filled with pad options.

The code for the end option is 255, and its length is 1 octet.

```
 Code
+-----+
| 255 |
+-----+
```

3.3. Subnet Mask

The subnet mask option specifies the client's subnet mask as per RFC 950 [5].

If both the subnet mask and the router option are specified in a DHCP reply, the subnet mask option MUST be first.

The code for the subnet mask option is 1, and its length is 4 octets.

```
 Code   Len        Subnet Mask
+-----+-----+-----+-----+-----+-----+
|  1  |  4  |  m1 |  m2 |  m3 |  m4 |
+-----+-----+-----+-----+-----+-----+
```

3.4. Time Offset

The time offset field specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC). The offset is expressed as a signed 32-bit integer.

The code for the time offset option is 2, and its length is 4 octets.

```
 Code   Len        Time Offset
+-----+-----+-----+-----+-----+-----+
|  2  |  4  |  n1 |  n2 |  n3 |  n4 |
+-----+-----+-----+-----+-----+-----+
```

TOC

3.5. Router Option

The router option specifies a list of IP addresses for routers on the client's subnet. Routers SHOULD be listed in order of preference.

The code for the router option is 3. The minimum length for the router option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  3  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.6. Time Server Option

The time server option specifies a list of RFC 868 [6] time servers available to the client. Servers SHOULD be listed in order of preference.

The code for the time server option is 4. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  4  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.7. Name Server Option

The name server option specifies a list of IEN 116 [7] name servers available to the client. Servers SHOULD be listed in order of preference.

The code for the name server option is 5. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  5  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.8. Domain Name Server Option

The domain name server option specifies a list of Domain Name System (STD 13, RFC 1035 [8]) name servers available to the client. Servers SHOULD be listed in order of preference.

The code for the domain name server option is 6. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  6  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.9. Log Server Option

The log server option specifies a list of MIT-LCS UDP log servers available to the client. Servers SHOULD be listed in order of preference.

The code for the log server option is 7. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  7  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.10. Cookie Server Option

The cookie server option specifies a list of RFC 865 [9] cookie servers available to the client. Servers SHOULD be listed in order of preference.

The code for the log server option is 8. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  8  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

TOC

3.11. LPR Server Option

The LPR server option specifies a list of RFC 1179 [10] line printer servers available to the client. Servers SHOULD be listed in order of preference.

The code for the LPR server option is 9. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  9  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.12. Impress Server Option

The Impress server option specifies a list of Imagen Impress servers available to the client. Servers SHOULD be listed in order of preference.

The code for the Impress server option is 10. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  10 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.13. Resource Location Server Option

This option specifies a list of RFC 887 [11] Resource Location servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 11. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  11 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

TOC

3.14. Host Name Option

This option specifies the name of the client. The name may or may not be qualified with the local domain name (see section 3.17 for the preferred way to retrieve the domain name). See RFC 1035 for character set restrictions.

The code for this option is 12, and its minimum length is 1.

```
 Code   Len                 Host Name
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  12 |  n  |  h1 |  h2 |  h3 |  h4 |  h5 |  h6 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.15. Boot File Size Option

This option specifies the length in 512-octet blocks of the default boot image for the client. The file length is specified as an unsigned 16-bit integer.

The code for this option is 13, and its length is 2.

```
 Code   Len   File Size
+-----+-----+-----+-----+
|  13 |  2  |  l1 |  l2 |
+-----+-----+-----+-----+
```

3.16. Merit Dump File

This option specifies the path-name of a file to which the client's core image should be dumped in the event the client crashes. The path is formatted as a character string consisting of characters from the NVT ASCII character set.

The code for this option is 14. Its minimum length is 1.

```
 Code   Len      Dump File Pathname
+-----+-----+-----+-----+-----+-----+---
|  14 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```

3.17. Domain Name

This option specifies the domain name that client should use when resolving hostnames via the Domain Name System.

The code for this option is 15. Its minimum length is 1.

```
 Code   Len        Domain Name
+-----+-----+-----+-----+-----+-----+--
|  15 |  n  |  d1 |  d2 |  d3 |  d4 |  ...
+-----+-----+-----+-----+-----+-----+--
```

3.18. Swap Server

This specifies the IP address of the client's swap server.

The code for this option is 16 and its length is 4.

```
 Code   Len    Swap Server Address
+-----+-----+-----+-----+-----+-----+
|  16 |  n  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

3.19. Root Path

This option specifies the path-name that contains the client's root disk. The path is formatted as a character string consisting of characters from the NVT ASCII character set.

The code for this option is 17. Its minimum length is 1.

```
 Code   Len      Root Disk Pathname
+-----+-----+-----+-----+-----+-----+---
|  17 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```

3.20. Extensions Path

A string to specify a file, retrievable via TFTP, which contains information which can be interpreted in the same way as the 64-octet vendor-extension field within the BOOTP response, with the following exceptions:

- the length of the file is unconstrained;
- all references to Tag 18 (i.e., instances of the BOOTP Extensions Path field) within the file are ignored.

The code for this option is 18. Its minimum length is 1.

```
 Code   Len      Extensions Pathname
+-----+-----+-----+-----+-----+-----+---
|  18 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```

4. IP Layer Parameters per Host

This section details the options that affect the operation of the IP layer on a per-host basis.

4.1. IP Forwarding Enable/Disable Option

This option specifies whether the client should configure its IP layer for packet forwarding. A value of 0 means disable IP forwarding, and a value of 1 means enable IP forwarding.

The code for this option is 19, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  19 |  1  | 0/1 |
+-----+-----+-----+
```

4.2. Non-Local Source Routing Enable/Disable Option

This option specifies whether the client should configure its IP layer to allow forwarding of datagrams with non-local source routes (see Section 3.3.5 of [4] for a discussion of this topic). A value of 0 means disallow forwarding of such datagrams, and a value of 1 means allow forwarding.

The code for this option is 20, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  20 |  1  | 0/1 |
+-----+-----+-----+
```

4.3. Policy Filter Option

This option specifies policy filters for non-local source routing. The filters consist of a list of IP addresses and masks which specify destination/mask pairs with which to filter incoming source routes.

Any source routed datagram whose next-hop address does not match one of the filters should be discarded by the client.

See [4] for further information.

The code for this option is 21. The minimum length of this option is 8, and the length MUST be a multiple of 8.

```
 Code   Len         Address 1                  Mask 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  21 |  n  |  a1 |  a2 |  a3 |  a4 |  m1 |  m2 |  m3 |  m4 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
        Address 2                  Mask 2
+-----+-----+-----+-----+-----+-----+-----+-----+---
|  a1 |  a2 |  a3 |  a4 |  m1 |  m2 |  m3 |  m4 | ...
+-----+-----+-----+-----+-----+-----+-----+-----+---
```

4.4. Maximum Datagram Reassembly Size

This option specifies the maximum size datagram that the client should be prepared to reassemble. The size is specified as a 16-bit unsigned integer. The minimum value legal value is 576.

The code for this option is 22, and its length is 2.

```
 Code   Len      Size
+-----+-----+-----+-----+
|  22 |  2  |  s1 |  s2 |
+-----+-----+-----+-----+
```

4.5. Default IP Time-to-live

This option specifies the default time-to-live that the client should use on outgoing datagrams. The TTL is specified as an octet with a value between 1 and 255.

The code for this option is 23, and its length is 1.

```
 Code   Len   TTL
+-----+-----+-----+
|  23 |  1  | ttl |
+-----+-----+-----+
```

4.6. Path MTU Aging Timeout Option

This option specifies the timeout (in seconds) to use when aging Path MTU values discovered by the mechanism defined in RFC 1191 [12]. The timeout is specified as a 32-bit unsigned integer.

The code for this option is 24, and its length is 4.

```
 Code   Len           Timeout
+-----+-----+-----+-----+-----+-----+
|  24 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

4.7. Path MTU Plateau Table Option

This option specifies a table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. The minimum MTU value cannot be smaller than 68.

The code for this option is 25. Its minimum length is 2, and the length MUST be a multiple of 2.

```
 Code   Len     Size 1      Size 2
+-----+-----+-----+-----+-----+-----+---
|  25 |  n  |  s1 |  s2 |  s1 |  s2 | ...
+-----+-----+-----+-----+-----+-----+---
```

5. IP Layer Parameters per Interface

This section details the options that affect the operation of the IP layer on a per-interface basis. It is expected that a client can issue multiple requests, one per interface, in order to configure interfaces with their specific parameters.

5.1. Interface MTU Option

This option specifies the MTU to use on this interface. The MTU is specified as a 16-bit unsigned integer. The minimum legal value for the MTU is 68.

The code for this option is 26, and its length is 2.

```
 Code   Len      MTU
+-----+-----+-----+-----+
|  26 |  2  |  m1 |  m2 |
+-----+-----+-----+-----+
```

5.2. All Subnets are Local Option

This option specifies whether or not the client may assume that all subnets of the IP network to which the client is connected use the same MTU as the subnet of that network to which the client is directly connected. A value of 1 indicates that all subnets share the same MTU. A value of 0 means that the client should assume that some subnets of the directly connected network may have smaller MTUs.

The code for this option is 27, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  27 |  1  | 0/1 |
+-----+-----+-----+
```

5.3. Broadcast Address Option

This option specifies the broadcast address in use on the client's subnet. Legal values for broadcast addresses are specified in section 3.2.1.3 of [4].

The code for this option is 28, and its length is 4.

```
 Code   Len     Broadcast Address
+-----+-----+-----+-----+-----+-----+
|  28 |  4  |  b1 |  b2 |  b3 |  b4 |
+-----+-----+-----+-----+-----+-----+
```

5.4. Perform Mask Discovery Option

This option specifies whether or not the client should perform subnet mask discovery using ICMP. A value of 0 indicates that the client should not perform mask discovery. A value of 1 means that the client should perform mask discovery.

The code for this option is 29, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  29 |  1  | 0/1 |
+-----+-----+-----+
```

5.5. Mask Supplier Option

This option specifies whether or not the client should respond to subnet mask requests using ICMP. A value of 0 indicates that the client should not respond. A value of 1 means that the client should respond.

The code for this option is 30, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  30 |  1  | 0/1 |
+-----+-----+-----+
```

5.6. Perform Router Discovery Option

This option specifies whether or not the client should solicit routers using the Router Discovery mechanism defined in RFC 1256 [13]. A value of 0 indicates that the client should not perform router discovery. A value of 1 means that the client should perform router discovery.

The code for this option is 31, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  31 |  1  | 0/1 |
+-----+-----+-----+
```

5.7. Router Solicitation Address Option

This option specifies the address to which the client should transmit router solicitation requests.

The code for this option is 32, and its length is 4.

```
 Code   Len            Address
+-----+-----+-----+-----+-----+-----+
|  32 |  4  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

5.8. Static Route Option

This option specifies a list of static routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are listed in descending order of priority.

The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination.

The default route (0.0.0.0) is an illegal destination for a static route. See section 3.5 for information about the router option.

The code for this option is 33. The minimum length of this option is 8, and the length MUST be a multiple of 8.

```
 Code   Len         Destination 1           Router 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  33 |  n  |  d1 |  d2 |  d3 |  d4 |  r1 |  r2 |  r3 |  r4 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
        Destination 2           Router 2
+-----+-----+-----+-----+-----+-----+-----+-----+---
|  d1 |  d2 |  d3 |  d4 |  r1 |  r2 |  r3 |  r4 | ...
+-----+-----+-----+-----+-----+-----+-----+-----+---
```

6. Link Layer Parameters per Interface

This section lists the options that affect the operation of the data link layer on a per-interface basis.

6.1. Trailer Encapsulation Option

This option specifies whether or not the client should negotiate the use of trailers (RFC 893 [14]) when using the ARP protocol. A value of 0 indicates that the client should not attempt to use trailers. A value of 1 means that the client should attempt to use trailers.

The code for this option is 34, and its length is 1.

```
 Code   Len  Value
```

```
+-----+-----+-----+
|  34 |  1  | 0/1 |
+-----+-----+-----+
```


6.2. ARP Cache Timeout Option

This option specifies the timeout in seconds for ARP cache entries. The time is specified as a 32-bit unsigned integer.

The code for this option is 35, and its length is 4.

```
 Code   Len           Time
+-----+-----+-----+-----+-----+-----+
|  35 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

6.3. Ethernet Encapsulation Option

This option specifies whether or not the client should use Ethernet Version 2 (RFC 894 [15]) or IEEE 802.3 (RFC 1042 [16]) encapsulation if the interface is an Ethernet. A value of 0 indicates that the client should use RFC 894 encapsulation. A value of 1 means that the client should use RFC 1042 encapsulation.

The code for this option is 36, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  36 |  1  | 0/1 |
+-----+-----+-----+
```

7. TCP Parameters

This section lists the options that affect the operation of the TCP layer on a per-interface basis.

7.1. TCP Default TTL Option

This option specifies the default TTL that the client should use when sending TCP segments. The value is represented as an 8-bit unsigned integer. The minimum value is 1.

The code for this option is 37, and its length is 1.

```
 Code   Len   TTL
+-----+-----+-----+
|  37 |  1  |  n  |
+-----+-----+-----+
```


7.2. TCP Keepalive Interval Option

This option specifies the interval (in seconds) that the client TCP should wait before sending a keepalive message on a TCP connection. The time is specified as a 32-bit unsigned integer. A value of zero indicates that the client should not generate keepalive messages on connections unless specifically requested by an application.

The code for this option is 38, and its length is 4.

```
 Code   Len           Time
+-----+-----+-----+-----+-----+-----+
|  38 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

7.3. TCP Keepalive Garbage Option

This option specifies the whether or not the client should send TCP keepalive messages with a octet of garbage for compatibility with older implementations. A value of 0 indicates that a garbage octet should not be sent. A value of 1 indicates that a garbage octet should be sent.

The code for this option is 39, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  39 |  1  | 0/1 |
+-----+-----+-----+
```

8. Application and Service Parameters

This section details some miscellaneous options used to configure miscellaneous applications and services.

8.1. Network Information Service Domain Option

This option specifies the name of the client's NIS [17] domain. The domain is formatted as a character string consisting of characters

from the NVT ASCII character set.

The code for this option is 40. Its minimum length is 1.

```
 Code   Len      NIS Domain Name
+-----+-----+-----+-----+-----+-----+---
|  40 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```


8.2. Network Information Servers Option

This option specifies a list of IP addresses indicating NIS servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 41. Its minimum length is 4, and the length MUST be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  41 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.3. Network Time Protocol Servers Option

This option specifies a list of IP addresses indicating NTP [18] servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 42. Its minimum length is 4, and the length MUST be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  42 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.4. Vendor Specific Information

This option is used by clients and servers to exchange vendor-specific information. The information is an opaque object of n octets, presumably interpreted by vendor-specific code on the clients and servers. The definition of this information is vendor specific. The vendor is indicated in the class-identifier option. Servers not equipped to interpret the vendor-specific information sent by a client MUST ignore it (although it may be reported). Clients which do not receive desired vendor-specific information SHOULD make an

TOC

attempt to operate without it, although they may do so (and announce they are doing so) in a degraded mode.

If a vendor potentially encodes more than one item of information in this option, then the vendor SHOULD encode the option using "Encapsulated vendor-specific options" as described below:

The Encapsulated vendor-specific options field SHOULD be encoded as a sequence of code/length/value fields of identical syntax to the DHCP options field with the following exceptions:

1) There SHOULD NOT be a "magic cookie" field in the encapsulated vendor-specific extensions field.

2) Codes other than 0 or 255 MAY be redefined by the vendor within the encapsulated vendor-specific extensions field, but SHOULD conform to the tag-length-value syntax defined in section 2.

3) Code 255 (END), if present, signifies the end of the encapsulated vendor extensions, not the end of the vendor extensions field. If no code 255 is present, then the end of the enclosing vendor-specific information field is taken as the end of the encapsulated vendor-specific extensions field.

The code for this option is 43 and its minimum length is 1.

```
Code   Len   Vendor-specific information
+-----+-----+-----+-----+---
|  43 |  n  |  i1 |  i2 | ...
+-----+-----+-----+-----+---
```

When encapsulated vendor-specific extensions are used, the information bytes 1-n have the following format:

```
 Code   Len   Data item        Code   Len   Data item       Code
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  T1 |  n  |  d1 |  d2 | ... |  T2 |  n  |  D1 |  D2 | ... | ... |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

8.5. NetBIOS over TCP/IP Name Server Option

The NetBIOS name server (NBNS) option specifies a list of RFC 1001/1002 [19] [20] NBNS name servers listed in order of preference.

The code for this option is 44. The minimum length of the option is 4 octets, and the length must always be a multiple of 4.

```
 Code   Len           Address 1              Address 2
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
|  44 |  n  |  a1 |  a2 |  a3 |  a4 |  b1 |  b2 |  b3 |  b4 | ...
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
```


8.6. NetBIOS over TCP/IP Datagram Distribution Server Option

The NetBIOS datagram distribution server (NBDD) option specifies a list of RFC 1001/1002 NBDD servers listed in order of preference. The code for this option is 45. The minimum length of the option is 4 octets, and the length must always be a multiple of 4.

```
 Code   Len          Address 1              Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
|  45 |  n  |  a1 |  a2 |  a3 |  a4 |  b1 |  b2 |  b3 |  b4 | ...
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
```

8.7. NetBIOS over TCP/IP Node Type Option

The NetBIOS node type option allows NetBIOS over TCP/IP clients which are configurable to be configured as described in RFC 1001/1002. The value is specified as a single octet which identifies the client type as follows:

Value	Node Type
0x1	B-node
0x2	P-node
0x4	M-node
0x8	H-node

In the above chart, the notation '0x' indicates a number in base-16 (hexadecimal).

The code for this option is 46. The length of this option is always 1.

```
 Code   Len  Node Type
+-----+-----+-----------+
|  46 |  1  | see above |
+-----+-----+-----------+
```

8.8. NetBIOS over TCP/IP Scope Option

The NetBIOS scope option specifies the NetBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002. See [19], [20], and [8] for character-set restrictions.

The code for this option is 47. The minimum length of this option is 1.

```
 Code   Len       NetBIOS Scope
+-----+-----+-----+-----+-----+-----+----
|  47 |  n  |  s1 |  s2 |  s3 |  s4 | ...
+-----+-----+-----+-----+-----+-----+----
```

8.9. X Window System Font Server Option

This option specifies a list of X Window System [21] Font servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 48. The minimum length of this option is 4 octets, and the length MUST be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+---
|  48 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |   ...
+-----+-----+-----+-----+-----+-----+-----+-----+---
```

8.10. X Window System Display Manager Option

This option specifies a list of IP addresses of systems that are running the X Window System Display Manager and are available to the client.

Addresses SHOULD be listed in order of preference.

TOC

The code for the this option is 49. The minimum length of this option is 4, and the length MUST be a multiple of 4.

```
 Code   Len         Address 1               Address 2

+-----+-----+-----+-----+-----+-----+-----+-----+---
|  49 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |   ...
+-----+-----+-----+-----+-----+-----+-----+-----+---
```


9. DHCP Extensions

This section details the options that are specific to DHCP.

9.1. Requested IP Address

This option is used in a client request (DHCPDISCOVER) to allow the client to request that a particular IP address be assigned.

The code for this option is 50, and its length is 4.

```
 Code   Len          Address
+-----+-----+-----+-----+-----+-----+
|  50 |  4  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

9.2. IP Address Lease Time

This option is used in a client request (DHCPDISCOVER or DHCPREQUEST) to allow the client to request a lease time for the IP address. In a server reply (DHCPOFFER), a DHCP server uses this option to specify the lease time it is willing to offer.

The time is in units of seconds, and is specified as a 32-bit unsigned integer.

The code for this option is 51, and its length is 4.

```
 Code   Len         Lease Time
+-----+-----+-----+-----+-----+-----+
|  51 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

9.3. Option Overload

This option is used to indicate that the DHCP "sname" or "file" fields are being overloaded by using them to carry DHCP options. A DHCP server inserts this option if the returned parameters will exceed the usual space allotted for options.

If this option is present, the client interprets the specified additional fields after it concludes interpretation of the standard option fields.

The code for this option is 52, and its length is 1. Legal values for this option are:

Value	Meaning
1	the "file" field is used to hold options
2	the "sname" field is used to hold options
3	both fields are used to hold options

Code	Len	Value
52	1	1/2/3

9.4. DHCP Message Type

This option is used to convey the type of the DHCP message. The code for this option is 53, and its length is 1. Legal values for this option are:

Value	Message Type
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNAK
7	DHCPRELEASE

Code	Len	Type
53	1	1-7

9.5. Server Identifier

This option is used in DHCPOFFER and DHCPREQUEST messages, and may optionally be included in the DHCPACK and DHCPNAK messages. DHCP servers include this option in the DHCPOFFER in order to allow the client to distinguish between lease offers. DHCP clients indicate which of several lease offers is being accepted by including this option in a DHCPREQUEST message.

The identifier is the IP address of the selected server.

The code for this option is 54, and its length is 4.

```
 Code   Len            Address
+-----+-----+-----+-----+-----+-----+
|  54 |  4  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

9.6. Parameter Request List

This option is used by a DHCP client to request values for specified configuration parameters. The list of requested parameters is specified as n octets, where each octet is a valid DHCP option code as defined in this document.

The client MAY list the options in order of preference. The DHCP server is not required to return the options in the requested order, but MUST try to insert the requested options in the order requested by the client.

The code for this option is 55. Its minimum length is 1.

```
 Code   Len   Option Codes
+-----+-----+-----+-----+---
|  55 |  n  |  c1 |  c2 | ...
+-----+-----+-----+-----+---
```

9.7. Message

This option is used by a DHCP server to provide an error message to a DHCP client in a DHCPNAK message in the event of a failure. A client may use this option in a DHCPDECLINE message to indicate the why the client declined the offered parameters. The message consists of n

octets of NVT ASCII text, which the client may display on an available output device.

The code for this option is 56 and its minimum length is 1.

```
 Code   Len     Text
+-----+-----+-----+-----+---
|  56 |  n  |  c1 |  c2 | ...
+-----+-----+-----+-----+---
```


9.8. Maximum DHCP Message Size

This option specifies the maximum length DHCP message that it is willing to accept. The length is specified as an unsigned 16-bit integer. A client may use the maximum DHCP message size option in DHCPDISCOVER or DHCPREQUEST messages, but should not use the option in DHCPDECLINE messages.

The code for this option is 57, and its length is 2. The minimum legal value is 576 octets.

```
 Code   Len     Length
+-----+-----+-----+-----+
|  57 |  2  |  l1 |  l2 |
+-----+-----+-----+-----+
```

9.9. Renewal (T1) Time Value

This option specifies the time interval from address assignment until the client transitions to the RENEWING state.

The value is in units of seconds, and is specified as a 32-bit unsigned integer.

The code for this option is 58, and its length is 4.

```
 Code   Len         T1 Interval
+-----+-----+-----+-----+-----+-----+
|  58 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

TOC

9.10. Rebinding (T2) Time Value

This option specifies the time interval from address assignment until the client transitions to the REBINDING state.

The value is in units of seconds, and is specified as a 32-bit unsigned integer.

The code for this option is 59, and its length is 4.

```
 Code   Len         T2 Interval
+-----+-----+-----+-----+-----+-----+
|  59 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```


9.11. Class-identifier

This option is used by DHCP clients to optionally identify the type and configuration of a DHCP client. The information is a string of n octets, interpreted by servers. Vendors and sites may choose to define specific class identifiers to convey particular configuration or other identification information about a client. For example, the identifier may encode the client's hardware configuration. Servers not equipped to interpret the class-specific information sent by a client MUST ignore it (although it may be reported).

The code for this option is 60, and its minimum length is 1.

```
Code   Len   Class-Identifier
+-----+-----+-----+-----+---
|  60 |  n  |  i1 |  i2 | ...
+-----+-----+-----+-----+---
```

9.12. Client-identifier

This option is used by DHCP clients to specify their unique identifier. DHCP servers use this value to index their database of address bindings. This value is expected to be unique for all clients in an administrative domain.

Identifiers consist of a type-value pair, similar to the

It is expected that this field will typically contain a hardware type and hardware address, but this is not required. Current legal values for hardware types are defined in [22].

TOC

The code for this option is 61, and its minimum length is 2.

```
Code   Len   Type  Client-Identifier
+-----+-----+-----+-----+-----+---
|  61 |  n  |  t1 |  i1 |  i2 | ...
+-----+-----+-----+-----+-----+---
```

10. Extensions

Additional generic data fields may be registered by contacting:

Internet Assigned Numbers Authority (IANA)
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, California 90292-6695

or by email as: iana@isi.edu

Implementation specific use of undefined generic types (those in the range 61-127) may conflict with other implementations, and registration is required.

11. Acknowledgements

The authors would like to thank Philip Almquist for his feedback on this document. The comments of the DHCP Working Group are also gratefully acknowledged. In particular, Mike Carney and Jon Dreyer from SunSelect suggested the current format of the Vendor-specific Information option.

RFC 1497 is based on earlier work by Philip Prindeville, with help from Drew Perkins, Bill Croft, and Steve Deering.

12. References

[1] Droms, R., "Dynamic Host Configuration Protocol", RFC 1531, Bucknell University, October 1993.

[2] Reynolds, J., "BOOTP Vendor Information Extensions", RFC 1497, USC/Information Sciences Institute, August 1993.

[3] Croft, W., and J. Gilmore, "Bootstrap Protocol", RFC 951, Stanford University and Sun Microsystems, September 1985.

[4] Braden, R., Editor, "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, USC/Information Sciences Institute, October 1989.

TOC

[5] Mogul, J., and J. Postel, "Internet Standard Subnetting Procedure", STD 5, RFC 950, USC/Information Sciences Institute, August 1985.

[6] Postel, J., and K. Harrenstien, "Time Protocol", STD 26, RFC 868, USC/Information Sciences Institute, SRI, May 1983.

[7] Postel, J., "Name Server", IEN 116, USC/Information Sciences Institute, August 1979.

[8] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, USC/Information Sciences Institute, November 1987.

[9] Postel, J., "Quote of the Day Protocol", STD 23, RFC 865, USC/Information Sciences Institute, May 1983.

[10] McLaughlin, L., "Line Printer Daemon Protocol", RFC 1179, The Wollongong Group, August 1990.

[11] Accetta, M., "Resource Location Protocol", RFC 887, CMU, December 1983.

[12] Mogul, J. and S. Deering, "Path MTU Discovery", RFC 1191, DECWRL, Stanford University, November 1990.

[13] Deering, S., "ICMP Router Discovery Messages", RFC 1256, Xerox PARC, September 1991.

[14] Leffler, S. and M. Karels, "Trailer Encapsulations", RFC 893, U. C. Berkeley, April 1984.

[15] Hornig, C., "Standard for the Transmission of IP Datagrams over Ethernet Networks", RFC 894, Symbolics, April 1984.

[16] Postel, J. and J. Reynolds, "Standard for the Transmission of IP Datagrams Over IEEE 802 Networks", RFC 1042, USC/Information Sciences Institute, February 1988.

[17] Sun Microsystems, "System and Network Administration", March 1990.

[18] Mills, D., "Internet Time Synchronization: The Network Time Protocol", RFC 1305, UDEL, March 1992.

[19] NetBIOS Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP transport: Concepts and Methods", STD 19, RFC 1001, March 1987.

[20] NetBIOS Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP transport: Detailed Specifications", STD 19, RFC 1002, March 1987.

[21] Scheifler, R., "FYI On the X Window System", FYI 6, RFC 1198, MIT Laboratory for Computer Science, January 1991.

[22] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1340, USC/Information Sciences Institute, July 1992.

13. Security Considerations

Security issues are not discussed in this memo.

14. Authors' Addresses

Steve Alexander
Lachman Technology, Inc.
1901 North Naper Boulevard
Naperville, IL 60563-8895

Phone: (708) 505-9555 x256
EMail: stevea@lachman.com

Ralph Droms
Computer Science Department
323 Dana Engineering
Bucknell University
Lewisburg, PA 17837

Phone: (717) 524-1145
EMail: droms@bucknell.edu

8.7 RFC 1350

Network Working Group
Request For Comments: 1350
STD: 33
Obsoletes: RFC 783

K. Sollins
MIT

July 1992

THE TFTP PROTOCOL (REVISION 2)

Status of this Memo

This RFC specifies an IAB standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "IAB Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Summary

TFTP is a very simple protocol used to transfer files. It is from this that its name comes, Trivial File Transfer Protocol or TFTP. Each nonterminal packet is acknowledged separately. This document describes the protocol and its types of packets. The document also explains the reasons behind some of the design decisions.

Acknowlegements

The protocol was originally designed by Noel Chiappa, and was redesigned by him, Bob Baldwin and Dave Clark, with comments from Steve Szymanski. The current revision of the document includes modifications stemming from discussions with and suggestions from Larry Allen, Noel Chiappa, Dave Clark, Geoff Cooper, Mike Greenwald, Liza Martin, David Reed, Craig Milo Rogers (of USC-ISI), Kathy Yellick, and the author. The acknowledgement and retransmission scheme was inspired by TCP, and the error mechanism was suggested by PARC's EFTP abort message.

The May, 1992 revision to fix the "Sorcerer's Apprentice" protocol bug [4] and other minor document problems was done by Noel Chiappa.

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Office of Naval Research under contract number N00014-75-C-0661.

1. Purpose

TFTP is a simple protocol to transfer files, and therefore was named the Trivial File Transfer Protocol or TFTP. It has been implemented on top of the Internet User Datagram protocol (UDP or Datagram) [2]

so it may be used to move files between machines on different networks implementing UDP. (This should not exclude the possibility of implementing TFTP on top of other datagram protocols.) It is designed to be small and easy to implement. Therefore, it lacks most of the features of a regular FTP. The only thing it can do is read and write files (or mail) from/to a remote server. It cannot list directories, and currently has no provisions for user authentication. In common with other Internet protocols, it passes 8 bit bytes of data.

Three modes of transfer are currently supported: netascii (This is ascii as defined in "USA Standard Code for Information Interchange" [1] with the modifications specified in "Telnet Protocol Specification" [3].) Note that it is 8 bit ascii. The term "netascii" will be used throughout this document to mean this particular version of ascii.); octet (This replaces the "binary" mode of previous versions of this document.) raw 8 bit bytes; mail, netascii characters sent to a user rather than a file. (The mail mode is obsolete and should not be implemented or used.) Additional modes can be defined by pairs of cooperating hosts.

Reference [4] (section 4.2) should be consulted for further valuable directives and suggestions on TFTP.

2. Overview of the Protocol

Any transfer begins with a request to read or write a file, which also serves to request a connection. If the server grants the request, the connection is opened and the file is sent in fixed length blocks of 512 bytes. Each data packet contains one block of data, and must be acknowledged by an acknowledgment packet before the next packet can be sent. A data packet of less than 512 bytes signals termination of a transfer. If a packet gets lost in the network, the intended recipient will timeout and may retransmit his last packet (which may be data or an acknowledgment), thus causing the sender of the lost packet to retransmit that lost packet. The sender has to keep just one packet on hand for retransmission, since the lock step acknowledgment guarantees that all older packets have been received. Notice that both machines involved in a transfer are considered senders and receivers. One sends data and receives acknowledgments, the other sends acknowledgments and receives data.

Most errors cause termination of the connection. An error is signalled by sending an error packet. This packet is not acknowledged, and not retransmitted (i.e., a TFTP server or user may terminate after sending an error message), so the other end of the connection may not get it. Therefore timeouts are used to detect such a termination when the error packet has been lost. Errors are

caused by three types of events: not being able to satisfy the request (e.g., file not found, access violation, or no such user), receiving a packet which cannot be explained by a delay or duplication in the network (e.g., an incorrectly formed packet), and losing access to a necessary resource (e.g., disk full or access denied during a transfer).

TFTP recognizes only one error condition that does not cause termination, the source port of a received packet being incorrect. In this case, an error packet is sent to the originating host.

This protocol is very restrictive, in order to simplify implementation. For example, the fixed length blocks make allocation straight forward, and the lock step acknowledgement provides flow control and eliminates the need to reorder incoming data packets.

3. Relation to other Protocols

As mentioned TFTP is designed to be implemented on top of the Datagram protocol (UDP). Since Datagram is implemented on the Internet protocol, packets will have an Internet header, a Datagram header, and a TFTP header. Additionally, the packets may have a header (LNI, ARPA header, etc.) to allow them through the local transport medium. As shown in Figure 3-1, the order of the contents of a packet will be: local medium header, if used, Internet header, Datagram header, TFTP header, followed by the remainder of the TFTP packet. (This may or may not be data depending on the type of packet as specified in the TFTP header.) TFTP does not specify any of the values in the Internet header. On the other hand, the source and destination port fields of the Datagram header (its format is given in the appendix) are used by TFTP and the length field reflects the size of the TFTP packet. The transfer identifiers (TID's) used by TFTP are passed to the Datagram layer to be used as ports; therefore they must be between 0 and 65,535. The initialization of TID's is discussed in the section on initial connection protocol.

The TFTP header consists of a 2 byte opcode field which indicates the packet's type (e.g., DATA, ERROR, etc.) These opcodes and the formats of the various types of packets are discussed further in the section on TFTP packets.

```
 ---------------------------------------------------
| Local Medium  |  Internet  |  Datagram  |  TFTP  |
 ---------------------------------------------------
```

Figure 3-1: Order of Headers

4. Initial Connection Protocol

A transfer is established by sending a request (WRQ to write onto a foreign file system, or RRQ to read from it), and receiving a positive reply, an acknowledgment packet for write, or the first data packet for read. In general an acknowledgment packet will contain the block number of the data packet being acknowledged. Each data packet has associated with it a block number; block numbers are consecutive and begin with one. Since the positive response to a write request is an acknowledgment packet, in this special case the block number will be zero. (Normally, since an acknowledgment packet is acknowledging a data packet, the acknowledgment packet will contain the block number of the data packet being acknowledged.) If the reply is an error packet, then the request has been denied.

In order to create a connection, each end of the connection chooses a TID for itself, to be used for the duration of that connection. The TID's chosen for a connection should be randomly chosen, so that the probability that the same number is chosen twice in immediate succession is very low. Every packet has associated with it the two TID's of the ends of the connection, the source TID and the destination TID. These TID's are handed to the supporting UDP (or other datagram protocol) as the source and destination ports. A requesting host chooses its source TID as described above, and sends its initial request to the known TID 69 decimal (105 octal) on the serving host. The response to the request, under normal operation, uses a TID chosen by the server as its source TID and the TID chosen for the previous message by the requestor as its destination TID. The two chosen TID's are then used for the remainder of the transfer.

As an example, the following shows the steps used to establish a connection to write a file. Note that WRQ, ACK, and DATA are the names of the write request, acknowledgment, and data types of packets respectively. The appendix contains a similar example for reading a file.

1. Host A sends a "WRQ" to host B with source= A's TID, destination= 69.

2. Host B sends a "ACK" (with block number= 0) to host A with source= B's TID, destination= A's TID.

At this point the connection has been established and the first data packet can be sent by Host A with a sequence number of 1. In the next step, and in all succeeding steps, the hosts should make sure that the source TID matches the value that was agreed on in steps 1 and 2. If a source TID does not match, the packet should be discarded as erroneously sent from somewhere else. An error packet should be sent to the source of the incorrect packet, while not disturbing the transfer. This can be done only if the TFTP in fact receives a packet with an incorrect TID. If the supporting protocols do not allow it, this particular error condition will not arise.

The following example demonstrates a correct operation of the protocol in which the above situation can occur. Host A sends a request to host B. Somewhere in the network, the request packet is duplicated, and as a result two acknowledgments are returned to host A, with different TID's chosen on host B in response to the two requests. When the first response arrives, host A continues the connection. When the second response to the request arrives, it should be rejected, but there is no reason to terminate the first connection. Therefore, if different TID's are chosen for the two connections on host B and host A checks the source TID's of the messages it receives, the first connection can be maintained while the second is rejected by returning an error packet.

5. TFTP Packets

TFTP supports five types of packets, all of which have been mentioned above:

opcode	operation
1	Read request (RRQ)
2	Write request (WRQ)
3	Data (DATA)
4	Acknowledgment (ACK)
5	Error (ERROR)

The TFTP header of a packet contains the opcode associated with that packet.

```
          2 bytes     string    1 byte     string   1 byte
          -------------------------------------------------
         | Opcode |  Filename  |   0  |    Mode    |   0  |
          -------------------------------------------------
```

Figure 5-1: RRQ/WRQ packet

RRQ and WRQ packets (opcodes 1 and 2 respectively) have the format shown in Figure 5-1. The file name is a sequence of bytes in netascii terminated by a zero byte. The mode field contains the string "netascii", "octet", or "mail" (or any combination of upper and lower case, such as "NETASCII", NetAscii", etc.) in netascii indicating the three modes defined in the protocol. A host which receives netascii mode data must translate the data to its own format. Octet mode is used to transfer a file that is in the 8-bit format of the machine from which the file is being transferred. It is assumed that each type of machine has a single 8-bit format that is more common, and that that format is chosen. For example, on a DEC-20, a 36 bit machine, this is four 8-bit bytes to a word with four bits of breakage. If a host receives a octet file and then returns it, the returned file must be identical to the original. Mail mode uses the name of a mail recipient in place of a file and must begin with a WRQ. Otherwise it is identical to netascii mode. The mail recipient string should be of the form "username" or "username@hostname". If the second form is used, it allows the option of mail forwarding by a relay computer.

The discussion above assumes that both the sender and recipient are operating in the same mode, but there is no reason that this has to be the case. For example, one might build a storage server. There is no reason that such a machine needs to translate netascii into its own form of text. Rather, the sender might send files in netascii, but the storage server might simply store them without translation in 8-bit format. Another such situation is a problem that currently exists on DEC-20 systems. Neither netascii nor octet accesses all the bits in a word. One might create a special mode for such a machine which read all the bits in a word, but in which the receiver stored the information in 8-bit format. When such a file is retrieved from the storage site, it must be restored to its original form to be useful, so the reverse mode must also be implemented. The user site will have to remember some information to achieve this. In both of these examples, the request packets would specify octet mode to the foreign host, but the local host would be in some other mode. No such machine or application specific modes have been specified in TFTP, but one would be compatible with this specification.

TOC

It is also possible to define other modes for cooperating pairs of

hosts, although this must be done with care. There is no requirement that any other hosts implement these. There is no central authority that will define these modes or assign them names.

```
 2 bytes     2 bytes      n bytes
 ----------------------------------
| Opcode |   Block #  |   Data     |
 ----------------------------------
```

Figure 5-2: DATA packet

Data is actually transferred in DATA packets depicted in Figure 5-2. DATA packets (opcode = 3) have a block number and data field. The block numbers on data packets begin with one and increase by one for each new block of data. This restriction allows the program to use a single number to discriminate between new packets and duplicates. The data field is from zero to 512 bytes long. If it is 512 bytes long, the block is not the last block of data; if it is from zero to 511 bytes long, it signals the end of the transfer. (See the section on Normal Termination for details.)

All packets other than duplicate ACK's and those used for termination are acknowledged unless a timeout occurs [4]. Sending a DATA packet is an acknowledgment for the first ACK packet of the previous DATA packet. The WRQ and DATA packets are acknowledged by ACK or ERROR packets, while RRQ

```
 2 bytes     2 bytes
 ---------------------
| Opcode |   Block #  |
 ---------------------
```

Figure 5-3: ACK packet

and ACK packets are acknowledged by DATA or ERROR packets. Figure 5-3 depicts an ACK packet; the opcode is 4. The block number in an ACK echoes the block number of the DATA packet being acknowledged. A WRQ is acknowledged with an ACK packet having a block number of zero.

```
          2 bytes     2 bytes      string    1 byte
          ----------------------------------------
         | Opcode |  ErrorCode |   ErrMsg   |   0  |
          ----------------------------------------
```

Figure 5-4: ERROR packet

An ERROR packet (opcode 5) takes the form depicted in Figure 5-4. An ERROR packet can be the acknowledgment of any other type of packet. The error code is an integer indicating the nature of the error. A table of values and meanings is given in the appendix. (Note that several error codes have been added to this version of this document.) The error message is intended for human consumption, and should be in netascii. Like all other strings, it is terminated with a zero byte.

6. Normal Termination

The end of a transfer is marked by a DATA packet that contains between 0 and 511 bytes of data (i.e., Datagram length < 516). This packet is acknowledged by an ACK packet like all other DATA packets. The host acknowledging the final DATA packet may terminate its side of the connection on sending the final ACK. On the other hand, dallying is encouraged. This means that the host sending the final ACK will wait for a while before terminating in order to retransmit the final ACK if it has been lost. The acknowledger will know that the ACK has been lost if it receives the final DATA packet again. The host sending the last DATA must retransmit it until the packet is acknowledged or the sending host times out. If the response is an ACK, the transmission was completed successfully. If the sender of the data times out and is not prepared to retransmit any more, the transfer may still have been completed successfully, after which the acknowledger or network may have experienced a problem. It is also possible in this case that the transfer was unsuccessful. In any case, the connection has been closed.

7. Premature Termination

If a request can not be granted, or some error occurs during the transfer, then an ERROR packet (opcode 5) is sent. This is only a courtesy since it will not be retransmitted or acknowledged, so it may never be received. Timeouts must also be used to detect errors.

I. Appendix

Order of Headers

```
                                                  2 bytes
    -------------------------------------------------------------
   |  Local Medium  |  Internet  |  Datagram  |  TFTP Opcode  |
    -------------------------------------------------------------
```

TFTP Formats

```
   Type   Op #     Format without header

          2 bytes    string   1 byte     string   1 byte
          ------------------------------------------------
   RRQ/  | 01/02 |  Filename  |   0  |    Mode    |   0  |
   WRQ    ------------------------------------------------
          2 bytes    2 bytes       n bytes
          ---------------------------------
   DATA  | 03    |   Block #  |    Data    |
          ---------------------------------
          2 bytes    2 bytes
          --------------------
   ACK   | 04    |   Block #  |
          --------------------
          2 bytes  2 bytes        string    1 byte
          ----------------------------------------
   ERROR | 05    |  ErrorCode |   ErrMsg   |   0  |
          ----------------------------------------
```

Initial Connection Protocol for reading a file

1. Host A sends a "RRQ" to host B with source= A's TID, destination= 69.

2. Host B sends a "DATA" (with block number= 1) to host A with source= B's TID, destination= A's TID.

Error Codes

Value	Meaning
0	Not defined, see error message (if any).
1	File not found.
2	Access violation.
3	Disk full or allocation exceeded.
4	Illegal TFTP operation.
5	Unknown transfer ID.
6	File already exists.
7	No such user.

Internet User Datagram Header [2]

(This has been included only for convenience. TFTP need not be implemented on top of the Internet User Datagram Protocol.)

Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Values of Fields

Source Port	Picked by originator of packet.
Dest. Port	Picked by destination machine (69 for RRQ or WRQ).
Length	Number of bytes in UDP packet, including UDP header.
Checksum	Reference 2 describes rules for computing checksum. (The implementor of this should be sure that the correct algorithm is used here.) Field contains zero if unused.

Note: TFTP passes transfer identifiers (TID's) to the Internet User Datagram protocol to be used as the source and destination ports.

References

[1] USA Standard Code for Information Interchange, USASI X3.4-1968.

[2] Postel, J., "User Datagram Protocol," RFC 768, USC/Information Sciences Institute, 28 August 1980.

[3] Postel, J., "Telnet Protocol Specification," RFC 764, USC/Information Sciences Institute, June, 1980.

[4] Braden, R., Editor, "Requirements for Internet Hosts -- Application and Support", RFC 1123, USC/Information Sciences Institute, October 1989.

Security Considerations

Since TFTP includes no login or access control mechanisms, care must be taken in the rights granted to a TFTP server process so as not to violate the security of the server hosts file system. TFTP is often installed with controls such that only files that have public read access are available via TFTP and writing files via TFTP is disallowed.

Author's Address

Karen R. Sollins
Massachusetts Institute of Technology
Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02139-1986

Phone: (617) 253-6006

EMail: SOLLINS@LCS.MIT.EDU

```
Sollins                                                        [Page 11]
```

8.8 Install Instructions

```
                    I N S T A L L A T I O N

Overview of the installation process
====================================

Due to it's nature this package requires at least two computer systems. One
acts as a server, and at least one other will be setup as a diskless client.
Therefore this installation guide is divided into four sections:

        1.) Compilation and installation of utility programs on the server
        2.) Create a netbootable image of the target operating system
        3.) Setup of the server
        4.) Setup of the client including building the bootrom

The server has to support TCP/IP and certain protocols based on this network
standard. Most likely this will be a Unix-type server. Though it's probably
possible to also use servers running OS/2 or Windows-NT, for example, all
server related programs in this package can currently only be compiled on
a Unix-type host. This requirement is independed of the operating system
which is later booted on the diskless client. Therefore even if you want
to boot MS-DOS on your client(s) you need at least one Unix-type computer
for program compilation and generation of all boot files. Lateron when all
necessary files are built you can use any server you want.

This package contains two main parts:

        1.) The bootrom source and binaries. This part gets installed on
            the diskless client. All binaries except for utility programs
            are already precompiled. There are no further user changeable
            or adjustable options in the sources so you don't have to have
            access to the 16 bit development tools in order to use the boot-
            rom. You can just use the binaries provided.
            In order for the bootrom to access the network card in your
            diskless client you need a driver. Currently the bootrom only
            supports so called packet drivers, which are normally used on
            MS-DOS systems to interface a network stack with the hardware.
            With this package only the packet driver binaries are required,
            so you don't need to recompile anything here as well. You can
            find precompiled packet drivers for many popular network cards
```

TOC

on any SimTel FTP mirror (it's called Crynwr packet driver collection), and for those of you without internet access some of those packet driver binaries are included with this package. Another good source for a packet driver for your network card might be it's manufacturer. At least the well known manufacturers (3Com and SMC for example) provide packet drivers for their complete product line. Those manufacturer provided packet drivers are usually faster and easier to install than those from the Crynwr collection, and can sometimes determine the hardware configuration at runtime, which the Crynwr drivers can't. However, there is a limitation in that you can only use packet drivers which are COM-type executables. EXE-type programs are not supported yet.

2.) A set of programs to generate netbootable images on the server. These programs are called mknbi-<os>, where <os> identifies the operating system which is lateron running on the diskless client. Currently only Linux and MS-DOS are supported.

There is another requirement which should not leave unnoted. Although you can build a bootrom with slightly limited functionality which is less than 16kB in size, the usual size for a bootrom will be between 16kB and 32kB. Therefore when you go shopping for a network card you should try to get one which is able to support 32kB EPROM's. This is standard on almost all cards from major manufacturers, but most cheap NE2000 are known to allow only a maximum of 16kB. Also note that some network cards from 3Com and SMC allow you to select ROM sizes of 32kB and more with their configuration programs, but can physically support only 16kB!

Compilation and installation of utility programs on the server

This package uses GNU's autoconf to configure the compilation process of the utility programs. You shouldn't have any problems to compile these programs on any Unix-type system.

1.) Cd into the netboot directory and run ./configure. It's a configuration script generated by autoconf and checks for header files and system specific details. The mknbi utility programs contain some Intel assembler modules which lateron run on the diskless client. If you want to assemble these modules you need as86 and ld86, which you can get for free for Unix systems. However, there are preassembled files available so you actually don't need these two programs. configure checks for their existence and creates the Makefiles accordingly.
For an explanation of the switches available to configure just run it with the --help option. Some additional switches are available:

```
            --disable-mknbi-linux
            --disable-mknbi-dos

        Choose these options if you don't want to create any of the
        corresponding mknbi utility programs. There is also another
        configure option:

            --enable-bootrom

        Use this option only if you want to recompile the bootrom
        itself. If you want to use the precompiled binaries, you don't
        need to specify this switch. See the file INSTALL.bootrom
        about how to recompile the bootrom.

    2.) Check that all generated Makefiles and the config.h are correct
        for your system.

    3.) Compile all programs with

            make clean
            make

        This will compile all programs without those which you disabled
        during the configuration stage. IMPORTANT NOTE: Some Makefiles
        use ifdef, which not every make program understands. If you
        get an error from make (usually in the form: "missing delimiter")
        then get and install GNU make on your system! Especially System V
        systems are known to have this deficiency.

    4.) If you want to permanently install the utility programs on
        your server you can run

            make install

        This will also install the corresponding man pages for later
        reference. However, it's perfectly ok to skip this step and
        run the mknbi program from their source directories. But please
        note that they are just called "mknbi" within their source
        directories. Therefore if you read further down to run mknbi-dos,
        you have to use "./mknbi-dos/mknbi" instead if you didn't install
        the programs using 'make install'.
```

Create a netbootable image of the target operating system

This step of the installation process depends on which operating you want to boot on your diskless clients. Everything described in this chapter does not depend on working on a Linux system. You can use any

TOC

UNIX type system to create the netbootable images.

Linux: With Linux you have far too many options to list them all in this text. Please refer to the mknbi-linux man page for all details. I will only describe the most common ways to setup a diskless Linux client here.
First you have to decide where the Linux client is going to mount it's root filesystem from. This can either be a directory on an NFS server or a ram disk. Setup your Linux kernel accordingly. To use a root filesystem on an NFS server you should include TCP/IP network support into the kernel together with support for NFS file-systems. You cannot load this NFS support using a module as it has to be available at bootup. Additionally you also have to select NFSROOT support during kernel configuration. However, you don't need BOOTP or RARP support. Accordingly if you want to use ramdisk support the filesystem type you are going to use on the ramdisk has to be permanently compiled into the kernel. Also initrd has to be included in that case.

1.) Configuring for NFS root filesystem.

Next copy your Linux kernel into the current directory and run mknbi-linux:

```
mknbi-linux -d rom -i rom -k zImage -o bootImage
```

This supposes that your kernel image is called zImage, and gives you a netbootable image named bootImage.

2.) Configuring for root filesystem on ramdisk

If you want to use a ramdisk as a root device you have to create a ramdisk image first. Probably the easiest way to setup such an image is to use a floppy, though you can also use the loopback device if you are working on a Linux host. First format the floppy and make a filesystem on it. Next copy all programs and files onto it which you want to have on the root filesystem of the diskless client lateron. You should then test your root floppy. To do this copy your kernel onto another floppy with dd and set it's root device to floppy using rdev:

```
dd if=zImage of=/dev/fd0
rdev /dev/fd0 /dev/fd0
```

Now boot your diskless client using this boot disk. After the kernel started up, it will ask you to insert the root floppy and to press enter. Your root floppy will be mounted.
If everything works as you intended, you can now create a netbootable image. Re-insert the root floppy into your server system (or whereever this netboot directory is located), and type:

```
dd if=/dev/fd0 of=ramImage
```

TOC

```
gzip -9 ramImage
mknbi-linux -d ram -i rom -r ramImage.gz -k zImage -o bootIm
```

Like above this will now give you a file bootImage with the netbootable Linux kernel image in it.

MS-DOS: To boot DOS on your diskless client you have to have MS-DOS Version 5.0 or higher. Windows-95 has an internal DOS called version 7.0, so it should be no problem to use it as well. Older MS-DOS versions will definitely not work. I haven't had a chance to test any other DOS like Novell-DOS or DR-DOS. Give them a try, and tell me.

First you have to create a directory which contains all the files the client will see on it's boot drive (either A: or C:). This can either be the root directory on a DOS floppy or any directory on the system on which you installed mknbi-dos. In the first case it has to be a floppy which contains a bootable DOS system, i.e. which has been created with

```
format a: /s
```

on a DOS system. If the directory resides on a UNIX system, you have to copy the two system files msdos.sys and io.sys, which are part of MS-DOS, into it by yourself. To do this I recommend using mread of the MTools, which are freely available for almost every UNIX system.

After you created the directory or floppy which lateron becomes the clients boot drive, you should copy all other necessary files into it. This will probably include programs to setup a network environment on the client. When editing text files for the client please note that they usually have to be in MS-DOS format with lines ending in Carriage-Return/Linefeed instead of just Linefeed as it is common on UNIX systems. When you are finished setting up the clients boot directory, first get a copy of the floppy disk image, and then run mknbi-dos to create a netbootable image:

```
dd if=/dev/fd0 of=fdImage
mknbi-dos -r fdImage -o bootImage
```

This assumes that you inserted the boot floppy into the fd0 drive of your UNIX system, and will create a file named bootImage. If you used a UNIX directory, substitute fdImage with it's name. mknbi-dos will automatically detect wether it is a directory, an ordinary file or a block device.

By default mknbi-dos creates a netbootable image, which lateron mounts the ram disk as the A: drive on your client. If you want to mount the ram disk as C: instead, you should include the '-c' switch to the call of mknbi-dos.
The difference between mounting the ram disk as a floppy (A:) or hard disk (C:) is, that with the floppy option the ram disk can be removed lateron, maybe after a network redirector has been

TOC

```
        loaded, which makes the ram disk obsolete. This is not possible
        with a virtual hard disk drive. On the other hand side, when using
        the ram disk as C: you can specify a different ramdisk size with
        the '-s' option. Please refer to the man page for mknbi-dos for
        further information.
```

Setup of the server

Setup of the server depends on the kind of server you are using. Therefore all further explanations in this chapter can only serve as a general guide. You should consult your server's documentation as the final authority.

When the bootrom starts on the client it first tries to query a bootp server for information like IP numbers and the name of the boot image file. Such a bootp server program is usually called bootpd. Most sun servers use a program called bootparamd instead. Note that you _cannot_ use bootparamd as a substitute for bootpd as both programs use different protocols. Install a publicly available bootpd instead on your sun.
Next you should copy the bootImage file, which you have created in the previous step above, into a publicly accessible directory (called /boot for example). If you want to boot more than one diskless client you can use the same bootImage file for every client. However, if you configured for a ramdisk (with Linux or DOS) and the ramdisk image contains different files or information for every client, you will obviously also need a different bootImage file for each client.
Then you need to setup a boot description file for bootpd, which is usually called /etc/bootptab. Consult your server's documentation for further information. However, the entries in this file will usually look something like this for every diskless client:

```
client1:hd=/boot:vm=auto:ip=192.109.225.66:\
        :ht=ethernet:ha=004001417173:\
        :bf=bootImage-client1:rp=/boot/client1/root
```

'hd' specifies the home directory and 'bf' is the name of the bootImage file, which you created in the previous step. Therefore the full pathname for the bootImage file for the diskless system called "client1" will be

```
        /boot/bootImage-client1
```

with this sample entry. The 'ip' tag specifies the IP address of the client, 'ht' the type of the network the client is attached to, and 'ha' it's hardware address. The 'vm=auto' tag tells bootpd to use the same vendor encoding as the bootrom. If your diskless client is going to use it's root filesystem via NFS you should also specify the directory on the server which gets mounted lateron with the 'rp' tag. However, if your diskless client uses a ramdisk,

TOC

you can omit 'rp'. When you choose to use the standard bootrom with ANSI display driver (see below for further information) you could also setup a menu for letting the user select different boot image files. See the additional file INSTALL.menu about how to use this feature. But I recommend to first use the standard way of setting up the bootptab file as described above. You can always add a user menu lateron.
Of course you should also remember to get bootpd running on the server, either on bootup from /etc/rc or some similar mechanism, or from inetd. Again, see your server's documentation about how to do this.

The next step preformed by the bootrom after querying the bootp server is to load in the boot image file specified by the 'hd' and 'bf' tags in /etc/bootptab. To do this a protocol named tftp is used. Therefore you will next have to setup a daemon process for this protocol on your server. Such a daemon is usually called tftpd, and you should again remember to get tftpd running, usually via inetd. Since the TFTP protocol is very insecure access to the tftpd server is usually restricted, either within tftpd itself, or with a TCP/IP wrapper like tcpd. tcpd for example uses host access control tables which are stored in /etc/hosts.allow and /etc/hosts.deny. See tftpd(8), tcpd(8) and hosts_access(5) as well as your server's documentation for further information.

If you selected a ramdisk for the diskless client's root directory you are now finished with the server setup. But if your client is going to use NFS (either directly like with booting Linux, or by using programs included on the ram disk) you should now setup everything which is necessary for mounting an NFS directory on the server. This usually involves running several programs: portmap, mountd, nfsd and optionally ugidd. portmap usually doesn't require editing any configuration files. But for mountd and nfsd you need to specify the permissions which allow the client to access the required directories on the server. These permissions are usually set with a file called /etc/exports. Typically it looks like this for our sample client:

```
#
#  Export directories for client1 (diskless workstation)
#
/boot/client1/root              client1(rw,link_absolute)
/boot/client1/usr               client1(rw,link_absolute)
```

If you use 'map-daemon' to map UID and GID numbers on the server you should remember to also configure and run ugidd on the server. Please consult your server's documentation for further information regarding setup of NFS exports. You might also want to check out the portmap(8), nfsd(8), mountd(8) and ugidd(8) man pages. Also remember that access to any of these services might be restricted with tcpd on your server.

Another important step is to fill up the root directory for the diskless client. It has to contain all files necessary for the client to startup and mount further directories via NFS (like a /usr filesystem as specified in the /etc/exports example above). How to setup this root directory is far beyond the scope of this documentation. Just one hint: if your server is _not_ running Linux, you should be aware of major/minor number assignments in the /boot/client1/root/dev directory. For example, simply using mknod on an AIX server will eventually give

TOC

you wrong major/minor number when the directory is later exported to a Linux diskless client. With some configurations AIX will add a certain offset to all major numbers which makes them unusable for Linux. Refer to your server's manuals for further information. You might also find some useful hints in the file Documentation/nfsroot.txt in the Linux source tree, if your diskless client is booting Linux.

Setup of the client including building the bootrom

Until now you only had to work on the server (with the exception of maybe booting your diskless client from a diskette to check the correctness of the root filesystem). As the last step we can now go on and setup the diskless client itself.

The first step is to configure the network card in the diskless client. For this refer to the manual which came with the network card. Some cards require setting of jumpers. Others have setup programs which have to be run. After configuring the network interface write down all necessary hardware parameters like I/O addresses, memory addresses, interrupt line number or DMA channel numbers, as you might need this information lateron in the configuration process.

Next change into the netboot directory on your UNIX system (where this documentation file is in) and type

```
make bootrom
```

This will compile all necessary utility programs and then run the configuration program. It will first ask you which bootrom kernel you want to use. The minimal kernel is necessary for network cards which only allow up to 16 kB ROM size, and kernel86 can be used to boot on 16-bit systems (older than 386), for example for booting MS-DOS. Unless you have any special requirements you should choose the standard kernel. Then you have to specify the packet driver to use for your network card. You can either choose one of the supplied drivers, or provide your own. If you want to provide your own driver you have to give the full path name of the packet driver binary on your server, and also specify all necessary options to run it. Don't specify any options here which switch the packet driver into windows mode or which allow it to work for diskless systems. Those options are for Novell network bootroms only, and are not necessary for this bootrom.
If you use one of the drivers in the list shown, the configuration program will ask you about all necessary hardware information to run the packet driver which you selected. This usually includes the I/O address of the network card, it's interrupt number and a DMA channel number. Note that only that information is requested which is really necessary. You should have your network card information handy when

TOC

entering this information. Some packet drivers are able to determine hardware related information at runtime and therefore don't require any further information.

If you did not select the minimal kernel, the configuration program is next going to ask you wether you want to include some additional drivers. First it lets you select the ANSI display driver. This will allow you to draw nice menus on the screen with the standard bootrom kernel. You can then select the packet driver debugging program. It's an additional module to trace network problems and is usually not required. It shows you the first couple bytes of all packets (where the UDP/IP headers are encoded) going through the packet driver during boot time of the diskless client. Only select this debugging module if you run into problems during the initial network boot process of the bootrom _and_ you know how to decode the UDP/IP header information. The configuration program will also ask you about any additional modules you want to install into the bootrom. These modules have to be standard DOS COM-type programs, and can, for example, preset the network card to a special state before the packet driver starts, or setup a serial line to support booting over a PPP or SLIP connection (the Crynwr packet driver collection also contains a SLIP packet driver which is not provided in this package). However note that the total size of the resulting bootrom image can't be larger than 64kB.

After you answered all questions the configuration program is creating the bootrom according to your specifications. It first combines the bootrom kernel with all selected modules, then compresses the resulting file and adds the bootrom startup code. When the configuration program has finished you will find two new files in the current directory:

```
image.flo - this file can be written onto a floppy using dd
image.rom - image to be burned into an EPROM
```

You should now copy image.flo onto a floppy using

```
dd if=image.flo of=/dev/fd0
```

and then boot your diskless client using this floppy. If you have setup everything (including your network card) you will see the bootrom code starting, querying the bootp server and loading the boot image file. When everything works as required you can then go on and burn the file image.rom into an EPROM. Please consult the manual of your EPROM burner how to do this. It usually requires converting the image file into a special format (Intel or Motorola hex format for example). Insert the EPROM into the socket on your network card and turn on the diskless system. You should now see the bootrom coming up.
Another way of getting the bootrom code into your client is using the Flash-EPROM card (called FlashCard), for which you can find a schematic and PCB layout in this package. You can use image.rom directly to burn it into FlashCard - there is no hex conversion necessary. About how to use and program the FlashCard see the documentation in the FlashCard directory.

In case you want to create new bootroms without always having the sources

TOC

around, you can now install the binaries created during the configuration step with the command

```
        make bootrom_install
```

This will copy all necessary binaries for creating new bootroms into the directory $prefix/lib/netboot where $prefix is either /usr/local or the prefix you specified with running GNU configure. The typical path would be /usr/local/lib/netboot. It will also install the makerom script into $prefix/bin, so you just have to type makerom to create a new bootrom.

Appendix: Recompiling the bootrom
========

If you want to recompile the bootrom for some reason, checkout the file INSTALL.bootrom for further information. However, you don't need to re-compile the bootrom in order to just use it!

8.9 Troubleshoot Problems

T R O U B L E S H O O T I N G

If you run into any problem during installation or when using this package, please first read the following text and all other relevant documentation. Especially you should consult your server's documen-tation if you run into problems setting up your server. Also refer to your network card's user manual or the documentation for the operating systems of the diskless clients accordingly. However, if you still can't solve the problem on your own, you can send me an email to

gero@gkminix.han.de

Users able to speak German can send me the mail in german. Otherwise please write in english. I already received some emails in so poor english that I haven't been able to even understand the problem. I can't help you in that case. And please excuse me that I can't answer questions sent to me by standard mail or telephone calls. I just don't have the time for dealing with that.
If you decided to send me an email please describe your problem as exactly as possible. It usually helps to send me relevant portions of configuration files (I have to pay for my internet access by myself so please keep quotings as short as possible). Especially with problems with the bootrom it usually helps to _exactly_ write down the screen output, not only but including any error messages. Also state as exact as possible how you created the problem so that I can try to simulate

TOC

it on my own hardware.
Additionally please note that I can't help you with every problem with your server, as there are so many different systems on the market. The same is true for problems with network cards. I just don't have the financial capabilities to buy any card on the market for testing. Personally I'm using NE2000 and WD8013 cards, so I can probably help you with those.
If you find a problem which looks like a bug in the code I really appreciate a short notice from you. And if you have a fix for the bug I would even more appreciate your message.
Besides contacting me directly there also exists a mailing list related to network booting which you can subscribe to. Write a mail with the message 'subscribe netboot' in it's body to majordomo@baghira.han.de (the subject of the mail doesn't matter). The readers of the mailing list should also be able to help you with any problem you might have while setting up a diskless client. And besides that I'm also going to announce any new version of this netboot package to the mailing list.

Problem: My operating system OS/XY is not supported by netboot

I would gladly provide support for every operating system on the market, but I don't have the resources for doing this. However, if you want a particular operating system to be supported, you should get in contact with me. In any case you will have to provide me with a valid and licensed copy of that operating system. You are also invited to write your own boot loader, and send it to me for inclusion into netboot under the terms of the GNU GPL.

Problem: While trying to build a bootrom I get a compiler error

The installation scripts require to compile a couple of utility programs which are only required during building the bootrom. They should compile on any Unix-type system, so if you get an error please report it to me, even when you are able to fix it yourself, so that I can include a patch for future releases.

Problem: I get a an error from make saying something like "missing delimiter"

Some of the Makefiles use ifdef's, which older make programs don't understand. Even some more "modern" systems like SCO Open-Server 5 have this problem. In that case you will have to get and install GNU make on your system (which is the better choice anyway).

Problem: The bootrom doesn't startup at all

TOC

Either you have a floppy in your diskette drive or you have a hard disk installed with a partition marked as active, and the bootrom has been built so that it lets the BIOS look for active partitions first. Both conditions let the system boot from the bootable media instead of using the bootrom. Just remove the floppy or use fdisk to mark all partitions as unbootable (e.g. inactive). Alternatively you can also build the bootrom so that it does not allow the BIOS to look for bootable partitions. The program which actually creates the bootrom ('makerom', it gets called when you run 'make bootrom') will ask you about this right after selecting the bootrom kernel image.

Problem: The bootrom behaves strange during startup, and may even hangup the whole system

If you compiled the mknbi programs on a system with big endian byte order (like Motorola or PPC systems) this might indicate that the configuration program couldn't find the correct byte order. It might also be that there is a bug in the byte ordering code. Some systems like SPARCs also do not allow data accesses at misaligned addresses. 'configure' should usually find out about these conditions. In any case, if 'configure' is not able to properly detect what kind of system you are using, edit the file config.h by hand and try it again. Please report this condition, and also note which system you used for installation.

Problem: The packet driver is not able to start properly

First check what error message the packet driver prints. Usually this problem is a result of an incorrect setup of the network card, so check that it uses an I/O address, interrupt line and DMA channel (if applicable) of it's own, and that the packet driver uses the correct values. Another common problem with ethernet cards which use shared memory (like WD80?3 cards) is an overlapping of this shared memory with the rom area used by the bootrom. Select a different shared memory address in that case. If that's ok you should next check that you configured the packet driver correctly with the bootrom configuration program. Usually the packet driver prints out what it expects the hardware to look like so you can use this information to check up your setup.

Problem: The bootrom tells me that there is not enough memory but I have xx megabytes installed

This problem is a result of the fact that the BIOS starts the bootrom in the processor's real mode. The bootrom is therefore only able to access the lower 1 megabyte of memory, regardless

TOC

of how much you installed. And 384kB of this is reserved for ROM's and the video memory, so there is only 640kB left. Unfortunately some systems even reserve memory from these lower 640kB for internal BIOS data. This is called extended BIOS data area, and known to be used on most PS/2 systems. But also some other BIOSes use such an extended BIOS data area, which is usually selectable in the system's setup. Therefore you should try to deselect such a feature. If that's not possible you are out of luck - sorry.

Problem: The bootrom doesn't receive a bootp answer and just hangs printing dots

First you should check if bootpd runs on your server or is started properly from inetd. Then check that the server's /etc/bootptab is setup correctly. Especially the hardware address and the client's IP address and name have to be correct.
Most bootp servers have the ability to write debugging information into a log file. Use that feature to verify that your server really receives bootp requests from the client's bootrom and sends out a valid answer. Also check for error messages in the log file. Even if your bootpd doesn't write into a seperate log file it might use syslog on your system, so find the log file name from your syslogd configuration file and check for errors.
If you are able to use a network tracing program like tcpdump you can check if the bootrom sends out correct requests and that the server is answering correctly. In that case it is more likely to be a problem in the bootrom, so you should create a new bootrom image with the packet driver debugging module included. You should then see the bootrom's request packets going out, and the server's answers coming in. If there are no packets coming in although you verified that the server is sending out correct replies there might be a problem with your network card. Did you set it up correctly, is a cable connected (no kidding, those things really happen)?
If everything fails try to boot the diskless client with the intended operating system and try to access the network card using that operating system's tools.
If the server is not sending out answer packets, but the bootpd logfiles indicates correct answers, it might be a problem with the arp setup on your server. Normally arp shouldn't be a concern for you. However, some older versions of bootpd for Linux had problems here, which could be solved by setting the kernel arp table manually.

Problem: The bootrom did get a bootp answer but is not able to load the bootimage file

This is likely to be a problem with the tftpd setup on the server. Does tftpd run when you startup the bootrom code? If not check that inetd is configured correctly. Also there might be a TCP/IP

wrapper running on your server which might prohibit access to the tftp service (which is known to be very insecure and therefore a candidate for getting started by an internet security wrapper like tcpd). Check any access configuration files for tcpd. Furthermore tftpd has to be able to access the bootimage file. It usually runs as a user with very low priviliges because of security reasons and might not be allowed to read the bootimage file, so you should check and set the bootimage file's permissions correctly.

Problem: The boot image loader reports an error

Congratulations! You just discovered a bug in the boot loader. Please report it to me.

Problem: When I'm using the bootrom menu to load a Unix system off the local hard disk, it reports some weird error messages to me (especially, SCO Unix says that it's not able to open boot device). However, booting without the bootrom works without a problem.

Some operating systems, especially Unix like systems, read the partition table after booting and try to find their own boot partition. When using the bootrom, it's not necessary to mark the Unix partition as bootable, so the Unix startup loader fails. To solve this problem, mark the Unix partition active with some fdisk program. To avoid that it starts running instead of the bootrom, create the bootrom so that it does not allow the BIOS to search for boot partitions on the installed hard disks (the 'makerom' program, which gets run when you do a 'make bootrom', will ask you about this right after selecting a kernel image).

Problem: I'm loading Linux onto my diskless client and the kernel tells me to insert a root floppy and press enter

First you should check that you built your kernel correctly. It should have support for the root filesystem built in. If you want to use an NFS mounted directory as root the kernel should have TCP/IP support installed. Also it has to have a driver for your network card built in, and NFS and NFSROOT have to be both specified. When using a ramdisk it's support has to be compiled in as well as support for the filesystem with which you formatted the ramdisk image. Please note that the loaded kernel is not able to use modules at bootup time (only _after_ the root filesystem has been mounted, but not before), so everything has to be compiled in.

If the kernel is not able mount it's root via NFS, this might have many different reasons. It requires all addresses in the /etc/bootptab file to be correct, and the access rights on the

server have to be set correctly - not only in /etc/exports but also the permissions for the directory to get mounted. If that's correct check that a portmapper is running on the server, and that it registered the mountd and nfsd services correctly. You can usually do this by running the command

```
rpcinfo -p
```

Note that services are only listed here if their associated server process is really running. The rpcinfo output should then look something like this:

```
program vers proto   port
 100000    2   tcp    111  portmapper
 100000    2   udp    111  portmapper
 100003    2   udp   2049  nfs
 100003    2   tcp   2049  nfs
 100005    1   udp    663  mountd
 100005    1   tcp    665  mountd
```

However, the port numbers might be different.

When the kernel starts mounting the NFS root directory it prints out the name of that directory on the server. It should be the same as the one configured in /etc/bootptab. Check that it's correct. If not you can try to use the -d option with mknbi-linux to specify the name explicitely.

If the kernel gets an error from the server's nfsd, it prints a number which is defined according to the NFS protocol. The most commonly occurring numbers are:

```
 1  -  permission denied to access directory
 2  -  directory doesn't exist
 5  -  I/O error on server filesystem
13  -  nfsd is unable to access directory
20  -  path name is not a directory
63  -  path name is too long
```

Note that some nfsd and mountd programs only read /etc/exports on startup. If you changed this file afterwards, you will have to restart both daemons. Additionally, with nfsd versions for Linux earlier than 2.1 you will have problems with special files like UNIX domain sockets or block/character special files on your NFS partitions. You should therefore use the latest available versions.

Problem: The Linux kernel mounts it's root correctly but doesn't give me a login prompt.

1.) This might be the result of an incorrect setup of the root file-system (see No. 2 below). However, it's also possible that your

TOC

server reported the wrong major/minor numbers for the console device even though you specified them correctly in the NFS mounted root directory. I know of this problem with AIX and HP-UX servers, but there might exist others as well which don't transfer special devices via NFS as Linux requires it. One solution to solve this problem is to boot the diskless client with a ramdisk image as it's root, and then mount the should-be-root directory on the server using NFS. Then you can create the special files in the dev directory using Linux's mknod program, and use the NFS root mounting bootimage again.
Another way is to try to find out, how the server operating system encodes major/minor numbers on it's own filesystem. For example, HP-UX uses a 32 bit device number, with the 8 highest bits being the major number, and the lower 24 bits being the minor device number:

```
major << 24 | minor   ==>   aaaaaaaabbbbbbbbbbbbbbbbbbbbbbb
```

In this representation (a) means a bit of the major number, and (b) means a bit of the minor number. Linux uses the following scheme instead:

```
major << 8 | minor    ==>   0000000000000000aaaaaaaabbbbbbbb
```

The NFS protocol now transfers these 32 bits just as they are, without any further interpretation regarding major/minor numbers. That means, that all relevant bits in the Linux representation fit into the minor number on HP-UX. Therefore, if you create a device on the HP-UX server, you have to alway give it a major number of zero and compute the minor number the way mentioned above for Linux. For example, to let Linux see a device 5/2 in it's NFS-mounted /dev directory, you can compute the minor device number on HP-UX as

```
5 << 8 | 2    ==>  1282
```

So the device to create on the HP-UX server is 0/1282. This will let Linux see 5/2 after the filesystem is mounted with NFS.

2.) Another reason for this problem might be that the init process doesn't get started at all. This can be a result of incorrect shared libraries, which the client might see but without a proper ld.so.cache file. Or the shared libraries are not reachable by the client at all. Bruce Janson and Markus Gutschke collected a good list of possibilities, which you should check out:

- you do not have a private copy of the /, /etc, /var, ... directories

- your /dev directory is missing entries for /dev/zero and/or /dev/null or is sharing device entries from a server that uses different major and minor numbers (i.e. a server that is not running Linux - see above).

- your /lib directory is missing libraries (most notably lib and/or libm*) or does not have the loader files ld*.so*

- you neglected to run ldconfig to update /etc/ldconfig.cache or you do not have a configuration file for ldconfig.

- your /etc/inittab and/or /etc/rc.d/* files have not been customized for the clients.

- your kernel is missing some crucial compile-time feature (such as NFS filesystem support, booting from the net, trans-name (optional), ELF file support, networking support, driver for your ethernet card).

- missing init executable (in one of the directories known by the kernel: /etc, /sbin, ?)

- missing /etc/inittab

- missing /dev/tty?

- missing /bin/sh

- system programs that insist on creating/writing to files outside of /var (mount and /etc/mtab* is the canonical example)

Problem: Can't compile the bootrom

Please get in touch with me if you encounter any problems while recompiling the bootrom.

The Linux Distribution HOWTO

Eric S. Raymond <esr@thyrsus.com>

v6.3, 8 May 1999

This document is intended to help new users choose a Linux distribution, and to help experienced users track the state of the Linux market. It does not aim to be a complete list of Linux distributions for all platforms, but instead focuses on leading English-language Intel distributions available on CD-ROM and accessible to Linux novices.

1. Introduction

- 1.1 New versions of this document
- 1.2 Recent Changes
- 1.3 Overview of the Linux Market
- 1.4 Editorial Recommendations

2. Distributions

- 2.1 Caldera OpenLinux
- 2.2 Debian Linux Distribution
- 2.3 Linux Pro
- 2.4 Red Hat Linux
- 2.5 Trans-Ameritech Linuxware
- 2.6 Slackware
- 2.7 S.u.S.E.

3. Submissions To This Document

- 3.1 Types of Submissions
- 3.2 How to submit
- 3.3 Submission Guidelines

4. Administrivia

- 4.1 Terms of Use
- 4.2 Acknowledgements

1. Introduction

There is no single distribution of the Linux software. Instead, there are many such distributions, available both via anonymous FTP and by mail order on CD-ROM.

The purpose of this document is to provide short summaries of the English-language Linux CD-ROM distributions, and to provide pointers for the reader to find more information. A German Distributions HOWTO was formerly maintained by Marco Budde but seems to have disappeared. We are not aware of any distributions in languages other than English and German.

The information presented here is not complete; there are other Linux distributions than are listed here. By the nature of open source software, anyone that changes anything in their Linux installation and makes it available to others somehow is `a distribution'. A document such as this has to pick somewhere to draw the line. I have picked `available on CD'. This is by no means the definition of a Linux distribution. There are numerous small distributions that offer things the big distros don't, not the least of which is smallness itself.

The information presented here is not complete; there are other Linux distributions than are listed here. If you are associated with a CD-ROM distribution we don't list, please see Submissions To This Document near the end of this document for information on making a submission. It's easy and should take less then five minutes.

For a more complete list of distributions (albeit with sparser information on each) see the Linux HQ Distributions List.

Disclaimer: We make absolutely *no* guarantee as to the correctness of the information, prices, and ordering details given in this document. Check the last-modified field of each to get an idea of its currency, then go to the vendor's web page for up-to-date information. Furthermore, unless otherwise stated the Linux software comes with *ABSOLUTELY NO WARRANTY*.

Your editor tries to stick to facts in most of this HOWTO, but he has some opinions on the state of the Linux market. If you care what they are, you can read them under Editorial Recommendations.

Disclosure: I (esr) have no financial connection to any Linux vendor, nor have I accepted any renumeration or perquisites from any vendor other than free product for review (and one T-shirt from Red Hat).

1.1 New versions of this document

This document will be posted monthly to the newsgroups `comp.os.linux.answers`. The document is archived on a number of Linux FTP sites, including `metalab.unc.edu` in `pub/Linux/docs/HOWTO`.

You can also view the latest version of this HOWTO on the World Wide Web via the URL

TOC

http://metalab.unc.edu/mdw/HOWTO/Distribution-HOWTO.html.

Feel free to mail any questions or comments about this HOWTO to Eric S. Raymond, `esr@snark.thyrsus.com`. Please do *not* send me general Linux questions or requests for help in choosing a distribution unless you're willing to hire me at normal consulting rates; I don't have time to deal with them, and I try to put everything I know about choosing a distribution in this document.

1.2 Recent Changes

Yggdrasil Plug-And-Play and Craftworks Linux seem no longer to be available for sale, and have been dropped. DOSLINUX has also been dropped due to its specialist nature.

1.3 Overview of the Linux Market

In the beginning (say, 1993), a Linux distribution was something you downloaded off the Internet onto floppies. Installation was a laborious process and repeated frustrations due to bad media were common.

Then came cheap CD-ROM drives and the CD-ROM, a medium ideally suited for shipping large volumes of operating-system software cheaply. There's a whole mini-industry now built around commercial CD-ROM Linuxes, and (because the vendors have actual cash flow to fund support and marketing) they increasingly dominate the Linux world. Debian is now the only significant non-commercial release, and even it seems to be propagated largely by shovelware CD-ROMs.

Most of the CD-ROM distributions (including Slackware, Yggdrasil and Red Hat) are still available for FTP from the home sites of their developers. But if you have a CD-ROM drive and a few dollars, you will have many more distributions and more support options to choose from (and you'll usually get some useful paper documentation). For more on the details of installation, see the Linux Installation HOWTO, http://metalab.unc.edu/mdw/HOWTO/Installation-HOWTO.html.

Prices for CD-ROM distributions of Intel Linuxes start at $20 and top out at a whole $50 (and the extra few dollars can buy real value). Many vendors sell subscription deals that will lower your cost-per-CD for regular updates over the subscription period.

Price correlates with features and quality pretty well (as one would expect in a very competitive market). Your editor recommends paying the few extra dollars for a top-drawer original CD-ROM distribution; this will pay off in fewer installation and administration hassles down the road.

Making good choices is much simpler than it used to be. In 1995-96 the Linux market underwent a serious shakeout, with a very few commercial distributions emerging as leaders while weaker ones disappeared or stagnated. The toll among general-purpose non-commercial distributions has been even fiercer; essentially, only Debian survives in this role.

As a result, the three-tier structure of primary distribution builders, value-added repackagers, and bottom-feeding CD shovellers that used to define the market has nearly collapsed. To be competitive in 1997, a Linux outfit (whether commercial or noncommercial) has to offer reasonable support and behave like a primary distribution builder, whether it's really one or not. So as long as you look for a recent freeze date, it is pretty hard to get stuck with a dud distribution these days.

1.4 Editorial Recommendations

Last section, the facts. In this section, my opinions (for whatever they're worth -- and remember the caveat about free advice). There is no substitute for doing your own evaluation based on experience and the data in this guide, and these are intended more to illuminate my possible biases than as a guide to what you should do.

From the beginnings of the Linux CD-ROM industry in 1993 to Fall 1995, Yggdrasil was the king of the hill -- it essentially founded the CD-ROM market and then set the standard for everybody else. I used Yggdrasil, and I recommended it over commercial System V versions for its superior documentation, large collection of applications, and enlightened policy of sending free releases to freeware authors and dedicating part of the price of each CD-ROM to financially supporting free software. But Yggdrasil hasn't issued a new release since 1995 and they've been left behind by the market.

I now run Red Hat Linux and am quite satisfied with it. Red Hat's RPM technology currently gives it, IMO, a technical edge over any other vendor. They've made most of the right moves at the right times and I consider them the current market leader.

If you're ideologically wedded to using a non-commercial distribution, Debian seems to me to be the clear choice, the only one left with a serious support team behind it.

These opinions should certainly not be interpreted as an unconditional endorsement; different Linux distributions are optimized for different needs, and yours may well be best served by some other distribution (especially if, unlike me, you're mainly a DOS user and are looking for a distribution tuned for dual-boot systems and being launched from DOS).

Furthermore, industry standing is volatile. By the time you read this, Red Hat or Debian may well have fallen off their games and been displaced by hungrier newcomers.

---

2. Distributions

All these distributions are available on CD-ROM (some of the non-commercial ones only show up on network-archive snapshots). Most of these are available for free over the network (but the commercial ones won't support you if you buy this way). They are custom-assembled, rather than just being re-packagings of a pre-existing network release.

They are listed in alphabetical order.

2.1 Caldera OpenLinux

Distributor:

Caldera, Inc.
633 South 550 East
Provo, Utah 84606

Net: info@caldera.com
WWW: http://www.caldera.com
FTP: ftp://ftp.caldera.com
Vox: (801)-377-7678
Fax: (801)-377-8752

Provider's Description:

Caldera is shipping OpenLinux 1.2 in a three-tiered family of products: OpenLinux Lite, OpenLinux Base, and OpenLinux Standard.

OpenLinux Lite is a freely available evaluation of the OpenLinux product (it includes a 90-day evalutation of the Caldera integrated Desttop--the full licensed version of the desktop is included in the Base and Standard products). OpenLinxu Lite can be downloaded from Caldera's ftp site, or obtainded on CD for the cost of shipping. It is also designed for use by book and software publishers to provide a commercially stable Linux offering with their books and software products.

OpenLinux Base contains everything that is in Lite plus it is a complete, easy-to-install Linux operating system, with integrated Desktop, Internet client and server components, and Intranet connectivity. OpenLinux Base includes Netscape Navigator and full documentation.

OpenLinux Standard includes what is in Base plus more high-end, name brand, commercial software components (such as Netscape FastTrack Web Server, Netscape Navigator Gold Web browser and authoring tools, ADABAS D database, StarOffice productivity suite, Caldera OpenDOS, and NetWare client and administration tools).

All three tiers of the OpenLinux 1.2 release are currently available.

TOC

With Caldera OpenLinux, you can see a whole world of powerful new computing, even the pc of a UNIX workstation on your PC! By using a Linux kernel, OpenLinux gives you speed and stability that compare to systems costing thousands of dollars. Open Linux also gives you:

- Real multi-tasking runs many text or graphical applications at once; no application can cause another to crash, unlike less robust systems.
- The Linux 2.0 kernel gives super-fast TCP/IP access to the Internet, plus loadable device drivers, for the most efficient use of your system resources.
- Easy-to-follow, quick installation with menu-based choices and auto-detection of your hardware make installation a snap on most systems. The Getting Started Guide (included) provides a step-by-step illustrated procedure, with troubleshooting tips.

No other Linux system includes a complete, integrated desktop interface with configuration tools and graphical editor. Its powerful drag-and-drop capabilities come pre-configured with thousands of icons and application settings. The Desktop provides a convenient place where you can store your favorite programs and data files for instant access. The Desktop gives you:

- Drag-and-drop launching of programs with data files.
- An integrated graphical text editor.
- Easy access to an array of powerful system configuration tools.
- A configurable icon bar, custom layouts, and preference settings.
- Powerful file management using drag-and-drop or graphical dialogs.

Ordering:

By email with PGP key, by phone, by fax, or by conventional mail. Full ordering details are at http://www.caldera.com/mpro/orinfo/orderinfo.html.

Support:

Caldera is one of the only Linux distributers to offer E-mail and telephone support. OpenLinux Base (E-mail only) and Standard (Telephone and E-mail) users are fully supported (Standard users receive 5 free incident calls during the first 30 days). There is no technical support for the Lite product.

OpenLinux Base and Standard ship with full licensed versions of the Caldera Desktop, which do not expire. OpenLinux Lite ships with a 90-day evaluation version of the Desktop.

Last Freeze Date:

May 1998 (?)

Entry last modified:

13 August 1998

2.2 Debian Linux Distribution

TOC

Distributor:

Software in the Public Interest, Inc. PO Box 1326
Boston, Ma. 02117 USA

Net: info@debian.org
WWW: http://www.debian.org
FTP: ftp://ftp.debian.org/debian

Provider's Description:

Debian Linux is the result of a volunteer effort to create a high-quality non-commercial Linux distribution. Debian Linux 2.0 is a complete and fully-functional Unix-compatible operating system for the personal computer. The system uses ELF executables and libc6 (unlikeprevious Debian versions, which used libc5). At present, Debian Linux is available for the IBM PC architecture (386/486/Pentium) and m68k, with versions for Sparc and Alpha promised for 2.1. ARM, PPC, and UltraSparc ports are underway.

Debian Linux is an easy-to-upgrade distribution that mostly consists of freely redistributable software from a number of different sources and systems. Support for shadow passwords is available, although it should be considered experimental in the 1.1 release. Non-free packages may also obtained, making Debian a well-rounded system.

The benefits of Debian Linux are its upgradability, well-defined dependencies between packages, and its open development. It is the only distribution of Linux that is being developed co-operatively by many individuals through the Internet, in the same spirit as Linux and other open-source operating systems. More than 400 package maintainers are working on over 1500 packages and improving Debian Linux. A sophisticated bug tracking system allows users to easily report bugs and security concerns which are quickly dealt with by the Debian community. A new release of the package will soon occur on well-known ftp archives.

Debian Linux is a very dynamic distribution. Snap-shot releases are made about every three months, the ftp archives are updated daily.

For more information about Debian Linux, please refer to the files at ftp://ftp.debian.org/debian/doc/ or visit our World Wide Web page at http://www.debian.org/.

If you're interested in joining this project you are invited to subscribe to either debian-user@lists.debian.org or debian-devel@lists.debian.org by sending a mail to debian-user-request@lists.debian.org (or debian-devel-request respectively) with the word "help" in the subject line.

Internet Access:

Debian Linux is available via anonymous FTP from ftp://ftp.debian.org/debian/ and on a mirror near to you.

Last Freeze Date:

24 July 1998 (2.0)

Entry last modified:

13 Aug 1998

2.3 Linux Pro

Distributor:

WorkGroup Solutions, Inc.
P.O. Box 460190
Aurora, CO 80046-0190

Vox: (303)-699-7470
Fax: (303)-699-2793
Net: info@wgs.com (orders)
FTP: ftp://ftp.wgs.com/pub2/wgs

Provider's Description:

The charter of the WGS Linux Pro CD is different from all other distributions currently on the market. We are actively pursuing penetration into the commercial marketplace for Linux. Everything we do is aimed at increasing the size of the Linux community, and helping Linux to become THE one true operating system, not just an operating system.

WGS Linux Pro consists of our Main "Stable" Linux CD, for which we select what we consider to be the best Linux distribution available. Then we make bug fixes & minor enhancements. This CD often contains software older than the latest, under the theory that "the latest is not always the greatest". Together with this approach we provide technical support as well as support agreements. For this reason WGS Linux Pro is considered a primary Linux distribution. Optionally, and in addition to our primary CD, we supply supplemental CDs containing all the latest Linux software (including current archives of the tsx-11, metalab, and Red Hat sites). Virtually anything you will get with anyone's Linux product can be found on these supplemental CDs. We continually add more to these supplemental CDs as software becomes available. Even with all this, pricing on our product line is very favorable when correctly compared to other Linux distributions. Also optional (included with Linux Pro+) is a professionally produced and printed manual containing the Linux documentation project and more!

Current version is 4.0a; kernel is 1.2.13 with installable 2.0.x on the CD.

Internet Access:

Look in http://www.wgs.com for a list of WGS products, and latest information on the CD-ROM.

Ordering:

WGS Linux Pro is available on CD from dealers and distributors worldwide, and soon from our FTP site.

You may call, fax, email, or mail us, to purchase, or request additional information. We will be

happy to respond to any question you have. We accept American Express, Visa, Discover, and MasterCard, or purchase orders with approval of credit.

Miscellaneous:

Complete information is available on our FTP site.

WGS publishes a free E-Mail newsletter on Linux and FlagShip to which you may subscribe by just emailing us your request.

In addition we carry Motif, FlagShip, A line of books, and other products to make your experience with Linux more complete and fulfilling. Check us out!

Last Freeze Date:

1 October 1996.

Entry last modified:

13 March 1997

Editor's comment:

They describe their current distribution as "Red Hat 3.03 plus".

2.4 Red Hat Linux

Distributor:

Red Hat Software
3201 Yorktown Rd, Suite 123 DeKalb Center
Durham, NC 27713

Vox: (800) 546-7274 or (919) 572-6500
Fax: (919) 572-6726
Net: redhat@redhat.com
WWW: http://www.redhat.com
FTP: ftp://ftp.redhat.com

Provider's Description:

Red Hat 6.0 is based on the 2.2 kernel and is available for Intel, Sparc and Alpha platforms.

Four installation innovations make Red Hat the easiest Linux to install ever. Our graphical installation mode brings you straight up into X for most of the installation. Our boot disk creation script eases selection of the proper boot disk, and it saves your existing network configuration information and XF86Config so you don't have to configure TCP/IP or X! Our FTP install allows you to install simply by downloading 3 floppy disk images -- the rest is done automatically! And finally, the installation includes seamless support for PCMCIA devices -- install Red Hat on your laptop as easily as on your desktop machine!

After installing Red Hat Linux once, you will never need to reinstall Linux again! The new RPM

packaging system is sophisticated enough to allow upgrading to new Red Hat releases without reinstalling your system - no partitioning, no backing up all your files, no headaches.

Ordering:

Available directly from Red Hat Software and most distributors of Linux related products. Please contact Red Hat Software for ordering details (on-line ordering is available through Red Hat's Web). Suggested retail price is for the Intel version is $49.95, for the Sparc and Alpha versions $99.95.

Last Freeze Date:

10 Mar 1999

Entry last modified:

7 Jun 1999.

Editor's Comments:

The big selling point of this distribution is RPM, the Red Hat Package Manager. This piece of software is a remarkable advance; it allows you to cleanly install and de-install applications and operating-system components, including the kernel and OS base itself. RPM is now used as well by essentially all other distributions except Debian.

2.5 Trans-Ameritech Linuxware

Distributor:

Trans-Ameritech
2342A Walsh Avenue
Santa Clara, CA 95051

Net: info@trans-am.com
WWW: http://www.trans-am.com
Vox: (408)-727-3883
Fax: (408)-727-3882
BBS: (408)-980-9840

Provider's Description:

Trans-Ameritech has published 10 releases of Linux by early 1997. Traditionally, Trans-Ameritech has set new standards for combining ease of use and straightforward installation, even for a first-time user, with its line of LinuxWare CD-ROMs. This is a flexible, easy-to-install operating system geared toward those interested in learning Unix as well as technical people, students and home PC users.

- New Linux users will appreciate a Windows-based set up program from the CD.
- To minimize the possibility of hardware conflicts many extra kernels are provided for different configurations. They are usable for installation and normal operation.
- Many on-line documents are provided for quick reference, including the Linux

Documentation Project files in source, dvi and ps formats.

Many applications are included:

- MS-Windows based X configuration program
- C/C++,Pascal and Ada compilers as well as converters fron Fortran
- TCP/IP networking, UUCP,SLIP,CSLIP,PPP
- Internet access with binaries and sources for FTP,Telnet,News and E-mail
- Multiple terminals and X-Windows environment
- Gnu and international versions of the ispell spell-cheker
- The communications apps:term,minicom,Seyon (X-Windows based)
- Editors:elvis(vi clone),joe,jove,Emacs
- PostScript clone ghostscript
- Object oriented GNU Smalltalk and the Smalltalk interface to X
- TCL/Tk (Powerful scripting language with Motif-like X interface)
- Programs for electrical engineers and ham enthusiasts
- Interviews libraries,include files and doc Word-processor and idraw drawing program
- Typesetting:TeX,LaTeX,xdvi,dvips,Metafont,groff
- Andrew multimedia word prodessor with hyperlinks
- FAX send and receive on either class 1 or class 2 fax modems
- DOOM for game enthusiasts

All the sources are available on the CD-ROM. The often needed sources are uncompressed and can be used directly from the CD-ROM.

An uncompressed Linux filesystem is available for references and disk space conversaion. You can run programs directly from the CD-ROM! There is a large info directory for on-line reference and many manpages.

For hacker's reference, uncompressed FreeBSD and netBSD sources are provided.

Our distribution is targeted for Windows and DOS users who want an easy migration path upwards, rather than for UNIX experts.

Ordering:

Email orders are taken at order@trans-am.com

The price for our current release of LinuxWare 2.5 is $19.95

The price for a package order: LinuxWare 2.5,Supplement 5 and Supplement 4 (Ultimate LinuxWare Bundle) in one shipment is $30

If you order with a credit card (VISA, MC, AmEx, Discovery), please, indicate the card number, expiration date and your mailing address.

Shipping and handling in US: single CD-ROM or Ultimate LinuxWare Bundle is $5(first class US

TOC

mail).

Overseas shipping is $8 for single CD-ROM , $12 for the bundle.

COD is available in the US only for $4.50. California residents, please add 7.75% sales tax.

Annual subscription (4 releases) is available for $80 plus S&H (note: there are 4 shipments in a subscription). Example: subscription in US is:$80+$5x4=$100. Subscription in Europe/Japan etc. $80 + $8x4 = $112.

Miscellaneous:

If you have any further questions,please contact us through E-mail: info@trans-am.com or order@trans-am.com.

Last Freeze Date:

January 1997 (Supplement 5)

Entry last modified:

13 August 1998

2.6 Slackware

Distributor:

Walnut Creek CDROM
4041 Pike Lane, Suite D
Concord, CA 94520

Net: info@cdrom.com (information), order@cdrom.com (orders), support@cdrom.com (support).

Provider's Description:

Ftp.cdrom.com is the home of Slackware Linux. We are the publishers of the Official Slackware Linux CDROM. Our distribution is a 4-disc set with the current version being Slackware 96. The current disc is based on the 2.0.34 kernel.

Slackware Linux is a full featured distribution of the Linux operating system designed for 386/486 computers with a 3.5" floppy and CD-ROM drive.

Internet Access:

WWW: http://www.cdrom.com/titles/os/slack96.htm
FTP: ftp:ftp.cdrom.com/pub/linux/slackware

Ordering:

The cost of the CD-ROM set from Walnut Creek is $39.95.

Last Freeze Date:

July 1998 (3.5)

Entry last modified:

13 August 1998.

2.7 S.u.S.E.

Distributor:

S.u.S.E., Inc.
458 Santa Clara Ave
Oakland CA 94610 USA

Net: info@suse.com
WWW: http://www.suse.com
FTP: ftp://ftp.suse.com
Vox: +1-510-835-7873 Fax: +1-510-835-7875 fax

Provider's Description:

S.u.S.E. Linux offers users of all levels, from novice to expert, quick and easy entry into world of Linux and Unix. Resources such as a menu-driven installation from CD-ROM, a modular boot diskette, 400-page reference book and S.u.S.E.'s own system administration tool, YaST allow one to quickly get Linux installed and running, and to keep it running smoothly.

Furthermore, S.u.S.E. offers a series of supported X servers, which support the newest graphics cards. These servers are created in cooperation with the XFree86 (tm) development team. See the following URL for more details: http://www.suse.de/XSuSE.

New in S.u.S.E. Linux 5.3:

- Kernel 2.0.35
- KDE 1.0
- GIMP 1.0
- SaX (SuSE advanced XF86 configuration tool)

Quick overview of other features:

- The Linux OS, over 875 software packages, including complete sources and live file system on 4 CD-ROMs
- 400 page reference book
- Packages in RPM format; TGZ packages installable
- System V compatible boot concept, compliance with file system standard
- Installation and administration utility 'YaST' configures network, ISDN, e-mail, printer, X Window System
- S.u.S.E. X servers which support new graphics cards
- Administration of file systems, users, groups
- Automatic configuration of several window managers

- Online help system and documentation
- Demo mode, either directly from CD-ROM, or on DOS partition

Ordering:

You can download from ftp://ftp.suse.com/pub/SuSE-Linux. For the CD-ROM with support, order vphone, secure WWW order form, fax, or post. Price is $49.95, or $34.95 per edition by subscription. Use order code LSUE520 for the English version, LSUD520.

Also available: OSF Motif 2.1 for $129.95 (price for update to be determined; order code: LMEA210

Support:

60 days of free installation support are included with each purchase of S.u.S.E. Linux. See the web page for contact info and conditions.

Last Freeze Date:

Unknown.

Entry last modified:

13 Aug 1998 (5.3)

---

3. Submissions To This Document

I encourage anyone and everyone who maintains a Linux distribution or mail order service, to submit information on their service to this HOWTO. It's easy and fun, and it's free advertising. This document is posted to many places and is archived (see the next section).

3.1 Types of Submissions

We are interested in submissions for:

- Complete distributions of Linux software, available either via anonymous FTP, UUCP, or mail order. By `complete distribution' we mean any set of software which can be used to build a complete Linux system from scratch.
- Layered products or individual software packages available only via mail order. If your software package is available via anonymous FTP, chances are people can find it. Software products only available via mail-order include commercial things such as Motif, and any other commercial software ported to Linux.
- Any other *Linux-specific* goods available via mail order, such as Linux-specific books and documentation, T-shirts, and assorted paraphrenalia. Note: our interest in the latter categories is marginal -- this is a Linux Distributions HOWTO, not a catalogue of cutesy Linux junk. Whatever you're selling will have to be (a) very useful, (b) very original, or (c) very funny, to get listed here.

If the number of submissions for services and layered products is large, I'll create a separate HOWTO for these items.

3.2 How to submit

To submit an entry to this HOWTO, please send mail to `esr@snark.thyrsus.com` with the following information. This format is not machine-parsable; any of the fields may be any length that you wish, but I'd like to keep each entry down to, say, 50 lines.

Name:

> *Name of service or distribution*

Distributor:

> *Name of company, person, etc. who distributes/maintains the service or distribution. Should include mail, email, phone contact information, and (if possible) a WWW reference.*

Provider's Description:

> *Description of the distribution or service that you provide. If this is a software distribution, please include information such as what software is included, versions, general overview of installation,*

requirements, and so on.

Internet Access:

Where your service or distribution is available over the Internet; typically a WWW or FTP address.

Ordering:

How to order your distribution or service, if applicable. Include prices, shipping information, methods of payment, etc.

Support:

Information on support terms and support contracts.

Miscellaneous:

Anything else that you find relevant.

Last Freeze Date:

Last freeze date of the current version(s). Also, your estimated update frequency.

3.3 Submission Guidelines

Please keep your entry as short as possible. If you need to include extensive information, please make a reference to where one can FTP or mail to get more information on your distribution; these entries are only meant to be *pointers* to where one can find information on your service or distribution.

If you provide more than one service or distribution, please use *separate entries* for each.

I may edit your entries for conciseness and brevity, if I find any irrelevant information, or if the entry is overly verbose. Otherwise the content should remain the same.

When making submissions to the Distribution-HOWTO, you grant implicit permission for me to use the entries in other materials, such as books from the LDP, and other online documents. For example, information from the Distribution-HOWTO may be included in a published Linux book. If you do not want me to include your entry in materials other than the Distribution-HOWTO, please say so.

---

4. Administrivia

4.1 Terms of Use

This document is copyright 1997 by Eric S. Raymond. You may use, disseminate, and reproduce it freely, provided you:

- Do not omit or alter this copyright notice.
- Do not omit or alter or omit the version number and date.
- Do not omit or alter the document's pointer to the current WWW version.
- Clearly mark any consdensed, altered or versions as such.

These restrictions are intended to protect potential readers from stale or mangled versions. If you think you have a good case for an exception, ask me.

4.2 Acknowledgements

This document was originated by Bill Riemers. Matt Welsh maintained the second version. Erik Troan then maintained the document to release 3.0.

In January 1995, Eric Raymond, while unaware of the existence of this document, began to develop a similar FAQ in the format of his now-discontinued "PC-clone UNIX Software Buyer's Guide", which had covered mainly System V UNIXes and BSD/OS.

In March 1995, Eric approached Erik about cooperating on a merged version. In early April 1995, Erik went to work for Red Hat Software, and (wishing to avoid a conflict of interest) handed the document to Eric. Eric merged in a lot of new information and added several new fields to the distribution entries.

Accordingly, this document has been a sort of serial collaboration. The editorial `we' generally tags observations by all the maintainers; `I' is Eric (the current one) speaking.

We are delighted to acknowledge the contributions of all the Linux users and Internet hackers who have contributed information and feedback.

The Linux Emacspeak HOWTO

Jim Van Zandt, jrv@vanzandt.mv.com

v.1.6, 10 April 1999

This document describes how a blind user can use Linux with a speech synthesizer to replace the video display. It describes how to get Linux running on your own PC, and how to set it up for speech output. It suggests how to learn about Unix.

1. Introduction

2. Stage 1. DOS with speech

- 2.1 Getting Linux on CDROM
- 2.2 Getting Linux by FTP
- 2.3 Linux Documentation

3. Stage 2. Terminal to remote UNIX system

- 3.1 Learning Unix
- 3.2 Learning Emacs

4. Stage 3. Terminal to local Linux system

- 4.1 Installing Linux
- 4.2 Learning Unix System Administration

5. Stage 4. Emacspeak under Linux

- 5.1 Emacspeak Introduction - Speech Enabled Normal Commands
- 5.2 Emacspeak Introduction - New Commands
- 5.3 Emacspeak Introduction - Using the Help System

TOC

6. Footnotes and References

- 6.1 JAWS
- 6.2 TELIX
- 6.3 COMMO
- 6.4 SimTel
- 6.5 InfoMagic
- 6.6 Walnut Creek
- 6.7 Red Hat
- 6.8 Craftwork
- 6.9 Yggdrasil
- 6.10 Emacs for DOS
- 6.11 GNU Mirror Sites
- 6.12 Emacspeak with Earlier Slackware Releases

7. Frequently Asked Questions (FAQ)

- 7.1 Why does it say "space" after each character?
- 7.2 Why the high pitched tone from a DECtalk?
- 7.3 Why the "No library autorevert in search path" message?
- 7.4 How do I change Emacspeak defaults?
- 7.5 What if an error message is not spoken?
- 7.6 The terminal emulator is the most problematic portion of Emacspeak.
- 7.7 In a term buffer, the first half screen is not spoken
- 7.8 Term mode gets flaky after a long session.

8. Legalese

1. Introduction

Emacspeak is an Emacs subsystem that allows the user to get feedback using synthesized speech.

Screen reading programs allow a visually impaired user to get feedback using synthesized speech. Such programs have been commercially available for well over a decade. Most of them run on PC's under DOS, and there are now a few screen-readers for the Windows platform. However, screen-readers for the UNIX environment have been conspicuous in their absence.

This means that most visually impaired computer users face the additional handicap of being DOS-impaired :-)

Emacspeak is an emacs subsystem that provides basic speech access. Emacspeak will always have the shortcoming that it will only work under Emacs. This said, there is very little that cannot be done inside Emacs, so it's not a real shortcoming:-) Within Emacs, you can open a "shell window" where you can run commands and examine their output, even output which has scrolled out of the window. Emacs provides special modes for running certain commands. For example, it can parse error messages printed by a compiler and open a separate edit window with the cursor at the point of the error. It can also run a debugger and keep a separate edit window open at the point in the source code corresponding to the program counter.

Emacspeak does have a significant advantage: since it runs inside Emacs, a structure-sensitive, fully customizable editor, Emacspeak often has more context-specific information about what it is speaking than its commercial counterparts. In this sense, Emacspeak is not a "screenreader", it is a subsystem that produces speech output. A traditional screen-reader speaks the content of the screen, leaving it to the user to interpret the visually laid-out information. Emacspeak, on the other hand, treats speech as a first-class output mode; it speaks the information in a manner that is easy to comprehend when listening.

This initial version provides a basic speech subsystem for Emacs; using Emacs' power and flexibility, it has proven straightforward to add modules that customize how things are spoken, e.g. depending on the major/minor mode of a given buffer. Note that the basic speech functionality provided by Emacspeak is sufficient to use most Emacs packages effectively; adding package-specific customizations makes the interaction much smoother. This is because package-specific extensions can take advantage of the current context.

Emacspeak will only work with emacs. However, emacs can be used to run any program that has a command-line interface (ls, cd, rm, adduser, etc.). In general use M-x shell to run such programs.

You can even run those like less or lynx which use escape sequences to control the appearance of the screen. The key to this is term mode, which you get with the emacs command M-x term.

Let me insert here some explanation, because the word "terminal" is used in several senses. The original terminal was of course an electromechanical device that translated keystrokes into character codes and received the same codes and printed the characters. Eventually these were replaced by entirely electronic "dumb terminals" which also implemented escape sequences to clear the screen, move the cursor, etc. Unix adapted to these by introducing an abstract notion of a "terminal" with very general capabilities. A program like a text editor issues generic commands, which are translated into the commands suitable for your terminal. The choice of translation is controlled by the variable "TERM" in the environment. One common value for this variable is "vt100", since many terminals implement the same commands as the DEC VT100. The Emacs command `M-x term' starts a virtual terminal with its own set of cursor commands. It notifies the operating system of this by setting the environment variable "TERM" to the value "eterm" (which is why it is sometimes called eterm mode).

Note that this Unix "terminal" is concerned only with escape commands for the display. Let me just mention a couple of other kinds of "terminal" to distinguish them from the Unix concept: For DOS, there are "terminal emulators" like COMMO and TELIX that simulate a terminal, and can be used to connect to a remote computer via a serial port (and usually a modem). That is, they accept keystrokes and send the character codes out a serial port, and receive character codes from the serial port and display them. Unix has similar programs, such as minicom and seyon. Both DOS and Unix have another program called "telnet" which can establish a similar connection to a separate computer using the Internet (that is, a TCP/IP connection rather than a serial connection). The program "kermit" is available for both DOS and Unix, and can establish either kind of connection. However, none of these should be confused with the eterm mode of emacs. There is nothing about eterm mode that implies a connection to a remote computer.

Note that in most cases there is an emacs mode that works better than the separate program in an eterm buffer. That is, instead of "less", just open a buffer on the file. Instead of "lynx", use w3. Instead of "man", use `M-x man', and type the name of the program in the minibuffer.

Remember, though: ordinary command-line oriented programs are best run in shell mode, which starts with the command `M-x shell'.

Emacs is a large program, but it does not all have to be in RAM, because Linux has virtual memory. You can designate a swap partition, so that programs (or parts of programs) can be swapped out when they are not being used. You can comfortably run emacs with 8 MB of ram plus 8 MB of swap space.

This document is limited to the following:

- Linux (not Free BSD)
- The Slackware distribution (not Red Hat, Debian, etc.)
- Speech output only (not Braille - see the Access HOWTO)
- DECtalk (Dectalk Express and MultiVoice), DoubleTalk, LiteTalk, Braille 'n Speak, Type 'n Speak, Braille Lite, Apollo, and Accent synthesizers (--not the SmarTalk, a sound card, etc.)
- Use of Emacs, with T. V. Raman's Emacspeak package, to drive the synthesizer.

The use of adaptive technology with Linux, and in particular, using adaptive technology to make Linux accessible to those who could not use it otherwise, is covered in the Linux Access HOWTO.

If you would like to help extend this document to cover one or more of the other alternatives, or point me to a discussion somewhere else, please contact me.

The most recent version of this document should be available in several formats at ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/ and its mirrors (see below). The plain text version should be available at ftp://leb.net/pub/blinux/doc/.

Emacspeak was written by T. V. Raman `raman@adobe.com`. Emacspeak has a Web page at http://www.cs.cornell.edu/Info/People/raman/emacspeak/emacspeak.html.

Emacspeak supports several speech synthesizers. The software required depends on which you have.

If you have a DECtalk Express or MultiVoice, you need the basic Emacspeak package, tcl (an interpreter), and tclx (extensions for tcl). You can get the source package for Emacspeak from the Emacspeak web page, or a binary package in one of the popular distributions of Linux (Slackware, Red Hat, or Debian). I build each of these packages. Since I normally run Debian, the Debian package will be available a little sooner than the others. At this writing, the most recent release of Emacspeak is version 7.0. Here are some URLs:

`http://leb.net/pub/blinux/emacspeak/cornell.mirror`

ftp://leb.net/pub/blinux/emacspeak/blinux/emacspeak-6.0-2.i386.rpm

Note: there are European mirrors of the blinux site which you should use if they closer:

ftp://ftp.uni-muenster.de/share/public6/
ftp://ftp.gwdg.de/pub/linux/misc/blinux/

For the internal DECtalk PC, you also need a kernel driver which is available from ftp://leb.net/pub/blinux/dectalk_pc-0.91.tgz

Extract it with the command: `tar xfz dectalk_pc-0.91.tgz' and then follow the installation instructions found in the README file. This driver apparently works with both the original DECTALK-pc and the new DECTALK-PC II.

For the external DoubleTalk, LiteTalk, Accent, or Apollo synthesizers, or for the Braille 'n Speak, Type 'n Speak, or Braille Lite devices in "speech box" mode, you need the Emacspeak package and a separate "speech server" which comes in the emacspeak-ss package:

ftp://leb.net/pub/blinux/emacspeak/blinux/emacspeak-ss-0.5.tar.gz

The DoubleTalk and LiteTalk are also supported by a separate speech server which has been packaged for Red Hat and Debian systems:

http://www.mv.com/ipusers/vanzandt/emacspeak-dt-tcl-1.2.tar.gz
ftp://leb.net/pub/blinux/emacspeak/blinux/emacspeak-dt-1.2-1.i386.rpm
http://www.mv.com/ipusers/vanzandt/emacspeak-dt_0.27-1_i386.deb
http://www.mv.com/ipusers/vanzandt/emacspeak-dt-tcl_1.2-1_all.deb

For the internal DoubleTalk, you need three pieces of software: the basic Emacspeak package, the emacspeak-ss or emacspeak-dt package, and a device driver. Here are some URLs for the device driver:

ftp://leb.net/pub/blinux/emacspeak/blinux/dtlk-1.12.tar.gz
http://www.mv.com/ipusers/vanzandt/dtlk-1.12.tar.gz
http://www.mv.com/ipusers/vanzandt/dtlk_1.12-1_i386.deb

There is a server for the Braille 'n Speak, Braille Lite, and Type 'n Speak devices used in "speech box" mode:

ftp://leb.net/pub/blinux/emacspeak/blinux/emacspeak-bs-tcl-1.2.tar.gz
ftp://leb.net/pub/blinux/emacspeak/blinux/emacspeak-bs-tcl_1.2-1_all.deb
ftp://leb.net/pub/blinux/emacspeak/blinux/emacspeak-bs-1.2-1.i386.rpm

Once again, the emacspeak-ss server handles all the synthesizers other than the DECtalk. You should use that one if you don't mind building and installing a source package.

Computer hardware, Unix user commands, Unix system administration, Emacs, and Emacspeak are each substantial subjects. Attempting to learn all of them at once is likely to lead to frustration. Instead, I suggest that the new user go through a sequence of stages, learning about only one system at a time.

---

2. Stage 1. DOS with speech

Most blind computer users have speech synthesizers with a screen reader program like JAWS [JAWS]. (References in this format refer to entries in the "Footnotes and References" section below.) Using this setup, install and become familiar with some terminal emulator like Telix [TELIX] or Commo [COMMO], which are available from the SimTel archive [SimTel] among others.

2.1 Getting Linux on CDROM

If you have or can borrow a CDROM drive, I recommend you get one of the many good distributions of Linux on that medium. The instructions below are for the Slackware distribution. I am most familiar with disks from InfoMagic [InfoMagic]. Another source is Walnut Creek [Walnut Creek] (where the whole idea of inexpensive CDROMs full of programs from Internet archives got its start). Distributions other than Slackware are available from Red Hat [Red Hat], Craftwork [Craftwork], and Yggdrasil [Yggdrasil]. As a rule, these CDROMs use the "ISO 9660" format, which can be read under DOS. (They also use the "Rock Ridge extensions" which add extra files in each directory. Linux uses the extra information to give you long filenames, both upper and lower case characters in filenames, and file permissions.)

2.2 Getting Linux by FTP

Another way to get Linux and its documentation is by FTP over the Internet. The home site for the Slackware distribution is Walnut Creek [Walnut Creek]. It is also carried by sunsite and many of it mirror sites. Here is a partial list:

- USA (home site) ftp://ftp.cdrom.com/pub/linux/slackware
- UK/Europe ftp://src.doc.ic.ac.uk/public/Mirrors/ftp.cdrom.com/pub/linux/slackware-3.1
- Japan ftp://ftp.cs.titech.ac.jp/pub/os/linux/slackware
- Taiwan ftp://NCTUCCCA.edu.tw/OS/Linux/Slackware
- Hong Kong ftp://ftp.cs.cuhk.hk/pub/slackware
- USA ftp://metalab.unc.edu/pub/Linux/distributions/slackware
- USA ftp://uiarchive.cso.uiuc.edu/pub/systems/linux/sunsite/distributions/slackware

More sites are listed in the INFO-SHEET:

```
http://metalab.unc.edu/pub/Linux/welcome.html
ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/INFO-SHEET
ftp://uiarchive.cso.uiuc.edu/pub/systems/linux/sunsite/docs/HOWTO/INFO-SHEET
```

2.3 Linux Documentation

Read the Linux documentation. I will quote here the file names and locations on the first disk of InfoMagic's December 1996 "Developer's Resource" set of six CDROMs, as seen under DOS. Other CDROM sets should have similar information, though perhaps differently arranged. The Slackware distribution is on disk 2 of the set. Matt Welsh's step by step guide to installing Slackware is in `\doc\install-\install-.002`. (This is a 245 page book!)

More general information is in the Linux "Frequently Asked Questions" list in

`\docs\linux.faq\linux-fa.asc`. Longer descriptions are in "HOWTO" documents (of which this is one). They are found in `\docs`. Note particularly `\docs\hardware`, which lists which kinds of hardware are supported by Linux, `\docs\meta-faq`, which points to sources of information (that is, a more extensive version of this paragraph), and `\help\index`, which is a list of the HOWTO documents with short descriptions. The Linux installation HOWTO, `\docs\installation`, is another (much shorter, somewhat older) version of Matt Welsh's installation instructions.

One note on reading the documentation. You may run into files with ASCII highlighting, where character-backspace-character stands for "bold", and underscore-backspace-character stands for "italics". One way to handle this is to use the `less` program, which displays these sequences in alternate colors. A DOS screenreader can, for example, search for such highlighted text. A DOS version of `less` can be obtained by FTP from the SimTel archive [SimTel]. Within the SimTel collection, look for directory msdos/textutil. For example, try ftp://ftp.coast.net/pub/SimTel/msdos/textutil.

I will suggest four alternatives for learning Emacs commands (see section [Learning Emacs]). The first option is to install Emacs under DOS and learn it while using the DOS screen reader. Where to get Emacs for DOS is a "frequently asked question" [Emacs for DOS].

The source code for Emacs (about 10 MB) can be gotten from ftp://prep.ai.mit.edu/pub/gnu/ (look for emacs-19.34b.tar.gz or similar), or from one of many mirrors of the GNU collection [Gnu Mirrors].

---

3. Stage 2. Terminal to remote UNIX system

Arrange for what is called a "shell account" on some Unix system. Most Internet Service Providers (ISPs) can provide this service. Use the terminal emulator program and a modem to dial in. Learn the basic Unix commands. If the system has Emacs installed, or you can persuade the system administrator to install it, this is your second chance to learn it. It is probably best to learn it at this point, because administering a Unix system (the next stage) will call for you to edit files. Therefore, I include here my suggestions for learning both Unix and Emacs.

3.1 Learning Unix

When you arrange for a shell account, or set up a new account on your own machine, you will have to decide on a username and a password. Your username will also be used in your email address, so try to find something short and memorable. Your password is important, and should be hard to guess. That usually means at least six characters, including at least one non-alphanumeric character.

When a Unix system is ready for you to log in, it normally displays a prompt ending with "login:". At this point you should type in your username. It will then prompt you for your password, and will turn off command echoing while you type it in.

The command to finish a terminal session is `logout`.

To learn about a command, use the `man` command to display its manual page. For example, learn more about the `cp` command by typing `man cp`. Of course, this helps only if you know or can guess the command name. However, each man page has a line near the beginning with the command name and a short description of what the command does. You can search a database of these lines using the command `apropos`. Thus, typing `apropos working` will list lines that include the word "working". Here's what that command prints here:

```
cd (n)                - Change working directory
pwd (n)               - Return the current working directory
Cwd (3pm)             - get pathname of current working directory
chdir (2)             - change working directory
fchdir (2)            - change working directory
get_current_dir_name (3) - Get current working directory
getcwd (3)            - Get current working directory
getwd (3)             - Get current working directory
pwd (1)               - print name of current/working directory
rcsclean (1)          - clean up working files
```

The numbers in parentheses are sections of the manual. User commands are in section 1. Functions called from within programs are in sections 2 and 3. Commands used mostly by the system administrator are in section 8. You can find out more about the on-line manual with the command `man man`.

Under Unix, commands normally accept options starting with a minus sign rather than the forward slash used

under DOS. In a path, directory names are separated by forward slashes rather than backward slashes. B operating systems have a "standard input", by default the keyboard, and a "standard output", by default t display screen. You can redirect the standard input using "<", and redirect the output using ">". You can use the output from one command as the input of another by separating the two commands with "|". This is called the "pipe" symbol.

The program that interprets your command is a "shell". Under DOS, COMMAND.COM is the shell. Most Unix shells are descendents of either the Bourne shell `sh` or the C shell `csh`. The shell most commonly used with Linux is the "Bourne again shell", or `bash`.

If a program gets "stuck", here is a sequence of keystrokes to try:

- Control-Q. You may have sent a control-S, which halts all output, without realizing it. The control-Q will restart it.
- Control-D, which signals "end of file" under Unix (similar to control-Z under DOS), in case the program expects input which you are not prepared to supply.
- Control-C is an interrupt, which may halt the program.
- Control-Z puts the program in the background. At this point you may simply log out, although you will be warned about the background process and will have to repeat the `logout` command. You can instead kill the process, as follows: Run `ps` with no arguments. It will list a header line, then one line for each of your processes. The first item on each line is the process id number, or PID. The command used to start the process (or at least the beginning of it) appears at the end of the line. If the PID were 117, you would kill the process with the command `kill -9 117`.
- If running Linux from the console, alt-2, or some other alt-number combination, will switch to a different virtual console. You can log in there just as if you had sat down to a different terminal.
- If your machine is connected to a network, you can log in from another machine.
- Control-alt-del should reboot the computer nondestructively.
- As a last resort, you can hit "reset" or cycle the power. This will leave the filesystems in an invalid state, since some buffers will not have been written to disk. The kernel will discover this while booting, and will take time to check and repair the filesystems. Actual data loss is unlikely unless you had something else going on at the time.

Guido Gonzato `Guido@ibogfs.cineca.it` has written an excellent guide to Linux for (former) DOS users, the DOS2Linux mini-HOWTO. You can probably find it where you found this document, or else at ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/mini/.

You can find general Unix information, including manual pages for several systems at http://www.cis.ohio-state.edu/hypertext/man_pages.html

There is a tutorial entitled "Beginning Unix and the C Shell" at http://www.eng.hawaii.edu:80/Courses/C.unix/page-03.html.

You can get general help from http://www.nova.edu/Inter-Links/UNIXhelp/TOP_.html or http://www.eecs.nwu.edu/unix.html

You can find a list of books on UNIX at http://www.eskimo.com/~cher/eskimospace/booklist.html.

3.2 Learning Emacs

TOC

When you start Emacs, you will normally list on the command line one or more files which you will be editing. To edit a file named "foobar" with Emacs, you would enter the command `emacs foobar`. If you enter the command `emacs` with no arguments, GNU Emacs will print out an introduction which includes the first five commands you need to learn, approximately as follows:

```
Type C-h for help;   (`C-' means use CTRL key.)
Type C-x u to undo changes.
Type C-h t for a tutorial on using Emacs.
Type C-h i to enter Info, which you can use to read GNU documentation.
To kill the Emacs job, type C-x C-c.
```

Note the way Emacs documentation refers to key combinations. C-h means hold the control key down while typing "h". You will also run into key combinations like M-v, which is pronounced "meta v". The tutorial suggests holding down the key labeled "edit" or "meta" then typing "v". I have never run across a keyboard with those keys, so I always use the escape key instead: typing "Esc" then "v" (two separate keystrokes). After using Emacs for a long time, I discovered that under Linux, the left "Alt" key works like a "meta" key. You may want to use this. On the other hand, some of these key combinations may conflict with your screen reader or communications program under DOS. Using the escape key is more reliable.

Three of the above commands start with C-h, which may be treated as a backspace by your communications program. In that case, you may access the help command using the long form M-x help. Conversely, you may find that pressing the backspace key starts the help command. This issue is treated in the Emacs FAQ, which is available within Emacs using C-h F or M-x help F. Look for the question "Why does the `Backspace' key invoke help?". In the mean time, you can end the help session with the command C-g. (This is the keyboard-quit command, which cancels any prefix keys you have typed.)

You may also find that C-s and C-q are unavailable because they are used for flow control (XON and XOFF). You should look at the question "How do I handle C-s and C-q being used for flow control?" in the FAQ. For the particular command C-x C-s (save buffer), you may substitute the command C-x s (save-some-buffers). The former command saves the current buffer, while the latter asks the user about each of the modified buffers.

Note in particular the command "C-h t" to start the Emacs tutorial. That is one the first things you will want to try. I will only make a couple of comments on the tutorial. To move the cursor, it gives the four commands C-f, C-b, C-p, and C-n (for forward, back, previous line, and next line). These commands always work. However, with a properly installed Emacs, the regular arrow keys should also work. Try them out and use them if you are more comfortable with them. Similarly, you may be able to use home, end, page down, and page up keys in place of the standard commands C-a, C-e, C-v, and M-v. Finally, all Emacspeak commands begin with C-e. Once you start using Emacspeak, you will have to type it twice to get the end of line function. (The "End" key should be unaffected by Emacspeak.)

---

4. Stage 3. Terminal to local Linux system

This arrangement again requires a DOS machine with a speech synthesizer and a terminal emulator program. However, instead of dialing up a remote computer, it is used as a terminal to a local computer running Linux. To get to this point, you need to install Linux on a machine. You may be able to prevail on a knowledgeable friend to help you with this. However, it is also possible to install it yourself with speech feedback for almost the whole procedure.

4.1 Installing Linux

First, some background. Even the simplest Unix system requires a program called the kernel and a root file system. The kernel has all the device drivers and resource management functions. One normally thinks of a "file system" as residing on a hard disk or floppy disk, but during an installation it is usually in ram. Linux is normally installed by writing a kernel image to a floppy disk, called the "boot floppy", configuring it to reserve a section of RAM for a ramdisk, then filling that ramdisk with data from a second floppy disk, called the "root floppy". As soon as both floppies have been read in, the user can log in as "root" and complete the installation. The sighted user logs in on the "system console", that is, the computer's own keyboard and video display. However, remember that Unix has been a multiprocessing operating system from the very beginning. Even this very primitive Unix system, running out of a small ramdisk, also supports logins from a terminal connected to a serial port. This is what a blind user can use.

To connect the two computers, you can use a "null modem", a serial cable that connects ground to ground, and transmit on each end to receive on the other. The cable that comes with the DOS application LapLink will work fine. It is particularly handy, in fact, because it has both a 9 pin and a 25 pin connector on each end. If you want to check a cable or have one made, here are the required connections:

For two 9 pin connectors, connect pin 2 (receive data) to pin 3, pin 3 (transmit data) to pin 2, and pin 5 (signal ground) to pin 5.

For two 25 pin connectors, connect pin 2 (receive data) to pin 3, pin 3 (transmit data) to pin 2, and pin 7 (signal ground) to pin 7.

For a 9 pin connector (first) to a 25 pin connector (second), connect pin 2 (receive data) to pin 2 (transmit data), pin 3 (transmit data) to pin 3 (receive data), and pin 5 (signal ground) to pin 7 (signal ground).

You may have noted that I have included no connections for the "handshaking" signals. During login, the serial port is handled by the program `agetty`. Recent versions of this program accept a -L switch which tells it not to expect modem control signals. The version in Slackware 3.0 does, but the one on the 3.0 (and earlier) installation root disks does not. However, Pat Volkerding has assured me that the root disks in the next release of Slackware will have the updated version of `agetty`. It is also possible to use the earlier root disks [Emacspeak with Earlier Slackware Releases].

Consult the documentation on your CDROM, or downloaded from an FTP site, and choose a boot disk with the proper kernel features for your hardware (IDE or SCSI, CDROM driver, etc.). I have the InfoMagic September 1996 "Developer's Resource" set of six CDROMs. Slackware 3.1 is on disk 1 of that set, mostly in the two directories slackwar and slakware. (Note the difference in spelling. You will access them in alphabetical order: first slackwar, then slakware.)

Documentation on the boot floppies is in \bootdsks.144\which.one. A copy of the DOS program for writing b images to a floppy, rawrite.exe, is in the same directory. Assuming the CDROM is the M drive under DOS, one might use these commands to write to a floppy disk in the A drive:

```
C>m:
M>cd \bootdsks.144
M>rawrite scsinet.s a:
```

Similarly, to write the "text" root disk:

```
C>m:
M>cd \rootdsks
M>rawrite text.gz a:
```

If you install from floppies, you should also copy the Emacspeak package onto a floppy with a command like this:

```
C>copy m:\contrib\emacspea.tgz a:
```

For the actual installation, proceed as follows: Use the null modem to connect the computer running DOS and equipped with speech output (which I will call the "DOS machine") to the computer into which you want to install Linux (the "Linux machine").

Boot the DOS machine, and start your terminal emulation program. Set it up for 9600 baud, no parity, eight data bits, 1 stop bit.

On the Linux machine, insert the "boot" disk and boot (power up, cntl-alt-del, or hit the reset switch). It should read the disk for five seconds or so, beep, and stop with the following text:

(Note: in the following, the large blocks of text quoted from the installation disks are preceded by "-- begin quote" and followed by "-- end quote". To skip to the end of a quote, you may search for two dashes starting in the first column. I have word wrapped some sections to limit the line lengths.)

```
-- begin quote

    Welcome to the Slackware96 Linux (v. 3.1.0) bootkernel disk!

    If you have any extra parameters to pass to the kernel, enter them at
    the prompt below after one of the valid configuration names (ramdisk,
    mount, drive2)

    Here are some examples (and more can be found in the BOOTING file):

      ramdisk hd=cyl,hds,secs    (Where "cyl", "hds", and "secs" are the
                                 number of cylinders, sectors, and heads
                                 on the drive.  Most machines won't need
                                 this.)

    In a pinch, you can boot your system with a command like:
      mount root=/dev/hda1
```

```
    On machines with low memory, you can use mount root=/dev/fd1 or mount
    root=/dev/fd0 to install without a ramdisk.  See LOWMEM.TXT for
    details.

    If you would rather load the root/install disk from your second
    floppy drive: drive2 (or even this: ramdisk root=/dev/fd1)

    DON'T SWITCH ANY DISKS YET!  This prompt is just for entering extra
    parameters.  If you don't need to enter any parameters, hit ENTER to
    continue.

    boot:
```

-- end quote

I have almost always been able to just hit "enter" at this point.

After your entry, the Linux machine should read the floppy for another twenty seconds or so, then boot the kernel. The first thing it prints is "Loading ramdisk...", which is somewhat misleading. In this case, "ramdisk" is actually the name of the kernel configuration.

Each device driver in the kernel displays a line or two. The particular disk I'm using (the "bare.i" bootdisk) displays more than one screen's worth. It is possible to type shift-page up to scroll the text back. On my machine, the boot messages are as follows:

-- begin quote

```
    Loading ramdisk.....
    Uncompressing Linux...done.
    Now booting the kernel
    Console: colour VGA+ 80x25, 1 virtual console (max 63)
    Calibrating delay loop.. ok - 35.94 BogoMIPS
    Memory: 23028k/24768k available (688k kernel code, 384k reserved,
     668k data)
    Swansea University Computer Society NET3.035 for Linux 2.0
    NET3: Unix domain sockets 0.12 for Linux NET3.035.
    Swansea University Computer Society TCP/IP for NET3.034
    IP Protocols: ICMP, UDP, TCP
    VFS: Diskquotas version dquot_5.6.0 initialized
    Checking 386/387 coupling... Ok, fpu using exception 16 error reporting.
    Checking 'hlt' instruction... Ok.
    Linux version 2.0.0 (root@darkstar) (gcc version 2.7.2) #1 Mon Jun 10
    21:11:56 CDT 1996
    Serial driver version 4.13 with no serial options enabled
    tty00 at 0x03f8 (irq = 4) is a 16550A
    PS/2 auxiliary pointing device detected -- driver installed.
    Ramdisk driver initialized : 16 ramdisks of 49152K size
    hda: IBM-DBOA-2720, 689MB w/64KB Cache, LBA, CHS=700/32/63
    ide0: at 0x1f0-0x1f7,0x3f6 on irq 14
```

```
    Floppy drive(s): fd0 is 1.44M
    Started kswapd v 1.4.2.2
    FDC 0 is a 8272A
    Partition check:
      hda: hda1 hda2 hda3
    VFS: Insert root floppy disk to be loaded into ramdisk and press ENTER
```

-- end quote

Some messages will of course be different on a machine with different hardware. Now, insert the "text" rootdisk and press ENTER. After it is read, the following is displayed on the console:

-- begin quote

```
    RAMDISK: Compressed image found at block 0
    JAVA Binary support v1.01 for Linux 1.3.98 (C)1996 Brian A. Lantz
    VFS: Mounted root (minix filesystem).
    INIT: version 2.60 booting
    none on /proc type proc (rw)
    INIT: Entering runlevel: 4


    Welcome to the Slackware Linux installation disk ,version 3.1.0-text!
    ### READ THE INSTRUCTIONS BELOW CAREFULLY! ###

    You will need one or more partitions of type "Linux native"
    prepared. It is also recommended that you create a swap partition
    (type "Linux swap") prior to installation. Most users can use the
    Linux "fdisk" utility to create and tag the types of all these
    partitions. OS/2 Boot Manager users, however, should create their
    Linux partitions with OS/2 "fdisk", add the bootable (root) partition
    to the Boot Manager menu, and then use the Linux "fdisk" to tag the
    partitions as type "Linux native".

    If you have 4 megabytes or less of RAM, you MUST ACTIVATE a swap
    partition before running setup. After making the partition with fdisk,
    use:

    mkswap /dev/<partition> <number of blocks> ; swapon /dev/<partition>

    Once you have prepared the disk partitions for Linux, type "setup" to
    begin the installation process.

    You may now login as "root".

    slackware login:
```

-- end quote

The program that prints the login prompt is called `agetty`. The Slackware 3.1 root disks are set up to allow logins only from the computer's own keyboard. You will have to reconfigure it to also allow logins from a serial port. This requires typing four lines on the Linux machine keyboard, with no voice feedback. If you realize you have made a mistake before hitting the carriage return, you can erase it with the backspace key. You can also discard what you have typed on a line with control-C. Here is what you type:

```
root
cat >>/etc/inittab
s1:45:respawn:/sbin/agetty 9600 ttyS0
control-D
init q
```

I will repeat that with explanations of what is going on.

First, type "root" and a single carriage return to log in (no password is needed). Next, you need to append one line to `/etc/inittab`. Type the following two lines:

```
cat >>/etc/inittab
s1:45:respawn:/sbin/agetty 9600 ttyS0
```

Finish each line with the "enter" key. Then type a control-D, which signals end of file to a Unix program. (Note: In the second line, the next to last character is an upper case "S". Everything else is in lower case.) This adds a line to the configuration file of the program `init`, to instruct it to use `agetty` to watch for logins on the first serial port on the Linux machine, called "COM1" under DOS, or "/dev/ttyS0" under Linux. To use the second port instead, change the last item on the above line to "ttyS1".

Then type

```
init q
```

which causes `init` to reread `/etc/inittab`. At this point the DOS machine should display the login prompt (the third of the blocks of text quoted above). On the DOS machine, type `root`, and finish the installation. (The next thing you should do is create and enable a swap partition.)

If you don't get the Slackware installation disk prompt, try the following:

- Type a single carriage return on the DOS machine.
- Recheck the terminal setup (9600 baud, no parity, eight data bits, 1 stop bit)
- Disconnect the null modem from the DOS machine. In its place, connect a modem which supports the Hayes "AT" commands. Type `AT` and a carriage return. You should get a reply of "OK" from the modem.

Once you get the above prompt on the DOS machine, you may type `root` and a carriage return to log in, and complete the installation like any other user. Of course, you must remember to include these packages: emacs, tcl, and tclX.

The installation script will offer to prepare a boot floppy. You should do this, since it is the most foolproof way to boot Linux. You will probably also want to install `lilo` (which is an abbreviation for "Linux loader") and/or `loadlin` (which is an abbreviation for "load Linux"). The installation script can install lilo. Loadlin is a DOS program that will let you boot from DOS to Linux. Install it on a DOS partition, and copy a compressed kernel file (usually named `zImage`) to the same partition. While running DOS, you may boot Linux with a command like `loadlin zimage root=/dev/hda3 ro/`. (I have assumed here that the kernel image is in the same directory as the loadlin program. You may find it more convenient to store kernel images in subdirectories named

for the kernel version.)

After the Slackware setup script finishes the main installation, it will tell you to restart by pressing cntl-alt-del. Before doing that, you should install emacspeak. It can be found with the other "contributed" software. In the InfoMagic set, it is in slackwar/contrib. Assuming you are installing Linux directly from a CDROM, the setup script will mount the CDROM under /CDROM, and you may install emacspeak with the following command:

```
# installpkg /CDROM/slackwar/contrib/emacspeak.tgz
```

If you install from floppies, insert the floppy you made earlier and type this:

```
# mount -tmsdos /dev/fd0 /floppy
# cp /floppy/emacspea.tgz /tmp/emacspeak.tgz
# installpkg /tmp/emacspeak.tgz
```

You should not install the package directly off the floppy disk, because the DOS filesystem will not allow the full filename, so the installpkg program will think the package name is "emacspea" and will store its records under that name.

If you have a DoubleTalk or LiteTalk speech synthesizer, you should also install the emacspeak-dt package.

Reboot the Linux machine with the new boot floppy, with the DOS machine still connected. You should get a login prompt on the DOS machine. Celebrate! After getting this system working, you need to learn emacs (third option) and Unix system administration.

4.2 Learning Unix System Administration

Mostly you will learn system administration as the need arises. First adding a user (yourself), then installing programs, and so forth. The exception to this is making backups, which you should learn **before** you need them.

Among the many programs you will need to learn are these:

adduser

Register a new user, including creating a home directory and adding an entry in /etc/passwd.

tar

Create and unpack `.tar` files, which are collections of files (something like `.zip` files). To list the contents of an archive, use `tar -tf foobar.tar`. For a more verbose listing, use `tar -tvf foobar.tar`. To unpack an archive, use `tar -xf foobar.tar`.

chmod

Change permissions of a file or directory.

chown

Change ownership of a file or directory.

find

Search directories recursively. For example, the command `find . -name '*alpha*' -print` means: search starting in the current directory (`.`) for a file whose name contains the string "alpha" (`-name '*alpha*'`), and print its path and name (`-print`). (With GNU find, the `-print` is optional.)

du

Display the amount of space occupied by files or subdirectories. For a file with "holes", this may be m
less than the length of the file.

df

Display filesystem capacities, free space, and where they are mounted.

mount

Display filesystems, where they are mounted, and the mount flags.

ifconfig

Configure and check internet protocol (IP) network interfaces, including Ethernet cards, SLIP links, and PLIP links.

route

Configure and check IP network routing, after the interface is configured.

ping

Check IP network connectivity, after the interfaces and routes are configured.

ftp

Transfer files across the Internet.

Here are some programs you may want to install:

agrep

Approximate grep searches for approximate, not exact, string matches (also called "fuzzy string searches").

archie

Search Internet archives for files.

flip

Convert text files between Unix and DOS formats.

glimpse

Fuzzy string searches in large collection of files (uses agrep).

lynx

Text mode web browser.

Here are some Web pages related to Unix system administration:

General information http://www.ensta.fr/internet/unix/sys_admin/ or http://www.sai.msu.su/sysadm.html

There is a Unix system administration tutorial at http://www.iem.ac.ru/sysadm.html

UnixWorld Online Magazine Home Page http://www.wcmh.com/uworld/

Internet Essentials for UNIX System Administrators Tutorial http://www.greatcircle.com/tutorials/ieusa.html

Pointers to Unix goodies available on the Internet http://www.ensta.fr/internet/unix/

Pointers to Unix system administration "goodies" available on the Internet

http://www.ensta.fr/internet/unix/sys_admin/

5. Stage 4. Emacspeak under Linux

The Slackware setup script for Emacspeak should create the needed environment variables and install a script `emacspeak` that starts emacs with emacspeak. This is your fourth option for learning Emacs. This is the first time you will be able to actually use Emacspeak. A short tutorial appears below. Within Emacs, you may type C-h C-e to get a list of the commands. To search for a command, use C-h a. To get an explanation for a key sequence, use C-h k. There is also an info file which is part of the Emacspeak distribution. Within emacs, you may type C-h i to open the directory to the info pages. Search for the emacspeak menu item by typing C-s emacspeak, then two carriage returns (one to terminate the search, and a second one to go to the info page. If you have the standalone info program installed, you can consult the info file with the command `info Emacspeak`.

5.1 Emacspeak Introduction - Speech Enabled Normal Commands

All of the normal Emacs movement commands will speak the relevant information after moving. Here are some of the cursor movement functions that have been speech enabled. Note that this list only enumerates a few of these speech enabled commands; the purpose of emacspeak is to speech-enable all of emacs and provide you spoken feedback as you work. Thus, this list is here only as a representative example of the kind of speech-enabling extensions Emacspeak provides.

`C-n' or `M-x next-line' or `down'

Moves the cursor to the next line and speaks it.

`C-p' or `M-x previous-line' or `up'

Moves the cursor to the previous line and speaks it.

`M-f' or `M-x forward-word' or

Moves the cursor to the next word and speaks it. Places point on the first character of the next work, rather than on the space preceding it (This is my personal preference).

`M-b' or `M-x backward-word'

Moves the cursor to the previous word and speaks it.

`M-C-b' or `M-x backward-sexp'

Moves the cursor to the previous sexp and speaks it. If the sexp spans more than a line, only the first line is spoken.

`M-<' or `M-x beginning-of-buffer'

Speaks line moved to.

`M->' or `M-x end-of-buffer'

Speaks line moved to.

`M-m' or `M-x back-to-indentation'

Speaks entire current line. A useful way of hearing the current line.

5.2 Emacspeak Introduction - New Commands

Emacspeak provides a number of commands for reading portions of the current buffer, getting status information, and modifying Emacspeak's state.

All of the commands are documented in the subsequent sections. They can be classified into types:

The first group of commands are for listening to chunks of information. The names of these commands all start with the common prefix `emacspeak-'. All Emacspeak commands are bound to the keymap EMACSPEAK-KEYMAP and are accessed with the key `Control e'. Thus, the Emacspeak command "emacspeak-speak-line" is bound to `l' in keymap EMACSPEAK-KEYMAP and can be accessed with the keystroke `Control-e l'.

Here are some of the commands for reading text:

`C-e c' or `M-x emacspeak-speak-char'

Speak current character, using the phonetic alphabet.

`C-e w' or `M-x emacspeak-speak-word'

Speak current word.

`C-e l' or `M-x emacspeak-speak-line'

Speak current line. With prefix `C-u', speaks the rest of the line from point. With negative prefix `C-u -', speaks from start of line to point. Voicifies if voice-lock-mode is on. Indicates indentation with a tone if audio indentation is in use. Indicates position of point with an aural highlight if option emacspeak-show-point is turned on --see command `M-x emacspeak-show-point'.

`C-e .' or `M-x emacspeak-speak-sentence'

Speak the current sentence.

The second category of commands provided by Emacspeak report status of various kinds.

`C-e k' or `M-x emacspeak-speak-current-kill'

Speak the current kill entry (which would be yanked by the next C-y).

`C-e =' or `M-x emacspeak-speak-current-column'

State the column where point is.

`C-e C-l' or `M-x emacspeak-speak-line-number'

State the line where point is.

`C-e m' or `M-x emacspeak-speak-line-number'

Speak the mode-line: the name of the buffer, how far point is into the current buffer as a percentage, and the major mode.

`C-e M' or `M-x emacspeak-speak-minor-mode-line'

TOC

Speak the minor modes that are in effect.

`C-e v' or `M-x emacspeak-speak-version'

Announce the emacspeak version.

The third category of commands provided by Emacspeak manipulate the state of the speech device. The names of these commands start with the common prefix `dtk-'. Many of these commands take the prefix `Control-e d'. Thus, the command "dtk-set-rate" is bound to `r' in keymap EMACSPEAK-DTK-SUBMAP and can be executed by pressing `Control e d r'.

`C-e s' or `M-x dtk-stop'

Stop speech now. In addition, any command that causes speech output will discard anything in the speech buffer.

`C-e d V' or `M-x emacspeak-dtk-speak-version'

Use this to find out which version of the Dectalk firmware you have.

`C-e d I' or `M-x dtk-toggle-stop-immediately-while-typing'

Toggle state of variable dtk-stop-immediately-while-typing. As the name implies, if true then speech flushes immediately as you type.

`C-e d i' or `M-x emacspeak-toggle-audio-indentation'

Toggle state of Emacspeak audio indentation. Specifying the method of indentation as `tone' results in the DECtalk producing a tone whose length is a function of the line's indentation. Specifying `speak' results in the number of initial spaces being spoken.

`C-e d k' or `M-x emacspeak-toggle-character-echo'

Toggle state of Emacspeak character echo (that is, whether typed characters are echoed).

`C-e d w' or `M-x emacspeak-toggle-word-echo'

Toggle state of Emacspeak word echo (initially on).

`C-e d l' or `M-x emacspeak-toggle-line-echo'

Toggle state of Emacspeak line echo (that is, whether typed text is echoed after typing enter).

`C-e d p' or `M-x dtk-set-punctuations'

Set punctuation state. Possible values are `some', `all', or `none'.

`C-e d q' or `M-x dtk-toggle-quiet'

Toggle state of the speech device between being quiet and talkative. Useful if you want to continue using an emacs session that has emacspeak loaded but wish to make the speech shut up.

`C-e d R' or `M-x dtk-reset-state'

Restore sanity to the Dectalk. Typically used after the Dectalk has been power cycled.

`C-e d SPC' or `M-x dtk-toggle-splitting-on-white-space'

Toggle state of emacspeak that decides if we split text purely by clause boundaries, or also include whitespace.

`C-e d r' or `M-x dtk-set-rate'

TOC

Set speaking rate for the dectalk.

`C-e d s' or `M-x dtk-toggle-split-caps'

Toggle split caps mode. In split caps mode, a transition from lower case to upper case is treated like the beginning of a new word. This is useful when reading Hungarian notation in program source code.

`C-e d v' or `M-x voice-lock-mode'

Toggle Voice Lock mode (initially off). When Voice Lock mode is enabled, text is voiceified as you type it, as follows:

- Comments are spoken in `voice-lock-comment-personality`; (That is a variable whose value should be a personality name.)
- Strings are spoken in `voice-lock-string-personality`.
- Documentation strings are spoken in `voice-lock-doc-string-personality`.
- Function and variable names in their defining forms are spoken in `voice-lock-function-name-personality`.
- Certain other expressions are spoken in other personalities according to the value of the variable `voice-lock-keywords`.

5.3 Emacspeak Introduction - Using the Help System

When you press C-h to get the help index, the screen will appear, but Emacspeak will not speak the window. The only thing spoken is "Type one of the options listed or Space to scroll:".

Here is the menu that Emacspeak is not speaking:

```
-- begin quote


    You have typed C-h, the help character.  Type a Help option:
    (Use SPC or DEL to scroll through this text.  Type q to exit the Help
    command.)

    a  command-apropos.  Give a substring, and see a list of commands
            (functions interactively callable) that contain
            that substring.  See also the  apropos  command.
    b  describe-bindings.  Display table of all key bindings.
    c  describe-key-briefly.  Type a command key sequence;
            it prints the function name that sequence runs.
    f  describe-function.  Type a function name and get documentation of
    it.
    C-f Info-goto-emacs-command-node.  Type a function name;
            it takes you to the Info node for that command.
    F  view-emacs-FAQ.  Shows emacs frequently asked questions file.
    i  info. The  info  documentation reader.
    k  describe-key.  Type a command key sequence;
```

```
        it displays the full documentation.
C-k Info-goto-emacs-key-command-node.  Type a command key sequence;
        it takes you to the Info node for the command bound to that
key.
l  view-lossage.  Shows last 100 characters you typed.
m  describe-mode.  Print documentation of current major mode,
        which describes the commands peculiar to it.
n  view-emacs-news.  Shows emacs news file.
p  finder-by-keyword. Find packages matching a given topic keyword.
s  describe-syntax.  Display contents of syntax table, plus
explanations
t  help-with-tutorial.  Select the Emacs learn-by-doing tutorial.
v  describe-variable.  Type name of a variable;
        it displays the variable's documentation and value.
w  where-is.  Type command name; it prints which keystrokes
        invoke that command.
C-c print Emacs copying permission (General Public License).
C-d print Emacs ordering information.
C-n print news of recent Emacs changes.
C-p print information about the GNU project.
C-w print information on absence of warranty for GNU Emacs.

-- end quote
```

Suppose you type "a", for command-apropos.

The next spoken prompt is "Apropos command (regexp):"

Now you type some word you think is part of an emacs command, like "visit".

The help system will display the first section of the help, but will leave the cursor in the other window. The spoken text is "Type C-x 1 to remove help window. M-C-v to scroll the help." At this point, I think it's more helpful to move point to the other window with C-x o, then you can use regular navigation commands to speak the help text. You can delete the help window with C-x 0, which will also put point back where it was.

The complete menu displayed by help-for-help is also visible if you do a describe function on help-for-help. In a future version of Emacspeak, Raman plans to add a message to that effect when the user presses C-h ?

6. Footnotes and References

6.1 JAWS

Job Access With Speech (JAWS) is a screen reader which runs under Microsoft MSDOS. JAWS for Windows supports Windows 95, Windows 98 and Windows NT. Both are products of Henter-Joyce, Inc., 2100 62nd Avenue North, St. Petersburg, FL 33702, telephone: 800-336-5658. For details, see http://www.hj.com.

6.2 TELIX

TELIX is a shareware terminal emulator for MSDOS. It can be obtained by FTP from the SimTel archive [SimTel]. Within the SimTel collection, look for directory msdos/telix. For example, try ftp://ftp.coast.net/pub/SimTel/msdos/telix. The latest version of the program itself is in the four files tlx322-1.zip, tlx322-2.zip, tlx322-3.zip, and tlx322-4.zip.

6.3 COMMO

COMMO is another shareware terminal emulator for DOS. In the SimTel archive [SimTel], it is in directory msdos/commprog, file `commo66.zip`. For example, try ftp://ftp.coast.net/pub/SimTel/msdos/commprog/commo66.zip.

6.4 SimTel

The SimTel archive is maintained by Keith Petersen `w8sdz@Simtel.Net`. CD-ROM copies of Simtel.Net collections are available from Walnut Creek CDROM [Walnut Creek]. The primary ftp sites are ftp://ftp.simtel.net/pub/simtelnet, and oak.oakland.edu://pub/simtelnet. There are many mirror sites, as listed in the following table:

- US, ALL (primary) ftp://ftp.simtel.net/pub/simtelnet
- US, California ftp://ftp.cdrom.com/pub/simtelnet
- US, California ftp://ftp.digital.com/pub/micro/pc/simtelnet
- US, California ftp://ftp.lib.sonoma.edu/pub/simtelnet
- US, Illinois ftp://uiarchive.cso.uiuc.edu/pub/systems/pc/simtelnet
- US, Massachusetts ftp://ftp.bu.edu/pub/mirrors/simtelnet
- US, Michigan ftp://oak.oakland.edu/pub/simtelnet

TOC

- US, New York ftp://ftp.rge.com/pub/systems/simtelnet
- US, Oklahoma ftp://ftp.ou.edu/pub/simtelnet
- US, Oregon ftp://ftp.orst.edu/pub/simtelnet
- US, Pennsylvania ftp://ftp.epix.net/pub/simtelnet
- US, Utah ftp://ftp.cyber-naut.com/pub/simtelnet
- US, Virginia ftp://mirrors.aol.com/pub/simtelnet
- Argentina ftp://ftp.satlink.com/pub/mirrors/simtelnet
- Australia ftp://ftp.iniaccess.net.au/pub/simtelnet
- Australia ftp://sunsite.anu.edu.au/pub/pc/simtelnet
- Austria, Vienna ftp://ftp.univie.ac.at/mirror/simtelnet
- Belgium ftp://ftp.linkline.be/mirror/simtelnet
- Belgium ftp://ftp.tornado.be/pub/simtelnet
- Bulgaria ftp://ftp.eunet.bg/pub/simtelnet
- Brazil ftp://ftp.iis.com.br/pub/simtelnet
- Brazil ftp://ftp.unicamp.br/pub/simtelnet
- Canada, Ottawa ftp://ftp.crc.doc.ca/systems/ibmpc/simtelnet
- Canada, Vancouver ftp://ftp.direct.ca/pub/simtelnet
- Chile ftp://sunsite.dcc.uchile.cl/pub/Mirror/simtelnet
- China ftp://ftp.pku.edu.cn/pub/simtelnet
- Czech Republic ftp://ftp.eunet.cz/pub/simtelnet
- Czech Republic ftp://ftp.zcu.cz/pub/simtelnet
- Czech Republic ftp://pub.vse.cz/pub/simtelnet
- Finland ftp://ftp.funet.fi/mirrors/ftp.simtel.net/pub/simtelnet
- France ftp://ftp.grolier.fr/pub/simtelnet
- France ftp://ftp.ibp.fr/pub/simtelnet
- Germany ftp://ftp.mpi-sb.mpg.de/pub/simtelnet
- Germany ftp://ftp.rz.ruhr-uni-bochum.de/pub/simtelnet
- Germany ftp://ftp.tu-chemnitz.de/pub/simtelnet
- Germany ftp://ftp.uni-heidelberg.de/pub/simtelnet
- Germany ftp://ftp.uni-magdeburg.de/pub/mirrors/simtelnet
- Germany ftp://ftp.uni-paderborn.de/pub/simtelnet
- Germany ftp://ftp.uni-trier.de/pub/pc/mirrors/Simtel.net
- Germany ftp://ftp.rz.uni-wuerzburg.de/pub/pc/simtelnet

TOC

- Greece ftp://ftp.ntua.gr/pub/pc/simtelnet
- Hong Kong ftp://ftp.cs.cuhk.hk/pub/simtelnet
- Hong Kong ftp://ftp.hkstar.com/pub/simtelnet
- Hong Kong ftp://sunsite.ust.hk/pub/simtelnet
- Ireland ftp://ftp.iol.ie/pub/simtelnet
- Israel ftp://ftp.huji.ac.il/pub/simtelnet
- Italy ftp://cis.utovrm.it/simtelnet
- Italy ftp://ftp.flashnet.it/pub/simtelnet
- Italy ftp://ftp.unina.it/pub/simtelnet
- Italy ftp://mcftp.mclink.it/pub/simtelnet
- Japan ftp://ftp.iij.ad.jp/pub/simtelnet
- Japan ftp://ftp.riken.go.jp/pub/simtelnet
- Japan ftp://ftp.saitama-u.ac.jp/pub/simtelnet
- Japan ftp://ftp.u-aizu.ac.jp/pub/PC/simtelnet
- Japan ftp://ftp.web.ad.jp/pub/simtelnet
- Japan ftp://ring.aist.go.jp/pub/simtelnet
- Japan ftp://ring.asahi-net.or.jp/pub/simtelnet
- Latvia ftp://ftp.lanet.lv/pub/mirror/simtelnet
- Malaysia ftp://ftp.jaring.my/pub/simtelnet
- Malaysia ftp://ftp.mimos.my/pub/simtelnet
- Mexico ftp://ftp.gdl.iteso.mx/pub/simtelnet
- Netherlands ftp://ftp.euro.net/d5/simtelnet
- Netherlands ftp://ftp.nic.surfnet.nl/mirror-archive/software/simtelnet
- New Zealand ftp://ftp.vuw.ac.nz/pub/simtelnet
- Norway ftp://ftp.bitcon.no/pub/simtelnet
- Poland ftp://ftp.cyf-kr.edu.pl/pub/mirror/Simtel.Net
- Poland ftp://ftp.icm.edu.pl/pub/simtelnet
- Poland ftp://ftp.man.poznan.pl/pub/simtelnet
- Portugal ftp://ftp.ip.pt/pub/simtelnet
- Portugal ftp://ftp.ua.pt/pub/simtelnet
- Romania ftp://ftp.sorostm.ro/pub/simtelnet
- Singapore ftp://ftp.nus.sg/pub/simtelnet
- Slovakia ftp://ftp.uakom.sk/pub/simtelnet

- Slovenia ftp://ftp.arnes.si/software/simtelnet
- South Africa ftp://ftp.is.co.za/pub/simtelnet
- South Africa ftp://ftp.sun.ac.za/pub/simtelnet
- South Korea ftp://ftp.nuri.net/pub/simtelnet
- South Korea ftp://ftp.sogang.ac.kr/pub/simtelnet
- South Korea ftp://sunsite.snu.ac.kr/pub/simtelnet
- Spain ftp://ftp.rediris.es/mirror/simtelnet
- Sweden ftp://ftp.sunet.se/pub/simtelnet
- Switzerland ftp://sunsite.cnlab-switch.ch/mirror/simtelnet
- Taiwan ftp://ftp.ncu.edu.tw/Packages/simtelnet
- Taiwan ftp://nctuccca.edu.tw/mirror/simtelnet
- Thailand ftp://ftp.nectec.or.th/pub/mirrors/simtelnet
- UK, Edinburgh ftp://emwac.ed.ac.uk/mirrors/simtelnet
- UK, London ftp://ftp.demon.co.uk/pub/simtelnet
- UK, Lancaster ftp://micros.hensa.ac.uk/pub/simtelnet
- UK, London ftp://sunsite.doc.ic.ac.uk/packages/simtelnet

6.5 InfoMagic

Simtel Ste D-InfoMagic is at 11950 N. Highway 89, Flagstaff AZ 86004, telephone 800-800-6613 or 520-526-9565, fax 520-526-9573, email: info@infomagic.com, web: http://www.infomagic.com.

6.6 Walnut Creek

Walnut Creek CDROM has many useful CDROMs. They are at 4041 Pike , Concord, CA 94520, USA. Telephone (800) 786-9907 or (510) 674-0783, or FAX (510) 674-0821. email: orders@cdrom.com. Web: http://www.cdrom.com/

6.7 Red Hat

Red Hat Software: telephone 800-454-5502 or 203-454-5500, fax: 203-454-2582, email: sales@redhat.com. Web: http://www.redhat.com.

TOC

6.8 Craftwork

CraftWork Solutions, 4320 Stevens Creek Blvd, Suite 170, San Jose CA 95129, telephone 800-985-1878, email: info@craftwork.com, web: http://www.craftwork.com.

6.9 Yggdrasil

Yggdrasil Computing, 4880 Stevens Creek Blvd., Suite 205, San Jose CA 95129-1024, telephone 800-261-6630 or 408-261-6630, fax: 408-261-6631, email: info@yggdrasil.com, web: http://www.yggdrasil.com.

6.10 Emacs for DOS

From the Emacs FAQ of November 11, 1996:

`--begin quote`

93: Where can I get Emacs for my PC running MS-DOS?

A pre-built binary distribution of Emacs 19.34 should be available by the beginning of November 1996 from the Simtel archives, the main site of which is at

ftp://ftp.simtel.net/pub/simtelnet/gnu/djgpp/v2gnu/

If you prefer to compile Emacs for yourself, you will need a 386 (or better) processor, and are running MS-DOS 3.0 or later. According to Eli Zaretskii `eliz@is.elta.co.il` and Darrel Hankerson `hankedr@dms.auburn.edu`, you will need the following:

Compiler: djgpp version 1.12 maint 1 or later. Djgpp 2.0 or later is recommended, since 1.x is being phased out. Djgpp 2 supports long filenames under Windows 95.

You can get the latest release of djgpp by retrieving all of the files in

ftp://ftp.simtel.net/pub/simtelnet/gnu/djgpp

Gunzip and tar:

The easiest way is to use "djtar" which comes with djgpp v2.x, because it can open gzip'ed tarfiles (i.e., those ending with ".tar.gz") in one step. Djtar comes in "djdev201.zip", from the URL mentioned above.

Utilities: make, mv, sed, rm.

All of these utilities are available at

ftp://ftp.simtel.net/pub/simtelnet/gnu/djgpp/v2gnu

16-bit utilities can be found in GNUish:

ftp://ftp.simtel.net/pub/simtelnet/gnu/gnuish

The file INSTALL in the top-level directory of the Emacs source contains some additional information regarding Emacs under MS-DOS. In addition, the file etc/MSDOS contains some information on the differences between the Unix and MS-DOS versions of Emacs.

For the most comprehensive information on running GNU Emacs on a PC, see the file prepared by Michael Ernst `mernst@theory.lcs.mit.edu` at

ftp://theory.lcs.mit.edu/pub/emacs/pc-emacs.gz

For a list of other MS-DOS implementations of Emacs (and Emacs look-alikes), consult the list of "Emacs implementations and literature," available at

ftp://rtfm.mit.edu/pub/usenet/comp.emacs/

Note that while many of these programs look similar to Emacs, they often lack certain features, such as the Emacs Lisp extension language.

```
--end quote
```

6.11 GNU Mirror Sites

The GNU collection at ftp://prep.ai.mit.edu/pub/gnu is mirrored at many sites. Mirrors in USA include these:

- ftp://labrea.stanford.edu/pub/gnu
- ftp://wuarchive.wustl.edu/systems/gnu
- ftp://ftp.kpc.com/pub/mirror/gnu
- ftp://f.ms.uky.edu/pub3/gnu
- ftp://jaguar.utah.edu/gnustuff
- ftp://ftp.hawaii.edu/mirrors/gnu
- ftp://uiarchive.cso.uiuc.edu/gnu
- ftp://uiarchive.cso.uiuc.edu/pub/gnu
- ftp://ftp.cs.columbia.edu/archives/gnu/prep
- ftp://gatekeeper.dec.com/pub/GNU
- ftp://ftp.uu.net/systems/gnu

6.12 Emacspeak with Earlier Slackware Releases

If you want to install Slackware 3.0 or earlier, you will need to prepare a full null modem cable, including modem control signals.

For two DB25 (25 pin) connectors, the required connections are:

- 1 (Frame Ground) - 1 (Frame Ground)
- 2 (Receive Data) - 3 (Transmit Data)
- 3 (Transmit Data) - 2 (Receive Data)
- 4 (Request To Send) - 5 (Clear To Send)
- 5 (Clear To Send) - 4 (Request To Send)
- 6 (Data Set Ready) - 20 (Data Terminal Ready)
- 7 (Signal Ground) - 7 (Signal Ground)
- 8 (Carrier Detect) - 20 (Data Terminal Ready)
- 20 (Data Terminal Ready) - 6 (Data Set Ready)
- 20 (Data Terminal Ready) - 8 (Carrier Detect)

For two DB9 connectors, the connections are:

- 1 (Carrier Detect) - 4 (Data Terminal Ready)
- 2 (Receive Data) - 3 (Transmit Data)
- 3 (Transmit Data) - 2 (Receive Data)
- 4 (Data Terminal Ready) - 6 (Data Set Ready)
- 4 (Data Terminal Ready) - 1 (Carrier Detect)
- 5 (Signal Ground) - 5 (Signal Ground)
- 6 (Data Set Ready) - 4 (Data Terminal Ready)
- 7 (Request To Send) - 8 (Clear To Send)
- 8 (Clear To Send) - 7 (Request To Send)
- 9 (Ring Indicator) not connected

For a DB9 (listed first) to a DB25 (second), the connections are:

- 1 (Carrier Detect) - 20 (Data Terminal Ready)
- 2 (Receive Data) - 2 (Transmit Data)
- 3 (Transmit Data) - 3 (Receive Data)
- 4 (Data Terminal Ready) - 6 (Data Set Ready)
- 4 (Data Terminal Ready) - 8 (Carrier Detect)
- 5 (Signal Ground) - 7 (Signal Ground)
- 6 (Data Set Ready) - 20(Data Terminal Ready)
- 7 (Request To Send) - 5 (Clear To Send)
- 8 (Clear To Send) - 4 (Request To Send)
- 9 (Ring Indicator) not connected

7. Frequently Asked Questions (FAQ)

7.1 Why does it say "space" after each character?

Your DECtalk Express has old firmware. Use the Emacspeak command `C-e d V' to find out your version. You should be running a version no older than 4.2bw from March 1995. If you have an earlier version, you can find an updated version at http://www.ultranet.com/~rongemma/tips_upd.htm, a WWW site maintained by Ron Jemma of the Dectalk Group at DEC. Alternatively, you can send email to Anne Nelson at DEC`nelson@dectlk.enet.dec.com`. The most recent version at this writing is 4.3 release AA X01 May 20 1996.

7.2 Why the high pitched tone from a DECtalk?

On occasion when reading the DECtalk will produce high pitch tones that last for several words or more, if this happens in a buffer it will often repeat within the same buffer.

The problem is due to remaining bugs in the Dectalk firmware. When emacspeak produces tones, especially when split caps is on, the dtk sometime goes into squealing mode.

If you notice this happening in particular text documents, just turn off split caps mode locally with `C-e d s'.

7.3 Why the "No library autorevert in search path" message?

What is the significance of the message "No library autorevert in search path" which is spoken (or at least queued for speech) when emacspeak 7.0 starts?

It's completely insignificant. autorevert is a new minor mode provided by emacs 20 and emacspeak looks for it and if found speech-enables it.

7.4 How do I change Emacspeak defaults?

What is the best way to change Emacspeak defaults for things such as speech rate, default punctuation mode, etc.?

Just put the needed changes in your .emacs file. ".emacs" is a file in your home directory that is loaded by emacs when it starts (shortly after reading emacspeak). It is the place to do personal modifications.

Here is an example which resets the speech rate and redefines a function (courtesy of Greg

Priest-Dorman priestdo@cs.vassar.edu):

```
;;
(dtk-set-rate 380 1)
;;
(defun emacspeak-speak-time ()
  "Speak the time. "
  (interactive)
  (dtk-speak  (format-time-string "%A %B %e, %I %M %p"
      (current-time) )))
;;
```

7.5 What if an error message is not spoken?

An error message was displayed on the screen, but not reported by Emacspeak

In general, if emacs hits an error and you don't get to hear the message because it went by too quickly, here is what you should do:

1) Switch to the *Messages* buffer and see if you see the message there. 99% of the time it will be there.

2) If some portion of emacs or an emacs application throws an error without leaving an appropriate message in the *Messages* buffer, then turn on emacs' debugging facility by `M-x set-variable ret' `debug-on-error ret' `t ret' this sets debugon-on-error to true now repeat what caused the error you will be placed in a backtrace buffer.

Warning: do not read the entire contents of the backtrace buffer at one shot with emacspeak. Go line by line. (courtesy of T. V. Raman `raman@Adobe.COM`)

7.6 The terminal emulator is the most problematic portion of Emacspeak.

It's not an emacspeak problem-- it's a terminal problem. Getting a terminal emulator working correctly inside emacs is not a joke-- and eterm is an excellent piece of work, but depending on how some old app behaves you will (or should always expect to) see some weirdness.

After all, remember all of these UNIX terminal apps work from all kinds of terminals ranging from vt52's to new devices.

7.7 In a term buffer, the first half screen is not spoken

When starting a term buffer, no output is spoken in line mode by Emacspeak until the cursor has made it about half way down the screen

I've not seen this-- and I am running emacs 20.

When you start term it is in character mode. When you switch to line mode make sure the cursor is at the end of the buffer-- otherwise what you send will never go to the shell.

Also, in general use M-x shell (comint-mode) for running things you would run in line mode in eterm-- it's been around longer and far more robust. M-x term specializes in its character-mode of terminal emulation and may not always take care of everything in line mode. (courtesy of T. V. Raman `raman@Adobe.COM`)

7.8 Term mode gets flaky after a long session.

Sometimes, after long sessions with a term buffer, I will experience similar problems as do you with spech in the term becoming very flaky. Output may or may not be spoken (I have not determined the pattern).

if the term becomes flaky, enter character mode and issue a reset. the above is usually a result of running some curses based app that leaves the terminal in a screwy state.

Many curses based apps follow slightly different conventions on where they leave the cursor while you edit-- 60-70% of them leave the cursor just after the character you typed with a null character after point. Others leave a space character there which is what you hear.

Experiment between setting TERM to vt100 and eterm. Depending on whether the app was linked against curses or ncurses (former uses TERMCAP and latter uses TERMINFO) you will get different results.

---

8. Legalese

Copyright (c) 1998 by James R. Van Zandt `jrv@vanzandt.mv.com`. This document may be distributed under the terms set forth in the LDP license at http://metalab.unc.edu/LDP/COPYRIGHT.html.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at `tjbynum@metalab.unc.edu` or `linux-howto@metalab.unc.edu`.

Linux-Esperanto-HOWTO

Kelkaj helpindikoj por uzi Esperanton sub Linukso

Kompilita de Wolfram Diestel (wolfram@steloj.de) helpita de aliaj.

v1.30, August 1999

1. Antauxvortoj kaj dankoj

2. Kodoj

- 2.1 koderoj deksesume
- 2.2 koderoj dekume
- 2.3 koderoj okume

3. Konzolo (teksta moduso)

4. Metamail

5. X-fenestra sistemo

- 5.1 Tiparoj
- 5.2 Tajpi E-signojn

6. KDE

7. Netscape Communicator

TOC

8. Xterm

9. Redaktiloj

- 9.1 Redaktiloj kun Latin-3/Unikod-subteno
- 9.2 Vi (Elvis, Vim)

10. Redakti per Emakso (Emacs)

- 10.1 Emakso 19.34
- 10.2 Emakso 20.* kaj MULE

11. Retposxto per Xfmail 1.3

12. Diskutgrupoj per Knews

13. TeX / LaTeX

14. Esperanto-subteno en internaciigitaj programoj

15. Esperanto-signoj en HTML-tekstoj

---

1. Antauxvortoj kaj dankoj

Cxar tiu cxi teksto estas verkita per linuxdoc-sgml, gxi ne povas uzi esperantajn signojn.

La avantagxo estas, ke la teksto aperas en la HOWTO-kolekto (anstataux en la mini-HOWTO-kolekto) kaj estas havebla en pluraj formoj (teksto, TTT-teksto, Postskript-teksto, k.a.)

La malavantagxo estas, ke mi devas kodi la esperantajn signojn per la x-maniero kaj el tio povus estigxi miskomprenoj cxe la agordekzemploj. Nu, mi provas rimarkigi tion cxe la koncernaj lokoj.

Solvojn kaj gravajn informojn donis: Eduard Werner (Emakso 19.34, TeX), Marko Rauhamaa (xmodmap, TeX, mailcap), Edmund Grimley Evans (TeX, konzolo), Lee Sau Dan (setfont, Emakso 20, MULE, TeX), Richard Hable (Vi, Xfmail, Knews, Netscape), Albert Reiner (TeX) kaj Axel Belinfante (Wily, Sam). Dankon al ili kaj cxiuj aliaj, kiuj helpis kompili tiun cxi dosieron.

Aktuala versio de tiu cxi dokumento trovigxas cxiam sub kielfari.sgml, kielfari.html.

2. Kodoj

Ekzistas tri internaciaj normoj por signokodoj, kiuj enhavas la esperantajn signojn, la du kutime uzataj estas ISO-8859-3 (Latin-3) kaj Unikodo (Unicode). La unua normo estas por okbitaj signoj kaj la dua por deksesbitaj. Por povi transporti unikodajxojn tra okbitaj kaj sepbitaj medioj ekzistas la kodoj UTF-8 kaj UTF-7.

Cxar Unikodo samtempe difinas koderojn por la signoj de cxiuj lingvoj de la mondo, gxi ricevas pli kaj pli da subteno de programarproduktantoj.

Alie ol ekz. Vindozo NT, Linukso interne uzas okbitajn kodojn. Subteno de Unikodo precipe estas en sistemkomponentoj, kiuj zorgas pri eligado de tekstoj sur la ekrano aux kiuj rilatas al intersxangxo de informoj kun la ekstera mondo. Do estas iom da Unikodo en la konzolo, iom da en X-fenestroj, iom da en la traktado de informoj venantaj per retposxto aux el TTT kaj iom da rilate al dosiersistemoj de KKDoj (Joliet-dosiersistemo). Ofte tiuj komponentoj uzas la okbitan unikodformon UTF-8.

Cxar la subteno de Unikodo en Linukso estas iom malkohera, gxenerale oni preferas ISO-8859-3. Sed supozeble la premisoj sxangxigxos gxis la fino de 1999.

Ampleksajn informojn pri la unuopaj kodoj kun E-signoj en internaciaj normoj kaj en specialaj programoj, vi trovas cxe Edmundo.

Foje vi bezonas la eblecon konverti esperantajn tekstojn de UTF-8 al Latin-3. Por tio vi povas uzi la sekvan Perl-programeton.

```
#!/usr/bin/perl
while (<>) {
  s/\304\210/\306/g;
  s/\304\211/\346/g;
  s/\304\234/\330/g;
  s/\304\235/\370/g;
  s/\304\244/\246/g;
  s/\304\245/\266/g;
  s/\304\264/\254/g;
  s/\304\265/\274/g;
  s/\305\234/\336/g;
  s/\305\235/\376/g;
  s/\305\254/\335/g;
  s/\305\255/\375/g;
  print;
};
```

Por konverti en la alia direkto, intersxangxu la du kolumnojn inter /../../.

TOC

Por presi Latin-3ajn tekstojn vi povas uzi solvon de Albert Reiner. Temas pri simpla programeto en] uzanta TeX-on por la presado.

Pro kompleteco tie cxi ni donas ankoraux la koderojn de la E-signoj laux la plej vastigxintaj normoj. UTF-8, kiu kodas unikodajn signojn per okbitaj koderoj, estas ekzemple uzata en HTML-pagxoj.

2.1 koderoj deksesume

litero	Latin-3	Unikodo	UTF-8
Cx	C6	0108	C4 88
cx	E6	0109	C4 89
Gx	D8	011C	C4 9C
gx	F8	011D	C4 9D
Hx	A6	0124	C4 A4
hx	B6	0125	C4 A5
Jx	AC	0134	C4 B4
jx	BC	0135	C4 B5
Sx	DE	015C	C5 9C
sx	FE	015D	C5 9D
Ux	DD	016C	C5 AC
ux	FD	016D	C5 AD

2.2 koderoj dekume

litero	Latin-3	Unikodo	UTF-8
Cx	198	0264	196 136
cx	230	0265	196 137
Gx	216	0284	196 156
gx	248	0285	196 157
Hx	166	0292	196 164
hx	182	0293	196 165
Jx	172	0308	196 180
jx	188	0309	196 181
Sx	222	0348	197 156
sx	254	0349	197 157
Ux	221	0364	197 172
ux	253	0365	197 173

2.3 koderoj okume

TOC

litero	Latin-3	Unikodo	UTF-8
Cx	306	0410	304 210
cx	346	0411	304 211
Gx	330	0434	304 234
gx	370	0435	304 235
Hx	246	0444	304 244
hx	266	0445	304 245
Jx	254	0464	304 264
jx	274	0465	304 265
Sx	336	0534	305 234
sx	376	0535	305 235
Ux	335	0554	305 254
ux	375	0555	305 255

3. Konzolo (teksta moduso)

Latin-3-tiparojn por la teksta moduso vi trovas sub la nomoj /usr/lib/kbd/consolefonts/iso03.f?? Por funkciigi unu el ili tajpu ekzemple:

```
setfont iso03.f16
```

Konzolaj tiparoj havas aux 256 aux 512 tipojn, tamen tiparo povas havi, kaj prefere havu, unikodan indekson. La oficiala loko por konzolaj tiparoj estas /usr/share/consolefonts, sed multaj distribuoj ankoraux uzas /usr/lib/kbd/consolefonts. Utila tiparo estas ekzemple `LatArCyrHeb-16.psf`, kiu enhavas multajn literojn el pluraj alfabetoj, inkluzive de la esperantaj literoj. Por funkciigi tiun tiparon tajpu:

```
setfont LatArCyrHeb-16.psf
```

Konzolo povas funkcii per UTF-8 aux per okbita signaro, kaj eblas sxalti inter la du modusoj per eskapokodoj. Por sxalti al la UTF-8-moduso tajpu:

```
echo -e '\033%G'
```

Poste la konzolo komprenos UTF-8, do eblos ekzemple legi diverslingvajn TTT-pagxojn en UTF-8 per `lynx`.

Por sxalti al la okbita moduso tajpu:

```
echo -e '\033%@'
```

En tiu okazo vi devas ankaux difini la uzatan 8-bitan signaron. Bedauxrinde, en la nuna momento, la komandoj por tion fari estas diversaj, depende de la linuks-distribuo. Sercxu, cxu vi havas komandon `consolechars`, kaj cxu vi havas dosieron sub la nomo `/usr/share/consoletrans/iso03*`. Eventuale vi devas aktualigi vian pakon `kbd` aux anstatauxigi gxin per `console-tools`.

Vian preferatan solvon vi kompreneble aldonu en via startskripto por ne cxiam devi tajpi gxin.

4. Metamail

La programeto Metamail estas uzata de aliaj programoj, aparte retposxtaj kaj diskutgrupaj legiloj, se ili ne scias trakti iun MIME-tipon. En la dosiero `/etc/mailcap` trovigxas informoj, kiel montri tiajn MIME-tipojn. Por povi legi tekstojn koditajn laux UTF-8, ni aldonos la sekvajn du liniojn al la dosiero `/etc/mailcap`:

```
text/plain; /plena/pado/al/utf8-l3 %s; \
test=test "`echo %{charset} | tr '[A-Z]' '[a-z]'`" = utf-8;\
copiousoutput
text/utf8; /plena/pado/al/utf8-l3 %s; copiousoutput
```

La duan linion ni aldonis, cxar kelkaj programoj provas mem trakti mesagxojn en la formo text/plain, ecx se ili ne havas tauxgan tiparon por gxi, ekzemple cxe `Content-Type: text/plain; charset="utf-8"`.

La programeto `utf8-l3` trovigxas en la 2a cxapitro "Kodoj".

5. X-fenestra sistemo

5.1 Tiparoj

Por trovi Latin-3-tiparojn kaj rilatajn informojn por la X-fenestra sistemo vi rigardu en la Virtuala Esperanto-Biblioteko - sub la rubriko "Komputadaj fakoj".

Cxe la tiparoj vi trovas indikojn kiel instali ilin. Per:

```
xlsfonts -fn *iso8859-3
```

vi povas kontroli cxu la tiparoj (sukcese) estas instalitaj en via sistemo. La proporciaj *Adobe*-tiparoj bone tauxgas por vidado en Communicator, Knews, Xfmail...). Por Xterm kaj la Vi-variantoj vi uzu la tiparojn de Alain Fagot.

La plej multajn X-ajn programojn vi povas voki tiel:

```
<programo> -fn <tiparnomo>
```

por ke ili uzu la tiparon *tiparnomo*.

Alternativo al la supre menciitaj tiparoj estas la uzado de *TrueType*-tiparoj. Por tiuj tiparoj, originale uzataj de Vindozo kaj Makintosho ekzistas biblioteko *FreeType*, kiu permesas uzi ilin ankaux kun Linukso. Surbaze de tiu biblioteko ekzistas du tiparserviloj *xfstt* kaj *xfsft*, kiuj servas *TrueType*-tiparojn al X-fenestroj. La necesajn *TrueType*-tiparojn vi povas aux preni el ekzistanta Vindozo 9x/NT-instalo aux ekzemple de la mikrosofta servilo. Atentu, ke vi bezonas unikodajn *TrueType*-tiparojn. La okbitaj normale ne enhavas la esperantajn signojn aux en maltauxgaj lokoj. Unikodaj estas ekz. la kutimaj *Courier New*, *Arial*, *Times New Roman* kaj *Lucida Sans Unicode*.

xfstt

Xfstt estas relative simple instalebla solvo, kiu tamen havas la malavantagxon, ke gxi ne realigas la tutan norman tiparservan protokolon.

Normale xfstt uzas Unikodon, kies unuaj 256 signoj estas la samaj kiel cxe ISO-8859-1. Ekde la versio 0.9.8 oni povas rebildigi la unikodajn signojn al aliaj kodoj. Ekzemple per la komandlinio:

```
xfstt --encoding iso8859-3,koi8-r,windows-1251
```

xfsft

Xfsft estas pli norma, sed pli malfacile instalebla solvo. Gxi estas fliko por X-fenestroj. Do oni devas elpaki la fontaron de X, poste fliki gxin per xfsft kaj rekompili. Por Linux ankaux ekzistas jam kompilita versio. La programon Xfs (X-Font-Server) vi plej konvene lancxas en propra "init"-skripto. La tiparojn oni poste instalas same kiel aliajn X-tiparojn.

TOC

Same kiel cxe xfstt oni povas servi unikodan tiparon laux diversaj kodoj. Ekzemple la linioj

```
arial.ttf -microsoft-arial-medium-r-normal--0-0-0-0-p-0-iso8859-1
arial.ttf -microsoft-arial-medium-r-normal--0-0-0-0-p-0-iso8859-2
arial.ttf -microsoft-arial-medium-r-normal--0-0-0-0-p-0-iso8859-3
```

en `fonts.dir` servas la tiparon Arial en la kodoj Latin-1, Latin-2 kaj Latin-3. Atentu, ke en la versio 1.0.1 estis eraro pri Latin-3, kiu en la sekvaj versioj estas korektita.

5.2 Tajpi E-signojn

La X-fenestro-sistemo permesas difini reagojn al klavpremoj de la "kompona klavo" (plej ofte tiu estas difinita kiel `AltGr`) plus alia klavo.

Ekzemple sur germana klavaro `AltGr+q` donas la signon '@'

Per la programo `xmodmap` oni povas difini, ke

```
AltGr+c -> cx
AltGr+Shift+c -> Cx
...
```

Por tio metu en vian uzanto-dosierujon dosieron ~/.Xmodmap. (Ofte gxi jam estas tie kaj vi nur bezonas modifi gxin.) Kaj aldonu la sekvajn liniojn:

```
keycode  54 = C NoSymbol AE
keycode  42 = G NoSymbol Ooblique
keycode  43 = H NoSymbol paragraph brokenbar
keycode  44 = J NoSymbol onequarter notsign
keycode  39 = S NoSymbol Thorn
keycode  30 = U NoSymbol Yacute
```

Ne cxiuj programoj gxuste traktas tion. Alia ebleco (vi povas uzi ambaux variantojn samtempe) do estas, meti la 12 esperantoliterojn sur la 12 funkcio-klavojn:

```
keysym F1 = F1 ae
keysym F2 = F2 AE
keysym F3 = F3 oslash
keysym F4 = F4 Ooblique
keysym F5 = F5 paragraph
keysym F6 = F6 brokenbar
keysym F7 = F7 onequarter
keysym F8 = F8 notsign
keysym F9 = F9 thorn
keysym F10 = F10 Thorn
keysym F11 = F11 yacute
keysym F12 = F12 Yacute
```

Certigu, ke dum la starto de la X-fenestro-sistemo lancxigxas la ordono `xmodmap ~/.Xmodmap`. Norm en /usr/X11R6/lib/X11/xinit/xinitrc jam trovigxas linioj kiel

```
usermodmap=$HOME/.Xmodmap
if [ -f $usermodmap ]; then
  xmodmap $usermodmap
fi
```

---

6. KDE

La X-fenestra labormedio KDE dekomence estas evoluigita por uzado kun multaj lingvoj. La versioj 1.x bazigxas sur okbitaj signokodoj. Laux la respondeculoj, de la versio 2.0 gxi interne uzos Unikodon (fine de 1999).

La jxus aperinta versio 1.1 enhavas esperantan adapton por multaj programoj. Vi devas elekti en la lingvoagordilo `Esperanton` por vidi la menuojn, dialogojn ktp. en Esperanto. Por tio aktuale estas uzata Latin-3.

Ekzistas ankaux iom da subteno por UTF-8. Tamen nur tiel, ke gxi provas rebildigi la unikodajn signojn al signoj de la aktuale uzata tiparo. Do rigardante UTF-8-koditan esperantan TTT-pagxon uzante Latin-3-tiparon, vi vidas la Esperanto-signojn, sed ekzemple neniujn cirilajn signojn sampagxe. Krome pro la rebildigo la distancoj al la najbaraj signoj ne estas gxustaj.

7. Netscape Communicator

Ofta bezono de Esperanto-tiparoj estigxas dum la legado de interretaj pagxoj kaj diskutgrupaj artikoloj. Netscape Communicator ja permesas difini kaj uzi multajn diversajn kodojn por rigardi diverslingvajn dokumentojn, sed bedauxrinde gxi ne rekte subtenas Latin-3-kodon. Anstatauxe vi povas uzi la kodon "User-defined". Por tio elektu en la menuo "Edit Preferences", iru al la dialogo "Appearance Fonts", elektu tie la kodon "User-defined" kaj alordigu konvenajn Latin-3-tiparojn.

Nun vi povas elekti en la menuo "View Encoding User-Defined" por legi dokumenton koditan laux normo Latin-3.

Tamen ekzistas pagxoj, kiujn Netscape Communicator cxiam montras per la Latin-1-tiparo. Por rigardi cxiujn pagxojn kun Latin-3-tiparo, vi povus en la dialogo "Appearance Fonts" elekti Latin-3-tiparon por la kodo "Western (iso-8859-1)". Tio ne estas simpla, cxar Netscape por tiu kodo nur akceptas tiparojn, kies nomoj finigxas per "-1". Tial vi devas aldoni kelkajn liniojn en la dosiero "fonts.alias" de la dosierujo kun Latin-3-tiparoj. Por cxiu Latin-3-tiparo, kiun Netscape uzu, aldonu liniojn kiel la sekvajn:

```
"-adobe-helvetica-medium-r-normal--8-80-75-75-p-46-iso8859-3" 3adobe-helvetica-8
"-adobe-helvetica3-medium-r-normal--8-80-75-75-p-46-iso8859-1" 3adobe-helvetica-8
```

Poste, vi povas alordigi la tiparon kiel "Helvetica3 (Adobe)" al la kodo "Western (iso-8859-1)".

Bedauxrinde oni ne povas legi la esperantajn literojn en unikodaj tekstoj. Unikodaj tekstoj estas aperigitaj tiel, ke la signoj estas prenitaj el konvenaj okbitaj tiparoj. Tiel vi bone povas legi cirilajn, grekajn ecx cxinajn literojn, sed pro la nesuficxa subteno de Latin-3 anstataux supersignaj literoj vi vidas nur demandosignon :-(

Cetere mi rimarkis eraron en Netscape Communicator 4.0x Se oni legas diskutgrupon, oni komence ne vidas Esperanto-signojn, kvankam kiel kutima kodo estas elektita "User-Defined". Sed post elekto de iu alia kodo kaj posta reelekto de la kodo "User-Defined" vi vidas la e-ajn signojn.

8. Xterm

Por vidi esperantajn signojn en Xterm, vi povas meti la sekvan linion en la dosieron `~/.Xresources`:

```
xterm*font: <tiparnomo>
```

Kiel <*tiparnomo*>n donu iun konvenan Latin-3-tiparon. (t.e. unu el la nomoj ricevitaj per `xlsfonts -fn *iso8859-3`).

Aux voku Xterm rekte tiel:

```
xterm -fn <tiparnomo>
```

Tio funkcias ankaux en kelkaj aliaj programoj, ekz. en Emakso. Se vi ofte bezonas jen vian nacian jen Latin-3an kodon, estas rekomendinde difini alinomon en .bashrc:

```
alias eterm='xterm -fn fixed-l3' # simile por aliaj programoj
```

9. Redaktiloj

9.1 Redaktiloj kun Latin-3/Unikod-subteno

Redaktiloj, kiuj kapablas trakti Latin-3 kaj/aux Unikodon estas la Vi-variantoj Vim kaj Elvis kaj redaktiloj Yudit, Wily, Sam, SSam. UTF-respekta pakajxo de regulaj esprimoj estas cxe http://www.westley.demon.co.uk/src/libutf-2.10.tar.gz, kiun ekzemple bezonas SSam.

Pli da informoj pri Wily kaj Sam vi trovas ankaux en la arhxivo de posxtlisto pri Wily.

Al Emakso ni dedicxas apartan cxapitron.

9.2 Vi (Elvis, Vim)

Kun la Vi-variantoj Vim kaj Elvis en la aktualaj versioj vi povas senprobleme uzi la samajn latin-3-tiparojn kiel kun Xterm.

Se vi lancxas ilin en propra fenestro, uzu la elekton -fn <tiparnomo>:

```
elvis -fn fixed-l3
vim -g -fn fixed-l3
```

Por tajpi esperantajn signojn, vi povas aux uzi la metodon laux la 3a cxapitro aux uzi la sekvajn difinojn:

```
digraph G^ X
digraph g^ x
digraph H^ &
digraph h^ 6
digraph J^ ,
digraph j^ <
digraph C^ F
digraph c^ f
digraph S^ ^
digraph s^ ~
digraph U^ ]
digraph u^ }
set digraph
```

Metu tiujn difinojn en dosieron, ekz. `~/.elvis.lat3` kaj lancxu:

```
elvis -fn fixed-l3 +source~/.elvis.lat3
```

TOC

Nun vi povas tajpi ekz. la literon gx per la klavoj ^Kg^ (Ctrl-k g cirkumflekso) aux cirkumflekso, retropasxo, g.

Vim kaj Elvis vi trovas jen ankaux en Interreto.

---

10. Redakti per Emakso (Emacs)

10.1 Emakso 19.34

Vidi E-signojn

Por vidi E-signojn en Emakso, vi devas certigi, ke Emakso uzas Latin-3-tiparon. Tion vi povas atingi per aldono de la sekva linio en la dosiero `~/.Xdefaults`:

```
Emacs.font: <tiparnomo>
```

Kiel <tiparnomo>n indiku iun konvenan Latin-3-tiparon.

Same kiel cxe Xterm kaj kelkaj aliaj programoj funkcias:

```
emacs -fn <tiparnomo>
```

Tajpi

Se vi uzas Emakson sub X-fenestroj vi per la solvo realigita sub cxapitro 3 bone povas tajpi. Sed ekzistas alia alternativo, la moduso "iso-accents-mode". Por instrui al tiu moduso la lingvon Esperanto, aldonu tauxgajn difinojn en la dosiero iso-acc.el.

```
("esperanto"
 (?^ (?C . ?\306) (?G . ?\330) (?H . ?\246) (?J . ?\254)
      (?S . ?\336) (?U . ?\335) (?c . ?\346) (?g . ?\370)
      (?h . ?\266) (?j . ?\274) (?s . ?\376) (?u . ?\375))
 (?' (?C . ?\306) (?G . ?\330) (?H . ?\246) (?J . ?\254)
      (?S . ?\336) (?U . ?\335) (?c . ?\346) (?g . ?\370)
      (?h . ?\266) (?j . ?\274) (?s . ?\376) (?u . ?\375))
)
```

La sintakson divenu de la jam enhavitaj lingvoj, gxi iom varias en la diversaj Emaks-versioj. La supra ekzemplo difinas la esperantosignojn por la kombinoj ^+<*litero*> kaj '+<*litero*>. Poste kompilu la dosieron al iso-acc.elc. Ekde nun vi povas funkciigi viajn tajperojn per la komandoj:

```
M-x iso-accents-mode
M-x iso-accents-customize
  language: Esperanto
```

Atentu, ke se la cirkumflekso cxe vi estas senpasxa klavo (t.e. gxi aperas nur post premo de

TOC

^(cirkumflekso) kaj alia klavo), vi ne povas uzi gxin en tiu Emaks-moduso, anstatauxe uzu alian sigr ekz. '+c por tajpi cx. Kelkaj ecx tajpas c+x por cx.

Por pli facile atingi tiun tajpmoduson, metu la sekvajn liniojn en la dosieron `~/.emacs`:

```
(defun esperanto-mode ()
  (interactive)
  (iso-accents-mode)
  (iso-accents-customize "esperanto")
)
(global-set-key "\C-ce" 'esperanto-mode)
```

Poste per premo de C-c e (t.e. Ctrl+c e) vi trovigxas en Esperanto-moduso. Estus avantagxe, difini ankaux tian klavmallongigon por reatingi normalan moduson.

10.2 Emakso 20.* kaj MULE

notoj: Pri tiuj Emakso-variantoj Lee Sau Dan donis la sekvajn indikojn:

- Emakso 20.* kaj MULE normale povas trovi la tauxgajn tiparojn, se ili jam estas instalitaj en la X-servilo aux tiparserviloj.
- Emakso 20.* povas montri latin-3-literojn se vi uzas gxin sur teksta ekrano kun funkciigita latin-3-a tiparo. Informu Emakson pri tio per la klavpremoj:

```
C-x C-m t iso-8859-3 RET
```

- Por tajpado de Esperantosignoj, klavu unu el:

```
C-u C-\ esperanto-postfix
C-u C-\ esperanto-prefix
C-u C-\ esperanto-alt-postfix
```

- Utilas ankaux

```
M-x set-language-environment latin-3 RET
```

aux se vi ofte bezonas tion, metu en .gnu-emacs ekzemple:

```
(set-language-environment "latin-3")
(setq default-input-method "esperanto-postfix")
```

11. Retposxto per Xfmail 1.3

Xfmail estas retposxtprogramo kun plena MIME-subteno. Oni agordas gxin ene de la grafika surfaco. Sub la menuo "Misc" vi povas elekti diversajn menupunktojn por la konfigurado.

La menupunkto "Appearance" kondukas al listo, kie oni povas rilatigi kodojn kaj tiparojn. Unue premu "Add" kaj indiku la plenan nomon de latin-3-tiparo. Ekz.

```
-adobe-helvetica-medium-r-normal--12-120-75-75-p-67-iso8859-3
```

La tiparnomo nun aperas en la maldekstra listo. Por rilatigi gxin al la kodo "iso-8859-3" vi devas unue elekti tiun kodon en la dekstra listo kaj poste elekti la tiparon en la maldekstra listo. Nun premu "OK".

Xfmail enhavas internan redkatilon per kies menuo "Language" vi povas elekti la gxustan kodon.

Se vi volas uzi eksteran redaktilon vi povas iri per la menupunkto "Config Misc" kaj la butono "View/Edit" al pagxo, kie vi povas indiki gxin, ekzemple per:

```
elvis -fn <tiparnomo> +source~/.elvis.lat3
```

Jen vi trovas Xfmail en la reto.

Atentu ankoraux alian aferon. Estas ne cxiam facile sendi okbitan mesagxon. Cxar kelkaj serviloj ankoraux funkcias sepbite kaj simple ignoras la okan biton tiel, ke viaj E-literoj farigxas iuj aliaj signoj. Se vi tamen volas sendi okbite kaj uzas lokan Sendmail, vi startu gxin kun la elekto -B8BITMIME.

Por trasendi okbitajn mesagxojn tra sepbitaj medioj ekzistas la normo "quoted printable". Tiu rekodas okbitajn signoj per signaro `=NN`, cxe kio NN estas la kodero de la signo. Mi rekomendas ensxalti tiun rekodon. Per la menupunkto "Config Display" kaj la butono "Charsets" vi povas aperigi pagxon, kiu ebligas ensxalti "Quoted-printable" kaj ankaux elekti la kodon, kiun XFMail automate ensxaltu, kiam vi skribas novan mesagxon aux respondas al retmesagxo.

12. Diskutgrupoj per Knews

Knews (Karl's threaded newsreader for X) estas legilo por diskutgrupoj, kiu kapablas rekoni kaj skribi gxustajn MIME-kapliniojn pri la signokodo. Por uzi Latin-3 oni devas aldoni kelkajn liniojn al la agordaj dosieroj.

Dum la instalo de Knews, la dosiero Knews.ad estas kopiata el la fontdosierujo al /usr/X11R6/lib/X11/app-defaults/Knews.

Aldonu en tiu dosiero la sekvajn liniojn:

```
Knews.iso-8859-3.bodyFont:      \
          -*-helvetica-medium-r-normal-*-*-120-*-*-*-*-iso8859-3
Knews.iso-8859-3.quoteFont:     \
          -*-helvetica-medium-r-normal-*-*-120-*-*-*-*-iso8859-3
Knews.iso-8859-3.headerFont:    \
          -*-helvetica-medium-r-normal-*-*-120-*-*-*-*-iso8859-3

Knews.editCommand: elvis +%i %s -fn fixed-l3 +source~/.elvis.lat3
```

Anstataux Elvis vi povas ankaux uzi ekzemple Emakson kiel redaktilo.

Por povi legi ankaux esperantajn afisxojn kun malgxusta kaplinio pri kodo aux koditajn laux UTF-8, estas avantagxe aldoni sub

```
Knews.baseTranslations: #override \n\
```

la linion:

```
c    <Key> e:          mime-hack(text/plain, 8bit) \n\
c s  <Key> u:          mime-hack(text/utf8, 8bit)  \n\
```

Atentu, la \n\, kiu finu cxiun linion en la alineo krom la lastan.

Nun per Ctrl-e oni povas sxalti en cxiu afisxo al la implicita kodo. Per Ctrl-Shift-u vi povas sxalti al la MIME-tipo text/utf8. Cxar Knews ne scias trakti gxin, gxi demandas Metamail kaj montras la tekston kiel difinita en /etc/mailcap. En la sekcio pri Metmail ni difinis, ke text/utf8 estu konvertata al Latin-3. Tial sxangxu ankaux la linion pri Knews.copiousoutput al:

```
Knews.copiousOutput:  exec xterm \
      -fn 'fixed-l3' -e /bin/sh -c '(%C) | less'
```

Anstataux `fixed-l3` enmetu la de vi preferatan Latin-3-tiparon.

La implicitan kodon (uzatan cxe Ctrl-e) oni difinas en la dosiero `~/.knews/config-*` (* estas la uzata diskutgrupa servilo. Tiu dosiero estas auxtomate kreata cxe la unua lancxo de Knews. Aldonu la sekvan linion:

```
*esperanto*defaultCharset:  iso-8859-3
```

Nun en cxiu diskutgrupo, kiu enhavas la nomon "esperanto", estas uzata la kodo Latin-3 kiel implicita.

Jen vi trovas Knews en Interreto.

13. TeX / LaTeX

Por produkti la supersignajn literojn en TeX, vi normale tajpas:

```
\^C \^c \^G \^g \^H \^h \^J \^{\j} \^S \^s \u{U} \u{u}
```

La \j cetere estas 'j' sen la punkto. Tamen, se vi skribas longajn tekstojn, estas iom tede skribi tiujn ordonojn. Do estas avantagxe krei dosieron esperanto.sty kun la sekva enhavo:

```
\let\mathcaret=^

\def\esperantocaret#1{
  \ifx#1u\u{#1}\else
  \ifx#1U\u{#1}\else
  \ifx#1j\^{\j}\else
  \^{#1}\fi\fi\fi
}

\catcode`\^=\active

\def^{\ifmmode\let\next\mathcaret
    \else\let\next\esperantocaret
    \fi\next}
```

Tio permesas skribi ^C ^c ... ^U ^u por la esperantaj literoj. Iom komplikas la aferon, ke la signo ^ estas ankaux uzata por supraj indicoj en la matematika moduso, t. e. por matematikaj formuloj. Do vi ne povas uzi tiujn mallongigojn en formuloj ekzemple. (Pro tio ankaux la konfuzigaj linioj kun "mathcaret" en la supraj difinoj)

Tamen la supra solvo povas kauxzi diversajn problemojn kun makrooj aux en matematikaj formuloj. Pro tio Albert Reiner sendis al mi pli bonan solvon, kiu uzas la grado-signon anstataux la tegmenteto:

```
\gdef\espon{\catcode`\*\active\relax}%
\gdef\espoff{\catcode`\*12\relax}%
%
{%
\catcode`\@11   % por kasxi kelkajn makrojn
\espon          % ensxalti Esperanton por la difinoj
\gdef*{\relax\esper@nto}%
\gdef\esper@nto#1{%
\ifx #1C\^C%
\else\ifx #1c\^c%
```

```
\else\ifx #1G\^G%
\else\ifx #1g\^g%
\else\ifx #1H\^H%
\else\ifx #1h\^h%
\else\ifx #1J\^J%
\else\ifx #1j\^\j%
\else\ifx #1S\^S%
\else\ifx #1s\^s%
\else\ifx #1U\u U%
\else\ifx #1u\u u%
\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi}%
}%
\espon          % ensxalti Esperanton
```

(rim.: En la supra ekzemplo anstatauxigu la steleton * per la gradosigno . La sintakso de la SGML-dokumenttipo ne traktas gradosignon en tiu loko gxuste.)

Se vi ebligis la uzadon de Latin-3 en via sxatata redaktilo, vi povas ankaux rekte traduki la Latin-3-kodojn al la esperantaj signoj:

```
\catcode`\cx=\active \defcx{\^c}
\catcode`\gx=\active \defgx{\^g}
\catcode`\hx=\active \defhx{\^h}
\catcode`\jx=\active \defjx{\^\j}
\catcode`\sx=\active \defsx{\^s}
\catcode`\ux=\active \defux{\u u}
\catcode`\Cx=\active \defCx{\^C}
\catcode`\Gx=\active \defGx{\^G}
\catcode`\Hx=\active \defHx{\^H}
\catcode`\Jx=\active \defJx{\^J}
\catcode`\Sx=\active \defSx{\^S}
\catcode`\Ux=\active \defUx{\u U}
```

Noto: En la supra ekzemplo vi devas anstatauxigi la esperantajn literojn cx ... Ux en cxiu linio kaj maldekstre kaj dekstre per la gxustaj literoj laux Latin-3-kodo.

Cetere pli ampleksa dosiero esperant.sty estas enhavata en la Babel-pakajxo por LaTeX2e. Simple kontrolu per `locate esperant`, cxu gxi eble jam estas instalita cxe vi. Gxi ekzemple ankaux kreas esperantajn titolojn en la enhavtabelo kaj indeksoj.

Por uzi la pakajxon, tajpu

sub TeX:

```
\input{esperant.sty}
```

sub LaTeX:

```
\usepackage[esperanto]{babel}
```

Ioma problemo estas, ke en tiparoj aliaj ol la normala "Computer Modern" ne trovigxas la senpunkta j. Por kasxi la punkton super j ekzistas dosiero dotlessj.sty. Detalaj informoj pri gxi trovigxas cxe Edmundo.

---

14. Esperanto-subteno en internaciigitaj programoj

Kelkaj Gnu-programoj kaj aliaj estas internaciigitaj. T.e. iliaj mesagxoj povas aperi en diversaj lingvoj. Tamen la Esperanto-tradukoj de tiuj programoj nur nun estas komencataj. Sed ja vi povas kontroli, cxu en via sistemo ekzistas dosierujo /usr/share/locale/eo aux cxu cxe la fontdistribuoj de viaj programoj estas iuj dosieroj po/eo.po. Se jes, la programoj kapablas montri esperantajn mesagxojn. Por vidi tiujn mesagxojn, vi devas difini la variablon LANG aux LANGUAGE en la cxirkauxajxo de la programo.

Por unuopa ordono vi povas ekzemple tajpi tiel:

```
LANG=eo <programo>
```

Aux vi povas dauxre difini la preferatajn lingvojn, se vi metas en vian startskripton ion kiel

```
export LANG=eo:de:en  # en bash

setenv LANG eo:de:en  # en tcsh
```

Tio signifas: lauxeble uzu Esperanton, alikaze la germanan lingvon, aux se ankaux tiu ne estas subtenata de la programo, uzu la anglan.

Forumo por la adaptado de linuksaj programoj al Esperanto cetere estas la retposxtlisto eo@li.org, al kiu vi povas aligxi per sendo de la vorto `subscribe` en la korpo de mesagxo al `eo-request@li.org`.

15. Esperanto-signoj en HTML-tekstoj

Vi povas uzi la Latin-3-koderojn por montri E-signojn en HTML-teksto. La normo volas, ke tiukaze vi metu la sekvan linion en la kapon (<head>...</head>) de via dosiero:

```
<meta http-equiv="Content-type" content="text/html; charset=ISO-8859-3">
```

Bedauxrinde Netscape Communicator kaj eble ankaux aliaj TTT-legiloj misinterpretas tiun linion kaj malhelpas, ke ecx per via "User-defined"-kodo vidigxos E-signoj. Alikaze en TTT-legiloj kiuj bone atentas la normon (ekz. Kfm), vi vidas la esperantaj signojn nur, se tiu linio trovigxas en la dosiero.

Pri kodado per Unikodo, legu la informpagxon cxe Aaron Irvine. Pro la problemoj de Netscape Communicator rilate al E-signoj en Unikodo ni malrekomendas la uzon de Unikodo en pure esperantaj HTML-tekstoj. Por multlingvaj tekstoj eble ne ekzistas pli tauxga alternativo. Sed por Esperanto uzu prefere Latin-3 ankoraux iom da tempo.

Finnish-HOWTO

Pekka Taipale, pjt@iki.fi

v1.1f, 14 February 1996

This document describes how to finlandize Linux - Finnish keyboard mapping, time zone, 8-bit clean settings for applications, etc. Except for this abstract, the text is in Finnish because that is the benefit of the intended audience. This document may be reproduced at the usual Linux HOWTO terms. If you use it commercially, you're welcome to do so but I'd like to be notified. If you have questions, contact me at my e-mail address.

1. Johdanto

- 1.1 Mikä HOWTO?
- 1.2 Tekijänoikeuksista ja kopioinnista
- 1.3 Tekstin esitystavasta
- 1.4 Kiitokset

2. Näppäimistöasetukset

- 2.1 Perusnäppäimistö
- 2.2 ISO Latin-1 -merkistö
- 2.3 Näppäimistökartan parantelua

3. Pääteasetukset

4. Ympäristön lokaaliasetukset

5. Aikavyöhykeasetukset

TOC

6. X11R6

7. Eräiden sovellusten asetukset

- 7.1 bash
- 7.2 tcsh
- 7.3 emacs
- 7.4 telnet
- 7.5 rlogin
- 7.6 joe
- 7.7 less
- 7.8 irc
- 7.9 elm
- 7.10 pine
- 7.11 TeX

8. Posti, nyyssit ja 8-bittiset merkit

- 8.1 Perusratkaisu: 8-bittiset merkit viesteihin
- 8.2 MIME Quoted Printable-koodaus

9. Aiheeseen liittyvää materiaalia

---

1. Johdanto

1.1 Mikä HOWTO?

Tämä dokumentti kertoo, mitä pitäisi tehdä Linuxin suomettamiseksi sen jälkeen kun käyttöjärjestelmä perussovelluksineen on asennettu koneeseen jostain Linux-jakelusta. Näin saadaan suomalaiset kirjaimet, näppäimistö ja muut paikalliset asetukset toimimaan oikein. Lähtökohtana on Slackware- tai RedHat-jakelusta asennettu Linux-järjestelmä, mutta ohjeista on todennäköisesti apua myös muiden Linux-jakeluversioiden käyttäjille.

Varsin suuri osa ohjeista pätee Linuxin lisäksi muissakin Unix-tyyppisissä järjestelmissä. Sama toimii myös toisinpäin: jos et löydä etsimääsi tietoa tästä dokumentista, tutki muita FAQ- jne. tiedostoja (ks. esim. Linuxin HOWTO-dokumentit sekä viitteet [UNIXHelp] ja [faqs]). Tämän dokumentin viitteissä on, mikäli mahdollista, käytetty kotimaisia peilauksia ulkomailla sijaitsevien alkuperäisten tiedostovarastojen asemesta.

Tässä ohjeessa ei ole yleensä eritelty root-käyttäjän ja tavallisen käyttäjän tekemiä asetuksia. Jotkut ohjeista ovat siis ainoastaan ylläpitäjän heiniä ja vaikuttavat koneen kaikkiin käyttäjiin, jotkut taas ovat tavallisen käyttäjän itselleen tekemiä asetuksia. Linux-koneissa ylläpitäjä ja käyttäjä ovat usein - mutta eivät aina - sama henkilö. Hyvin monet asetukset voi tehdä molemmilla tavoilla; esim. kaikkien käyttäjien oletukset hakemistossa `/etc` oleviin tiedostoihin ja käyttäjän omat asetukset kotihakemistossa `$HOME` oleviin pisteellä alkaviin tiedostoihin.

1.2 Tekijänoikeuksista ja kopioinnista

Copyright (c) 1996 by Pekka Taipale. Tätä dokumenttia saa kopioida ja levittää muuttamattomana vapaasti. Jos levität muutettua tai lyhennettyä versiota dokumentista, tätä tekijänoikeusmerkintää ei saa poistaa. En ota vastuuta mahdollisista virheistä enkä varsinkaan muiden muuttamista versioista. Jos teet dokumenttiin lisäyksiä tai korjauksia, merkitse selvästi oma työsi. Parasta olisi, jos lähettäisit korjausehdotuksesi minulle osoitteeseen `pjt@iki.fi` jolloin ne voidaan ottaa mukaan alkuperäiseen tekstiin.

1.3 Tekstin esitystavasta

Tämä HOWTO on kirjoitettu SGML-muodossa. HTML-, TeX- ja tekstiversiot generoidaan SGML:stä Linuxdoc-SGML -ohjelmalla [Linuxdoc]. Tämän tekstin ulkopuolelle osoittavat viittaukset (HTML-versiossa linkit) on koottu yhteen paikkaan, ks. luku Aiheeseen liittyvää materiaalia. Muualla dokumentissa olevat linkit ovat ristiinviittauksia dokumentin sisällä. Näin on menetelty siksi, että olisi helpompi hahmottaa, milloin HTML-version linkin seuraaminen on viemässä lukemaan jotakin muuta dokumenttia ja milloin se on ristiinviittaus tämän HOWTO:n sisällä.

1.4 Kiitokset

Kiitos kaikille kommentteja ja korjauksia esittäneille. Erityisesti mainittakoon Tuomas Aura, Tuomas Eerola, Timo Korvola, Jaakko Ruohio, Jani Tammi ja Lars Wirzenius.

---

2. Näppäimistöasetukset

2.1 Perusnäppäimistö

Tässä kerrotaan tekstikonsolin näppäimistöasetuksista. X-ympäristöstä on tietoa luvussa X11R6.

Jos näppäimistösi on Linuxin asennuksen jäljiltä amerikkalaista mallia, olet ohittanut näppäimistökartan valinnan asennusohjelmassa. Näppäimistökartaksi kannattaa valita ISO Latin-1-merkkejä (ks. kohta alla) tuottava kartta *fi-latin1*. Seitsemänbittinen *fi*-kartta on poistumassa käytöstä eikä sitä kannata käyttää jos siihen ei ole erityistä tarvetta.

Slackware-järjestelmissä näppäimistökartan voi asettaa lisäämällä tiedostoon `/etc/rc.d/rc.keymap` tai `/etc/rc.d/rc.local` seuraava rivi:

```
/usr/bin/loadkeys fi-latin1       # Finnish latin1 keymap
```

RedHatissa asetus taas on tiedostossa `/etc/sysconfig/keyboard`, jossa pitää olla seuraava rivi:

```
KEYTABLE=/usr/lib/kbd/keytables/fi-latin1.map
```

Debian-järjestelmissä asetus tehdään tiedostoon `/etc/rc.boot/console`.

Näppäimistökarttatiedostot ovat yleensä hakemistossa `/usr/lib/kbd/keytables`. Mikäli fi-latin1 -karttaa ei löydy tästä hakemistosta, etsi esim. Slackware-jakelun keytbls-paketti. Ruotsalaiset käyttävät samaa näppäimistökarttaa kuin suomalaiset.

2.2 ISO Latin-1 -merkistö

Suomenkielinen teksti kirjoitetaan Linux-ympäristössä ISO Latin-1 -merkistöllä, joka käyttää 7-bittisiä ASCII-merkkejä englannin perusaakkosten näyttämiseen ja 8-bittisiä ns. ISO-skandeja suomalaisiin kirjaimiin å, ä, ö jne. Myös mm. MS Windows käyttää samaa merkistöä. Lisätietoja löytyy Timo Kiravuon merkistöohjeesta [skandit] ja Jukka Korpelan ISO-Latin-1-merkkejä käsittelevästä tekstistä [ISO-latin1].

Jos fi-latin1 -näppäimistöllä ä:n ja ö:n painalluksista tulee d- ja v-kirjaimia, hukkuu kahdeksas bitti jonnekin. Ks. Pääteasetukset-kohta ( ) sekä rlogin ( ) ja telnet ( ). Jos skandien painaminen aiheuttaa piippauksia tai mitään ei tapahdu, sovellus (esim. komentotulkki) ei huoli 8-bittisiä merkkejä. Katso sovelluksien kohdalta komentotulkkisi asetukset (kohdat bash ja tcsh).

2.3 Näppäimistökartan parantelua

Timo Korvola ehdottaa:

Gravea, tildeä ja sirkumfleksia tarvitaan sen verran usein, että elämän helpottamiseksi voi tehdä fi-latin1.map:in lataamisen jälkeen vaikkapa näin:

```
loadkeys << '@loadkeys'
  keycode 5 = four currency dollar
  keycode 13 = dead_acute grave
  keycode 27 = dead_diaeresis asciicircum asciitilde
  keycode 97 = Compose
@loadkeys
```

Tämä korjaa myös shift 4:n tuottamaan valuuttamerkin siten kuin näppäimistössä lukee. Oikeasta controlista tehdään compose, jolla voi tuottaa vaikkapa è:n näppäilemällä compose ` e.

Konsolin näppäinkoodit saa helposti selville ohjelmalla `showkey`.

3. Pääteasetukset

8-bittisten suomalaisten merkkien toimiminen vaatii, että pääteyhteys on 8-bittinen. Yleensä Linux-koneen näppäimistöllä ja päätelinjoilla näin onkin, mutta mikäli ei ole, asetus saadaan päälle laittamalle seuraava komento komentotulkin alustustiedostoon (`.profile`, `.login` jne):

```
stty pass8
```

8-bittisten merkkien toimiminen rlogin- ja telnet-yhteyksien yli on selitetty alempana sovelluksien kohdalla (ks. kohdat rlogin ja telnet).

4. Ympäristön lokaaliasetukset

Lokaaliasetusten tarkoitus on saada ohjelmat noudattamaan esimerkiksi eri kielten erilaisia aakkosjärjestyssääntöjä ja muita käytäntöjä. Lokaalit vaikuttavat mm. kellonaikojen ja päivämäärien tulostusmuotoon ja tietueiden lajittelujärjestykseen eri ohjelmia käytettäessä - mikäli ohjelmat on tehty asianmukaisella tavalla ja kirjastot tukevat kyseistä lokaalia.

Uudehkojen Linux-jakelujen (Slackware 3.0 ja RedHat 2.1) kirjastot tukevat dokumenttien mukaan yleistä eurooppalaista Latin1-merkistölokaalia ISO-8859-1. Oikea lokaaliasetus Suomessa on `finnish.iso88591`. Lokaali asetetaan ympäristömuuttujien LC_CTYPE, LC_TIME jne. avulla (ks. `man 7 locale`).

Voi olla viisasta asettaa suomalainen lokaali jo nyt, niin että spesifisen suomalaisen lokaalituen mahdollisesti joskus ilmestyessä kirjastoihin se otetaan käyttöön. Lokaalin asetus voi olla myös epäviisasta, koska esim. päivämäärätulostuksen muodon muuttuminen voi muuttaa komentotiedostojen toimintaa ja rikkoa siten ohjelmia. LC_TIME-lokaalin POSIX-asetuksella date-komennon tulostus on muotoa `Wed Aug 9 13:33:39 EETDST 1995` kun taas suomalaisella lokaalilla se olisi muotoa `9.8.1995 13:33:39`.

LC_ALL- tai LC_TIME-asetuksen tekeminen ei siis ole välttämättä järkevää, mutta LC_CTYPE ja LC_COLLATE kannattaa asettaa suomalaiseksi. Esim. tcsh vaatii LC_CTYPE-asetuksen, jotta 8-bittiset merkit toimisivat.

Jos käytät komentotulkkina csh-varianttia (kuten tcsh), haluat todennäköisesti tehdä asetuksen tiedostossa `/etc/csh.cshrc` tai `$HOME/.cshrc`:

```
setenv LC_CTYPE finnish.iso88591
```

Jos taas käytät sh-varianttia (kuten bash tai pdksh), kirjoita tiedostoon `/etc/profile` tai `$HOME/.profile`:

```
export LC_CTYPE=finnish.iso88591
```

5. Aikavyöhykeasetukset

Linux-koneen CMOS-kello voi osoittaa joko UTC-aikaa - kuten UNIX-koneissa on yleensä tapana - tai paikallista aikaa. (UTC on kansainvälinen normaaliaika, suunnilleen sama kuin Greenwich Mean Time eli GMT.) Linux-jakelut olettavat yleensä kellon käyvän paikallista aikaa, koska MS-DOS ja muut tavalliset PC-käyttöjärjestelmät eivät tunne aikavyöhykkeitä. Tällöin Linuxin käynnistystiedostoissa - esim. `/etc/rc.d/rc.S` (Slackware) tai `/etc/rc.d/rc.sysinit` (RedHat) - on komento `clock -s`. Jos kello on UTC-ajassa, komennon pitää olla `clock -u -s`.

Kun CMOS-kello on UTC-ajassa, käyttöjärjestelmä huolehtii ajan näyttämisestä oikein mm. kesäaikaan siirryttäessä, kunhan aikavyöhyke on asetetty oikein. Jos CMOS-kello on paikallisessa ajassa, kesä- ja talviaikoihin siirtymiset on tehtävä käsin. CMOS-kellon aikaa voi muuttaa joko CMOS-setupista käsin tai sitten Linuxissa clock-komennolla (ks. `man 8 clock`).

Suomen aikavyöhyke on EET (Eastern European Time). Aikavyöhykkeen voi asentaa käyttöjärjestelmän asennuksen yhteydessä; mikäli et tehnyt sitä tai teit sen väärin, voit tehdä asetuksen root-tunnuksella komennolla

```
ln -s /usr/lib/zoneinfo/Europe/Helsinki /usr/lib/zoneinfo/localtime
```

(Vanhemmissa Linux-jakeluissa ei ole Helsinki-vyöhykettä; käytä EET:tä).

Yleensä systeemin oletusaikavyöhykkeen asettaminen riittää. Jos yksittäinen käyttäjä tarvitsee erilaisen aikavyöhykkeen, sen voi asettaa käyttäjäkohtaisesti TZ-muuttujalla. POSIX-käytäntö on, että TZ-asetus on muotoa EET-2EETDST, mikä tarkoittaa, että aikavyöhyke on EET, se on kaksi tuntia edellä GMT:tä, ja kesäaikaan siirrytään EETDST-säännön mukaan. Asetus saadaan aikaan komentotulkin alustustiedostossa, esim. bash-käyttäjille lisäämällä kotihakemiston `.profile`-tiedostoon rivi

```
export TZ=EET-2EETDST
```

6. X11R6

XFree86 saa näppäimistökarttansa Linuxin konsolilta. Useimpien näppäinten (ml. ä ja ö) pitäisi toimia suoraan oikein, kun konsolin näppäimistökartta on asetettu. "Kuolleiden" näppäinten määrityksiä (dead_acute, dead_diaeresis ym.) ei kuitenkaan saada konsolilta, joten lopputulosta täytyy hieman paikkailla.

Jos kone on konfiguroitu käynnistämään xdm eli käyttäjä ei kirjoittaudu sisään virtuaalikonsolille vaan suoraan X11:aan, voi olla tarpeen asettaa koko suomalainen näppäimistökartta `.Xmodmap`-tiedostossa.

Mikäli AltGr-näppäin ei toimi (merkit {, [,], }, \ jne.), tarkista, että X:n konfigurointitiedostossa (`/etc/XF86Config` tai `/usr/lib/X11/XF86Config`) on rivi

```
RightAlt    ModeShift
```

ja että sitä ei ole kommentoitu pois #-merkillä.

X:n käynnistysskriptit ajavat yleensä xmodmap-ohjelman, joka muuttaa X:n näppäimistökarttaa käyttäjän `.Xmodmap`-tiedoston perusteella. `.Xmodmap`-tiedosto voi näyttää esim. tältä:

```
keycode  13 = 4 currency dollar
keycode  21 = acute grave
keycode  35 = dead_diaeresis asciicircum asciitilde
```

Tässä näppäimeen, jonka näppäinkoodi on 13 (näppäimistön päälohkon numeronäppäin 4) sidotaan merkit 4, valuuttamerkki ja $, jotka tulevat, kun ko. näppäintä painetaan yksinään, shiftin kanssa ja AltGr-näppäimen kanssa.

Näppäimeen 21 (Backspace-näppäimen vieressä) sidotaan hipsumerkit ja näppäimeen 35 (å:n ja Enterin välissä) sidotaan "kuolleet pisteet" (¨) sekä sirkumfleksi ("hattumerkki") ja tilde (~). Voit kirjoittaa esim. saksalaisen ü-kirjaimen painamalla ensin kuolleita pisteitä ja sitten u:ta. .Xmodmap-tiedoston muutokset saa heti käyttöön komennolla

```
xmodmap .Xmodmap
```

Ks. `man xmodmap`. Huomaa, että emacs saattaa kohdella näppäimiä poikkeavasti. X:n alla näppäinkoodit saa selville ohjelmalla `xev`.

Esimerkki X11-näppäimistökartasta

Tämän voi sijoittaa `.Xmodmap`-tiedostoon.

TOC

```
keycode   8 =
keycode   9 = Escape
keycode  10 = 1 exclam
keycode  11 = 2 quotedbl at
keycode  12 = 3 numbersign sterling
keycode  13 = 4 dollar dollar
keycode  14 = 5 percent
keycode  15 = 6 ampersand
keycode  16 = 7 slash braceleft
keycode  17 = 8 parenleft bracketleft
keycode  18 = 9 parenright bracketright
keycode  19 = 0 equal braceright
keycode  20 = plus question backslash
keycode  21 = apostrophe grave
keycode  22 = Delete
keycode  23 = Tab
keycode  24 = Q
keycode  25 = W
keycode  26 = E
keycode  27 = R
keycode  28 = T
keycode  29 = Y
keycode  30 = U
keycode  31 = I
keycode  32 = O
keycode  33 = P
keycode  34 = aring Aring
keycode  35 = dead_diaeresis asciicircum asciitilde
keycode  36 = Return
keycode  37 = Control_L
keycode  38 = A
keycode  39 = S
keycode  40 = D
keycode  41 = F
keycode  42 = G
keycode  43 = H
keycode  44 = J
keycode  45 = K
keycode  46 = L
keycode  47 = odiaeresis Odiaeresis
keycode  48 = adiaeresis Adiaeresis
keycode  49 = section onehalf
```

TOC

```
keycode  50 = Shift_L
keycode  51 = apostrophe asterisk
keycode  52 = Z
keycode  53 = X
keycode  54 = C
keycode  55 = V
keycode  56 = B
keycode  57 = N
keycode  58 = M
keycode  59 = comma semicolon
keycode  60 = period colon
keycode  61 = minus underscore
keycode  62 = Shift_R
keycode  63 = KP_Multiply
keycode  64 = Alt_L
keycode  65 = space
keycode  66 = Caps_Lock
keycode  67 = F1
keycode  68 = F2
keycode  69 = F3
keycode  70 = F4
keycode  71 = F5
keycode  72 = F6
keycode  73 = F7
keycode  74 = F8
keycode  75 = F9
keycode  76 = F10
keycode  77 = Num_Lock
keycode  78 = Scroll_Lock
keycode  79 = Home KP_7 KP_7 Home
keycode  80 = Up KP_8 KP_8 Up
keycode  81 = Prior KP_9 KP_9 Prior
keycode  82 = KP_Subtract
keycode  83 = Left KP_4 KP_4 Left
keycode  84 = Begin KP_5 KP_5 Begin
keycode  85 = Right KP_6 KP_6 Right
keycode  86 = KP_Add
keycode  87 = End KP_1 KP_1 End
keycode  88 = Down KP_2 KP_2 Down
keycode  89 = Next KP_3 KP_3 Next
keycode  90 = Insert KP_0 KP_0 Insert
keycode  91 = Delete KP_Decimal KP_Decimal Delete
keycode  92 = 0x1007ff00
keycode  93 =
keycode  94 = less greater bar
keycode  95 = F11
```

```
keycode  96 = F12
keycode  97 = Home
keycode  98 = Up
keycode  99 = Prior
keycode 100 = Left
keycode 101 = Begin
keycode 102 = Right
keycode 103 = End
keycode 104 = Down
keycode 105 = Next
keycode 106 = Insert
keycode 107 = Delete
keycode 108 = KP_Enter
keycode 109 = Control_R
keycode 110 = Pause
keycode 111 = Print
keycode 112 = KP_Divide
keycode 113 = Alt_R Mode_switch
keycode 114 = Break
keycode 115 =
```

7. Eräiden sovellusten asetukset

Tähän ei ole kerätty kaikkia mahdollisia sovelluksia; lisäapua voi löytyä esim. German-HOWTO:sta [german].

7.1 bash

Tee kotihakemistoosi tiedosto `.inputrc`, ja laita sinne nämä rivit:

```
set meta-flag on
set convert-meta off
set output-meta on
```

Tämän jälkeen komentotulkki huolii 8-bittiset merkit ja näyttää ne oikein.

bash lukee käynnistyessään useita eri tiedostoja. `.profile`, `.bash_login` ja `.bash_profile` luetaan ainoastaan sisäänkirjoittautumisen (login) yhteydessä; `.bashrc` taas suoritetaan aina kun bash käynnistyy (myös esim. xterm-ikkunassa).

7.2 tcsh

tcsh käyttää lokaaliasetusta LC_CTYPE. Tee kotihakemistoosi tiedosto `.tcshrc`, ja laita sinne rivi

```
setenv LC_CTYPE finnish.iso88591
```

Huomaa, että RedHat 2.1:n mukana tuleva tcsh ei jostain syystä toimi 8-bittisten merkkien kanssa oikein. Voit joko kääntää tcsh:n itse, tai kopioida tcsh-binäärin esim. Slackware 3.0:sta.

7.3 emacs

Tee tai editoi kotihakemistossasi tiedostoa `.emacs`. Laita sinne rivit:

```
(standard-display-european 1)
(set-input-mode (car (current-input-mode))
                (nth 1 (current-input-mode))
                'accept-8bit-input)
```

```
(require 'iso-syntax)
```

Huomaa, että GNU Emacs 19.25, joka on mm. Slackware-jakelun versiossa 2.0.0, on rikki eikä eurooppalaisten merkkien tuki toimi. Jos käytät tätä versiota, poista se ja asenna uudempi versio. Myös emacs 19.30 toimii väärin jos se on käännetty vanhalla gcc:llä (esim. 2.6.3).

7.4 telnet

Luo kotihakemistoosi tiedosto `.telnetrc` ja tee sinne oma rivi kullekin koneelle johon otat yhteyttä telnetillä. Rivillä on koneen nimi sekä komento `set outbinary`. Jos esim. otat yhteyttä koneeseen masina paikallisessa verkossa sekä koneeseen hissi.kone.fi Internetin yli, `.telnetrc`-tiedostosi pitäisi näyttää tältä:

```
masina set outbinary
hissi.kone.fi set outbinary
```

Voit myös antaa käynnistettäessä option `-8` tai asettaa binäärimoodin telnet-istunnon aikana painamalla telnetin escape-merkkiä (ei ESC-näppäin vaan telnetin escape character, oletuksena `C-]` joka löytyy yleensä näppäilemällä Ctrl-5) ja antamalla komennon. Ks. `man telnet`.

7.5 rlogin

rlogin-komennossa pitää antaa optio `-8` jotta 8-bittiset merkit kulkisivat yhteyden läpi. Ks. `man rlogin`.

7.6 joe

joe tarvitsee option `-asis` jotta se kohtelisi merkkejä 8-bittisinä.

Poista välilyönti joerc-tiedostosta sen rivin alusta, jolla lukee `-asis`. Kaikkien käyttäjien yhteinen joerc-tiedosto on yleensä `/usr/lib/joe/joerc` ja käyttäjäkohtainen tiedosto `$HOME/.joerc`.

7.7 less

Aseta ympäristömuuttuja LESSCHARSET arvoon latin1:

```
export LESSCHARSET=latin1
```

tai vastaava C-shellin komento.

TOC

7.8 irc

Tämän kirjoitti Jani Tammi (jantam@utu.fi).

Laita `.ircrc`-tiedostoon rivi:

```
        set eight_bit_characters on
```

Tämän lisäksi saatat tarvita myös komennon

```
        /set translation latin_1
```

Huomaa, että tämä komento annetaan irc:issä, eikä sitä kirjoiteta `.ircrc`-tiedostoon.

7.9 elm

Laita tiedostoon $HOME/.elm/elmrc seuraavat rivit:

```
charset = iso-8859-1
displaycharset = iso-8859-1
textencoding = 8bit
```

7.10 pine

Laita tiedostoon `/usr/local/lib/pine.conf` tai `$HOME/.pinerc` seuraava rivi:

```
character-set=ISO-8859-1
```

7.11 TeX

Suomalaiset aakkoset ja tavutuksen saa TeXissä toimimaan parhaiten käyttämällä babel-pakettia, joka on siis syytä valita asennettavaksi TeXiä asennettaessa.

TeXistä on kaksi Linux-järjestelmissä yleisesti käytettyä jakelua: NTeX ja teTeX. NTeX tulee Slackwaren ja teTeX RedHatin mukana. teTeX on näistä helpompi konfiguroida.

TeXin hakemistopolut ja konfigurointi vaihtelevat jakelusta toiseen. Voit joutua tekemään asiat jakelusta

riippuen hyvinkin paljon tästä ohjeesta poikkeavalla tavalla.

NTeX

Tässä opasti Tuomas Aura (Tuomas.Aura@hut.fi):

LaTeXissa (2e) ei ole valmiina suomenkielistä tavutusta, vaan se on käännettävä itse. Tämä tapahtuu seuraavasti (root-käyttäjänä):

- Etsi tiedosto `language.dat` (Slackwaren NTeXissä `/usr/lib/texmf/tex/latex/cfg/language.dat`) ja editoi sitä. Valitse enintään 4 kieltä ja poista niiden edestä kommenttimerkki %. Slackware 3.0:n NTeX:issä on kielen nimi "finnish" kirjoitettu väärin: "finish". Tämä on syytä korjata (tiedoston `hyphen.finish` nimessä se ei haittaa).
- Samassa hakemistossa on `lthyphen.cfg` joka on korvattava babel-versiolla:

```
rm lthyphen.cfg
ln -s lthyphen.babel lthyphen.cfg
```

- Poista tavutuksen amerikkalainen oletustiedosto:

```
rm /usr/lib/texmf/tex/tex/macros/hyphen.tex
```

- Etsi tiedosto `latex.ltx` ja mene samaan hakemistoon (NTeXissä `/usr/lib/texmf/tex/latex/latex-base`). Aja `initex`:

```
initex latex.ltx
```

- Näin syntyy tiedosto `latex.fmt`. Kopioi se LaTeXin ini-hakemistoon, esim. NTeXissä:

```
mv latex.fmt /usr/lib/texmf/ini/
```

Nyt pitäisi suomen tavutuksen toimia, kunhan dokumentin otsikot ovat kunnossa.

teTeX

teTeX:in konfigurointi on helppoa:

- Ajetaan `texconfig`
- Valitaan valikosta HYPHEN
- Editori aukeaa editoimaan `language.dat`-tiedostoa, poistetaan kommentti sen rivin alusta, jossa lukee `finnish`
- Uusi LaTeX-formaatti luodaan automaattisesti

LaTeX-otsikot

Suomalaisen LaTex-dokumentin alkuriveillä pitäisi lukea jotain tämän tapaista:

```
\documentclass[a4paper,finnish]{article}
\usepackage{t1enc,isolatin1,babel}
```

LaTeX2e <1994/12/01> -formaatilla ja uudemmilla toimii myös hienompi ratkaisu:

```
\documentclass[a4paper, finnish]{article}
\usepackage{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
```

Näillä otsikoilla pitäisi suomalaisen tavutuksen ja kirjainten toimia oikein, mikäli TeX-jakelu on asennettu ja konfiguroitu oikein. Tässä ei neuvota LaTeX-dokumenttien tekemistä tarkemmin. Jukka Korpelan TKK:lle tekemästä Tex-ohjeesta [texref] voi olla apua.

8. Posti, nyyssit ja 8-bittiset merkit

8.1 Perusratkaisu: 8-bittiset merkit viesteihin

Posti- ja nyyssiviesteissä (USENET news ja vastaavat paikalliset järjestelmät) on järkevintä lähettää 8-bittisiä ISO Latin1-skandeja silloin kun kirjoitetaan suomeksi tai ruotsiksi.

Postiohjelmat kuten elm ja pine näyttävät Latin1-merkit oikein, kun ne on konfiguroitu sovellusten kohdalla kuvatulla tavalla (ks. kohdat elm ja pine). Jäljellä on kuitenkin vielä yksi ongelma: miten saada postissa mitä kummallisimmissa muodoissa tulevat viestit ISO-8859-1:n mukaisiksi?

8-bittiset ISO Latin1-merkit kirjoitetaan viestiin sellaisenaan ja viestin otsikossa ilmoitetaan, että sisältö on 8-bittistä tekstiä. Tämä tapahtuu liittämällä lähteviin viesteihin seuraavat MIME-standardin [RFC1521] mukaiset otsikot. Tämä edellyttää, että kaikki viestiä välittävät koneet kykenevät käsittelemään 8-bittistä tietoa viestin rungossa. Suomi on jo jokseenkin läpeensä 8-bittinen, mutta erityisesti USA:sta löytyy vielä runsaasti 7-bittisiä postijärjestelmiä, jotka ovat postistandardin RFC822 mukaisia [RFC822]. Näiden kanssa on yleensä tyydyttävä käyttämään QP-koodausta (ks. MIME Quoted Printable-koodaus). Tavallisesti seuraavat otsikot ja 8-bittiset merkit kuitenkin riittävät:

```
Mime-Version: 1.0
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: 8bit
```

Esim. elm-ohjelmassa nämä otsikot laitetaan tiedostoon `$HOME/.elm/elmheaders`. emacs-pohjaisilla posti- ja nyyssisovelluksilla asetus tapahtuu helpoiten laittamalla seuraavat rivit `.emacs`-tiedostoon:

```
(setq mail-default-headers "Mime-Version: 1.0\n\
  Content-Type: text/plain; charset=ISO-8859-1\n\
  Content-Transfer-Encoding: 8bit\n")
```

8.2 MIME Quoted Printable-koodaus

Yleisin ongelma viestien vastaanotossa ovat MIME Quoted Printable -koodatut merkit [RFC1521]. Näitä kutsutaan usein sattuvasti myös nimellä Quoted Unreadable.

QP-koodattuja viestejä ei kannata lähettää, jos 8-bittisten merkkien lähettäminen sellaisenaan vain on

mahdollista, koska varsin monet lukijat n=E4kev=E4t QP-viestit t=E4m=E4n n=E4k=F6isin=E4. QP-koodatut - samoin kuin kokonaan binääridataa sisältävät Base64-koodatut viestit - saa purettua lukukelpoisiksi metamail-ohjelmalla [metamail]. metamail tulee valmiina ainakin RedHat 2.1-jakelun mukana.

elm (ainakin versio 2.4 PL24) osaa käyttää metamailia suoraan, mikäli metamail on polun varrella eli ajettavissa. emacs-pohjaiset posti- ja nyyssisovellukset saattavat tarvita tiedon metamailin sijainnista `.emacs`-tiedostossa esim. näin:

```
(setq metamail-program-name "/usr/local/bin/metamail")
```

9. Aiheeseen liittyvää materiaalia

Seuraavat dokumentit voivat olla hyödyllistä luettavaa näiden ongelmien kanssa painiessasi:

1. [skandit] **Suomalaiset merkit tietokoneissa**, Timo Kiravuo: http://www.nixu.fi/~kiravuo/skandit.html
2. [ISO-latin1] **ISO-Latin-1-merkistöstä**, Jukka Korpela: http://www.hut.fi/~jkorpela/ISO.html
3. [texref] **TeX-ladontaohjelma**, Jukka Korpela: http://www.hut.fi/~jkorpela/softa/tex.html
4. [danish] **Danish-HOWTO**, Thomas Petersen: http://www.funet.fi/pub/Linux/doc/HOWTO/Danish-HOWTO
5. [german] **German-HOWTO**, Thomas Ploss: http://www.funet.fi/pub/Linux/doc/HOWTO/German-HOWTO
6. [RFC822] **Format of Internet Text Messages**, David Crocker: http://www.funet.fi/pub/doc/rfc/rfc822.txt
7. [RFC1521] **MIME, Part One: Format of Internet Message Bodies**, N. Borenstein & N. Freed: http://www.funet.fi/pub/doc/rfc/rfc1521.txt
8. [RFC1522] **MIME, Part Two: Message Header Extensions**, K. Moore: http://www.funet.fi/pub/doc/rfc/rfc1522.txt
9. [metamail] **Metamail-ohjelma**: ftp://ftp.funet.fi/pub/unix/mail/metamail/mm2.7.tar.Z
10. [linuxdoc] **Linuxdoc-SGML**, Matthew D. Welsh: ftp://ftp.funet.fi:pub/mirrors/sunsite.unc.edu/pub/Linux/utils/text/linuxdoc-sgml-1.4.tar.gz
11. [unixhelp] **UNIXHelp for Users**, University of Edinburgh: http://unixhelp.ed.ac.uk/
12. [faqs] **FAQ Collection**, Thomas Fine: http://www.cis.ohio-state.edu/hypertext/faq/usenet/FAQ-List.html

Firewalling and Proxy Server HOWTO

Mark Grennan, `markg@netplus.net`

v0.4, 8 November 1996

This document is designed to teach the basics of firewall systems and give you some detail on setting up both a filtering and proxy firewall on a Linux based PC. An HTML version of this document is available at http://okcforum.org/~markg/Firewall-HOWTO.html

1. Introduction

- 1.1 Feedback
- 1.2 Disclaimer
- 1.3 Copyright
- 1.4 My Reasons for Writing This
- 1.5 TODO
- 1.6 Further Readings

2. Understanding Firewalls

- 2.1 Drawbacks with Firewalls
- 2.2 Types of Firewalls

3. Setting up the Firewall

- 3.1 Hardware requirements

4. Firewalling Software

- 4.1 Available packages
- 4.2 The TIS Firewall Toolkit vs SOCKS

TOC

5. Preparing the Linux system

- 5.1 Compiling the Kernel
- 5.2 Configuring two network cards
- 5.3 Configuring the Network Addresses
- 5.4 Testing your network
- 5.5 Securing the Firewall

6. IP filtering setup (IPFWADM)

7. Installing the TIS Proxy server

- 7.1 Getting the software
- 7.2 Compiling the TIS FWTK
- 7.3 Installing the TIS FWTK
- 7.4 Configuring the TIS FWTK

8. The SOCKS Proxy Server

- 8.1 Setting up the Proxy Server
- 8.2 Configuring the Proxy Server
- 8.3 Working With a Proxy Server
- 8.4 Drawbacks with Proxy Servers

9. Advanced Configurations

- 9.1 A large network with emphasis on security

---

1. Introduction

This original Firewall-HOWTO was written by David Rudder, **drig@execpc.com**. I'd like to thank him for allowing me to update his work.

Firewalls have gained great fame recently as the ultimate in Internet Security. Like most things that gain fame, with that fame has come misunderstanding. This HOWTO will go over the basics of what a firewall is, how to set one up, what proxy servers are, how to set up proxy servers, and the applications of this technology outside of the security realm.

1.1 Feedback

Any feedback is very welcome. **PLEASE REPORT ANY INACCURACIES IN THIS PAPER!!!** I am human, and prone to making mistakes. If you find any, fixing them is of my highest interest. I will try to answer all e-mail, but I am busy, so don't get insulted if I don't.

My email address is ***markg@netplus.net***

1.2 Disclaimer

I AM NOT RESPONSIBLE FOR ANY DAMAGES INCURRED DUE TO ACTIONS TAKEN BASED ON THIS DOCUMENT. This document is meant as an introduction to how firewalls and proxy servers work. I am not, nor do I pretend to be, a security expert. I am just some guy who has read to much and likes computers more than most people. Please, I am writing this to help get people acquainted with this subject, and I am not ready to stake my life on the accuracy of what is in here.

1.3 Copyright

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have any questions, please contact Mark Grennan at <markg@netplus.net>.

1.4 My Reasons for Writing This

Even though there were a lot of discussions on comp.os.linux.* over the past year about firewalling, I found it difficult to find the information I needed to setup a firewall. The original version of this HOWTO was helpful but still lacking. I hope this beefed up version of David Rudder's Firewall HOWTO will give everyone the information they need to create a functioning firewall in hours, not weeks.

I also feel I should return something to the Linux community.

1.5 TODO

- Give some instructions on how to setup the clients
- Find a good UDP proxy server that works with Linux

1.6 Further Readings

- The NET-2 HOWTO
- The Ethernet HOWTO
- The Multiple Ethernet Mini HOWTO
- Networking with Linux
- The PPP HOWTO
- TCP/IP Network Administrator's Guide by O'Reilly and Associates
- The Documentation for the TIS Firewall Toolkit

Trusted Information System's (TIS) web site has a great collection of documentation on firewalls and related meterial. **http://www.tis.com/**

Also, I am working on a security project called I am calling *Secure Linux*. On the *Secure Linux* web site I am gathering all the information, documemtation and programs you need to create a trusted Linux system. Email me if you would like information.

---

2. Understanding Firewalls

A firewall is a term used for a part of a car. In cars, firewalls are physical objects that separate the engine from the passengers. They are meant to protect the passenger in case the car's engine catches fire while still providing the driver access to the engine's controls.

A firewall in computers is a device that protects a private network from the public part (the internet as a whole).

The firewall computer, from now on named "firewall", can reach both the protected network and the internet. The protected network can't reach the internet, and the internet can not reach the protected network.

For someone to reach the internet from inside the protected network, they must telnet to firewall, and use the internet from there.

The simplest form of a firewall is a dual homed system. (a system with two network connections) If you can TRUST ALL your users, you can simple setup a Linux (compile it with IP forwarding/gatewaying turned OFF!) and give everyone accounts on it. The can then login to this system and telnet, FTP, read mail, and use any other service you provided. With this setup, the only computer on your private network that knows anything about the outside world is the firewall. The other system on your protected network dont even need a default route.

This needs re-stating. For the above firewall to work **YOU MUST TRUST ALL YOUR USERS!** I don't recommend it.

2.1 Drawbacks with Firewalls

The problem with filtering firewalls are they inhibit the access to your network from the internet. Only services on systems that have pass filters can be accessed. With a proxy server users can login to the firewall and then access any system within the private network they have access to.

Also, new types of network clients and servers a coming out almost daily. When they do you must find a new way to allow controled access before these services can be used.

2.2 Types of Firewalls

There are two types of firewalls.

1. IP or Filtering Firewalls - that block all but selected network traffic.
2. Proxy Servers - that make the network connections for you.

TOC

IP Filtering Firewalls

An IP filtering firewall works at the packet level. It is designed to control the flow of packets based the source, destination, port and packet type information contained in each packet.

This type of firewall is very secure but lacks any sort of useful logging. It can block people from accessing private system but it will not tell you who accessed your public systems or who accessed the internet from the inside.

Filtering firewalls are absolute filters. Even if you want to give someone on outside access to your private servers you can not without giving everyone access to the servers.

Linux has included packet filtering software in the kernel starting with version 1.3.x.

Proxy Servers

Proxy servers allow indirect internet access through the firewall. The best example of how this works is a person telneting to a system and then telneting from there to another. Only with a proxy server the process is automatic. When you connect to a proxy server with your client software, the proxy server starts it's client (proxy) software and passes you the data.

Because proxy servers are duplicating all the communications they can log every thing they do.

The great thing about proxy servers is that they are completely secure, when configured correctly. They will not allow someone in through them. There are no direct IP routes.

---

3. Setting up the Firewall

3.1 Hardware requirements

For our example, the computer is a 486-DX66 with 16 meg of memory and a 500 meg Linux partition. This system has two network cards one connected to our private LAN and the other connected to the a lan we will call the de-militarized zone (DMZ). The DMZ has a router connected to it with a connection to the internet.

This is a pretty standard setup for a business. You could use one network card and a modem with PPP to the internet. The point is, the firewall must have two IP network numbers.

I know a lot of people have small LANs at home with two or three computers on them. Something you might consider is putting all your modems in on Linux box (maybe an old 386) and connecting all of them to the internet with load balancing. With this setup when only one person was pulling data they would get both modems doubling the throughput. :-)

4. Firewalling Software

4.1 Available packages

If all you want is a filtering firewall, you only need Linux and the basic networking packages. One package that might not come with your distribution is the IP Firewall Administration tool.

(IPFWADM) Comes from **http://www.xos.nl/linux/ipfwadm/**

If you want to setup a poxy server you will need one of these packages.

1. SOCKS
2. TIS Firewall Toolkit (FWTK)

4.2 The TIS Firewall Toolkit vs SOCKS

Trusted Information System (**http://www.tis.com**) has put out a collection of programs designed to facilitate firewalling. The programs do basically the same thing as the SOCKS package, but with a different design strategy. Where Socks has one program that covers all Internet transactions, TIS has provided one program for each utility that wishes to use the firewall.

To contrast the two, let's use the example of world wide web and Telnet access. With SOCKS, you set up one configuration file and one daemon. Through this file and daemon, both telnet and WWW are enabled, as well as any other service that you have not disabled.

With the TIS toolkit, you set up one daemon for each WWW and telnet, as well as configuration files for each. After you have done this, other internet access is still prohibited until explicitly set up. If a daemon for a specific utility has not been provided (like talk), there is a "plug-in" daemon, but it is neither as flexible, nor as easy to set up, as the other tools.

This might seem a minor, but it makes a major difference. SOCKS allows you to be sloppy. With a poorly set up SOCKS server, someone from the inside could gain more access to the internet than was originally intended. With the TIS toolkit, the people on the inside have only the access the system administrator wants them to have.

SOCKS is easier to set up, easier to compile and allows for greater flexibility. The TIS toolkit is more secure if you want to regulate the users inside the protected network. Both provide absolute protection from the outside.

I will cover the installation and setup of both.

5. Preparing the Linux system

5.1 Compiling the Kernel

Start with a clean installation of your Linux distribution. (I used RedHat 3.0.3 and the examples here are based on this distribution.) The less software you have loaded the less holes, backdoors and/or bugs there will be to introduce security problems in your system, so load only a minimal set of applications.

Pick a stable kernel. I used the Linux 2.0.14 kernel for my system. So this documentation is based on it's settings.

You well need to recompile the Linux kernel with the appropriate options. At this point, you should look at the Kernel HOWTO, the Ethernet HOWTO, and the NET-2 HOWTO if you haven't done this before.

Here are the network related setting I know work in 'make config'

1. Under General setup
    1. Turn Networking Support ON
2. Under Networking Options
    1. Turn Network firewalls ON
    2. Turn TCP/IP Networking ON
    3. Turn IP forwarding/gatewaying OFF (UNLESS you wish to use IP filtering)
    4. Turn IP Firewalling ON
    5. Turn IP firewall packet loggin ON (this is not required but it is a good idea)
    6. Turn IP: masquerading OFF (I am not covering this subject here.)
    7. Turn IP: accounting ON
    8. Turn IP: tunneling OFF
    9. Turn IP: aliasing OFF
    10. Turn IP: PC/TCP compatibility mode OFF
    11. Turn IP: Reverse ARP OFF
    12. Turn Drop source routed frames ON
3. Under Network device support
    1. Turn Network device support ON
    2. Turn Dummy net driver support ON
    3. Turn Ethernet (10 or 100Mbit) ON
    4. Select your network card

Now you can recompile, reinstall the kernel and reboot. Your network card/s should show up in the boot-up sequence. If not, go over the other HOWTOs again until it is working.

5.2 Configuring two network cards

If you have two network cards in your computer, you most likely will need to add an append statement to your /etc/lilo.conf file to describe the IRQ and address of both cards. My lilo append statement looks like this:

```
append="ether=12,0x300,eth0 ether=15,0x340,eth1"
```

5.3 Configuring the Network Addresses

This is the real interesting part. Now you have a few decisions to make. Since we don't want the internet to have access to any part of the private network, we do not need to use real addresses. There are a number of internet addresses set aside for private networks. Because everyone needs more addresses and because these addresses can not cross the Internet they are a good choice.

Of these, 192.168.2.xxx, is set aside and we will use it in our examples.

Your proxy firewall will be a member of both networks and so it can pass the data through to and from the private network.

```
        199.1.2.10   __________    192.168.2.1
  _  __  _         \ |          | /          ______________
 | \/  \/ |        \| Firewall |/           |              |
/ Internet \--------|  System  |------------| Workstation/s |
\_/\_/\_/\_/        |__________|            |______________|
```

If your going to use a filtering firewall you can still use these numbers. You will need to use IP masquerading to make this happen. With this process the firewall will forward packets and translate them into "REAL " " IP address to travel on the Internet.

You must assign the real IP address to the network card on the Internet (out) side. And, assign 192.168.2.1 to the Ethernet card on inside. This will be your proxy/gateway IP address. You can assign all the other machines in the protected network some number in that 192.168.2.xxx range. (192.168.2.2 through 192.168.2.254)

Since I use RedHat Linux (Hey guys, want to give me a copy for the plugs? ;-) to configure the network at boot time I added a 'ifcfg-eth1' file in the /etc/sysconfig/network-scripts directory. This file is read during the boot process to set your network and routing tables.

Here is what my ifcfg-eth1 looks like;

```
#!/bin/sh
#>>>Device type: ethernet
#>>>Variable declarations:
DEVICE=eth1
IPADDR=192.168.2.1
NETMASK=255.255.255.0
NETWORK=192.168.2.0
BROADCAST=192.168.2.255
GATEWAY=199.1.2.10
ONBOOT=yes
#>>>End variable declarations
```

You can also use these scripts to automatically connect by modem to your provider. Look at the ipup-ppp script.

If your going to use a modem for your internet connection your outside IP address will be assigned for you by your provider at connect time.

5.4 Testing your network

TOC

Start by checking ifconfig and route. If you have two network cards your ifconfig should look something like:

```
#ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.0  Bcast:127.255.255.255  Mask:255.0.0.0
          UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
          RX packets:1620 errors:0 dropped:0 overruns:0
          TX packets:1620 errors:0 dropped:0 overruns:0

eth0      Link encap:10Mbps Ethernet  HWaddr 00:00:09:85:AC:55
          inet addr:199.1.2.10 Bcast:199.1.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
          Interrupt:12 Base address:0x310

eth1      Link encap:10Mbps Ethernet  HWaddr 00:00:09:80:1E:D7
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
          Interrupt:15 Base address:0x350
```

and your route table sould look like:

```
#route -n
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
199.1.2.0       *               255.255.255.0   U     1500   0       15 eth0
192.168.2.0     *               255.255.255.0   U     1500   0        0 eth1
127.0.0.0       *               255.0.0.0       U     3584   0        2 lo
default         199.1.2.10      *               UG    1500   0       72 eth0
```

Note: 199.1.2.0 is the Internet side of this firewall and 192.168.2.0 is the private side.

Now try to ping the internet from the firewall. I used to use nic.ddn.mil as my test point. It's still a good test, but has proven to be less reliable than I had hoped. If it doesn't work at first, try pinging a couple other places that are not connected to your LAN. If this doesn't work, then your PPP is incorrectly setup. Reread the Net-2 HOWTO, and try again.

Next, try pinging a host within the protected network from the firewall. All the computers should be able to ping each other. If not, go over the NET-2 HOWTO again and work on the network some more.

Then, try to ping the outside address of firewall from inside the protected network. (NOTE: this is not any of the 192.168.2.xxx IP numbers.) If you can, then you have not turned off IP Forwarding. Make sure this is the way you want it. If you leave it turned on you will have to go through the IP filtering section of this document as well.

Now try pinging the internet from behind your firewall. Use the same address that worked for you before. (I.E. nic.ddn.mil) Again, if you have IP Forwarding turned off, this should not work. But, if you have it turned on, it should.

If have IP Forwarding turned on and your using a "REAL" (not 192.168.2.*) IP address for your private network you can't ping the internet but you can ping the internet side your firewall, check if the next router up stream is routing packets for your private network address. (Your provider may have to do this for you.)

If you have assigned your protected network to 192.168.2.*, then no can packets can be routed to it anyway. If you have skipped ahead and you already have IP masquerading turn on, this test should work.

Now, you have your basic system setup.

5.5 Securing the Firewall

A firewall isn't any good if it is left wide open to attacks through a unused service. A "bad guy" could gain access to the firewall and modify it for their own needs.

Start by turning off any unneeded services. Look at /etc/inetd.conf file. This file controls what are called the "super server". It controls a bunch of the server daemons and starts them as they are requested.

Definitely turn off netstat, systat, tftp, bootp, and finger. To turn a service off, put # as the first character of the service line. When your done, send a SIG-HUP to the process by typing **"kill -HUP <pid>"**, where <pid> is the process number of inetd. This will make inetd re-read its configuration file (inetd.conf) and restart.

Test it out by telneting to port 15 on firewall, the netstat port. If you get an output of netstat, you have not restarted it correctly.

---

6. IP filtering setup (IPFWADM)

To start, you should have IP Forwarding turned on in your kernel and your system should be up and forwarding everything you send it. Your routing tables should be in place and you should be able to access everything, both from the inside out and from the outside in.

But, we're building a firewall so we need to start chocking down what everyone has access to.

In my system I created a couple of scripts to set the firewall forwarding policy and accounting policy. I call theses scripts from the /etc/rc.d scripts so my system is configured at boot time.

By default the IP Forwarding system in the Linux kernel forwards everything. Because of this, your firewall script should start by denying access to everything and flushing any ipfw rules in place from the last time it was run. This script will do the trick.

```
#
# setup IP packet Accounting and Forwarding
#
#   Forwarding
#
# By default DENY all services
ipfwadm -F -p deny
# Flush all commands
ipfwadm -F -f
ipfwadm -I -f
ipfwadm -O -f
```

Now we have the ultimate firewall. Nothing can get through. No doubt you have some services you need to forward so here are a few examples you should find useful.

```
# Forward email to your server
ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 192.1.2.10 25

# Forward email connections to outside email servers
ipfwadm -F -a accept -b -P tcp -S 196.1.2.10 25 -D 0.0.0.0/0 1024:65535

# Forward Web connections to your Web Server
/sbin/ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 196.1.2.11 80

# Forward Web connections to outside Web Server
/sbin/ipfwadm -F -a accept -b -P tcp -S 196.1.2.* 80 -D 0.0.0.0/0 1024:65535

# Forward DNS traffic
/sbin/ipfwadm -F -a accept -b -P udp -S 0.0.0.0/0 53 -D 196.1.2.0/24
```

You might also be interested in accounting for traffic going through your firewall. This script will count ever packet. You could add a line or to to account for packets going to just a single system.

```
# Flush the current accounting rules
ipfwadm -A -f
# Accounting
/sbin/ipfwadm -A -f
/sbin/ipfwadm -A out -i -S 196.1.2.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -A out -i -S 0.0.0.0/0 -D 196.1.2.0/24
/sbin/ipfwadm -A in -i -S 196.1.2.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -A in -i -S 0.0.0.0/0 -D 196.1.2.0/24
```

If all you wanted was a filtering firewall you can stop here. Enjoy :-)

---

7. Installing the TIS Proxy server

7.1 Getting the software

The TIS FWTK is avaible at **ftp://ftp.tis.com/**.

Don't make the mistake I did. When you ftp files from TIS, READ THE README's. The TIS fwtk is locked up in a hidden directory on their server. TIS requires you **send email to fwtk-request@tis.com** with only the word **SEND** in the body of the message to learn the name of this hidden directory. No subject is needed in the message. Their system will then mails you back the directory name (good for 12 hours) to download the source.

As I'm writing this TIS is releasing version 2.0 (beta) of the FWTK. This version seems to compile well (with a few exceptions) and everything is working for me. This is the version I will be covering here. When they release the final code I'll update the HOWTO.

To install the FWTK, create a fwtk-2.0 directory in your /usr/src directory. Move your copy of the FWTK (fwtk-2.0.tar.gz) to your this directory and untar it (tar zxf fwtk-2.0.tar.gz).

The FWTK does not proxy SSL web documents but there is an addon for it written by Jean-Christophe Touvet. It is avaible at **ftp://ftp.edelweb.fr/pub/contrib/fwtk/ssl-gw.tar.Z**. Touvet does not support this code.

I am using a modified version that includes access to Netscape secure news servers written by Eric Wedel. It is available at **ftp://mdi.meridian-data.com/pub/tis.fwtk/ssl-gw/ssl-gw2.tar.Z**.

In our example I will use Eric Wedel's version.

To install it, simply create a ssl-gw directory in your /usr/src/fwtk-2.0 directory and put the files in it.

When I installed this gateway it required a few changes before it would compile with the rest of the toolkit.

The first change was to the ssl-gw.c file. I found it didn't include a needed include file.

```
#if defined(__linux)
#include        <sys/ioctl.h>
#endif
```

Second it didn't come with a Makefile. I copied one out of the other gateway directories and replaced the gateway's name with ssl-gw.

7.2 Compiling the TIS FWTK

Version 2.0 of the FWTK compiles much easier then any of the older versions. I still found a couple of things that needed to be changed before the BETA version would compile cleanly. Hopefully these changes will be make in the final version.

To fix it up, start by changing to the /usr/src/fwtk/fwtk directory and coping the Makefile.config.linux file over the Makefile.config file.

DON'T RUN FIXMAKE. The instructions tell you to run this. If you do it will break the makefiles in each directory.

I do have a fix for fixmake. The problem is the sed script add a '.' and '' to the include line of ever Makefile. This sed script works.

```
sed 's/^include[        ]*\([^  ].*\)/include \1/' $name .proto > $name
```

Next we need to edit the Makefile.config file. There are two changes you may need to make.

The author set the source directory to his home directory. We are compiling our code in /usr/src so you should changed the FWTKSRCDIR variable to reflect this.

```
FWTKSRCDIR=/usr/src/fwtk/fwtk
```

Second, at least some Linux system us the gdbm database. The Makefile.config is using dbm. You might need to change this. I had to for RedHat 3.0.3.

```
DBMLIB=-lgdbm
```

The last fix is in the x-gw. The bug in the BETA version is in the socket.c code. To fix it remove these lines of code.

```
#ifdef SCM_RIGHTS  /* 4.3BSD Reno and later */
                     + sizeof(un_name->sun_len) + 1
#endif
```

If you added the ssl-gw to your FWTK source directory you will need to add it to the list of directory in the Makefile.

```
DIRS=   smap smapd netacl plug-gw ftp-gw tn-gw rlogin-gw http-gw x-gw ssl-gw
```

Now run **make**.

7.3 Installing the TIS FWTK

Run **make install**.

The default installation directory is /usr/local/etc. You could change this (I didn't) to a more secure directory. I chose to change the access to this directory to 'chmod 700'.

All last is left now is to configure the firewall.

7.4 Configuring the TIS FWTK

Now the fun realy begins. We must teach the system to call theses new services and create the tables to control them.

I'm not going to try to re-write the TIS FWTK manual here. I will show you the setting I found worked and explain the problems I ran into and how I got around them.

There are three files that make up these controls.

- /etc/services
  - Tells the system what ports a services is on.
- /etc/inetd.conf
  - Tells inetd what program to call when someone knocks on a service port.
- /usr/local/etc/netperm-table
  - Tells the FWTK services who to allow and deny service to.

To get the FWTK functioning, you should edit these files from the bottom up. Editing the services file without the inetd.conf or netperm-table file set correctly could make your system inaccessible.

The netperm-table file

This file controls who can access the services of the TIS FWTK. You should think about the traffic using the firewall from both sides. People outside your network should identify themselves before gaining access, but the people inside your network might be allowed to just pass through.

TOC

So people can identify themselves, the firewall uses a program called **authsrv** to keep a database of user IDs and passwo The authentication section of the netperm-table controls where the database is keep and who can access it.

I had some trouble closing the access to this service. Note the premit-hosts line I show uses a '*' to give everyone access. The correct setting for this line is " `authsrv: premit-hosts localhost` if you can get it working.

```
#
# Proxy configuration table
#
# Authentication server and client rules
authsrv:      database /usr/local/etc/fw-authdb
authsrv:      permit-hosts *
authsrv:      badsleep 1200
authsrv:      nobogus true
# Client Applications using the Authentication server
*:            authserver 127.0.0.1 114
```

To initialize the database, su to root, and run **./authsrv** in the /var/local/etc directory to create the administrative user record. Here is a sample session.

Read the FWTK documentation to learn how to add users and groups.

```
#
# authsrv
authsrv# list
authsrv# adduser admin "Auth DB admin"
ok - user added initially disabled
authsrv# ena admin
enabled
authsrv# proto admin pass
changed
authsrv# pass admin "plugh"
Password changed.
authsrv# superwiz admin
set wizard
authsrv# list
Report for users in database
user   group  longname           ok?    proto   last
------ ------ ------------------ -----  ------  -----
admin         Auth DB admin      ena    passw   never
authsrv# display admin
Report for user admin (Auth DB admin)
Authentication protocol: password
Flags: WIZARD
authsrv# ^D
EOT
#
```

The telnet gateway (tn-gw) controls are straight forward and the first you should set up.

In my example, I premit host from inside the private network to pass through without authenticating themselves. (permit-hosts 19961.2.* -passok) But, any other user must enter their user ID and password to use the proxy. (permit-hosts * -auth)

I also allow one other system (196.1.2.202) to access the firewall directly without going through the firewall at all. The two inetacl-in.telnetd lines do this. I will explain how these lines are called latter.

The Telnet timeout should be keep short.

```
# telnet gateway rules:
tn-gw:                  denial-msg      /usr/local/etc/tn-deny.txt
tn-gw:                  welcome-msg     /usr/local/etc/tn-welcome.txt
tn-gw:                  help-msg        /usr/local/etc/tn-help.txt
tn-gw:                  timeout 90
tn-gw:                  permit-hosts 196.1.2.* -passok -xok
tn-gw:                  permit-hosts * -auth
# Only the Administrator can telnet directly to the Firewall via Port 24
netacl-in.telnetd: permit-hosts 196.1.2.202 -exec /usr/sbin/in.telnetd
```

The r-commands work the same way as telnet.

```
# rlogin gateway rules:
rlogin-gw:    denial-msg      /usr/local/etc/rlogin-deny.txt
rlogin-gw:    welcome-msg     /usr/local/etc/rlogin-welcome.txt
rlogin-gw:    help-msg        /usr/local/etc/rlogin-help.txt
rlogin-gw:    timeout 90
rlogin-gw:    permit-hosts 196.1.2.* -passok -xok
rlogin-gw:    permit-hosts * -auth -xok
# Only the Administrator can telnet directly to the Firewall via Port
netacl-rlogind: permit-hosts 196.1.2.202 -exec /usr/libexec/rlogind -a
```

You shouldn't have anyone accessing your firewall directly and that includes FTP so don't put an FTP, server on you firewall.

Again, the permit-hosts line allows anyone in the protected network free access to the Internet and all others must authenticate themselves. I included logging of every file sent and received to my controls. (-log { retr stor })

The ftp timeout controls how long it will take to drop a bad connections as well as how long a connection will stay open with out activity.

```
# ftp gateway rules:
ftp-gw:                 denial-msg      /usr/local/etc/ftp-deny.txt
ftp-gw:                 welcome-msg     /usr/local/etc/ftp-welcome.txt
ftp-gw:                 help-msg        /usr/local/etc/ftp-help.txt
ftp-gw:                 timeout 300
ftp-gw:                 permit-hosts 196.1.2.* -log { retr stor }
ftp-gw:                 permit-hosts * -authall -log { retr stor }
```

Web, gopher and browser based ftp are contorted by the http-gw. The first two lines create a directory to store ftp and web documents as they are passing through the firewall. I make these files owned by root and put the in a directory accessible only by root.

The Web connection should be kept short. It controls how long the user will wait on a bad connections.

```
# www and gopher gateway rules:
http-gw:      userid          root
http-gw:      directory       /jail
http-gw:      timeout 90
http-gw:      default-httpd   www.afs.net
http-gw:      hosts           196.1.2.* -log { read write ftp }
http-gw:      deny-hosts      *
```

The ssl-gw is really just a pass anything gateway. Be carefull with it. In this example I allow anyone inside the protected network to connect to any server outside the network except the addresses 127.0.0.* and 192.1.1.* and then only on ports 443 through 563. Ports 443 through 563 are known SSL ports.

TOC

```
# ssl gateway rules:
ssl-gw:         timeout 300
ssl-gw:         hosts           196.1.2.* -dest { !127.0.0.* !192.1.1.* *:443:563 }
ssl-gw:         deny-hosts      *
```

Here is an example of how to use the plug-gw to allow connections to a news server. In this example I allow anyone inside the protected network to connect to only one system and only to it's news port.

The seconded line allows the news server to pass its data back to the protected network.

Because most clients expect to stay connected while the user read news, the timeout for a news server should be long.

```
# NetNews Pluged gateway
plug-gw:        timeout 3600
plug-gw: port nntp 196.1.2.* -plug-to 199.5.175.22 -port nntp
plug-gw: port nntp 199.5.175.22 -plug-to 196.1.2.* -port nntp
```

The finger gateway is simple. Anyone inside the protected network must login first and then we allow them to use the finger program on the firewall. Anyone else just gets a message.

```
# Enable finger service
netacl-fingerd: permit-hosts 196.1.2.* -exec /usr/libexec/fingerd
netacl-fingerd: permit-hosts * -exec /bin/cat /usr/local/etc/finger.txt
```

I haven't setup the Mail and X-windows services so I'm not including examples. If anyone has a working example, please send me email.

The inetd.conf file

Here is a complete /etc/inetd.conf file. All un-needed services have been commented out. I have included the complete file to show what to turn off, as well as how to setup the new firewall services.

```
#echo stream  tcp  nowait  root       internal
#echo dgram   udp  wait    root       internal
#discard      stream  tcp  nowait  root       internal
#discard      dgram   udp  wait    root       internal
#daytime      stream  tcp  nowait  root       internal
#daytime      dgram   udp  wait    root       internal
#chargen      stream  tcp  nowait  root       internal
#chargen      dgram   udp  wait    root       internal
# FTP firewall gateway
ftp-gw      stream  tcp  nowait.400  root  /usr/local/etc/ftp-gw  ftp-gw
# Telnet firewall gateway
telnet        stream  tcp  nowait      root  /usr/local/etc/tn-gw
/usr/local/etc/tn-gw
# local telnet services
telnet-a    stream  tcp  nowait      root  /usr/local/etc/netacl in.telnetd
# Gopher firewall gateway
gopher        stream  tcp  nowait.400  root  /usr/local/etc/http-gw
/usr/local/etc/http-gw
# WWW firewall gateway
http  stream  tcp  nowait.400  root  /usr/local/etc/http-gw /usr/local/etc/http-gw
# SSL firewall gateway
```

```
ssl-gw  stream  tcp     nowait  root /usr/local/etc/ssl-gw   ssl-gw
# NetNews firewall proxy (using plug-gw)
nntp    stream  tcp     nowait  root    /usr/local/etc/plug-gw plug-gw nntp
#nntp stream  tcp     nowait  root    /usr/sbin/tcpd  in.nntpd
# SMTP (email) firewall gateway
#smtp stream  tcp     nowait  root    /usr/local/etc/smap smap
#
# Shell, login, exec and talk are BSD protocols.
#
#shell        stream  tcp     nowait  root    /usr/sbin/tcpd  in.rshd
#login        stream  tcp     nowait  root    /usr/sbin/tcpd  in.rlogind
#exec stream  tcp     nowait  root    /usr/sbin/tcpd  in.rexecd
#talk dgram   udp     wait    root    /usr/sbin/tcpd  in.talkd
#ntalk        dgram   udp     wait    root    /usr/sbin/tcpd  in.ntalkd
#dtalk        stream  tcp     waut    nobody  /usr/sbin/tcpd  in.dtalkd
#
# Pop and imap mail services et al
#
#pop-2   stream  tcp  nowait  root  /usr/sbin/tcpd    ipop2d
#pop-3   stream  tcp  nowait  root  /usr/sbin/tcpd    ipop3d
#imap    stream  tcp  nowait  root  /usr/sbin/tcpd    imapd
#
# The Internet UUCP service.
#
#uucp    stream  tcp  nowait  uucp  /usr/sbin/tcpd  /usr/lib/uucp/uucico -l
#
# Tftp service is provided primarily for booting.  Most sites
# run this only on machines acting as "boot servers." Do not uncomment
# this unless you *need* it.
#
#tftp dgram   udp     wait    root    /usr/sbin/tcpd  in.tftpd
#bootps       dgram   udp     wait    root    /usr/sbin/tcpd  bootpd
#
# Finger, systat and netstat give out user information which may be
# valuable to potential "system crackers."  Many sites choose to disable
# some or all of these services to improve security.
#
# cfinger is for GNU finger, which is currently not in use in RHS Linux
#
finger        stream  tcp  nowait  root   /usr/sbin/tcpd  in.fingerd
#cfinger      stream  tcp  nowait  root   /usr/sbin/tcpd  in.cfingerd
#systat       stream  tcp  nowait  guest  /usr/sbin/tcpd  /bin/ps -auwwx
#netstat      stream  tcp  nowait  guest  /usr/sbin/tcpd  /bin/netstat -f inet
#
# Time service is used for clock syncronization.
#
#time stream  tcp  nowait  root  /usr/sbin/tcpd  in.timed
#time dgram   udp  wait    root  /usr/sbin/tcpd  in.timed
#
# Authentication
#
auth          stream  tcp  wait    root  /usr/sbin/tcpd  in.identd -w -t120
authsrv       stream  tcp  nowait  root  /usr/local/etc/authsrv authsrv
#
# End of inetd.conf
```

The /etc/services file

This is where it all begins. When a client connects to the firewall it connects on a known port (less then 1024). For example telnet connects on port 23. The inetd deamon hears this connection and looks up the name of these service in the /etc/services file. It then calls the program assigned to the name in the /etc/inetd.conf file.

Some of the services we are creating are not normally in the /etc/services file. You can assign some of them to any port you want. For example, I have assigned the administrator's telnet port (telnet-a) to port 24. You could assign it to port 2323 if you wished. For the administrator (YOU) to connect directly to the firewall you will need to telnet to port 24 not 23 and if you setup your netperm-table file, like I did, you will only be able to to this from one system inside your protected network.

```
telnet-a        24/tcp
ftp-gw          21/tcp           # this named changed
auth            113/tcp   ident    # User Verification
ssl-gw          443/tcp
```

8. The SOCKS Proxy Server

8.1 Setting up the Proxy Server

The SOCKS proxy server available from **ftp://sunsite.unc.edu/pub/Linux/system/Network/misc/socks-linux- src.tgz**. There is also an example config file in that directory called "socks-conf". Uncompress and untar the files into a directory on your system, and follow the instructions on how to make it. I had a couple problems when I made it. Make sure that your Makefiles are correct.

One important thing to note is that the proxy server needs to be added to /etc/inetd.conf. You must add a line:

```
socks  stream  tcp  nowait  nobody  /usr/local/etc/sockd  sockd
```

to tell the server to run when requested.

8.2 Configuring the Proxy Server

The SOCKS program needs two separate configuration files. One to tell the access allowed, and one to route the requests to the appropriate proxy server. The access file should be housed on the server. The routing file should be housed on every Un*x machine. The DOS and, presumably, Macintosh computers will do their own routing.

The Access File

With socks4.2 Beta, the access file is called "sockd.conf". It should contain 2 lines, a permit and a deny line. Each line will have three entries:

- The Identifier (permit/deny)
- The IP address
- The address modifier

The identifier is either permit or deny. You should have both a permit and a deny line.

The IP address holds a four byte address in typical IP dot notation. I.E. 192.168.2.0.

The address modifier is also a typical IP address four byte number. It works like a netmask. Envision this number to be 32 bits (1s or 0s). If the bit is a 1, the corresponding bit of the address that it is checking must match the corresponding bit in the IP address field. For instance, if the line is:

```
permit 192.168.2.23 255.255.255.255
```

it will permit only the IP address that matches every bit in 192.168.2.23, eg, only 192.168.2.3. The li

```
permit 192.168.2.0 255.255.255.0
```

will permit every number within group 192.168.2.0 through 192.168.2.255, the whole C Class domain. One should not have the line:

```
permit 192.168.2.0 0.0.0.0
```

as this will permit every address, regardless.

So, first permit every address you want to permit, and then deny the rest. To allow everyone in the domain 192.168.2.xxx, the lines:

```
permit 192.168.2.0 255.255.255.0
deny 0.0.0.0 0.0.0.0
```

will work nicely. Notice the first "0.0.0.0" in the deny line. With a modifier of 0.0.0.0, the IP address field does not matter. All 0's is the norm because it is easy to type.

More than one entry of each is allowed.

Specific users can also be granted or denied access. This is done via ident authentication. Not all systems support ident, including Trumpet Winsock, so I will not go into it here. The documentation with socks is quite adequate on this subject.

The Routing File

The routing file in SOCKS is poorly named "socks.conf". I say "poorly named" because it is so close to the name of the access file that it is easy to get the two confused.

The routing file is there to tell the SOCKS clients when to use socks and when not to. For instance, in our network, 192.168.2.3 will not need to use socks to talk with 192.168.2.1, firewall. It has a direct connection in via Ethernet. It defines 127.0.0.1, the loopback, automatically. Of course you do not need SOCKS to talk to yourself. There are three entries:

- deny
- direct
- sockd

Deny tells SOCKS when to reject a request. This entry has the same three fields as in sockd.conf, identifier, address and modifier. Generally, since this is also handled by sockd.conf, the access file, the modifier field is set to 0.0.0.0. If you want to preclude yourself from calling any place, you can do it here.

The direct entry tells which addresses to not use socks for. These are all the addresses that can be reached without the proxy server. Again we have the three fields, identifier, address and modifier. Our example would have

```
direct 192.168.2.0 255.255.255.0
```

Thus going direct for any on our protected network.

The sockd entry tells the computer which host has the socks server daemon on it. The syntax is:

```
sockd @=<serverlist> <IP address> <modifier>
```

Notice the @= entry. This allows you to set the IP addresses of a list of proxy servers. In our example, we only use one proxy server. But, you can have many to allow a greater load and for redundancy in case of failure.

The IP address and modifier fields work just like in the other examples. You specify which addresses go where through these. 6.2.3. DNS from behind a Firewall

Setting up Domain Name service from behind a firewall is a relatively simple task. You need merely to set up the DNS on the firewalling machine. Then, set each machine behind the firewall to use this DNS.

8.3 Working With a Proxy Server

Unix

To have your applications work with the proxy server, they need to be "sockified". You will need two different telnets, one for direct communication, one for communication via the proxy server. SOCKS comes with instructions on how to SOCKify a program, as well as a couple pre-SOCKified programs. If you use the SOCKified version to go somewhere direct, SOCKS will automatically switch over to the direct version for you. Because of this, we want to rename all the programs on our protected network and replace them with the SOCKified programs. "Finger" becomes "finger.orig", "telnet" becomes "telnet.orig", etc. You must tell SOCKS about each of these via the include/socks.h file.

Certain programs will handle routing and sockifying itself. Netscape is one of these. You can use a proxy server under Netscape by entering the server's address (192.168.2.1 in our case) in the SOCKs field under Proxies. Each application will need at least a little messing with, regardless of how it handles a proxy server.

MS Windows with Trumpet Winsock

Trumpet Winsock comes with built in proxy server capabilities. In the "setup" menu, enter the IP address of the server, and the addresses of all the computers reachable directly. Trumpet will then handle all outgoing packets.

Getting the Proxy Server to work with UDP Packets

The SOCKS package works only with TCP packets, not UDP. This makes it quite a bit less useful. Many useful programs, such as talk and Archie, use UDP. There is a package designed to be used as a proxy server for UDP packets called UDPrelay, by Tom Fitzgerald <fitz@wang.com>. Unfortunately, at the

time of this writing, it is not compatible with Linux.

TOC

8.4 Drawbacks with Proxy Servers

The proxy server is, above all, a `security device`. Using it to increase internet access with limited IP addresses will have many drawbacks. A proxy server will allow greater access from inside the protected network to the outside, but will keep the inside completely unaccessible from the outside. This means no servers, talk or archie connections, or direct mailing to the inside computers. These drawbacks might seem slight, but think of it this way:

- You have left a report you are doing on your computer inside a firewall protected network. You are at home, and decide that you would like to go over it. You can not. You can not reach your computer because it is behind the firewall. You try to log into `firewall` first, but since everyone has proxy server access, no one has set up an account for you on it.
- Your daughter goes to college. You want to email her. You have some private things to talk about, and would rather have your mail sent directly to your machine. You trust your systems administrator completely, but still, this is private mail.
- The inability to use UDP packets represents a big drawback with the proxy servers. I imagine UDP capabilities will be coming shortly.

FTP causes another problem with a proxy server. When getting or doing an `ls`, the FTP server opens a socket on the client machine and sends the information through it. A proxy server will not allow this, so FTP doesn't particularly work.

And, proxy servers run slow. Because of the greater overhead, almost any other means of getting this access will be faster.

Basically, if you have the IP addresses, and you are not worried about security, do not use a firewall and/or proxy servers. If you do not have the IP addresses, but you are also not worried about security, you might also want to look into using an IP emulator, like Term, Slirp or TIA. Term is available from **`ftp://sunsite.unc.edu`**, Slirp is available from **`ftp://blitzen.canberra.edu.au/pub/slirp`**, and TIA is available from marketplace.com. These packages will run faster, allow better connections, and provide a greater level of access to the inside network from the internet. Proxy servers are good for those networks which have a lot of hosts that will want to connect to the internet on the fly, with one setup and little work after that.

---

9. Advanced Configurations

There is one configuration I would like to go over before wrapping this document up. The one I have just outlined will probably suffice for most people. However, I think the next outline will show a more advanced configuration that can clear up some questions. If you have questions beyond what I have just covered, or are just interested in the versatility of proxy servers and firewalls, read on.

9.1 A large network with emphasis on security

Say, for instance, you are the leader of millisha and you wish to network your site. You have 50 computers and a subnet of 32 (5 bits) IP numbers. You need various levels of access within your network because you tell your followers different things. Therefore, you'll need to protect certain parts of the network from the rest.

The levels are:

1. The external level. This is the level that gets shown to everybody. This is where you rant and rave to get new volunteers.
2. **Troop** This is the level of people who have gotten beyond the external level. Here is where you teach them about the evail goverment and how to make bombs.
3. **Mercenary** Here is where the *real* plans are keep. In this level is stored all the information on how the 3rd world goverment is going to take over the world, your plans involving Newt Gingrich, Oklahoma City, lown care products and what realy is stored in that hangers at area 51.

The Network Setup

The IP numbers are arranged as:

- 1 number is 192.168.2.255, which is the broadcast address and is not usable.
- 23 of the 32 IP addresses are allocated to 23 machines that will be accessible to the internet.
- 1 extra IP goes to a linux box on that network
- 1 extra goes to a different linux box on that network.
- 2 IP #'s go to the router
- 4 are left over, but given domain names paul, ringo, john, and george, just to confuse things a bit.
- The protected networks both have the addresses 192.168.2.xxx

Then, two separate networks are built, each in different rooms. They are routed via infrared Ethernet so that they are completely invisible to the outside room. Luckily, infrared ethernet works just like normal ethernet.

These networks are each connected to one of the linux boxes with an extra IP address.

There is a file server connecting the two protected networks. This is because the plans for taking over the world involves some of the higher Troops. The file server holds the address 192.168.2.17 for the Troop network and 192.168.2.23 for the Mercenary network. It has to have different IP addresses because it has to have different Ethernet cards. IP Forwarding on it is turned off.

IP Forwarding on both Linux boxes is also turned off. The router will not forward packets destined for 192.168.2.xxx unless explicitly told to do so, so the internet will not be able to get in. The reason for turning off IP Forwarding here is so that packets from the Troop's network will not be able to reach the Mercenary network, and vica versa.

The NFS server can also be set to offer different files to the different networks. This can come in handy, and a little trickery with symbolic links can make it so that the common files can be shared with all. Using this setup and another ethernet card can offer this one file server for all three networks.

The Proxy Setup

Now, since all three levels want to be able to monitor the network for their own devious purposes, all three need to have net access. The external network is connected directly into the internet, so we don't have to mess with proxy servers here. The Mercenary and Troop networks are behind firewalls, so it is necessary to set up proxy servers here.

Both networks will be setup very similarly. They both have the same IP addresses assigned to them. I will throw in a couple of parameters, just to make things more interesting though.

1. No one can use the file server for internet access. This exposes the file server to viruses and other nasty things, and it is rather important, so its off limits.
2. We will not allow troop access to the World Wide Web. They are in training, and this kind of information retrieval power might prove to be damaging.

So, the sockd.conf file on the Troop's linux box will have this line:

```
deny 192.168.2.17 255.255.255.255
```

and on the Mercenary machine:

```
deny 192.168.2.23 255.255.255.255
```

And, the Troop's linux box will have this line

```
deny 0.0.0.0 0.0.0.0 eq 80
```

This says to deny access to all machines trying to access the port equal (eq) to 80, the http port. This will still allow all other services, just deny Web access.

Then, both files will have:

```
permit 192.168.2.0 255.255.255.0
```

TOC

to allow all the computers on the 192.168.2.xxx network to use this proxy server except for those tha have already been denied (ie. the file server and Web access from the Troop network).

The Troop's sockd.conf file will look like:

```
deny 192.168.2.17 255.255.255.255
deny 0.0.0.0 0.0.0.0 eq 80
permit 192.168.2.0 255.255.255.0
```

and the Mercenary file will look like:

```
deny 192.168.2.23 255.255.255.255
permit 192.168.2.0 255.255.255.0
```

This should configure everything correctly. Each network is isolated accordingly, with the proper amount of interaction. Everyone should be happy.

Now, take over the world!

Le French-HOWTO : Linux & la langue française

Écrit par Guylhem Aznar, guylhem@danmark.linux.eu.org

v1.30, August 1997

Copyright ((c)) Juillet 1997, Guylhem Aznar. Librement distribuable & modifiable selon les termes de la GPL.

1. Présentation
2. Les problèmes
3. Le vocabulaire : petit lexique
4. Réglage du clavier sous Linux
5. Réglage du clavier sous XFree
6. Réglage du clavier pour les applications en mode texte
7. Réglage du clavier pour les applications X
8. Remerciements

9. Fichiers inclus dans ce HOWTO

1. Présentation

Ce document a été écrit pour aider à configurer un ordinateur doté du système d'exploitation Linux pour des utilisateurs francophones ; il y est expliqué par exemple comment configurer périphériques & applications afin de prendre en compte les particularités & les spécificités de notre langue...

Toutefois, ce document ne traite pas de l'installation du système d'exploitation Linux ; il est implicitement supposé que vous avez correctement installé une distribution de Linux (RedHat, Debian ou Slackware, cette liste n'étant pas exhaustive) & que vous vous apprêtez maintenant à mieux la configurer ; veuillez donc vous référer à d'autres HOWTOs pour l'installation. Les français ne sont pas les seuls francophones !

Pour chaque manipulation dépendant de la « nationalité », quatre versions sont donc proposées : une canadienne française, une belge wallone, une française & une suisse romande. Le Grand Duché du Luxembourg utilisant les mêmes standards que la France, j'invite les lecteurs luxembourgeois à se reporter aux exemples français !

Toutefois seuls les fichiers destinés aux français (table de clavier en mode console, X...) sont inclus, puisque la majorité des lecteurs seront certainement français & que seuls ces fichiers présentent des améliorations par rapport aux défauts.

Envoyez-moi donc vos fichiers belges, canadiens, luxembourgeois ou suisses ; ils seront inclus avec plaisir s'ils se montrent plus utiles que ceux proposés par défaut !

2. Les problèmes

Le problème de tout utilisateur non-américain lors de l'installation de Linux est de lui faire comprendre que, justement, il n'est pas américain !

Cela a des conséquences sur :

- les caractères spéciaux que l'on utilise, accentués (É È À Ç) ou bien ligaturés (Æ , voire « e dans l'o » qui n'existe même pas en standard sous Linux ; j'espère bientôt présenter ici une solution simple pour en disposer !)
- la correspondance entre la sérigraphie des cabochons (touches) sur lesquels vous appuyez & les codes que Linux renvoie, celui-ci supposant un clavier QWERTY américain par défaut
- le format de page pris en charge par les programmes que vous utilisez pour imprimer, différent entre les États-Unis (14 x 8,5 pouces pour du papier dit légal) & l'Europe (29,7 x 21 cm pour des feuilles a4)

Tout étant préconfiguré pour un anglophone nord-américain -bien que ceux-ci ne représentent pas la majorité de la population mondiale- , il faut donc se faire reconnaître comme une « exception » par les différents programmes, par des options ou des variables à exporter voire dans les pires des cas une modification du code source suivie d'une recompilation.

Heureusement, « l'esprit » de Linux rend le système très ouvert à des modifications de toute sorte & comme vous allez le découvrir au fil de ce document, il est assez simple d'obtenir un résultat très correct.

Pour le moment, le futur de l'internationalisation de Linux porte sur la traduction des messages d'erreur, des programmes, des pages de manuel & de la documentation, alors si vous vous sentez l'âme d'un traducteur, n'hésitez pas à contacter l'équipe de traduction des HOWTOs ; pour celà, adressez-vous à Éric Dumas (dumas@freenix.fr ou dumas@Linux.EU.Org) & à xc@itp.fr .

3. Le vocabulaire : petit lexique

Tout d'abord, un mot sur le vocabulaire employé :

Des caractères sont dits « 8 bits » s'ils contiennent par exemple des accents ou des signes spéciaux (comme paragraphe), non inclus dans le standard `ASCII` 7 bits (sans accents ni caractères spéciaux).

Une table de clavier est un fichier permettant de faire correspondre l'empreinte physique des cabochons (touches) du clavier au résultat obtenu sur l'écran de l'ordinateur.

Les touches « Home », « End », « Page Up » & « Page Down » (dont les noms varient selon les claviers) sont les touches permettant respectivement d'aller en début/fin de ligne & de monter/descendre d'une page.

Une touche morte est une touche qui doit être suivie d'un espace pour obtenir l'accent circonflexe seul, ou dans le cas où elle est suivie d'une voyelle y positionne un accent, par exemple :

```
^ + e donne ê
```

Les touches mortes sont :

- pour les claviers belges & français, l'accent aigu (en AltGr de 1 pour les français & en AltGr de ù pour les belges), l'accent grave (en AltGr de 7 pour les français & en AltGr de carré/cube pour les belges), le tréma & l'accent circonflexe (tous deux à côté du P)
- pour les canadiens, les accent graves, circonflexes, les trémas & la cédille
- pour les suisses, les accent aigus, graves, circonflexes, les trémas & la tilde

Enfin, voici une petite explication de la différence entre GMT & UTC par Edgar Bonet Orozco :

GMT est l'heure solaire moyenne de Greenwich. L'heure est comptée à partir de midi et est donc décalée de 12 h par rapport au temps universel. GMT est basée sur la rotation de la Terre, ce n'est donc pas un temps régulier. En effet, la vitesse de rotation de la Terre subit des variations faibles mais assez cahotiques, en plus elle diminue sur le long terme.

UTC (temps universel coordonné) est compté à partir de minuit (toujours par rapport au méridien zéro). Il est basé sur des étalons atomiques, mais des secondes intercalaires sont ajoutées occasionnellement pour faire en sorte qu'il ne s'écarte jamais de plus de une seconde de l'heure solaire moyenne. Tu auras plus de précisions sur la détermination de UTC dans "http://opdaf1.obspm.fr/", et en particulier dans "http://opdaf1.obspm.fr/www/lexique.html".

C'est le temps UTC qui sert de référence aux différentes heures légales. L'acronyme "GMT" est employé 99% du temps pour désigner UTC, mais c'est une erreur (ou un abus de langage).

4. Réglage du clavier sous Linux

Il existe deux programmes pour configurer votre clavier : un pour la console : `loadkeys` & un pour XFree : `xkbd`.

Pour installer un clavier français sous Linux, tapez selon que vous soyez respectivement Français, Canadien Français, Belge Wallon ou Suisse Roman :

```
loadkeys french.map
loadkeys cf.map
loadkeys be-latin1.map
loadkeys sf-latin1.map
```

`french.map` est inclu dans ce HOWTO (au chapitre « Fichiers inclus ») & tous les autres fichiers se trouvent d'ordinaire dans le répertoire `/usr/lib/kbd/keytables`.

Si vous utilisez french.map & french (pour X window : cf. chapitre «Régler le clavier pour X window»), `N'UTILISEZ PLUS` :

```
stty erase ^H
ou
stty erase ^?
```

« stty erase » ne sert en effet qu'à établir une correspondance entre le code que renvoie une touche et la fonction `erase` (effacer le caractère).

Les tables fournies dans ce HOWTO fonctionnent correctement & ne nécessitent donc pas de `stty erase` !

D'habitude, loadkeys est exécuté par des scripts au démarrage ; par exemple dans `/etc/rc.d/init.d/keytable` ou `/etc/rc.d/rc.local` (...) ; la commande :

```
 find
/etc/rc.d -type f | xargs grep loadkeys
```

vous fournira sur le nom exact du fichier lançant loadkeys ; il vous suffit alors de l'éditer pour que vos modifications soient désormais prise en en compte à chaque démarrage.

Si vous rencontrez les messages d'erreur suivants :

```
loadkeys french
Loading /usr/lib/kbd/keytables/french.map
unknown keysym 'dead_cedilla'
/usr/lib/kbd/keytables/french.map:67: parse error
```

```
syntax error in map file
key bindings not changed
```

celà signifie que votre version de kbd (dont le programme loadkeys dépend) est trop vieille : il faut vous procurer une version plus récente du programme.

Il existait & existe encore des versions des tables de clavier pour ceux ne voulant ni touches mortes ni accents, or cela tombe actuellement en désuétude puisque la grande majorité des programmes prennent désormais en charge ces fameuses touches mortes et les caractères accentués.

De plus, les tables fournies avec ce HOWTO permettent de se servir des touches mortes ou de s'en passer si un programme les refuse, les deux cas ayant été prévus.

---

5. Réglage du clavier sous XFree

Pour XFree, votre fichier `XF86Config` doit contenir une section Keyboard :

```
Section "Keyboard"
Protocol        "Standard"
AutoRepeat      500 30
LeftAlt         Meta
RightAlt        ModeShift               # Important pour Alt-gr, mais dans les
                                        # versions récentes de XFree,
                                        # AltGr & RightAlt sont synonymes.
ScrollLock      Compose                 # Pour faire des caractères spéciaux.
                                        # Utiliser sinon une des 3 touches des
                                        # nouveaux claviers 105 touches...
RightCtl        Control


XkbKeycodes     "xfree86"       #
XkbSymbols      "french(pc102)" #
XkbGeometry     "pc(pc102)"     #  Tout est nécessaire pour xkb
XkbModel        "pc102"         #
XkbLayout       "french"        #
# ou « ca » ou « be » ou « fr_CH » ; si vous êtes canadien, remplacez
# « french(pc102) » par « us » (« quebec », compatible avec « french »,
# sera bientôt disponible)
EndSection
```

Pour cette partie Xkb, certains préfèrent :

```
XkbKeymap "xfree86(french)"
XkbKeycodes  "xfree86"
XkbSymbols   "en_US(pc101)+french"
XkbGeometry  "pc(pc101)"
```

« french » est disponible à la fin de ce fichier !

Mettre « french » dans `/usr/X11R6/lib/X11/xkb/symbols/french` & « french.map » dans `/usr/lib/kbd/keytables/french.map`.

Attention !

xkb n'est disponible que pour des versions de XFree supérieures ou égales à 3,2 !

Si vous n'avez pas de répertoire `/usr/X11R6/lib/X11/xkb/symbols/` , procurez-vous donc une version plus récente de X window !

Sans xkb, XFree se rabattra par défaut sur votre configuration de clavier pour Linux (avec loadkeys). Toutefois, la translation n'est pas parfaite & je vous recommanderais plutôt d'éviter cette option, aussi bien que l'ancien gestionnaire de clavier `Xmodmap`.

Il se peut aussi que sans aucune commande pour xkb, XFree vous considère comme utilisateur d'un clavier anglo-américain !

Il convient donc d'utiliser « fr » ou « french » avec xkb ; toutefois ce dernier permet d'utiliser de multiples améliorati pour disposer du plus de confort possible (gage de confort : je l'utilise actuellement :-)).

Par exemple, certains programmes comme Netscape (tm) ne savent pas encore gérer les touches mortes ... donc impossible de taper par exemple des ê ou des î sur les claviers belges & français, puisque la touche morte « accent circonflexe » est le seul moyen d'obtenir ces caractères !

« french » répond à ce problème en proposant une solution de remplacement : AltGr (voyelle) permet d'obtenir la voyelle avec un accent grave, AltGr+Shift (voyelle) permet d'obtenir la voyelle avec un tréma !

Voici la table complète des correspondances :

- AltGr (voyelle) : voyelle accent grave
- AltGr + Shift (voyelle) : voyelle tréma
- AltGr (touche au dessous de la voyelle) : voyelle majuscule accent grave
- AltGr + Shift (touche au dessous de la voyelle) : voyelle majuscule tréma
- AltGr + Shift (minuscule accentuée) : majuscule accentuée

Par exemple :

- AltGr (e) : ê
- AltGr + Shift (ù) : Ù

L'astuce marche aussi pour le c cédille :

- AltGr + Shift (ç) : Ç

Donc même dans les pires conditions (si aucune touche morte ne fonctionne), on peut quand même accéder à toutes les voyelles accentuées, majuscules & minuscules !

Vous y trouverez en plus les guillemets français (« »), les symboles employés en physique (ß, å, delta, phi ...), des signes de monnaies (yen, ¢) ou de copyright ([TM], (c)) ainsi que des lettres d'autres langues (Ø, Æ ...).

Tous ces signes s'obtiennent grâce à AltGr (lettre), par exemple AltGr (b) donne ß, AltGr (r) donne [TM], AltGr (y) donne yen...

Pour ce qui est du choix entre les deux... à vous de voir !

Un conseil : essayez l'un puis l'autre.

Une autre méthode pour obtenir des caratères 8 bits :

Dans `/usr/X11R6/lib/X11/locale/iso8859-1/Compose` se trouve une table de « correspondance » entre deux caractères & un résultat ; les deux caractères sont « mélangés » lors de l'appui sur la touche Compose : par exemple;

```
Compose e ^ a toutes les chances de vous donner ê
```

Pour utiliser la touche Compose, vous devez l'affecter à une touche !

Le paramétrage préconisé, réglé dans XF86Config, la fait correspondre à la touche « Arrêt Défil » par la commande :

```
ScrollLock Compose
```

dans la section keyboard.

Éditez votre fichier de configuration & remplacez par exemple :

```
key <TLDE> {        [     twosuperior,   threesuperior      ]       };
```

par :

```
key <TLDE> {        [     twosuperior,   threesuperior      ],
                        [     Compose                           ]        };
```

Si vous avez par exemple un clavier belge ou français, désormais appuyer sur la touche ²/³ (carré/cube) vous donnera normalement ² (carré), en majuscule ³ (cube) & en AltGr la fameuse fonction Compose ; vous pouvez bien sur décider d'attribuer Compose à une autre touche ; je vous recommande toutefois d'utiliser ScrollLock, bien moins utilisée que les autres touches.

NB : La touche ²/³ (carré/cube), peu utilisée, est par défaut assignée aux guillemets français (« »), ² & ³ restant accessible respectivement en AltGr & AltGr + Shift ainsi que par l'accent circonflexe suivi du chiffre.

Pour modifier ce comportement par défaut, il vous suffit avant d'utiliser les fichiers de configuration pour XFree & linux en mode console de repérer la ligne mise en commentaire au début du fichier et de supprimer ce commentaire !

---

6. Réglage du clavier pour les applications en mode texte

Avertissement :

Cette section est fortement « inspirée » pour le réglage d'emacs, zsh, tcsh, flex, less, rlogin, joe, col & TeX du Guide du Rootard & de documents rassemblés sous le nom de « Francophones-HOWTO » qui hélas n'a jamais été fini.

Les auteurs respectifs sont donc remerciés pour leurs contributions !

Tout d'abord :

Le huitième bit doit survivre à l'entrée du noyau, assurez-vous-en donc avec :

```
stty cs8 -istrip -parenb
```

Ensuite, il convient de s'assurer que l'application est apte à supporter les caractères 8 bits : par exemple, ispell n'est pas compilé pour des caractères 8 bits par défaut : il faut le recompiler sans l'option :

```
-NO8BIT
```

Enfin, charger une police de caractères adaptées :

```
setfont /usr/lib/kbd/consolefonts/xxx-NNN
```

- xxx repésente le type de police ; il doit s'agir d'une iso01 ou lat1 pour les francophones sinon il n'y a plus de caractères « étendus » (voyelles accentués, symboles de ponctuation comme paragraphe...)
- NNN représente la taille individuelle de chaque caractère ; 08 est presque illisible mais 16 est un peu gros... À régler selon les goûts de chacun.

Éric Dubreuil (edubreuil@aol.com) vous conseille pour sa part la police `lat1u-16.psf`, permettant d'avoir tous les caractères accentuées sans pour autant enlaidir `mc`, un gestionnaire de fichier très populaire mais très sensible à la police utilisée !

0) Linux

Linux en 'version française' existe... mais ne semble pas facile à obtenir.

Il faut en pratique disposer des sources de la libc & d'un minimum de compétences (langage C).

Voici donc, pour la bonne bouche, quelques renseignements collectés...

Laurent Chemla nous pilote : « Si j'ai suivi, il y a deux trucs qui font la meme chose, d'ou embrouille, les locales et un truc appelé nls.

Les locales, sous Linux, ça va par défaut dans /usr/lib/locale, et ça contient de quoi préciser entre autres le format des nombres (LC_NUMERIC), les retours des fonctions ctype (isalpha etc) (LC_CTYPE), le format des sommes d'argent (LC_MONETARY), le format des dates (LC_TIME) et encore d'autres détails. Sur certains Unix, il y a aussi LC_MESSAGES, qui contient les messages de l'OS dans la langue choisie par la variable d'environnement LC_MESSAGES. Pas sous Linux. Sous Linux, c'est nls qui fait ça, et de fait, nls gêre un fichier qui est dans /etc/locale/. (NDG : De nos jours, ils se trouvent plutôt dans /usr/lib/locale) D'où l'embrouille. Ils auraient mis ça dans /usr/lib/nls, comme tout le monde, ça aurait evite la confusion. Mais bon.

Donc, si l'on veut se tenir à jour d'nls, il suffit d'aller dans les sources de libc/nls, faire 'make french' et copier libc.dat dans /etc/locale/C/libc.dat et toc, tous les messages sont en francais. Y compris ceux des programmes qui n'utilisent pas setlocale(): c'est perror() qui fait le boulot (du moins je suppose).

Si l'on veut en plus que les programmes qui utilisent setlocale() utilisent nos règles françaises, là il faut aller dans les sources

TOC

de libc/locale et dans chacun des sous-répertoires jeter un oeil sur le fichier exemple, le modifier pour le français (c'est tout petits fichiers), et faire 'make', avant de copier le résultat dans /usr/lib/local/fr/ et de faire un 'export LC_ALL=fr'.

1) Les shells

Ces variables doivent être exportées à chaque login : mettez les donc dans `.bashrc` & `.bash_profile`, `.tcshrc` ou /etc/profile :

```
# pour tcsh, utiliser setenv au lieu de export, par exemple «setenv LANG fr»

# Les locales
LC_CTYPE=ISO-8859-1
LC_ALL=fr_FR
# (ou fr_BE, fr_CA, fr_CH, fr_LU ...)
LANG=fr
LESSCHARSET=latin1
export PATH PS1 LC_CTYPE LC_ALL LANG LESSCHARSET

# Les raccourcis : sans ll ou d, impossible de voir les fichiers comprenant des
# caractères 8 bits ...
alias q="cd .."
alias ll="ls --color=auto -a -N -l"
alias d="ls --color=auto -a -N"

# Au cas où del/backspace ne fonctionne pas, essayez l'un des deux :
#stty erase ^?
#stty erase ^H
```

N'oubliez pas : tous les fichiers ici cités dont le nom commence par « . » doivent se trouver dans votre répertoire maison !

a) Pour bash (version 1.13.*), créez un `.inputrc`:

```
# Permettre de rentrer & recevoir des caractères accentués
set meta-flag on
set convert-meta off
set input-meta on
set output-meta on

# Pas de bip audible mais visible
# set bell-style visible

# Pour faire marcher Home, End, Delete, Esc + Delete & Esc + Backspace
"\e[1~":beginning-of-line
"\e[3~":delete-char
"\e[4~":end-of-line
"\e\C-h": backward-kill-word
"\e\e[3~": kill-word
```

Celà vous permettra d'effacer des caractères vers l'avant avec DELETE (souvent nommée DEL ou SUPPR) et vers l'arrière avec BACKSPACE (touche située à côté d' « Entrée »), ceci n'étant pas prévu sur les tables de clavier par défaut « mimant » le comportement d'un clavier de console VT100 .

Les deux dernières options vous permettront d'utiliser Home & End pour aller au début & à la fin de la ligne en cours d'édition.

b) zsh se contente d'un `stty pass8`.

c) Pour tcsh, placer, dans le fichier `/etc/csh.login` (ou dans les fichiers `.tcshrc` des utilisateurs concernés) :

```
setenv LC_CTYPE iso_8859_1
setenv LANG fr
```

d) Pour tous les shells

Remplacer 'iso_8859_1' par 'iso-8859-1' si cela ne fonctionne pas, voire french.iso88591.fr_FR (ou fr_BR, fr_CA, fr_CH, fr_LU ...)

À propos de LANG, Pablo Saratxaga signale qu'il vaut mieux éviter fr_FR : « Par ailleurs je conseille un simple "fr" plutôt que fr_FR etc. Certains programmes n'aiment pas trop (ils ne passent pas vraiment par la resolution de locales mais lisent LANG directemment...) »

Si nls est installe, les routines correspondantes sont utilisees, sinon tcsh agit en iso_8859_1, quelle que soit les valeurs donnees a LANG & LC_CTYPE : cf. la section NATIVE LANGUAGE SYSTEM de tcsh(1).

D'après le Danish-HOWTO:

```
setenv LC_CTYPE ISO-8859-1; stty pass8
```

2) L'heure

Rien n'est plus facile que de laisser à Linux le soin de passer de l'heure d'été à l'heure d'hiver & réciproquement...

a) D'abord, quelle heure est-il :-) ?

Après avoir regardé sur la pendule la plus proche de vous, tapez :

```
clock
```

puis

```
date
```

La première heure est celle du bios, la seconde celle calculée par Linux à partir de `/etc/zoneconfig`.

Pour peu que vous ayez installé timezone, votre machine peut jongler aisément entre les différents fuseaux horaires... ce que les DOS ou Windows ne permettent pas !

b) L'heure du choix !

De là, deux possibilitées : soit vous décidez de laisser l'heure système à l'heure locale (peu pratique, excepté si vous hébergez aussi sur votre disque dur ces systèmes d'exploitation peu évolué, ne sachant même pas exploiter les fuseaux horaires), soit vous passez à l'heure de Greenwich, Linux se chargeant de l'adapter à votre fuseau horaire.

Dans ce cas, les grands voyages à l'étranger avec un portable tout neuf équipé de Linux se feront en toute simplicité : le dernier pargraphe vous expliquera comment changer de fuseau horaire facilement...

c) Le réglage

Pour procéder, regardez d'abord si l'heure système retournée par clock est l'heure locale ou l'heure de Greenwich (vous avez bien regardé une pendule comme je vous le conseille plus haut, n'est-ce pas ?).

Si clock vous retourne l'heure locale, il va falloir jongler un peu avec les fuseaux horaires, sinon passez directement à l'étape « réglage du fuseau ».

- Pour les français, belges wallons & suisses romans :

Vous êtes normalement dans une zone horaire MET (Medium European Time, soit GMT+1).

La zone « MET DST » correspond elle à l'heure d'été active (GMT + 2).

TOC

Selon les décisions futures de la cour européenne, l'heure d'été (daylight savings) risque d'être abandonné... ce qui compliquera cette explication !

- Pour les canadiens français :

Vous êtes dans une zone horaire GMT-4 si vous habitez autour de Montréal .

Pour les autres provinces, consulter un dictionnaire !

Premièrement, effacez le fichier /etc/localtime ou /usr/lib/zoneinfo/localtime (l'emplacement varie selon les distributions ; si vous avez les deux fichiers, effacez /etc/localtime : l'autre fichier doit certainement être un lien vers /etc/localtime).

Dès lors, clock & date doivent vous retourner la même heure...

- Pour les canadiens français : utilisez GMT+4
- Pour les français, belges wallons & suisses romans :

Si vous lisez ce document en été, le décalage de l'heure locale par rapport à l'heure de Greenwich est de deux heures, on va donc mettre temporairement le fuseau en GMT-2, soit vous lisez ce document en hiver & il convient de remplacer tous les « 2 » par des « 1 » dans les exemples suivants...

GMT-2 ?!!? Alors que le fuseau est GMT+2 ?

Ceci va vous retourner l'heure de Greenwich dont on a besoin pour régler le système : tapez :

```
ln -sf /usr/lib/zoneinfo/Etc/GMT-2 /etc/zoneconfig
```

```
clock
```

```
date
```

L'heure calculée correspond donc à l'heure de Greenwich, l'heure système à l'heure locale... soit l'inverse de ce que l'on veut.

Un simple : `clock -w`

Va alors mettre l'heure du système en heure de Greenwich, utilisant pour celà l'heure locale inversée volontairement avec l'heure système.

d) Régler le fuseau horaire :

Tapez alors : `ln -sf /usr/lib/zoneinfo/Europe/Votre-ville /etc/zoneconfig`

Et voilà ! Le système est à l'heure de Greenwich, la conversion vers le fuseau horaire local & l'heure d'été se faisant automatiquement.

Si vous habitez hors Europe, procédez de même en remplaçant les « -2 » par « X », X représentant l'opposé de votre décalage horaire (par exemple -4 au Québec, donc X=+4...)

e) Modifier le fuseau horaire :

Si vous partez à l'étranger, il est alors simple de modifier l'heure locale de votre portable : par exemple si vous partez pour Moscou :

```
ln -sf /usr/lib/zoneinfo/Europe/Moscow /etc/zoneconfig
```

Très simple & très utile : plus besoin de se souvenir du décalage horaire (plus ou moins X heures) puisque Linux se charge désormais de tout !

3) Emacs

Voici un fichier de configuration de base pour employer les accents. XEmacs les accepte lui par défaut sans broncher :-)

```
;;
```

```
;; Fichier .emacs: initialisation d'emacs
;; Tiré du Guide du Rootard
;;

(display-time)                      ;; Pour avoir l'heure dans la barre d'etat
(setq display-time-24hr-format t)  ;; Format 24 heures

;; Nouveaux modes
(autoload 'c++-mode   "cplus-md" "C++ Editing Mode" t)
(autoload 'perl-mode   "perl-mode" "Perl Editing Mode" t)
(autoload 'c-mode "c-mode" "C Editing Mode" t)
; mieux vaudrait utiliser le "cc-mode"

(autoload 'jargon-mode "jargon" "Jargon Mode" t)

;; Auto-Mode Settings : positionne le mode selon l'extension
(setq auto-mode-alist
(append '(("\.c$"  . c-mode)    ;; utilise le mode C++ meme pour C
("\.h$"  . c-mode)
("\.C$"  . c++-mode)
("\.H$"  . c++-mode)
("\.cc$" . c++-mode)
("\.C$"  . c++-mode)
("\.pl$" . perl-mode)           ;; Perl
("/tmp/snd\.[0-9]*" . text-mode);; Text (pour le mail)
("[Rr][Ee][0-9]*" . text-mode)
("\.ada$"  . ada-mode)          ;; Ada
("\.spec$"  . ada-mode)
("\.body$"  . ada-mode)
("makefile$" . makefile-mode)  ;; Makefile
("Makefile$" . makefile-mode)
("Imakefile$" . makefile-mode))
auto-mode-alist))

; Remapes varies a mettre ici
(global-set-key "\eg" 'goto-line)       ;; ESC G = Goto line

(put 'eval-expression 'disabled nil)

;; Accents...
(standard-display-european 1)
(load-library "iso-syntax")
(set-input-mode (car (current-input-mode))
              (nth 1 (current-input-mode))
              0)

;; Sous X-Window, textes en couleurs (C/C++/Shell/Makefile,etc)
(cond (window-system
(setq hilit-mode-enable-list  '(not text-mode)
hilit-background-mode   'light
hilit-inhibit-hooks     nil
hilit-inhibit-rebinding nil)
(require 'hilit19)
))
```

```
(if (not (equal window-system ""))
(global-set-key "\C-?" 'delete-char))
))
```

Le Keyboard-HOWTO conseille quand à lui :

```
(standard-display-european t)
(set-input-mode nil nil 1)
(require 'iso-syntax)
(load-library "iso-insert.el")
(define-key global-map [?\C-.] 8859-1-map)
```

4) less

`/etc/profile` ou les fichiers de démarrage du shell (comme `.bashrc` & `.bash_profile` dans votre répertoire maison) doivent contenir :

```
export LESSCHARSET=latin1
```

Le fichier `.lessrc` se trouve à la section « Fichier inclus ».

Pour s'en servir, taper :

```
lesskey -o .less .lessrc
```

N'oubliez pas de mettre le fichier .less dans votre répertoire maison : il permet d'employer les touches fléchées, Page_up, Page_down, Home & End.

5) ls

Utilisez l'option :

```
--color=auto
```

ou, plus simplement, définissez dans les fichiers de démarrage de votre shell les alias suivants permettant de voir les fichiers dont le nom comprend des caractères 8 bits :

```
alias ll="ls --color=auto -l -a -N"
alias d="ls --color=auto"
```

Un simple `d` remplacera le classique `ls`, en y ajoutant la couleur, alors que `ll` vous offrira une liste complète de tous les fichiers présents dans le répertoire, avec leurs attributs, leurs tailles...

Pablo Saratxaga conseille d'utiliser --color=auto car « --color=yes fera avoir des codes escape ansi même lorsqu'on redirige vers un pipe un fichier où que le termina ne supporte pas la couleur... ce n'est pas ce qu'on voudrait. »

6) flex

Donnez l'option :

```
-8
```

si l'analyseur généré doit accepter les entrees 8-bits (bien sur qu'il doit le faire !).

7) Elm

Rajoutez ces trois lignes dans votre `.elm/elmrc` :

```
charset = iso-8859-1
```

```
displaycharset = iso-8859-1

textencoding = 8bit
```

8) col

Assurez-vous :

1) qu'il a été corrigé et fasse un

```
setlocale(LC_CTYPE,"");
```

2) de définir

```
LC_CTYPE=ISO-8859-1
```

dans l'environment

9) Rlogin

Utilisez l'option :

```
-8
```

10) joe

```
ftp://sunsite.unc.edu:/pub/Linux/apps/editors/joe-2.8.tar.gz
```

ou plus récent devrait fonctionner après édition du fichier de configuration. Placez l'option -asis dans `/usr/lib/joerc` en première colonne.

11) TeX & LaTeX

Le plus simple consiste à employer GuTemberg, une distribution de LaTeX avec tous les défauts réglés pour des francophones par des francophones !

Sinon :

Pour LaTeX:

```
\documentstyle[isolatin]{article}
```

Pour LaTeX2e:

```
\documentclass{article}\usepackage{isolatin}
```

ou alors :

```
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
```

Au cas où votre distribution soit trop vieille pour l'inclure, `isolatin.sty` est disponible séparément sur :

```
ftp.vlsivie.tuwien.ac.at/pub/8bit
```

Pour gs & xdvi, il faut utiliser des options speciales (sur ma machine, ce sont des alias). En effet, ils sont souvent configurés pour un papier de taille anglo-saxonne.

Ces options sont les suivantes :

```
gs -sPAPERSIZE=a4 xdvi -paper a4 ghostview -a4
```

Pour que dvips convertisse les documents dans un format papier a4, il faut specifier dans le fichier config.ps (le chemin varie en fonction des versions de LaTeX) : /usr/lib/texmf/dvips/config.ps ou `.dvipsrc`

```
@ a4 210mm 297mm @+ ! %%DocumentPaperSizes: a4 @+ %%PaperSize: a4 @+
%%BeginPaperSize: a4 @+ a4
```

Sinon ajoutez à votre `.Xresources` :

```
XDvi.paper:              a4
Ghostview.pageMedia:     A4
```

12) Manuel

Tout d'abord, il faut vous procurer les pages de manuel Linux traduites en français sur :

`ftp.lip6.fr/pub/linux`

Si vous utilisez une distribution Debian, celles-ci sont présentes sur le cédérom : il suffit de les sélectionner lors de l'installation.

De là, deux possibilitées :

- soit vous ne voulez que des manuels en français, auquel cas il faut faire un :

  `export MANPATH=/usr/man/fr_FR (ou fr_BE, fr_CA, fr_CH, fr_LU...)`
- soit vous préférez par défaut des manuels en français et si la page n'existe pas, vous vous rabattez vers l'anglais, il suffir alors de taper :

  `export LANG=fr`

man ira chercher dans le « MANPATH » des pages en « LANG » par défaut et se rabattra sinon sur des pages en anglais

13) Correcteur orthographique ispell

Vous pouvez vous le procurer sur son site original :

`ftp.ireq-robot.hydro.qc.ca:/pub/ispell/francais-IREQ-1.3.tar.gz`

ou sur :

`ftp.lip6.fr/pub/ispell/francais`

Un fichier LISEZMOI explique pas à pas & en français l'installation.

14) Les claviers 105 touches

Aussi appelés « claviers Microsoft (tm) », ils ont trois touches de plus que les claviers normaux.

Comment s'en servir ?

En mode console, éditer `french.map` et décommenter les lignes en parlant !

De même pour `french` (pour Xwindow).

15) PGP

Tout d'abord, un avertissement : ce paragraphe n'est destiné qu'aux francophones ne vivant pas en France !

En effet, en France, en Iran, en Irak & dans quelques autres pays, l'utilisation de logiciels de cryptographie est considérée comme usage d'armes de guerre, ce qui est punissable par la loi !

Donc sautez ce paragraphe si votre législation locale ne permet pas d'utiliser des logiciels de cryptographie.

Premièrement, il faut se procurer pgp-international (à cause de problèmes de patentes et de législation sur l'export d'arm de guerre, il existe pgp-us & pgp-international, ce dernire étant doté d'un algorythme encore plus puissant) sur :

`ftp://ftp.ifi.uio.no/pub/pgp/`

Ensuite, mettre dans votre répertoire maison, dans un sous répertoire .pgp les fichiers :

```
config.txt
language.idx
pgpdoc1.txt
pgpdoc2.txt
language.txt
fr.hlp
en.hlp
pgp.hlp
```

Enfin, éditer le fichier `.pgp/config.txt` pour y rajouter :

```
Language = fr
CharSet = latin1
```

Lisez ensuite `.pgp/fr.hlp` pour apprendre à vous servir de pgp !

16) Impression en mode ascii

Pierre-Charles David (pcdavid@club-internet.fr) vous donne ce truc :

Pour l'impression de texte simple, beaucoup d'imprimantes attendent des lettres en codage IBM (celui utilisé par MS-DOS).

Si le fichier est codé en latin1, on doit utiliser 'recode latin1:ibmpc' (à placer dans les filtres lpr par exemple), sous peine de voir imprimer des 'alpha' à la place de 'à', et autres choses du même genre.

17) Divers

Pour les programmes utilisant curses (comme lynx) David Sibley dit:

« La version standard de curses utilise le huitième bit pour la vidéo inversee (voir le flag _STANDOUT défini dans /usr/include/curses.h). Cependant, ncurses semble fonctionner en 8-bits et affiche le iso-latin-8859-1 correctement. »

Pour les programmes utilisant groff (comme man), utilisez le :

`-Tlatin1`

au lieu de :

`-Tascii`

Les vieilles versions de man utilisent aussi col, & le point suivant s'applique aussi.

Une belle discussion sur le thème de l'ISO-8859-1 & sur « comment manipuler les caractères 8-bits » est disponible dans :

`ftp://grasp.insa-lyon.fr/pub/faq/fr/accents`

Pour perl, si tout d'un coup après une mise à jour il raconte :

```
(guylhem@danmark:guylhem)$ perl
perl: warning: Setting locale failed for the categories:
        LC_CTYPE LC_COLLATE
perl: warning: Please check that your locale settings:
        LC_ALL = "fr_FR",
        LC_CTYPE = "ISO-8859-1",
```

```
            LC_COLLATE = (unset),
            LANG = "fr"
        are supported and installed on your system.
    perl: warning: Falling back to the "C" locale.
    (guylhem@danmark:guylhem)$
```

Essayez cette solution de Thomas Parmelan (Thomas.Parmelan@efrei.fr) :

- en libc6 le format des fichiers binaires de définition des locales a (encore) changé : il faut les régénérer à partir des sources (WG15collection.tar.tz sur ftp.lip6.fr:/pub/linux/GCC par exemple) et des programmes "locale" et "localedef" qui sont fournis dans les sources de la libc.

- Sinon tu peux également essayer ceci (c'est comme cela que ça fonctionne chez moi en libc-5.4.x) :

```
unset LANG
unset LC_CTYPE
unset LC_COLLATE
export LC_ALL=fr_FR.ISO-8859-1
```

---

7. Réglage du clavier pour les applications X

1) Les xterminaux (xterm, nxterm, rxvt...)

Les faire fonctionner avec toutes les touches standard du clavier 102 touches relève du miracle, quant aux accents, seule une recompilation vous permettra de les utiliser !

Pour xterm, essayez de mettre dans votre `.Xdefaults` :

```
XTerm*eightBitInput: true
XTerm*eightBitOutput: true
```

Pablo Saratxaga conseille de rajouter dans `.inputrc` :

```
# pour rxvt
"\e[7~":beginning-of-line
"\e[8~":end-of-line
# pour xterm
"\e[H": beginning-of-line
"\e[F": end-of-line
```

Mais ces deux solutions ne vous garantissent pas de résoudre tous les problèmes de rxvt et de xterm ...

Pour les résoudre sans créer de fichiers dans `/usr/X11R6/lib/X11/app-defaults`, je tiens sinon à votre disposition un xiterm, un terminal X supportant *TOUTES* les touches des claviers standard, avec Home, End, Page_up (...) & même le pavé numérique, fonctionnant comme en mode console, i.e. Shift-Page_up permettant de faire défiler une page, voire mieux : le pavé numérique reconnaissant enfin la différence entre Verr. Num allumé & Verr. Num éteint.

X international terminal (xiterm pour les intimes) est bien sur sous GPL & mis à jour pour suivre parallèlement l'évolution de rxvt.

Sur http://www.linux-france.com/prj/xiterm & sur ftp://sunsite.unc.edu/pub (& bientôt ailleurs sans doute), vous trouverez donc xiterm, qui remplace votre xterm classique en y apportant la couleurs, les accents & les touches étendues.

Pour l'installer, compilez-le, devenez root & tapez :

```
    rm /usr/X11R6/bin/xterm
    rm /usr/X11R6/lib/X11/app-defaults/xterm
    rm /usr/X11R6/lib/X11/app-defaults/rxvt
    rm /usr/X11R6/lib/X11/app-defaults/Xterm
    rm /usr/X11R6/lib/X11/app-defaults/XTerm
    rm /usr/X11R6/lib/X11/app-defaults/Rxvt
    rm /usr/X11R6/lib/X11/app-defaults/RXvt
    cp xiterm /usr/local/bin

    ln -sf /usr/local/bin/xiterm /usr/X11R6/bin/xterm
    ln -sf /usr/local/bin/xiterm /usr/X11R6/bin/rxvt
```

2) Les applications Motif

Là, je n'ai malheureusement aucun remède miracle !

Utilisez les possibilités du fichier french au lieu des touches mortes (AltGr (voyelle) : voyelle accent circonflexe...) !

3) Le manuel

Pablo Saratxaga tient à votre disposition un xman modifié « pour qu'il supporte la variable « LANG » et puisse afficher les pages en d'autres langues qu'anglais ».

Il est disponible sur :

```
ftp.lip6.fr:/pub/linux/distributions/redhat-contrib/SRPMS/xman-3.1.6-3.src.rpm
ftp.lip6.fr:/pub/linux/distributions/redhat-contrib/i386/xman-3.1.6-3.i386.rpm »
```

---

8. Remerciements

Ce document ne serait pas ce qu'il est sans l'aide de Nat Makarévitch (nat@linux-france.com).

Je tiens aussi à remercier les personnes suivantes pour leurs contributions aux autres documents sur Linux en français (Guide du Rootard & Francophones-HOWTO) desquels la section « Configurer les applications en mode texte » s'inspire pour emacs, zsh, tcsh, flex, less, rlogin, joe, col & TeX.

Rappelons que les sites officiels du Guide du Rootard sont :

`http://www.freenix.fr/linux/Guide`

&

`ftp.lip6.fr:/pub/linux/french/docs/`

Par ordre alphabétique :

```
Jean-Michel Antoine
Michel Billaud
Stephane Bortzmeyer
Rémy Card
Xavier Cazin
Laurent Chemla
René Cougnenc
Bruno Cornec
Yann Dirson
Éric Dubreuil
Éric Dumas
Pierre Ficheux
Laurent Frigault
Frederic Lepied
Stephane Marzloff
Marc Merlin
Thomas Parmelan
Thomas Quinot
David Robert
Ollivier Robert
Benjamin Ryzman
Pablo Saratxaga
Julien Simon
J.M. Vansteene
```

Surtout, si je vous oublie, signalez-le moi !

De même si vous ne voulez plus figurer dans cette liste...

Ce document est en GPL, toutefois il vaudrait mieux éviter une multiplication des versions :-).

Envoyez-donc vos suggestions & modifications à guylhem@danmark.linux.eu.org

Je me ferai un plaisir de les intégrer dans les prochaines versions !

Signalez-moi aussi les erreurs : cet HOWTO est en perpétuel changement, avec des nouveautés incluses à chaque version ; des erreur peuvent facilement s'y glisser !

---

9. Fichiers inclus dans ce HOWTO

Comme signalé plus haut, cette section ne contient que .bashrc, .bash_profile (tous deux pour bash), .lessrc (pour less), french (pour X window), quebec (pour X window) & french.map .

Voici .bashrc (pour bash, à placer dans votre répertoire maison) :

```
# ~/.bashrc : executé par bash(1) pour les shells lancés après le login.

# Le path
PATH="/usr/sbin:/usr/bin:/sbin:/bin:/usr/X11R6/bin:/usr/local/bin"

# Le prompt
PS1="(\u@\h:\W)$ "

# Les locales
LC_CTYPE=ISO-8859-1
LC_ALL=fr_FR
# (ou fr_BE, fr_CA, fr_CH, fr_LU ..)
LANG=fr
LESSCHARSET=latin1
export PATH PS1 LC_CTYPE LC_ALL LANG LESSCHARSET

# Les raccourcis
alias q="cd .."
alias ll="ls --color=auto -a -N -l"
alias d="ls --color=auto -a -N"

# Au cas où del/backspace ne fonctionne pas, essayez l'un des deux :
#stty erase ^?
#stty erase ^H

# Refuser le talk & le write
#mesg n
# Permettre les coredumps
#umask 022
```

Voici .bash_profile (pour bash, à placer dans votre répertoire maison) :

```
# ~/.bash_profile: executé par bash(1) lors les logins.

# Le path
PATH="/usr/sbin:/usr/bin:/sbin:/bin:/usr/X11R6/bin:/usr/local/bin"

# Le prompt
PS1="(\u@\h:\W)login$ "

# Les locales
LC_CTYPE=ISO-8859-1
LC_ALL=fr_FR
# (ou fr_BE, fr_CA, fr_CH, fr_LU ...)
LANG=fr
```

```
LESSCHARSET=latin1
export PATH PS1 LC_CTYPE LC_ALL LANG LESSCHARSET

# Les raccourcis
alias q="cd .."
alias ll="ls --color=auto -a -N -l"
alias d="ls --color=auto -a -N"

# Au cas où del/backspace ne fonctionne pas, essayez l'un des deux :
#stty erase ^?
#stty erase ^H

# Refuser le talk & le write
#mesg n
# Permettre les coredumps
#umask 022
```

Voici .lessrc (pour less, à compiler avec lesskey) :

```
(taper ESC)[B   forw-line
(taper ESC)[A   back-line
(taper ESC)[6~  forw-scroll
(taper ESC)[5~          back-scroll
(taper ESC)[1~   goto-line
(taper ESC)[4~  goto-end
(taper ESC)[C    next-file
(taper ESC)[D    prev-file
(taper ESC)OA    back-line
(taper ESC)OB    forw-line
```

Pour rentrer un ESC (parfois nommé échap), sous vi, mettez vous en mode insertion avec i puis tapez ctrl+v & enfin esc.

Voici french (pour X11, à placer dans /usr/X11R6/lib/X11/kbd/keytables) :

```
// French keyboard description, (c) Guylhem Aznar, July 1997
// Released under GNU GENERAL PUBLIC LICENSE Version 2
// If you haven't received it, this license is available at :
// Free Software Foundation, Inc.  675 Mass Ave, Cambridge, MA 02139, USA

partial hidden alphanumeric_keys modifier_keys
xkb_symbols "basic" {

// Description *TOTALE* d'un clavier français, sans le comparer à un
// clavier américain par défaut : on y gagne plus de flexibilité, ce
// modèle devient plus personalisable ...

// Les touches é, è, ç, à & ù, présentes en mode minuscule (i.e. sans shift)
// ne sont pas disponibles en mode majuscule car elles sont respectivement
// remplacées par 2, 7, 9, 0 & % .
// Pour obtenir les majuscules accentuées É, È, Ç, À & Ù, vous pouvez soit
// utiliser AltGr + Shift (minuscule accentuée), soit utiliser les accent
// dits "morts", i.e. fonctionnant comme l'accent circonflexe & le tréma .
// Sont disponibles :  AltGr (1) : accent aigü, AltGr (7) : accent grave,
// AltGr (^) : cédille ; pour s'en servir, procéder comme avec l'accent
// circonflexe & le tréma : ex : AltGr (1) + e : é ; AltGr (1) + e : É ...
// La tilde (~) n'étant pas employée sur des lettres en français, elle reste
```

```
// en AltGr (2) une touche normale .
// Une tilde "morte" (avec la cédille "morte") est toutefois disponibles sur l
// touche accent circonflexe & tréma en AltGr + Shift .

// Au cas où les dead_keys ne fonctionnent pas (certains programmes les
// refusent), vous pouvez toujours utiliser pour obtenir les voyelles à accent
// circonflexe & à tréma :
// AltGr (voyelle) : voyelle-minuscule à accent circonflexe
// AltGr + Shift (voyelle) : voyelle-minuscule à tréma
// AltGr (en dessous de la voyelle) : voyelle-majuscule à accent circonflexe
// AltGr + Shift (en dessous de la voyelle) : voyelle majuscule à tréma

// Plusieurs lettres étrangères, utilisées comme signes mathématiques ou
// physiques, abbréviations (...) ont été rajoutées en AltGr :
// T->Phi;, r->[TM], y->yen, c->¢ & (c), f->señora, m->señor,
// g->`, h->', ù->±, x->æ s->ø, z->å, w->Delta;, b->ß

// La correspondance essaye de se faire la plus naturelle possible, en mettant
// ces touches là où l'on s'attend le plus de les trouver ...
// mais elle est loin d'être parfaite :^)

// Modifiez donc à loisir tous ces AltGr, et si vous pensez avoir trouvé la
// configuration ultime permettant d'accéder en mieux à toutes ces touches,
// mailez moi !
// Vôtre arrangement, s'il est vraiement plus pratique, sera inclu dans la
// prochaine version de la carte de correspondance du clavier !

    name[Group1]= "French";
    key <ESC>   {       [          Escape                        ]       };

// Deux possibilitées pour cette touche "au carré" (²) :
// * un mode mathématique où elle renvoie "au carré" en mode normal, "au cube"
// si elle est Shiftée et "puissance 1" en Alt-gr (pour parler par exemple de
// m.s-¹ ...)
// Toutefois, accent circonflexe suivi de 1, 2, 3, -, renvoyant respectivement
// ¹, ², ³ et -, cette touche est inutile !

// * un mode littéraire est donc proposé par défault, pour disposer des
// guillemets français (« & ») ; le ² et le ³ restant disponibles
// respectivement en AltGr et AltGr + Shift .

//   key <TLDE> {       [     twosuperior,   threesuperior      ],
//                      [     onesuperior                       ]       };

    key <TLDE>  {       [   guillemotleft,  guillemotright      ],
                        [     twosuperior,   threesuperior      ]       };

    key <AE01>  {       [       ampersand,               1      ],
                        [      dead_acute                       ]       };
    key <AE02>  {       [          eacute,               2      ],
                        [      asciitilde,          Eacute      ]       };
    key <AE03>  {       [        quotedbl,               3      ],
                        [      numbersign                       ]       };
    key <AE04>  {       [      apostrophe,               4      ],
                        [       braceleft                       ]       };
    key <AE05>  {       [       parenleft,               5      ],
```

TOC

```
                        [     bracketleft                          ]       };
key <AE06>  {           [           minus,                6      ],
                        [             bar                          ]       };
key <AE07>  {           [          egrave,                7      ],
                        [      dead_grave,           Egrave       ]       };
key <AE08>  {           [      underscore,                8      ],
                        [       backslash                          ]       };
key <AE09>  {           [        ccedilla,                9      ],
                        [     asciicircum,         Ccedilla       ]       };
key <AE10>  {           [          agrave,                0      ],
                        [              at,           Agrave       ]       };
key <AE11>  {           [      parenright,           degree       ],
                        [    bracketright                         ]       };
key <AE12>  {           [           equal,             plus       ],
                        [      braceright                         ]       };
key <BKSP>  {           [       BackSpace                         ]       };
key  <TAB>  {           [             Tab,     ISO_Left_Tab       ]       };
key <AD01>  {           [               a,                A       ],
                        [     acircumflex,       adiaeresis       ]       };
key <AD02>  {           [               z,                Z       ],
                        [           aring,             Aring       ]       };
key <AD03>  {           [               e,                E       ],
                        [     ecircumflex,       ediaeresis       ]       };
key <AD04>  {           [               r,                R       ],
                        [      registered                         ]       };
key <AD05>  {           [               t,                T       ],
                        [           thorn,             THORN       ]       };
key <AD06>  {           [               y,                Y       ],
                        [             yen                          ]       };
key <AD07>  {           [               u,                U       ],
                        [     ucircumflex,       udiaeresis       ]       };
key <AD08>  {           [               i,                I       ],
                        [     icircumflex,       idiaeresis       ]       };
key <AD09>  {           [               o,                O       ],
                        [     ocircumflex,       odiaeresis       ]       };
key <AD10>  {           [               p,                P       ],
                        [        paragraph                         ]       };
key <AD11>  {           [ dead_circumflex,   dead_diaeresis       ],
                        [    dead_cedilla,       dead_tilde       ]       };
key <AD12>  {           [          dollar,          sterling       ],
                        [        currency                         ]       };
key <RTRN>  {           [          Return                         ]       };
key <AC01>  {           [               q,                Q       ],
                        [     Acircumflex,       Adiaeresis       ]       };
key <AC02>  {           [               s,                S       ],
                        [          oslash,          Ooblique       ]       };
key <AC03>  {           [               d,                D       ],
                        [     Ecircumflex,       Ediaeresis       ]       };
key <AC04>  {           [               f,                F       ],
                        [     ordfeminine                         ]       };
key <AC05>  {           [               g,                G       ],
                        [       quoteleft                         ]       };
key <AC06>  {           [               h,                H       ],
                        [      quoteright                         ]       };
key <AC07>  {           [               j,                J       ],
                        [     Ucircumflex,       Udiaeresis       ]       };
```

TOC

```
    key <AC08>  {       [               k,               K      ],
                        [     Icircumflex,      Idiaeresis      ]       };
    key <AC09>  {       [               l,               L      ],
                        [     Ocircumflex,      Odiaeresis      ]       };
    key <AC10>  {       [               m,               M      ],
                        [       masculine                        ]       };
    key <AC11>  {       [          ugrave,         percent      ],
                        [       plusminus,          Ugrave      ]       };
    key <BKSL>  {       [        asterisk,              mu      ],
                        [     Greek_OMEGA                        ]       };
    key <LFSH>  {       [          Shift_L                      ]       };
    key <LSGT>  {       [           less,          greater      ],
                        [            bar                         ]       };
    key <AB01>  {       [               w,               W      ],
                        [            eth,             ETH      ]       };
    key <AB02>  {       [               x,               X      ],
                        [             ae,              AE      ]       };
    key <AB03>  {       [               c,               C      ],
                        [           cent                         ]       };
    key <AB04>  {       [               v,               V      ],
                        [       copyright                        ]       };
    key <AB05>  {       [               b,               B      ],
                        [          ssharp                        ]       };
    key <AB06>  {       [               n,               N      ],
                        [         notsign                        ]       };
    key <AB07>  {       [           comma,        question      ],
                        [    questiondown                        ]       };
    key <AB08>  {       [       semicolon,          period      ],
                        [        multiply                        ]       };
    key <AB09>  {       [           colon,           slash      ],
                        [        division                        ]       };
    key <AB10>  {       [          exclam,         section      ],
                        [      exclamdown                        ]       };
    key <RTSH>  {       [         Shift_R                        ]       };

    key <LCTL>  {       [       Control_L                        ]       };
//    key <LALT>        {       [           Alt_L,          Meta_L      ]       };
    key <SPCE>  {       [           space                        ]       };
//    key <RALT>        {       [           Alt_R,          Meta_R      ]       };
    key <RCTL>  {       [       Control_R                        ]       };

    key <FK01> {        [        F1             ]       };
    key <FK02> {        [        F2             ]       };
    key <FK03> {        [        F3             ]       };
    key <FK04> {        [        F4             ]       };
    key <FK05> {        [        F5             ]       };
    key <FK06> {        [        F6             ]       };
    key <FK07> {        [        F7             ]       };
    key <FK08> {        [        F8             ]       };
    key <FK09> {        [        F9             ]       };
    key <FK10> {        [        F10            ]       };
    key <FK11> {        [        F11            ]       };
    key <FK12> {        [        F12            ]       };

    key <PRSC> { type= "PC_SYSRQ", symbols[Group1]= [ Print, Execute ] };
    key <SCLK> {  [  Scroll_Lock        ]       };
```

```
        key <PAUS> { type= "PC_BREAK", symbols[Group1]= [ Pause, Break ] };
        key  <INS> {  [  Insert              ]        };
        key <HOME> {         [  Home                 ]        };
        key <PGUP> {         [  Prior                ]        };
        key <DELE> { [ Delete                ]                        };
        key  <END> {         [  End                  ]        };
        key <PGDN> {         [  Next                 ]        };

        key   <UP> {         [  Up                   ]        };
        key <LEFT> {         [  Left                 ]        };
        key <DOWN> {         [  Down                 ]        };
        key <RGHT> {         [  Right                ]        };

        key <NMLK> {         [  Num_Lock, Pointer_EnableKeys          ]        };
        key <KPDV> {         [  KP_Divide            ]        };
        key <KPMU> {         [  KP_Multiply          ]        };
        key <KPSU> {         [  KP_Subtract          ]        };

        key  <KP7> {         [  KP_Home,      KP_7   ]        };
        key  <KP8> {         [  KP_Up,        KP_8   ]        };
        key  <KP9> {         [  KP_Prior,     KP_9   ]        };
        key <KPAD> {         [       KP_Add          ]        };

        key  <KP4> {         [  KP_Left,      KP_4   ]        };
        key  <KP5> {         [  KP_Begin,     KP_5   ]        };
        key  <KP6> {         [  KP_Right,     KP_6   ]        };

        key  <KP1> {         [  KP_End,       KP_1   ]        };
        key  <KP2> {         [  KP_Down,      KP_2   ]        };
        key  <KP3> {         [  KP_Next,      KP_3   ]        };
        key <KPEN> {         [       KP_Enter        ]        };

        key  <KP0> {         [  KP_Insert,    KP_0   ]        };
        key <KPDL> {  [  KP_Delete, KP_Decimal ]    };

        modifier_map Shift  { Shift_L       };
        modifier_map Lock   { Caps_Lock     };
        modifier_map Control{ Control_L     };
    //    modifier_map Mod1   { Alt_L, Alt_R, Meta_L, Meta_R };
    //    modifier_map Mod2   { Num_Lock };
        modifier_map Mod3   { Mode_switch   };
    //    modifier_map Mod4   { Meta_L, Meta_R };
    //    modifier_map Mod5   { Scroll_Lock };
    };

    // Touches supplémentaires sur un clavier "Windows95"

    // ²/³ (carré/cube) par défaut puisqu'on a mis les guillemets français !
    //    key <LWIN> {        [      twosuperior, threesuperior        ],
    //                        [      onesuperior                       ]  };
    //    key <RWIN> {        [      Mode_switch       ]       };
    //    key <MENU> {        [      Multi_key         ]       };


    xkb_symbols "pc101" {
        key <RALT> { [ Mode_switch,  Multi_key ] };
```

```
        modifier_map Mod3 { Mode_switch };
    };

    default
    xkb_symbols "pc102" {
        key <RALT> { [ Mode_switch,  Multi_key ] };
        modifier_map Mod3 { Mode_switch };
    };

    xkb_symbols "pc104" {
        key <RALT> { [ Mode_switch ] };
        key <LWIN> { [ Meta_L ] };
        key <RWIN> { [ Multi_key ] };
        key <MENU> { [ Menu ] };
        modifier_map Mod3 { Mode_switch };
    };

    xkb_symbols "pc105" {
        key <RALT> { [ Mode_switch ] };
        key <RWIN> { [ Multi_key ] };
        modifier_map Mod3 { Mode_switch };
    };

    partial alphanumeric_keys
    xkb_symbols "nodeadkeys" {
        include "french(basic)"
        key <AE12> {        [           equal,            plus      ]       };
        key <AD11> {        [     asciicircum,       diaeresis      ]       };
        key <AD12> {        [          dollar,        sterling      ]       };
        key <AB07> {        [           comma,        question      ]       };
    };
```

Et enfin voici french.map (à placer dans /usr/lib/kbd/keytables) :

```
    # French keyboard description :
    #
    # Copyright (c) July 1997 Guylhem Aznar <guylhem@danmark.linux.eu.org>
    # Copyright (c) August 1997 Pierre-Charles David <pcdavid@club-internet.fr>
    #  Ajouté : Linefeed, Meta_Control_backslash, Control_backslash, Meta_backslash
    #
    #      This program is free software; you can redistribute it and/or
    #    modify it under the terms of the GNU General Public License version
    #            2 as published by the Free Software Foundation.
    #
    #      This program is distributed in the hope that it will be useful,
    #      but WITHOUT ANY WARRANTY; without even the implied warranty of
    #       MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    #               GNU General Public License for more details.
    #
    #      You should have received a copy of the GNU General Public License
    #         along with this program; if not, write to the Free Software
    #          Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
    #
    # Les définitions de alt & control (keycode) ne sont qu'optionelles ; n'ont
    # donc été redéfinis alt (nombre) et atl + shift (nombre) pour inverser le
    # comportement par défaut : touche = signe ; touche + shift = nombre, or ici
```

```
# alt touche = Meta_nombre ; alt + shift touche = Meta_signe
#
# Shift (pavé numérique) pour taper de l'héxadécimal !
# control alt suppr : rebooter
# control altgr suppr : arrêter
#
# ajouter pour ces deux fonction les lignes suivantes à /etc/inittab :
# ca:12345:ctrlaltdel:/sbin/shutdown -t1 -r now
# kb:12345:kbrequest:/sbin/halt

keycode   0 =

keycode   1 = Escape

keycode   2 = ampersand        one              dead_acute
        alt     keycode 2 = Meta_one
        alt     shift   keycode 2 = Meta_ampersand

keycode   3 = eacute           two              tilde            Eacute
        alt     keycode 3 = Meta_two
        alt     shift   keycode 3 = Meta_asciitilde

keycode   4 = quotedbl         three            numbersign
        alt     keycode 4 = Meta_three
        alt     shift   keycode 4 = Meta_quotedbl

keycode   5 = apostrophe       four             braceleft
        alt     keycode 5 = Meta_four
        alt     shift   keycode 5 = Meta_apostrophe

keycode   6 = parenleft        five             bracketleft
        alt     keycode 6 = Meta_five
        alt     shift   keycode 6 = Meta_parenleft

keycode   7 = minus            six              bar              hyphen
        alt     keycode 7 = Meta_six
        alt     shift   keycode 7 = Meta_minus

keycode   8 = egrave           seven            dead_grave       Egrave
        alt     keycode 8 = Meta_seven
        alt     shift   keycode  8 = Meta_grave

keycode   9 = underscore       eight            backslash        macron
        alt     keycode 9 = Meta_eight
        alt     shift   keycode 9 = Meta_underscore

keycode  10 = ccedilla         nine             circumflex       Ccedilla
        alt     keycode 10 = Meta_nine
        alt     shift   keycode 10 = Meta_asciicircum

keycode  11 = agrave           zero             at               Agrave
        alt     keycode 11 = Meta_zero
        alt     shift   keycode 11 = Meta_at

keycode  12 = parenright       degree           bracketright
        alt     keycode 12 = Meta_parenright
```

TOC

```
          alt     shift   keycode  12 = Meta_bracketright

keycode  13 = equal            plus             braceright
          alt     keycode 13 = Meta_equal
          alt     shift   keycode  13 = Meta_plus

keycode  14 = Delete           Remove
          alt     control keycode 14 = Boot
          altgr   control keycode 14 = KeyboardSignal

keycode  15 = Tab

keycode  16 = a                A                acircumflex     adiaeresis

keycode  17 = z                Z                aring           Aring

keycode  18 = e                E                ecircumflex     ediaeresis

keycode  19 = r                R                registered

keycode  20 = t                T                thorn           THORN

keycode  21 = y                Y                yen

keycode  22 = u                U                ucircumflex     udiaeresis

keycode  23 = i                I                icircumflex     idiaeresis

keycode  24 = o                O                ocircumflex     odiaeresis

keycode  25 = p                P                paragraph

keycode  26 = dead_circumflex  dead_diaeresis   dead_cedilla    dead_tilde

keycode  27 = dollar           pound            currency

keycode  28 = Return
          alt     keycode  28 = 0x080d
          altgr   keycode  26 = Linefeed

keycode  29 = Control

keycode  30 = q                Q                Acircumflex     Adiaeresis

keycode  31 = s                S                oslash          Oslash

keycode  32 = d                D                Ecircumflex     Ediaeresis

keycode  33 = f                F                ordfeminine

keycode  34 = g                G                acute

keycode  35 = h                H                periodcentered

keycode  36 = j                J                Ucircumflex     Udiaeresis
```

```
keycode  37 = k                K                Icircumflex     Idiaeresis

keycode  38 = l                L                Ocircumflex     Odiaeresis

keycode  39 = m                M                masculine

keycode  40 = ugrave           percent         plusminus        Ugrave
        alt     keycode 40 = Meta_semicolon

# Mode math
#keycode  41 = twosuperior      threesuperior   onesuperior     periodcentered
# Mode littéraire + math
keycode  41 = guillemotleft    guillemotright   twosuperior     threesuperior

keycode  42 = Shift

keycode  43 = asterisk         mu               onehalf

keycode  44 = w                W                +eth            +ETH

keycode  45 = x                X                +ae             +AE

keycode  46 = c                C                cent

keycode  47 = v                V                copyright

keycode  48 = b                B                ssharp

keycode  49 = n                N                notsign

keycode  50 = comma            question        questiondown

keycode  51 = semicolon        period          multiply

keycode  52 = colon            slash           division

keycode  53 = exclam           section         exclamdown

keycode  54 = Shift

keycode  55 = KP_Multiply      Hex_A           Hex_D

keycode  56 = Alt

keycode  57 = space
        control keycode  57 = nul
        alt     keycode  57 = Meta_space
        altgr   keycode  57 = nobreakspace

# Très utile pour tous ceux ne disposant pas de leurs deux mains
keycode  58 =  CapsShift
# Caps_Lock
# ou au choix CapsShift ou Caps_On !
        shift   keycode  58 = Shift_Lock
        control keycode  58 = Control_Lock
        alt     keycode  58 = Alt_Lock
```

TOC

```
          altgr   keycode  58 = AltGr_Lock

keycode  59 = F1                F11               Console_13
          alt     keycode  59 = Console_1
          control alt     keycode  59 = Console_1

keycode  60 = F2                F12               Console_14
          alt     keycode  60 = Console_2
          control alt     keycode  60 = Console_2

keycode  61 = F3                F13               Console_15
          alt     keycode  61 = Console_3
          control alt     keycode  61 = Console_3

keycode  62 = F4                F14               Console_16
          alt     keycode  62 = Console_4
          control alt     keycode  62 = Console_4

keycode  63 = F5                F15               Console_17
          alt     keycode  63 = Console_5
          control alt     keycode  63 = Console_5

keycode  64 = F6                F16               Console_18
          alt     keycode  64 = Console_6
          control alt     keycode  64 = Console_6

keycode  65 = F7                F17               Console_19
          alt     keycode  65 = Console_7
          control alt     keycode  65 = Console_7

keycode  66 = F8                F18               Console_20
          alt     keycode  66 = Console_8
          control alt     keycode  66 = Console_8

keycode  67 = F9                F19               Console_21
          alt     keycode  67 = Console_9
          control alt     keycode  67 = Console_9

keycode  68 = F10               F20               Console_22
          alt     keycode  68 = Console_10
          control alt     keycode  68 = Console_10

keycode  69 = Num_Lock          Bare_Num_Lock

keycode  70 = Scroll_Lock       Show_Memory       Show_Registers  Show_State

keycode  71 = KP_7              Hex_7
          alt     keycode  71 = Ascii_7

keycode  72 = KP_8              Hex_8
          alt     keycode  72 = Ascii_8

keycode  73 = KP_9              Hex_9
          alt     keycode  73 = Ascii_9

keycode  74 = KP_Subtract       Hex_B             Hex_E
```

```
            alt     keycode  74 = minus

   keycode  75 = KP_4             Hex_4
            alt     keycode  75 = Ascii_4

   keycode  76 = KP_5             Hex_5
            alt     keycode  76 = Ascii_5

   keycode  77 = KP_6             Hex_6
            alt     keycode  77 = Ascii_6

   keycode  78 = KP_Add           Hex_C            Hex_F
            alt     keycode  78 = plus

   keycode  79 = KP_1             Hex_1
            alt     keycode  79 = Ascii_1

   keycode  80 = KP_2             Hex_2
            alt     keycode  80 = Ascii_2

   keycode  81 = KP_3             Hex_3
            alt     keycode  81 = Ascii_3

   keycode  82 = KP_0             Hex_0
            alt     keycode  82 = Ascii_0

   keycode  83 = KP_Period         KP_Comma
            altgr   control keycode  83 = KeyboardSignal
            control alt     keycode  83 = Boot

   # SysReq = Alt+Print Screen = 84
   keycode  84 = Last_Console

   keycode  85 =

   keycode  86 = less             greater          bar              brokenbar

   keycode  87 = F11              F11              Console_23
            alt     keycode  87 = Console_11
            control alt     keycode  87 = Console_11

   keycode  88 = F12              F12              Console_24
            alt     keycode  88 = Console_12
            control alt     keycode  88 = Console_12

   keycode  89 =

   keycode  90 =

   keycode  91 =

   keycode  92 =

   keycode  93 =

   keycode  94 =
```

```
keycode  95 =

keycode  96 = KP_Enter

keycode  97 = Control

keycode  98 = KP_Divide

# Print Screen
keycode  99 = Control_backslash
        control keycode  99 = Control_backslash
        alt     keycode  99 = Control_backslash

keycode 100 = AltGr
        alt     keycode 100 = Compose

# Ctrl+Pause = Break = 101
keycode 101 = Break

keycode 102 = Home

keycode 103 = Up

keycode 104 = PageUp
        shift   keycode 104 = Scroll_Backward

keycode 105 = Left
        alt     keycode 105 = Decr_Console

keycode 106 = Right
        alt     keycode 106 = Incr_Console

keycode 107 = End

keycode 108 = Down
        alt     keycode 108 = Last_Console

# PageDown == Next
keycode 109 = PageDown
        shift   keycode 109 = Scroll_Forward

keycode 110 = Insert

keycode 111 = BackSpace       BackSpace      SAK
        alt     keycode 111 = Meta_backslash
        control keycode 111 = Control_backslash

keycode 112 = Macro

keycode 113 = F13

keycode 114 = F14

keycode 115 = Help
```

```
keycode 116 = Do

keycode 117 = F17

keycode 118 = KP_MinPlus

keycode 119 = Pause

keycode 120 =

keycode 121 =

keycode 122 =

keycode 123 =

keycode 124 =

# Fenêtre gauche = Alt puisque d'habitude c'est la position d'Alt !
keycode 125 = Alt

# Fenêtre droite = AltGr puisque d'habitude c'est la position d'AltGr !
keycode 126 = AltGr

# Menu = compose (mélanger des signes et des lettres ex : (compose) ' e = é
keycode 127 = Compose

string F1 = "\033[[A"
string F2 = "\033[[B"
string F3 = "\033[[C"
string F4 = "\033[[D"
string F5 = "\033[[E"
string F6 = "\033[17~"
string F7 = "\033[18~"
string F8 = "\033[19~"
string F9 = "\033[20~"
string F10 = "\033[21~"
string F11 = "\033[23~"
string F12 = "\033[24~"
string F13 = "\033[25~"
string F14 = "\033[26~"
string F15 = "\033[28~"
string F16 = "\033[29~"
string F17 = "\033[31~"
string F18 = "\033[32~"
string F19 = "\033[33~"
string F20 = "\033[34~"
string F21 = ""
string F22 = ""
string F23 = ""
string F24 = ""
string F25 = ""
string F26 = ""
string Home = "\033[1~"
string Insert = "\033[2~"
string Remove = "\033[3~"
```

```
string End = "\033[4~"
string PageUp = "\033[5~"
string PageDown = "\033[6~"
string Macro = "\033[M"
string Pause = "\033[P"

compose '`' 'A' to 'À'
compose '`' 'a' to 'à'
compose '\'' 'A' to 'Á'
compose '\'' 'a' to 'á'
compose '^' 'A' to 'Â'
compose '^' 'a' to 'â'
compose '~' 'A' to 'Ã'
compose '~' 'a' to 'ã'
compose '"' 'A' to 'Ä'
compose '"' 'a' to 'ä'
compose 'O' 'A' to 'Å'
compose 'o' 'a' to 'å'
compose '0' 'A' to 'Å'
compose '0' 'a' to 'å'
compose 'A' 'A' to 'Å'
compose 'a' 'a' to 'å'
compose 'A' 'E' to 'Æ'
compose 'a' 'e' to 'æ'
compose ',' 'C' to 'Ç'
compose ',' 'c' to 'ç'
compose '`' 'E' to 'È'
compose '`' 'e' to 'è'
compose '\'' 'E' to 'É'
compose '\'' 'e' to 'é'
compose '^' 'e' to 'ê'
compose '"' 'E' to 'Ë'
compose '"' 'e' to 'ë'
compose '`' 'I' to 'Ì'
compose '`' 'i' to 'ì'
compose '\'' 'I' to 'Í'
compose '\'' 'i' to 'í'
compose '^' 'I' to 'Î'
compose '^' 'i' to 'î'
compose '"' 'I' to 'Ï'
compose '"' 'i' to 'ï'
compose '-' 'D' to 'Ð'
compose '~' 'N' to 'Ñ'
compose '~' 'n' to 'ñ'
compose '`' 'O' to 'Ò'
compose '`' 'o' to 'ò'
compose '\'' 'O' to 'Ó'
compose '\'' 'o' to 'ó'
compose '^' 'O' to 'Ô'
compose '^' 'o' to 'ô'
compose '~' 'O' to 'Õ'
compose '~' 'o' to 'õ'
compose '"' 'O' to 'Ö'
compose '"' 'o' to 'ö'
compose '/' 'O' to 'Ø'
compose '/' 'o' to 'ø'
```

```
compose '`' 'U' to 'Ù'
compose '`' 'u' to 'ù'
compose '\'' 'U' to 'Ú'
compose '\'' 'u' to 'ú'
compose '^' 'u' to 'û'
compose '"' 'U' to 'Ü'
compose '"' 'u' to 'ü'
compose '\'' 'Y' to 'Ý'
compose '\'' 'y' to 'ý'
compose 's' 's' to 'ß'
compose 's' 'z' to 'ß'
# si quelqu'un veut bien y inclure des «compose» pour les autres caractères
# latin1 comme les guillemets français, le copyright (...) qu'il me les envoie !
```

Ftape-HOWTO

Claus-Justus Heine, <heine@math1.rwth-aachen.de>

v3.0, August 1998

This HOWTO discusses essential do's and dont's for the `ftape` floppy tape driver under Linux. It focusses on the newest version which is `ftape-4.02` at the time of this writing. This HOWTO is to be intended as first step help and source of information. The `ftape` driver interfaces to QIC-40, QIC-80, QIC-3010 and QIC-3020 compatible drives, and to the Iomega Ditto 2GB and Ditto Max drives. The QIC-3010 and QIC-3020 standards are also known as `Travan' (TR-2 and TR-3). These drives connect via the floppy disk controller (**FDC**) which may be either an internal FDC or inside of certain parallel port floppy tape drives. Please refer to the section Supported drives for further information. `ftape` **does not** cover SCSI or QIC-02 tape drives. DAT tape drives usually (always?) connect to a SCSI controller. This is but one of the Linux HOWTO documents. You can get an index of the HOWTOs from the Linux HOWTO index, while the real HOWTO's can be fetched (using `ftp`) from `sunsite.unc.edu:pub/Linux/doc/HOWTO` (this is the ``official'' place) or via the World Wide Web from the Linux Documentation Project home page.

1. Legalese

2. Revision History

3. The preliminaries

- 3.1 Other sources of information
- 3.2 Contacts
- 3.3 What is `ftape`

4. Getting and installing `ftape`

- 4.1 Getting `ftape`
- 4.2 Differences between `ftape-2.x`, `ftape-3.x` and `ftape-4.x` versions

- 4.3 Installing the driver with v2.0.x and earlier kernels
- 4.4 Installing the driver with v2.1.x and later kernels
- 4.5 Following the development of the `ftape` driver
- 4.6 Mixing `ftape` and floppies

5. The Care and Feeding of Tape and Tape Drives

- 5.1 Formatting
- 5.2 Retensioning
- 5.3 Drive Cleaning
- 5.4 Repairing de-spooled cartridges

6. Hardware support

- 6.1 Supported tape drives
- 6.2 Supported special controllers
- 6.3 Unsupported tape drives
- 6.4 Using an external tape drive with `ftape`
- 6.5 PCI motherboards and `ftape`

7. Backing up and restoring data

- 7.1 Writing an archive to a tape
- 7.2 Restoring an archive
- 7.3 Testing the archive
- 7.4 Putting more than one backup on a tape
- 7.5 Appending files to an archive
- 7.6 Mount/unmounting tapes

8. Creating an emergency boot floppy for `ftape`

9. Frequently Asked Questions

10. FAQ: "Compiling and installing Ftape" related questions !

- 10.1 What Ftape version should I use?
- 10.2 I'm having problems getting my XYZ drive to run under the 2.0.xx kernel with the built-in driver. How do I fix this?
- 10.3 I'm running Linux/SMP and the system just freezes when trying to access the Ftape devices!
- 10.4 Why does depmod complain about "undefined symbols"?
- 10.5 "insmod" says the kernel version is wrong
- 10.6 "insmod" says that kernel 1.2.0 and 1.2.0 differ
- 10.7 Trying to compile Ftape gives me the error "modversions.h: no such file or directory"
- 10.8 What is this versioned symbols stuff anyway?
- 10.9 I seem to be getting sftape instead of zftape. When I run "ftmt status" command, I get output that the Ftape docs says corresponds to sftape (/dev/qft0: Invalid argument). Why?
- 10.10 My Ditto DASH/FC-20/Exabyte Accelerator card works under Microsoft Windows, but I get a drive not found type of error in /var/log/messages when trying to use it under Linux.
- 10.11 Ftape DMA transfers gives ECC errors
- 10.12 Help! I'm getting 'dmaalloc() failed' in my syslog file.
- 10.13 Syslogd works overtime when running Ftape
- 10.14 How do I change the trace-level?
- 10.15 I'm having problems with Ftape. I'm using the latest version of Ftape from the Ftape Home Page and believe that I've located a real bug. What should I do?

11. FAQ: "Using Ftape" related questions !

- 11.1 How fast is Ftape ?
- 11.2 When I write to some of my tapes, they seem to spend a lot of time "shoe-shining," or repositioning instead of streaming. Is something wrong with my system?
- 11.3 Do I have to reboot to the DOS world to format tapes?
- 11.4 Is it possibly to format Ditto 2GB tapes with ftape?
- 11.5 Is it possibly to format Ditto Max or Max Pro tapes with ftape?
- 11.6 Ftape detects more bad sectors than DOS on QIC-3020 tapes
- 11.7 Is it ok that I'm not hearing the tape move when I do a fsf or a bsf with mt?
- 11.8 Why does my XYZ backup program complain about "Invalid argument" errors?

- 11.9 I/O errors and FDC - some explanations.
- 11.10 Why do I get "/dev/qft0: No such device" errors?
- 11.11 I get "device busy" when I make multiple backups on a tape using some script.
- 11.12 How do I "..." with tar?
- 11.13 What block-size should I use with tar ?
- 11.14 Where can I find the tar/mt/cpio/dd binaries - sources - manpages?
- 11.15 If I use tapers compression, is it a bad idea to use the compression with zftape, or would it be better to not use tapers compression, and let zftape do it?
- 11.16 How does zftape compression compare to say gzip -9?
- 11.17 I don't trust compression, but hear that the sftape interface is going away. What should I do?
- 11.18 Ftape says "This tape has no 'Linux raw format"
- 11.19 Can I exchange tapes with someone using DOS?
- 11.20 How does `mt eom' work when you've started overwriting a tape in the middle?
- 11.21 When I made backups before using taper, under the 2.0.29 ftape my drive didn't support fsf, under the new zftape it does, why would this be, and what exactly is fsf ?
- 11.22 What exactly is the difference between ftape, and zftape?
- 11.23 What is the difference between a rewinding, and non rewinding drive?
- 11.24 Can someone tell me how to use mt to rewind my TR-3 drive one record using zftape record, so I can verify it?
- 11.25 By non-rewinding, they mean that it doesn't automatically rewind, correct? It doesn't mean that under no circumstances it will rewind, right? I tried using /dev/zqft0, and it instantly rewinds the tape.
- 11.26 What is the difference between what mt considers a record and what it considers a file?
- 11.27 Reusing tapes with zftape without reformatting the tape.
- 11.28 This script implements a simple contents listing for the zftape package using the "MTIOCVOLINFO" ioctl.

12. FAQ: "Tape and Drivers" related questions !

- 12.1 What are good makers of Travan tapes?
- 12.2 Where can I obtain the QIC standards?
- 12.3 Is the Iomega Ditto 2GB drive supported?
- 12.4 Is the Iomega Ditto Max drive supported?
- 12.5 Is the Iomega Ditto Max Pro drive supported?

TOC

13. FAQ: Miscellaneous !

- 13.1 How to subscribe to the Ftape Mailing List?
- 13.2 How to un-subscribe from the Ftape Mailing List?
- 13.3 Links to related information.

14. Debugging the `ftape` driver

- 14.1 The kernel/`ftape` crashes on me when I do `...' - is that a bug?
- 14.2 OK, it's a bug ...ehhh... feature - How do I submit a report?

15. Contributions

---

1. Legalese

The Linux ftape-HOWTO may be reproduced and distributed in whole or in part, subject to the following conditions:


```
Copyright (c) 1993-1996 by Kai Harrekilde-Petersen
Email: khp@dolphinics.no

Copyright (c) 1996-1997 by Kevin Johnson
Email: kjj@pobox.com

Copyright (c) 1998 by Claus-Justus Heine
Email: heine@math1.rwth-aachen.de
```

The Linux ftape-HOWTO is a free document; you may reproduce and/or modify it under the terms of version 2 (or, at your option, any later version) of the GNU General Public License as published by the Free Software Foundation.

This HOWTO is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

The author encourages wide distribution of this document for personal or commercial use, provided that the above copyright notice remains intact and the provisions of the GNU General Public License are adhered to. The summary is that you may copy and distribute this document free of charge, or for a profit. No explicit permission is required from the author for reproduction of this document in any medium, physical or electronic.

Note that derivative works and translations of this document must be placed under the GNU General Public License, and the original copyright notice must remain intact. If you have contributed new material to this document, you must make the source code (e.g., SGML source) available for your revisions. Please make revisions and updates available directly to the author: Contact heine@math1.rwth-aachen.de via Internet e-mail. This will allow the author to merge updates and provide consistent revisions to the Linux community.

The author encourages distributors of Linux software in any medium to use the HOWTO as an installation and user guide. Given the copyright above, you are free to print and distribute copies of this document with your software. If doing so, you may wish to include a short ``installation supplement'' for your release, or modify the relevant sections of this book to reflect your product.

The author would like to know of any plans to publish and distribute this HOWTO commercially. In this way, we can ensure that you are kept up-to-date with new revisions. And, should a new version be right around the corner, you might wish to delay your publication of the HOWTO until it is available.

If you are distributing this HOWTO commercially, donations, royalties, and/or printed copies are gr
appreciated by the author. Contributing in this way shows your support for free software and the Lin
Documentation Project.

If you have questions or comments, please contact the author at

heine@math1.rwth-aachen.de

---

2. Revision History

version 3.0 (August, 1998)

- Additions to list of supported hardware.
- New section about differences between ftape versions.
- Pointers to the Ftape-FAQ and the Ftape manual.
- Updated to ftape-4.02.
- Additions to the FAQ.
- Update all URLs.

version 2.0 (March 15, 1997)

- Updated to ftape v2.11 and v3.xx
- Lots of updates.

version 1.9 (September 20, 1996)

- New maintainers of ftape and the HOWTO.
- A few minor formatting and spelling fixes.
- Updated for Linux v2.0.
- Started to integrate some of Andrew Martin's ftape info.

version 1.8 (May 22, 1996)

- Copyright policy changed to GNU GPL v2
- The maintainer's email address has changed.
- Updated to ftape-2.08
- `ftape` is now a part of the kernel distribution.

version 1.7.1 (February 13, 1996)

- Updated to ftape-2.06b

version 1.7 (January 28, 1996)

- Updated to ftape-2.06 and modules-1.3.57

version 1.6.2 (January 23, 1996)

- Connor TST3200R drive added
- Updated 2Mbps fdc information.

version 1.6.1 (January 16, 1996)

- minor corrections

version 1.6 (January 10, 1996)

- New maintainer of `ftape`
- updated to v2.05

- added new drives

3. The preliminaries

3.1 Other sources of information

ftape version 3

`ftape-3.x` came with a manual of its own, which is contained in the `ftape-3.04d` package available from the usual places. See Getting Ftape.

ftape version 4

`ftape-4.x` also has a documentation package `ftape-doc` which is available from the usual places. This Ftape-HOWTO, however, also focusses on `ftape-4.x` and is meant as an entry point to the available documentation. See Getting Ftape.

ftape-tools

The `ftape-tools` package (including useful utilities for `ftape`) comes with its own manual. See Getting Ftape.

Ftape-FAQ

The `Ftape-FAQ` is included wordly in this manual, but more recent versions may be found at http://www.correct.nl/~ftape.

3.2 Contacts

The maintainer of the source for `ftape` is Claus Heine <heine@math1.rwth-aachen.de>. He has a web page at http://www-math.math.rwth-aachen.de/~LBFM/claus/ftape/.

If you have a problem or questions about ftape, try posting to the `Linux Tape` mailing list `linux-tape@vger.rutger.edu` (see Following the ftape development below). There also used to be a newsgroup that mirrored the mailing list traffic but it has vanished some time ago.

I use `ftape` (it is my sole means of backing up on my linux box :-). I hesitate to make recommendations on what hardware to buy. See the section Supported drives and Unsupported drives for a list of supported and unsupported drives.

You should try to post a summary of your problems and its solution(s), after you've got it working, even if you only got it partially working. Please also send a copy copy of your solution to the Linux Tape mailing list at <linux-tape@vger.rutgers.edu> so that it can be added to the HOWTO and/or the FAQ.

If you receive this as part of a printed distribution or on a CD-ROM, please check out the Linux Documentation home page or ftp to ftp://sunsite.unc.edu:/pub/Linux/doc/HOWTO to see if there exists a

TOC

more recent version. This could potentially save you a lot of trouble.

If you email me, please include the string `ftape` in the subject line. This will help ensure the mail doesn't inadvertently get buried. But preferrably you should email to the Linux Tape mailing list at <linux-tape@vger.rutgers.edu> instead of contacting me directly.

3.3 What is `ftape`

`ftape` is a driver program that controls various low-cost tape drives that connect to the floppy controller.

`ftape` is not a backup program as such; it is a device driver, which allows you to use the tape drive (just like the SoundBlaster 16 driver let you use your sound card) through the device files `/dev/[n]qft[0-3]`.

`ftape` was originally written by Bas Laarhoven `<bas@vimec.nl>`, with ``a little help from his friends'' to sort out the ECC (Error Correcting Code) stuff. `ftape` is copyrighted by Bas under the GNU General Public License, which basically says: ``go ahead and share this with the world, just don't disallow other people from copying it further''.

`ftape` has undergone several changes since then. While the Linux-2.0.x kernel series still contains `ftape-2.08` the v2.1.x and soon the v2.2.* kernel series come with `ftape-3.x` (hopefully even with `ftape-4.02`, but this wasn't clear at the time of this writing) which differs in some points from the `ftape-2.x` driver. Since version `3.00` the `ftape` driver has been maintained by me (Claus-Justus Heine); it has been changed and improved in several respects and support for new hardware has been added.

`ftape` is quite stable, and has been that for some time now. It is reliable enough for critical backups (but it's always a good idea to check your backups, so you won't get a nasty surprise some day).

`ftape` supports drives that conform to the QIC-117 and one of the QIC-80, QIC-40, QIC-3010, and QIC-3020 standards as well as the Iomega Ditto 2GB and Ditto Max drives which no longer strictly conform to the QIC standards in all respects.

`ftape` can drive floppy tape drives that connect to the internal FDC as well as certain parallel port floppy tape drives.

`ftape` supports neither QIC-02, IDE (ATAPI), nor SCSI tape drives. SCSI drives are accessed as `/dev/[n]st[0-7]` and are supported by the kernel through the SCSI drivers. If you look for help on SCSI tape drives, you should read the `SCSI-howto`. ATAPI tape drives are supported by the kernel since 1.3.46. See section Supported drives and Unsupported drives for a list of supported and unsupported drives.

---

4. Getting and installing `ftape`

4.1 Getting `ftape`

The v2.0.x versions of the kernel include version 2.08 of `ftape` I recommend, however, that you grab the latest version of the full source code package for `ftape`. It is a newer version, includes files that are not included in the kernel v2.0.X distribution, and includes much better documentation about how to install `ftape`. The v2.1.x and later versions of the kernel include the version 3.04 of `ftape`.

I recommend that you download the latest stable version of `ftape` which is 4.02 at the time of this writing and is available from

http://www-math.math.rwth-aachen.de/~LBFM/claus/ftape/archives.html

as well as from

ftp://sunsite.unc.edu/pub/Linux/kernel/tapes/.

You probably should also grab the `ftape-doc` and the `ftape-tools` package that are available from the same locations.

If you still want to use the `ftape-2.08` which is shipped with the v2.0.x kernels, then you get a version of the driver which is really out of date and doesn't support QIC-3020 tapes at 2Mbps correctly, neither does it support the Ditto 2GB drives nor the Ditto Max drives nor any kind of parallel port tape drive. The section Supported drives gives detailed information about which version of the `ftape` driver supports which hardware.

4.2 Differences between `ftape-2.x`, `ftape-3.x` and `ftape-4.x` versions

`ftape-3.x` and `ftape-4.x` use the file system interface that was implemented for a branch release which was called `zftape`. Actually, the module that implements the **VFS** (`Virtual File System`) interface of `ftape-3.x` and `ftape-4.x` still is called `zftape.o` and its `C`-sources inside the kernel tree reside in `[/usr/src/linux/]drivers/char/ftape/zftape/`.

`ftape-2.x` (i.e. the version still contained in the v2.0.x kernel) uses another file system interface, that was implemented by `ftape's` original author Bas Larhoven.

File Marks

> The conceptional difference between `ftape-2.x` and later versions of `ftape` is the way *file marks* are implemented.

TOC

Floppy tape devices don't have real file marks.

> **File marks** are used to distinguish different backup sets if you write multiple backup sets to a tape. SCSI and QIC-150 tapes have *real* file marks, i.e. between two different backup sets there is a region on the tape that is written special data to so that the drive logic can detect that marker when the tape is wound with (possibly) high speed over those file marks.

Because the goal of `ftape's` file system interface was from the beginning on to provide an interface that could be used with standard Unix-like tape utilities (i.e. `mt`) the developers of `ftape` started to emulate file marks by storing the positions on the tape where a file mark should be located in certain fields of the header segments.

> **header segments** refers to a region at the beginning of the tape sized two times 29k to hold some important information about the tape format and size and some status information.

However, the `QIC` standards already designate a special region to store such information in, the so called **volume table segment**. Since `ftape-3.x` this *volume table segment* is used instead of using unused data fields in the header segment. As a result it is possible to use your tape cartridge with different operating systems in the sense that your Win or DOS backup program will realize that certain regions of the tape cartridge are already occupied with data, and `ftape-3.x` and later will detect the regions used by those DOS and Win programs. However, you can't extract a DOS backup set under Linux or extract a volume written by `ftape` under DOS, safe you write your own software to do that.

IOCTL interface

There are certain differences in the **IOCTL**

> This *IO control* interface is used by e.g. `mt` to rewind the tape or skip to the next file mark or do any other tape operation.

interface between `ftape-2.x` and `ftape-3.x` and later. A detailed description can be found in the `ftape-manual` contained in the `ftape-doc` package. See Getting Ftape.

Formatting

Formatting of cartridges is supported with `ftape-3.x` and later **only**. Please get the `ftape-tools` package that contains the `ftformat` program that interfaces to the driver to format cartridges. See Getting Ftape. The `ftape-tools` package comes with (more or less) detailed documentation, so the case of formatting cartridges is not dealt with in this document.

Compression

`ftape-3.x` supported user transparent on-the-fly compression in software. This feature (or *bug*) has vanished in `ftape-4.x` as it made further improvements concerning the realiability of backups very very hard. This means, `ftape-4.x` comes without compression support.

However **de-compression** of compressed archives produced with `ftape-3.x` **is** supported in order not to brake existing backup programs where a user-level filter would not suffice to preserve compatibility. Think, e.g., of `taper` which calls the `MTIOC` ioctls itself instead of relying on the `mt` program to perform tape operations.

The `ftape-manual` contained in the `ftape-doc` package contains much more detailed informat about `ftape`s file system interface as well as implementation notes which by far exceed the scope of this HOWTO. See Getting Ftape for informations about where to obtain the manual.

4.3 Installing the driver with v2.0.x and earlier kernels

The following section provides some useful information to get you going with the installation of v4.x which is **not** shipped with the kernel source tree yet but has to be downloaded separately, see the section Getting ftape above.

Once you've downloaded the source code (probably `ftape-4.02-tar.gz`), untar it. You can do this by determining what directory you want the source code to be located in. I recommend `/usr/src/` or `~/src`. When the tar file is extracted, it will dump everything into a `ftape-4.02` subdirectory, so that you'll end up, in the example I've given, with something like `/usr/src/ftape-4.02` or `~/src/ftape-4.02`.

NOTE: you cannot compile `ftape-4.02` into your v2.0.x kernel. Instead, configure your kernel to **not** compile the `ftape` driver and follow the installation instructions in the `ftape-4.02` distribution and install `ftape-4.02` as a module.

Read the `README` file. The `README` is required reading. It's the top of the tree, so to speak. If there are specific files that the `README` tells you to read then read them. It will make the process much less complicated.

Do NOT proceed with compiling the package until you have read the appropriate `README` files and the `INSTALL` file.

Afterwards you need to edit the `MCONFIG` file and configure you package according to your hardware. The `MCONFIG` file contains lots of explanations so it should be fairly easy to go along with it.

However, most of the hardware configuration can be done via setting parameters during module load time so most parameters specified in the file `MCONFIG` simply give the default configuration, but you don't need to recompile the driver to change IO addresses or interrupt settings. The file `INSTALL` and the file `modules/insert` contain examples how to specify the proper module parameters when loading the kernel modules, so I won't go into further detail here.

If you are using a Linux-v1.3.x kernel, you should consider moving to v2.0.x. v1.3.x was the development release prior to the production release v2.0.x.

4.4 Installing the driver with v2.1.x and later kernels

Comment

Maybe `ftape-4.02` will be included into the v2.2.x kernel, but this isn't clear at the time of this writing. This HOWTO will be revised appropriately when this has become clear. So long you have to

TOC

refer to the previous section Installing the driver with v2.0.x and earlier kernels and disregard the contents of this section.

The Linux kernel v2.1.x and later already include `ftape-4.x` so you don't need to download the `ftape-4.x` kernel driver package.

`ftape-4.x` as included in the v2.1.x versions of the kernel can be completely configured using the kernel configuration menus (either with `make menuconfig` or `make xconfig`. Also, there is online help available that documents each parameter setting which I won't repeat here.

The various boot- and loadtime parameter settings are explained in the file

`[/usr/src/linux/]Documentation/ftape.txt`

of the Linux-v2.1.x and later kernel distributions.

4.5 Following the development of the `ftape` driver

If you want to follow the development of the `ftape` driver, you should subscribe to the Linux Tape mailing list `linux-tape@vger.rutgers.edu`. To do so you need to send an email saying `` `subscribe linux-tape' `` (*in the body*) to `majordomo@vger.rutgers.edu`. When you subscribe, you will be sent a greeting mail, which will tell you how to submit real mails and how to get off the list again. **Store this email in a safe place**. Please.

Please note that I do not, repeat **DO NOT**, have any special powers with regard to this mailing list. If you're stuck on the list, don't bother to tell me that. I can only shrug and send you my sympathy (but that won't get you off the list).

4.6 Mixing `ftape` and floppies

If you use your floppy tape drive with the standard FDC then the floppy drive and the floppy tape drive cannot run concurrently as they share the same hardware, the FDC, and the `floppy` and the `ftape` driver do not talk to each other. Thus, if you have mounted a floppy and then try to access the tape drive, `ftape` will complain that it cannot grab IRQ6 and then die. This is especially a problem when designing a emergency disk for use with ftape. This solution is to either load the boot/root disk into a ramdisk and then unmount the floppy, or have two floppy drive controllers.

---

5. The Care and Feeding of Tape and Tape Drives

5.1 Formatting

Before a tape can be used, it must be formatted. The formatting process lays out sector information onto the tape. Other tape interfaces don't typically require formatting. The reason floppy tapes do is that they need to look like a floppy (kinda gross, but what the hey - it works :-).

Can I format my tapes under Linux?

Yes, you can, if you use `ftape-3.04d` or above. To format a floppy tape cartridge you need a user level tool called `ftformat` as well which is contained in the `ftape-tools` distribution (see section Getting ftape).

The `ftape-tools` package comes with its own manual, so I do not need to repeat here how to use `ftformat`.

Which formatting programs can I use under DOS?

The following are known to work:

- Colorado Memory System's software (`tape.exe`)
- Conner Backup Basics v1.1 and all Windows versions
- Norton Backup
- QICstream version 2
- Tallgrass FileSecure v1.52
- Escom Powerstream 3.0 (`qs3.exe` -- QICstream v3?)

These programs are known to be more or less buggy:

- Conner Backup Basics 1.0
- Colorado Windows tape program
- CP Backup (wastes tape space, but is OK apart from that)

As a general rule, most software under DOS should work. The Conner Backup Basics v1.0 has a parameter off by one (someone could not read the QIC-80 specs right!), which is corrected in version 1.1. However, `ftape` detects this, and will work around it. Dennis T. Flaherty (`<dennisf@denix.elk.miles.com>`) report that Conner C250MQ owners can obtain the new v1.1, by calling Conner at 1-800-4Conner (in the US) and ask for an upgrade (for a nominal fee for the floppy). The Windows versions should work fine. Some versions of Colorado's tape program for windows, has an off-by-one error in the number of segments. `ftape` also detect and work around that bug.

TOC

Central Point Backup can be used, but it wastes precious tape space when it encounters a bad spot on tape.

NOTE: If you are running a formatting software under DOS, which is not mentioned here, please mail the relevant info to me (<heine@math1.rwth-aachen.de>), so I can update the list.

5.2 Retensioning

QIC tapes are particularly sensitive to tape stretch. The reason is that floppy tapes are pre-formatted with sector information, whereas other tape types have their sync information written as the data is written to the tape. If the floppy tape stretches and the sync fields get out of sync the result will be read errors. The problem is worse with longer tapes.

It is a good idea to retension new tapes a few times before using them and before formatting them. You should also try retensioning the tape if you are start getting read errors. It might also be a good idea retension the tape before a backup.

5.3 Drive Cleaning

The coating on the tape is an oxide compound. As the tape is dragged across the tape head it has a tendency to leave tiny amounts of residue on the head. You should periodically use a tape cleaner - following the specs for the drive in question. Tape cleaners should be available from any distributer of tapes.

One more additional note about tape cleaning. You might want to clean the drive after the first use of a brand new tape. A brand new tape will typically leave quite a bit of residue the first time it's used.

Thanks to Neal Friedman for the explanation and suggestion that this information be included in the HOWTO.

5.4 Repairing de-spooled cartridges

In rare occasions it can happen that the tape drive doesn't detect the **EOT** (End Of Tape) markers correctly. These markers are simply holes in the tape which are detected by the tape drive with means of a little photo-transistor (or the like).

The manual of your tape drive will probably give you proper hints how to clean those EOT detectors.

However, if the EOT detection fails, then the tape drive despooles the cartridge because the tape isn't glued to the wheels, but hold by friction only.

There are detailed instructions how to fix such a despooled tape at the Iomega WWW pages at

http://www.iomega.com/support/techs/ditto/3006.html

and at the Hewlett Packard WWW pages at

http://www.hp.com/isgsupport/cms/docs/lpg12020.html

If the pages shouldn't be in the exact locations as given above, then please try to browse a little bit through the web pages of HP or Iomega until you find the needed information.

6. Hardware support

6.1 Supported tape drives

All drives that are both QIC-117 compatible *and* one of the QIC-40, 80, 3010, and 3020 standards should work. QIC-WIDE and Travan drives are also supported (TR-1 is just QIC-80 with 8mm tapes, while TR-2 and TR-3 is a.k.a QIC-3010 and 3020 respectively). Iomega Ditto 2GB and Ditto Max drives are supported, too, though they no longer conform to the QIC standards in every respect. Some parallel port tape drives are supported as well.

Some of the comments given below about possible problems with certain tape drives are very old, and I don't have access to all of the hardware, so I couldn't check everything.

Some of the reports below have been commented by me (<heine@math1.rwth-aachen.de>) like this:

> This is a comment.

Currently, the list of drives that are known to work with `ftape` is:

Alloy Retriever 250

Archive 5580i, XL9250i

Colorado DJ-10, DJ-20 (aka: Jumbo 120, Jumbo 250)

Colorado 1400

> <kosowsky@bellini.harvard.edu> reported a problem doing a 1G backup using taper.

Colorado Trakker parallel port tape drive

> Support added by Jochen Hoenicke <Jochen.Hoenicke@Informatik.Uni-Oldenburg.DE>.

HP Colorado T1000

> The problem reports are probably totally out-dated. In particular, the `zftape` the people talk about doesn't exist any more, and the `ftape` driver is the very `ftape-2.08`.

> Works with 3M Travan 400M (TR-1) tapes with 120M tapes. Also reported that mt dies, but with backups using tar it works ok. With cpio, ftape is recommended rather than zftape. (<millner@millner.bevc.blacksburg.va.us>)

> Problems have been reported with the drive continually stopping and starting with zftape (<75104.1756@compuserve.com>). This appears to be a problem with the tape going too fast for the computer; the DMA buffers are getting flushed before getting filled again. Newer versions of zftape don't do this any more is a suitably fast backup program or large DMA buffers are used (<millner@millner.bevc.blacksburg.va.us>).

Conner C250MQ(T)

TOC

The 250Q is reported to generate write error and frequent repositioning. (Frank Stuess at Naca Data Communications)

Write errors need not be caused by the tape drive, but also by bad tape cartridges. Frequent repositioning can be caused by bad cartridges, too, but can also be caused by overrun errors which would indicate that the FDC and DMA controller have problems to talk to each other.

Conner TSM420R, TSM850R

The 400 and 800 models only work with TR-1 tapes.

I don't know whether it was meant that named drives doesn't work with ordinary 120MB DC-2120 cartridges, or that TR-3 tapes can't be read. The tape drives weren't designed for the latter. So what.

Conner TST3200R

Works with TR-3 tapes at 1Mbps (ie. 1600M capacity only). Wirks with QIC-WIDE 400M tapes (Sony 5122's?) (<chris@cs.wmich.edu>). Works with TR3, QIC-3010, and QIC-3020 tapes. Comes with a 2MB FDC which the Promise 2300+ 1Mbps controller works (<kjh@pollux.usc.edu>).

Reported that the floppy disk can no longer read low-density floppies. May have to fiddle with IRQ/ports/dma channels (<chris@yakkocs.wmich.edu>).

The TST3200R works well with `ftape`.

Conner TST800R

The TST800R works with TR-1, Sony QW5122F (210M) and DC2120 tapes.

Works well with `ftape` since `ftape-2.07` at least. Used it myself until the drive died with a melted transistor. Probably caused by over-heating it previously.

Conner CTT3200

The CTT3200 is supposedly identical to the Iomega Ditto 3200. It works with the supplied 2Mbps controller, but reported not to work under DOS on some machines. (<jmorris@dtx.net>)

Conner 1.7G Tapestor (TSM1700R)

Works with QIC-WIDE tapes (<pschmidt@slip.net>). Partially works with QIS-3200. Using the HSC-2 controller, the DMA channel needs to be changed (incremented by 1, channel2?, Modify the Makefile). You then need to modify the ftape Makefile to reflect this change. However, ftape seems to be a bit flaky with this (no version number supplied) (<ttait@tiac.net>). It may not work at 2Mbps (QIC-3020) with the HSC controller. The tape died with a messages like "dumb tape stop" and has since been unreliable (<ttait@tiac.net>).

No recent informations available

Escom or Archive (Hornet) 31250Q

Exabyte EXB-1500

Work with QIC-3010 tapes.

Exabyte TR-3

TOC

Irwin 80SX, Insight 80Mb

Iomega 250

Iomega Ditto Tape Insider 420, 1700

Iomega Ditto Tape Insider 3200

This is the unit, that I use. The default jumper settings don't work. Leave the irq and ioport address at the default (6 and 0x370, respectfully), but change the DMA from 3 to 2. (Kevin Johnson <kjj@pobox.com>).

> Refer to the file `MCONFIG` of recent `ftape` distributions for other suggestions for ioport, irq and DMA channel.

May require the having `{0x08882, 80, wake_up_colorado, "Iomega 3200"},` added to `vendors.h` on older versions of `ftape`.

Problems reported with ftape 2.07 and kernel 1.12.13. With all sorts of combinations of accelerator, etc, the drive may (on some systems) only be accessed once (<erwin@box.nl>). Also, after the first access, the next use of the tape says it is write protected (<erwin@box.nl>, <M.J.Ammerlaan@dutiwy.twi.tudelft.nl>).

There has been one report of a problem where the tape got wound off the end of the spool.

> This may be caused by a dirty EOT sensor, and need not be a real hardware bug (except when it was a bug that dirtied the EOT sensor ...)

Another problem has been reported with writing archives (with dd) to the tape. It may start fine, but when the driver catches up with dd, it stops the tape and rewinds it to the beginning. Then it starts winding on through the tape ad infinitum. It appears to occur when the driver asks the tape to pause which should cause the tape to move back by 3 segments, but instead is moves back to the beginning of the tape. A bug fix submitted is reported to not solve the problem.

> Should have been fixed somewhere between `ftape-3.00` and `ftape-4.00`. Unluckily, the fast-skipping facilities of all Iomega floppy tape drives are **really** poor. Recent `ftape` versions work around this problem. I suggest getting the latest version of the `ftape` driver when you experience this problem.

Iomega Ditto 800 Insider

Works with Travan TR1, TR2, or DC2120 tapes (<klein@informatik.uni-rostock.de>).

Iomega Ditto 2GB

Support added by Jochen Hoenicke <Jochen.Hoenicke@Informatik.Uni-Oldenburg.DE> to `ftape-3.xx` and later.

Can't format cartridges, writing is only possible with special Ditto 2GB cartridges (hardware limitation, not a lacking feature of `ftape`).

Iomega Ditto Max

Iomega Ditto Max Pro

Supported since `ftape-4.00`. Thanks to Tim Jones <tjones@estinc.com>.

TOC

Can't format cartridges, writing is only possible with special Ditto Max cartridges (hardware limitation, not a lacking feature of `ftape`)

I wasn't able to get the Ditto Max to work with any other device than `/dev/[n]qft0`. I don't know whether this is a feature of the Ditto Max or the Ditto EZ controller I had plugged the Ditto Max into.

Comment

You don't need to buy a `Ditto Max Pro` to use the 5/10GB cartridges. With `ftape` there is no real difference between the `Ditto Max` and the `Ditto Max Pro`.

Iomega Ditto 800/3200/2GB/Max/Max Pro Easy (parallel port)

Supported since `ftape-4.00` with the `bpck-fdc` FDC driver.

Mountain FS8000

Reveal TB1400

Reported not to work with kernel 1.3.79 and ftape (no version given) or with kernel 1.2.13 and zftape 1.04 (<colin@colina.demon.co.uk>).

The mentioned `ftape` driver versions are out of date. If you still have such a beast try the more recent versions of the `ftape` driver.

Summit SE 150, SE 250

Tallgrass FS300

If you have a Tallgrass FS300 and an AHA1542B, you need to increase the bus-on / bus-off time of the 1542B. Antti Virjo (`<klanvi@uta.fi>`), says that changing `CMD_BUSON_TIME` to `4` and `CMD_BUSOFF_CMD` to 12 in `linux/drivers/scsi/aha1542.c` will do the trick.

Teac 800

Memorex tape drive backup system

Wangtek 3040F, 3080F

You can always check out the newest list of drives that are recognised by `ftape`, by looking in the file `vendors.h` in the `ftape` distribution.

Although I do not want to endorse one drive type over another, it has been reported that the Colorado DJ-20 drive is rather noisy, when compared to, say, a Conner C250MQ drive ('tis said that the Colorado is 5-10 times as noisy as the Conner drive. Since I have neither, I can't tell for sure).

If you have a drive that works fine, but it is not listed here, or if you have corrections to the above information, please send a mail to the HOWTO maintainer (`<heine@math1.rwth-aachen.de>`).

6.2 Supported special controllers

TOC

These dedicated high-speed tape controllers are supported by `ftape`:

- Colorado FC-10, FC-20
- Mountain MACH-2
- Iomega Tape Accelerator II
- 2Mbps controllers (using the i82078-1 fdc)
- Iomega Ditto EZ 4Mbps PnP controller

Colorado FC-10, FC-20

Support for the FC-10 controller has been merged into the `ftape` driver in version 1.12. See the `RELEASE-NOTES` and the `Makefile` files in the `ftape` distribution. Since of version 2.03 of `ftape`, the FC-20 controller will work, but only at 1Mbit/sec (check the Release notes!).

Mountain MACH-2

The support for the MACH-2 controller was added in `ftape-1.14d`.

Iomega Tape Accelerator II

To use the Iomega Tape Accelerator II (**not** to be mistaken as the Iomega Ditto Dash!), use `-DMACH2`, and set the right settings for I/O base, IRQ and DMA. This works (by the empirical testing of Scott Bailey <sbailey@xcc.mc.xerox.com>), with at least `ftape-2.02`.

Iomega Ditto Dash and other 2Mbps controllers

The Iomega Ditto Dash, and all other known 2Mbps controllers, use the Intel 82078-1 chip, which can run at 2Mbps. This is supported properly since `ftape-3.00`.

Iomega Ditto EZ PnP controller

This controller requires the use of e.g. the `isapnptools` package to configure it. You may get it from

http://www.roestock.demon.co.uk/isapnptools/

The controller will cause too many overrun errors when used at the highest possible speed of 4Mbps. Neither Tim Jones <tjones@estinc.com> nor I <heine@math1.rwth-aachen.de> have been able to find but a single system which could run the controller at 4Mbps. 3Mbps seems to be fine.

If you configure the Ditto EZ to use DMA 2 (the DMA channel used by the floppy controller) then your floppy drive will no longer work. It doesn't help to disable the controllers DMA gate (as is the case with other hight speed controllers) so this can't be helped from inside `ftape`.

TOC

6.3 Unsupported tape drives

- Some parallel port floppy tape drives still not work. Others do.
- Irwin AX250L / Accutrak 250. (not a QIC-80 drive)
- IBM Internal Tape Backup Unit (identical to the Irwin AX250L drive)
- COREtape light

The Irwin AX250L (and the IBM Internal Tape Backup Unit) does not work the `ftape`. This is because they only support QIC-117, but not the QIC-80 standard (they use Irwin's proprietary ``servoe (Rhomat)'' format). I know nothing about the Rhomat format, nor where to get any info on it. Sorry.

The COREtape light does not accept the initialisation commands, we're feeding it. This pretty much leaves the drive unusable.

6.4 Using an external tape drive with `ftape`

If you have a floppy controller which has a female DB37 connector on the bracket (and some means of delivering power to the drive), you can use it with `ftape`. OK, that sentence was not very obvious. Let's try it this way: Some FDC's (the very ancient one's), have a DB37 connector on the bracket, for connecting to external floppy drives.

If you make a suitable cable from the DB37 connector (on the FDC) to your external tape drive, you can get `ftape` to control your tape drive.

This is because that from a program's view there is no difference between the internal and the external connectors. So, from `ftape`'s point of view, they are identical.

- Pins 20-37: GROUND
- 1: +12 Volt (POWER)
- 2: +12 Volt return (GROUND)
- 3: +5 Volt return (GROUND)
- 4: +5 Volt (POWER)
- 5: 2
- 6: 8
- 7: 10
- 8: 12
- 9: 14
- 10: 16
- 11: 18
- 12: 20
- 13: 22
- 14: 24

TOC

- 15: 26
- 16: 28
- 17: 30
- 18: 32
- 19: 34

The power connector is of the "mini" type, sitting on 3.5" floppy drives. The idea appears to be that you plug one of the power connectors from the PSU to this connector on the board. If you want to use just a single cable, you might want to get a 50 wire cable, and use multiple wires for the power lines (and ground, for that matter).

I have received no confirmation from anyone that this works. Let me know your results if you try it.

6.5 PCI motherboards and `ftape`

Unfortunately, some PCI motherboards cause problems when running `ftape`. Some people have experienced that `ftape` would not run in a PCI based box, but ran flawlessly in a normal ISA based 386DX machine. If you have such a problem, please read the `README.PCI` file in the `ftape` distribution.

> A floppy disk controller needs the ISA bus DMA controller for its memory transfers. Seemingly the ISA DMA controller doesn't get control over the memory bus often enough on some PCI based systems.

---

7. Backing up and restoring data

This section describes some simple uses of `tar` and `mt`. Other examples can be found in the `ftape-manual` of the `ftape-doc` package. The `ftape-tools` contains some simple automated `DejaGnu`

Package for writing automated tests.

test-suites. See section Getting ftape for informations about where to download those additional packages from.

7.1 Writing an archive to a tape

You can use `tar', `dd', `cpio', and `afio'. You will need to use `mt' to get the full potential of your tapes and the `ftape` driver. For a start I'd recommend using `tar', as it can archive lots of directories and let you pick out separate files from an archive. `cpio` creates smaller archives and is more generally more flexible than `tar`, but is missing some features like volume labels. `afio' creates backups where each file is compressed individually and then concatenated. This will allow you to access the files ``after'' the point of the error. If you use `gzip`ped `tar` files, all data after the point of the error is lost! (to me, this is a pretty good reason for NOT using compression on backups). The choice of which is most appropriate depends on the situation and the features and malfeatures of each of the packages. I recommend taking a look at each package at reviewing the options that each provides. It's possible that this HOWTO may provide more detail on this subject at some point in the future.

There are more links pointing to backup software at http://www-math.math.rwth-aachen.de/~LBFM/claus/ftape/ in the `software` section of that page.

To make a backup of your kernel source tree using `tar`, do this (assuming you have the sources in `/usr/src/linux`):

```
# cd /usr/src
# tar cf /dev/ftape linux
```

This will not compress the files, but gives you a smoother tape run. If you want the compression (and you've got `tar` 1.11.2), you just include the `-z` flag(*), eg: `tar czf /dev/ftape linux'

For further instructions on how to use `tar`, `dd` and `mt` look at the man pages and the texinfo files that comes with the respective distributions.

(*) `tar` assumes that the first argument is options, so the `-' is not necessary, i.e. these two commands are the same: `tar xzf /dev/ftape' and `tar -xzf /dev/ftape'

TOC

7.2 Restoring an archive

OK, let us restore the backup of the kernel source you made in section Writing an archive to a tape above. To do this you simply say

```
tar xf /dev/ftape
```

If you used compression, you will have to say

```
tar xzf /dev/ftape
```

When you use compression, `gzip` will complain about trailing garbage after the very end of the archive (and this will lead to a `broken pipe' message). This can be safely ignored.

For the other utilities, please read the man page.

7.3 Testing the archive

tar has an option (`-d`) for detecting differences between two archives. To test your backup of the kernel source say

```
tar df /dev/ftape
```

If you do not have the man page for `tar`, you are not lost (yet); tar has a built-in option list: try `` `tar --help 2>&1 | less' ``

7.4 Putting more than one backup on a tape

To put more than one backup on a tape you must have the `mt` utility. You will probably have it already, if you got one of the mainline distributions (eg. Slackware or Debian).

Programs like `tar` and `cpio` generate a single Tape ARchive and know nothing about multiple files or positioning of a tape, it just reads or writes from/to a device. `mt` knows everything about moving the tape back and forth, but nothing about reading the data off the tape. As you might have guessed, combining `tar` or `cpio` with `mt` does the trick.

By using the `nqft[0-3]` (`nftape`) device, you can use `` `mt' `` to position the tape the correct place (`` `mt -f /dev/nqft0 fsf 2' `` means step over two ``file marks'', i.e. `tar` files) and then use `tar` or `cpio` to read or write the relevant data.

The most common use of the non-rewinding device is to append another backup to an existing tape. Here are the specific steps with a little explanation thrown in for good measure.

- Insert a tape into the drive. On some devices this may cause the tape to be rewound.
- Issue an End-of-Tape command to the NON-rewinding device.

```
mt -f /dev/n???? eof
```

The tape should now be positioned at the End-of-Data (**EOD**). The tape won't move unless a program opens the device, closes the rewinding device, removes the device driver from kernel memory (rmmod) or ejects the tape. Using `mt eof' may be faster on QIC tapes.

- The next tape operation will start at the EOD mark. If you perform a write, it will append a new `file'. If you perform a read it will fail with EOF. The EOD mark on most tape formats is actually two consecutive EOF marks, however, since version 3.xx `ftape` uses the volume table as specified in the `QIC-113` standard to emulate file marks, thus there aren't two consecutive file marks at EOD. Writing the EOF marks is handled by either the device driver or the hardware when a close() is performed.
- Here's where you write the actual data to the tape.
- Here's the important part. **Now rewind the tape**. Both `ftape` caches some information that belongs in the header segments on the tape and update those header segments **only when the tape is rewound**. This caching is necessary because rewinding the tape and updating the header segments takes a conspicuous amount of time. The drawback of this caching is that you will lose information if you have written to the tape and not rewound the device.

7.5 Appending files to an archive

``Is there a way to extend an archive -- put a file on the tape, then later, add more to the tape?"

No. The `tar` documentation will tell you to use `tar -Ar', but it does not work. This is a limitation of the current `ftape` driver.

7.6 Mount/unmounting tapes

Since a tape does not have a ``filesystem" on it, you do not mount / unmount the tape. To backup, you just insert the tape and run your `tar' command (or whatever you use to access the tape with).

---

8. Creating an emergency boot floppy for `ftape`

Comment

As of the time of this writing (August 1998) I remember that I have read about several emergency disk sets in the **c.o.l.a** (`comp.os.linux.announce`) news group since the time this section has been written. Some of those packages actually might produce rather sophisticated emergency boot floppy sets. Please check out yourself. I didn't try to create an emergency boot floppy with recent versions of `ftape`.

This section was written by Claus Tøndering `<ct@login.dknet.dk>`.

Once you are the happy owner of a tape drive and several tapes full of backups, you will probably ask yourself this question: ``If everything goes wrong, and I completely lose my hard disk, how do I restore my files from tape?"

What you need is an emergency floppy disk that contains enough files to enable you to boot Linux and restore your hard disk from tape.

The first thing you should do is to read ``The Linux Bootdisk HOWTO" written by Graham Chapman <grahamc@zeta.org.au>. That document tells you almost everything you need to know about making an emergency floppy boot kit. The paragraphs below contain a few extra pieces of information that will make your life a bit easier when you follow Graham Chapman's procedures:

- You don't really need `/etc/init`, `/etc/inittab`, `/etc/getty`, and `/etc/rc.d/*` on your floppy disk. If Linux doesn't find `/etc/init`, it will start `/bin/sh` on your console, which is fine for restoring your system. Deleting these files gives you extra space on your floppy, which you will probably need.
- Find a small version of `/bin/sh`. They are frequently available on the boot floppies that come with a Linux distribution. This again will give you extra space. I'd suggest `ash`, which is extremely small (approx 62Kbytes), and yet very `bash` compatible.
- The `/etc/fstab` you include on your floppy disk should look something like this:

  ```
  /dev/fd0        /               minix   defaults
  none            /proc           proc    defaults
  /dev/hda        /mnt            ext2    defaults
  ```

  Once you have booted from your floppy, give the command:

  ```
  mount -av
  ```

- Make sure your floppy drive is not mounted when you access the streamer tape! Otherwise you may get the following error message:

TOC

Unable to grab IRQ6 for ftape driver

This means that you **MUST** load the floppy into a RAMDISK. This has the unfortunate consequence that the programs needed to restore the files from the tape can not be located on a separate floppy disk. You have two options here:

1. You place `tar` (or `cpio` or `afio` or whatever other backup program you use) on your root floppy disk. (This is where you'll need all the extra space created in the steps above.)
2. Before you start restoring from tape, copy `tar` (or `cpio` or `afio` or whatever) to your hard disk and load it from there.

- Apart from your backup program, you will probably need `mt` on your root floppy as well.
- Make sure your ftape device (typically `/dev/nqft0`) is present on your boot floppy.
- Finally: **TRY IT OUT!** Of course, I don't recommend that you destroy your hard disk contents to see if you are able to restore everything. What I do recommend, however, is that you try booting from your emergency disks and make sure that you can at least make a file listing of the contents of your backup tape.

---

9. Frequently Asked Questions

Comment

This is the literal inclusion of the Ftape Frequently Asked questions collection which is maintained by Johan De Wit <jo@correct.nl> and which may be viewed on the web at http://www.correct.nl/~ftape. As Linuxdoc SGML doesn't include sub-sub-sections into the table of contents, I have prepended the word `FAQ` to the sections of the original FAQ document.

This FAQ collection might be slightly out of data as it was collected while version 3.04d of the `ftape` driver was the newest one. If any answer given in the FAQ contradicts any other statement of this HOWTO, then please disregard the answer in the FAQ and drop me (<heine@math1.rwth-aachen.de>) as well as the maintainer of the `Ftape-FAQ` (Johan De Wit <jo@correct.nl>) a note

You might encounter references to the following addresses while reading this document:

- The maintainer of the Ftape FAQ : Johan De Wit <jo@correct.nl>
- The Ftape maintainer : Claus-Justus Heine <claus@momo.math.rwth-aachen.de>
- The Ftape Home Page : <http://www-math.math.rwth-aachen.de/~LBFM/claus/ftape/>
- Mirrors of the Ftape Home Page :

  <http://www.torque.net/ftape/>

  Thanks to Grant R. Guenther <grant@torque.net>

  <http://www.info-systems.com/ftape/>

  Thanks to Jakob Curdes <jc@info-systems.com>

  <http://www.newwave.net/~joshg/ftape/>

  Thanks to Josh Goins <joshg@newwave.net>
- The Ftape HOWTO : <http://sunsite.unc.edu/LDP/HOWTO>
- The Ftape Mailing List : <linux-tape@vger.rutgers.edu>

There is surely quite a lot missing. Please feel free to improve this FAQ. The preferred way of doing this is to post to the Ftape Mailing List in case you have a question that isn't answered here.

Also, if you are already reading the list regularly and have the impression that some questions occur again and again, feel free to send that question and possibly an answer in the format indicated below to the maintainer of the *Ftape FAQ* AND to Ftape Mailing List.

If you make FAQ related postings, then please DON'T FORGET to prepend the word "**[FAQ]**" to the

TOC

subject of your posting. Please don't add the word "FAQ" to the subject if you post something that is related to the FAQ.

That's all for now.

Claus-Justus Heine.

---

10. FAQ: "Compiling and installing Ftape" related questions !

10.1 What Ftape version should I use?

Always the latest stable version which is _supposed_ to be available from ftp://sunsite.unc.edu/pub/Linux/kernel/tapes and http://www-math.math.rwth-aachen.de/~LBFM/claus/ftape/

At time of this writing the latest stable version is `ftape-4.02`.

<answer from Claus Heine>

10.2 I'm having problems getting my XYZ drive to run under the 2.0.xx kernel with the built-in driver. How do I fix this?

The default version of *Ftape* included in the 2.0.xx kernel sources is 2.08 or 2.09 and is very out of date. Please update the *Ftape drivers* to the latest from the Ftape Home Page.

<answer from Tim Jones>

10.3 I'm running Linux/SMP and the system just freezes when trying to access the Ftape devices!

You need to add `-D__SMP__` to the `KERNEL_OPT` variable in the file `MCONFIG`. In newer `ftape` versions you only need to uncomment a certain line in the `MCONFIG` file.

<answer from Claus Heine>

10.4 Why does depmod complain about "undefined symbols"?

Ignore the depmod error messages. The problem is that the *Ftape modules* have to be compiled without the version checksum feature (i.e. `CONFIG_MODVERSIONS`) with 2.0.* kernels. This causes no problem, even when the modules are used with a kernel that has support for this feature; only that `depmod` wrongly complains about undefined symbols. Ignore the complaints of `depmod` and try to

insert the modules despite of these complaints:

```
modprobe zftape
```

If this fails, something is wrong.

<answer from Claus Heine>

10.5 "insmod" says the kernel version is wrong

The `insmod` program can check the kernel version against the version that *Ftape* was compiled for in two ways: It can directly compare the kernel version number recorded in the *Ftape module* against the version of the running kernel, or, if both the kernel and *Ftape* is compiled with versioned symbols, compare the version of the used kernel symbols.

If you have upgraded your version of GCC to v2.7.0 or later, you must recompile the modules utilities with gcc v2.7.x.

Newer versions of `insmod` allows you to "force" insertion of a module into the kernel, even though the version string is incorrect.

<from the Ftape-Howto>

10.6 "insmod" says that kernel 1.2.0 and 1.2.0 differ

Did you remember to apply the `ksyms.c` patch to the kernel? If not, read the `README.linux-1.2` file in the source distribution.

<from the Ftape-Howto>

10.7 Trying to compile Ftape gives me the error "modversions.h: no such file or directory"

The `modversions.h` file is created when the kernel is compiled with the configuration item `CONFIG_MODVERSIONS` turned on. With this option enabled, the file will be created during the `make dep` step.

One more handy tip is that a `make mrproper` will remove `/usr/include/linux/modversions.h`. You will need to reconfig the kernel and do a `make dep` to get the file back.

<from the Ftape-Howto>

10.8 What is this versioned symbols stuff anyway?

TOC

When you say `yes' to `CONFIG_MODVERSIONS` during `make config', all the symbols exported by the kernel, i.e: the symbols that the loadable modules can "see", are augmented to include a checksum across the types of the call/return parameters. This allows `insmod` to detect whether the definition of a variable or function in the kernel has changed since the time when *Ftape* was compiled.

This ensures a high degree of safety, such that you do not crash the kernel because you used an outdated module with your kernel.

If you enable `CONFIG_MODVERSIONS` in the kernel, make sure you have

```
-DMODVERSIONS -include /usr/include/linux/modversions.h
```

uncommented in the MODULE_OPT line in the *Ftape* Makefile. Conversely, if you do not have CONFIG_MODVERSIONS enabled, make sure you have it commented out.

<from the Ftape-Howto>

10.9 I seem to be getting sftape instead of zftape. When I run "ftmt status" command, I get output that the Ftape docs says corresponds to sftape (/dev/qft0: Invalid argument). Why?

There are (at least) two possible sources of the problem:

- All Ftape-3.* versions prior to 3.04 install the modules into

  ```
  /lib/modules/misc
  instead of
  /lib/modules/uname -r/misc
  ```

  As `modprobe` searches in `/lib/modules/misc/` last there might be an old `ftape.o` module floating around in `/lib/modules/` *uname* *-r*`/misc` which `modprobe` finds first (`uname -r` stands for the kernel version). Remove the old `ftape.o` module.
- Your kernel has support for *Ftape* compiled in. Reconfigure your kernel without support for *Ftape* (`CONFIG_FTAPE`) and recompile and install it.

<answer from Claus Heins>

10.10 My Ditto DASH/FC-20/Exabyte Accelerator card works under Microsoft Windows, but I get a drive not found type of error in /var/log/messages

when trying to use it under Linux.

TOC

You are probably trying to use the same IRQ and DMA settings as your on-board FDC. This does not work in versions of *Ftape* prior to 3.03b. Please update the *Ftape Drivers* to the latest from the Ftape Home Page.

<answer from Tim Jones>

10.11 Ftape DMA transfers gives ECC errors

Sadly to say there are some SVGA cards and Ethernet cards that do not decode their addresses correct. This typically happens when the *Ftape* buffers are in the range `0x1a0000` to `0x1c0000`. Somehow, the DMA write cycles get clobbered and every other byte written gets a bad value (`0xff`). These problems are reported to happen with both SVGA and Ethernet cards. We know of at least one (bad?) ATI 16bit VGA card that caused this.

The easiest solution is to put the card in an 8bit slot (it is often not enough to reconfigure the card to 8bit transfers). Moving the *Ftape* buffer away from the VGA range is only a partial solution; All DMA buffers used in Linux can have this problem! Let us make this one clear: This has nothing to do with the *Ftape* software.

<from the Ftape-Howto>

10.12 Help! I'm getting 'dmaalloc() failed' in my syslog file.

You should only see this is you are trying to `insmod` the `ftape.o` module. Try running `swapout` first. It is provided with the standalone *Ftape* source. It doesn't appear in the *Ftape* source that's provided with the kernel.

Here's an example of how you can set your rc.local file to use it.

```
# Install the Floppy Tape Driver
if [ -f /boot/modules/`uname -r`/misc/ftape.o ]; then
   echo Installing ftape for Linux `uname -r`
   swapout
   insmod /boot/modules/`uname -r`/misc/ftape.o
fi
```

Please note that you won't have this type of problem if you compile the *Ftape driver* into the kernel.

<from the Ftape-Howto>

10.13 Syslogd works overtime when running Ftap

TOC

The compile-time options `NO_TRACE` and `NO_TRACE_AT_ALL` in *Ftape* control the amount of system logging. Add whichever is appropriate to the `FTAPE_OPT` line in the Makefile and recompile.

<from the Ftape-Howto>

10.14 How do I change the trace-level?

There are three ways you can do this (in order of personal preference).

While we're at it, here are the meanings of the various trace levels.

- 0 Bugs
- 1 + Errors
- 2 + Warnings
- 3 + Information
- 4 + More information
- 5 + Program flow
- 6 + FDC/DMA info
- 7 + Data flow
- 8 + Everything else

1. Using insmod to change trace-level If you are using the modules mechanism to load the *Ftape* driver, you can specify the tracing level as an option to the insmod command.

   ```
   /sbin/insmod ftape.o tracing=<tracing-level>
   ```

2. Using mt to change trace-level The *Ftape* driver has a hack in it that allows the `fsr` option in `mt` to be used to set the tracing level. `zftape` does not have this hack.

   ```
   mt -f /dev/ftape fsr <tracing-level>
   ```

   The use of the `fsr` command in `mt` is a *hack*, and will probably disappear or change with time.

3. Recompiling to change trace-level The file `tracing.c` contains a line `int tracing = 3;`. Change the 3 to whatever is appropriate and recompile.

<From the Ftape-Howto>

10.15 I'm having problems with Ftape. I'm using the latest version of Ftape from the Ftape Home Page and believe that I've located a real bug. What should I do?

Check the Ftape Home Page. for an even newer version. Then check the FAQ contained in the that package if your problem is listed there. Next, try to check if the manual that comes with the *Ftape distribution* mentions your problem.

There is no need to read the entire manual, simply check the "Concept Index" if it contains a keyword that might be related to your problem, then jump to the proper location in the manual.

If you are still convinced you've found a bug, then post a general question describing the problem to the Linux-Tape Mailing List , but do not attach your entire *Ftape* error-log. If we've seen the problem before, we'll let you know where the resolution effort stands. If we haven't, the Ftape maintainer will most likely request that you send him the entire *Ftape* error-log (snipped from your system messages file).

<answer from Tim Jones>

---

11. FAQ: "Using Ftape" related questions !

11.1 How fast is Ftape ?

You can achieve quite respectable backup and restore speeds with *Ftape*: a Colorado DJ-20 and an Adaptec 1542CF controller, has been measured at 4.25Mbyte/min sustained data transfer rate (no compression) across a 70Mbyte tar archive, while comparing the archive on the tape with data on an IDE disk. The speed of *Ftape* is mostly dependent on the data transfer rate of your FDC: The AHA1542CF has a ``post-1991 82077'' FDC, and it will push 1Mbit/sec at the tape drive. If you have an FDC which can only deliver 500Kbit/sec data rates, you will see half the transfer rate (well, roughly).

11.2 When I write to some of my tapes, they seem to spend a lot of time "shoe-shining," or repositioning instead of streaming. Is something wrong with my system?

There has been a few reports of "shoeshining". This is when the tape just seems to run back and forth endlessly. This has been seen on a Jumbo 250 (74407.3051@compuserve.com) and on an Iomega 250 Ditto Insider (tom@opus.cais.com). In the latter case it has been narrowed own to using an ELF Linux and running off a SCSI hard disk (connected to an Adaptec 1542cf). Please contact me if you have an update to this problem.

<from the Ftape-Howto>

Probably not. If you are backing up a large number of < 2K files, you're just going to have to live with it. In this event, the repositions are caused by file system access overhead. If you are backing up a normal system's files, this may be caused by slop or media stretching in the tape cartridge. By simply retensioning the tape, you should see this go away. Try

```
ftmt -f /dev/zqft0 reten
```

to retension the tape. If retensioning doesn't solve this, and it's only happening on certain tapes, it might be wise to replace the tapes in question.

<answer from Tim Jones>

If you use afio as your backup tool you can set it to write a very large number of buffers in one hit by using the -c flag. Make it large enough so that you supply enough data for most of a single end-to-end pass over the tape. For my system, the following streams quite nicely - stopping relatively few times per tape pass on an unloaded system:

```
find /usr/local -xdev -print | afio -o -v -f -b 10240 -c 800 /dev/qft0
```

In my case I'm writing 800 x 10240 bytes per tape write, i.e. about 8MB. haven't experimented that much with these settings - so someone might like to establish more optimal ones.

Presumably other backup utilities could be modified to use a similar technique.

<answer by Michael Hamilton>

GNU tar doesn't use buffering in this way. The commercial backup program "bru" is able to multi-buffer using shared memory; this works only when writing compressed archive with bru (regardless whether you use *Ftape's* builtin compression)

TOC

Another way to overcome the problem might be to use more dma buffers in the *Ftape kernel driver* like:

```
mt -f /dev/qft0 setdrvbuffer $((6&ast;32786))
```

$((6*32786)) should be expanded by your shell when using a Bourne compatible one. This has a negative impact on the system's memory pool: *Ftape's* dma buffers cannot be used by any other part of the kernel nor by any other application. And kernel memory cannot be swapped out. If you decide to use this kind of multi-buffering then you should unload the driver as soon as it isn't needed any longer.

<answer by Claus Heine>

11.3 Do I have to reboot to the DOS world to format tapes?

Not if you are using the latest version of the *Ftape drivers* from the Ftape Home Page.

To format a QIC-80, TR-1, TR-3, QICWide 3010 or 3020 tape, get the latest version of `ftape` and the latest version of the `ftape-tools` package (from the same location) and read the documentation of the `ftformat` utility which is included in the `ftape-tools` package.

Comment

Do not try to format Ditto 2GB tapes.

Comment

Do not try to format Ditto Max or Max Pro tapes.

<answers from Tim Jones and Claus Heine>

11.4 Is it possibly to format Ditto 2GB tapes with ftape?

It isn't possible to format `Ditto 2GB` tapes with `Ditto 2GB` tape drive, and it isn't possible at all to re-format `Ditto 2GB` tapes in a way that they still can be used by a `Ditto 2GB` tape drive.

This is a hardware limitation of the `Ditto 2GB` tape drive. It can't be helped at the software level, i.e. it isn't `ftape's` fault.

11.5 Is it possibly to format Ditto Max or Max Pro tapes with ftape?

No, the `Ditto Max` can't format tapes.

This is a hardware limitation of the `Ditto Max (Pro)` tape drive. It can't be helped at the software level, i.e. it isn't `ftape's` fault.

11.6 Ftape detects more bad sectors than DOS on QIC-3020 tapes

If you look at the difference, you will notice that *Ftape* always detects 2784 sectors more than DOS.

The number that *Ftape* reports is correct (of course `:-)`. Each correctly formatted QIC-3020 tape has 2784 sectors at

TOC

fixed positions that are marked in the bad sector map. To quote from the specs:

> Tracks 5,7,9,11,13,15,17,19,21,23,25 and 27 within 4 segments of either EOT or BOT are prone to increased error rates due to hole imprints. Therefore, these regions shall be mapped as bad at format time and entered in the bad sector map by indicating that all sectors within the identified segments are bad.

This gives 12 tracks * 2 * 4 segments * 29 sectors == 2784 sectors.

So *Ftape* choose to report the real number of sectors that cannot be used on the tape, while DOS gives a more optimistic number giving a better indication of tape quality. (*Ftape's* behavior might change in the future to detect correct formatting and display the separate numbers. It has rather low priority though).

QIC-3010 are alike QIC-3020 tapes regarding this.

<from the Ftape-Howto>

11.7 Is it ok that I'm not hearing the tape move when I do a fsf or a bsf with mt?

Yes. The driver merely updates an internal counter when those commands are issues. The tape should move to the proper location on the next read or write access to the tape drive.

<from the Ftape-Howto>

11.8 Why does my XYZ backup program complain about "Invalid argument" errors?

`zftape` requires the data to be written in multiples of a fixed minimal block size. This is a very usual behavior for a tape device. There are three ways to get rid of those errors:

- set *Ftape's* block size to the block size used by the backup program. The example below works for "afio":

  ```
  mt -f /dev/qft0 setblk 5120
  ```

- If you don't want to use *Ftape's* built in compression you can also use

  ```
  mt -f /dev/qft0 setblk 0
  ```

  to switch *Ftape* to variable block size mode and be able to write the data in arbitrary portions to the tape (BUT: the builtin compression doesn't work with this setting). When you intend to use "KBackup" then this is the only way to make it work together with *Ftape* (it _may_ work, don't know if it does)

- tell your backup program about *Ftape's* default block size of 10k (which is also the default of GNU tar). For "afio" you can use the following command line switch:

  ```
  afio -b 10k ...
  ```

You may want to read the section "Tape blocks" of the manual (use its "Concept index" to directly jump to that section)

When using GNU tar's builtin compression with GNU tar versions prior to tar-1.12 one needs to run tar with the `--block-compress` switch to `re-block` the output to the tape. Otherwise tar will compress the data it reads, and write it in arbitrary portions to the tape.

```
Example :
```

```
tar -czvf /dev/qft0 --block-compress /etc
```

WARNING: One shouldn't use tar's builtin compression with large backups as it makes the entire data stream one huge compressed block. If such archives are corrupted right at the beginning it will be very difficult to recover.

<answer by Claus Heine>

11.9 I/O errors and FDC - some explanations.

When you get next messages, this could be interesting for you !

- fdc-io.c (ft_handle_perpend) - Your FDC does not support QIC-3020.
- Cannot write to /dev/qft0: I/O error

The explanations:

"FDC" menas "Floppy Disk Controller". The problem is that your floppy disk controller must be able to support something that is called "perpendicular mode" to be able to read and write QIC-3020/QIC-3010 cartridges (i.e. TR-3 cartridges). To my knowledge all FDCs that are capable of at least 1Mbit/sec data transfer rate also support "perpendicular mode" ("perpendicular" refers to the direction of magnetization of the ferro-magnetic particles on the tape).

This means that you need to purchase another FDC. Either look around some computer stores and ask for an IO controller cards that is able to support 2.88 Mb floppies (which imlies 1Mbit data transfer rate and perpendicular mode).

Or get one of the so called "high speed" controllers that even support 2Mbit/sec data transfer rate. Those controllers are based on an Intel 82078 FDC. Iomega sells such a card under the name "Ditto Dash". I think Exabyte sells their 2Mbit controllers separately, too, whereas Seagate ships its TR-3 drives (i.e. the TST-3200) together with such a controller.

<answer from Claus Heine>

11.10 Why do I get "/dev/qft0: No such device" errors?

I assume that the following is the problem: The *Ftape* module is loaded OK into the kernel:

```
/usr/src/ftape-3.03b-970603# lsmod
 Module          Pages    Used by
 ftape              22            0
```

but then this happens:

```
$ ftmt -f /dev/qft0 status
ftmt: /dev/qft0: No such device
```

Solution You need to load the *zftape.o* module as well. With Ftape-3.* the *ftape.o* module doesn't implement the VFS interface. This is done by *zftape.o*.

<answer from Claus Heine>

11.11 I get "device busy" when I make multiple backups on a tape using some script.

The "device busy" messages can only occur while the *Ftape devices* are still held open by some program. As soon as the close() system call has completed the busy flag is cleared. May be "bru" or some other program has still forked off a

TOC

child that dies delayed?

Yes, this will reproduce the problem, it seems:

```
tar -cvvzf /dev/nqft0 --block-compress ; mt rewind
```

You can skip the "--block-compress" if using the most recent version of GNU tar.

However, this is not a bug of *Ftape*. It seems that the parent tar process exits before its child has closed the tape device. I know, however, from hacking the tar code ages ago, that tar properly waits for its parent to die.

However, the busy message simply means that the "busy" variable is still held at 1 (zftape/zftape-init.c). And this simply means that there still is a process hanging around that holds the tape device open.

I think I have it (only for the case of tar 'cause I have the source code.

If on uses tar with compression, then it forks a child which will become the compressor bei execing "gzip" or whatever. Before the call to execlp() the child will fork off a grand child of its parent tar. That grandchild will do the actual tape I/O.

```
tar - fork() - write to child tar
        |
      child tar - fork() - gzip (will pipe to grand child tar)
                     |
                   grand child tar - open archive.
```

Now, parent tar only waits for its child to die. gzip surely doesn't wait for the grand child as the gzip is a result of an execlp().

What I don't know is whether the grand child should be implicitly waited for by the parent tar, or if the wait() function also waits for grand childs.

But this seems to be the problem: the parent tar already has exited while its grandchild still is busy closing the archive. One hardly will notice this problem if the close() happens fast (i.e. regular files, block devices, also other tape devices?), but it isn't a bug in *Ftape*, but either in the backup programs or in the kernel or maybe libc exit code.

Don't know if the considerations above also apply to bru. If there is no grandchild and the parent process properly waits for its childs then there shouldn't be a problem.

<answer from Claus Heine>

11.12 How do I "..." with tar?

These are really `tar` questions: Please read the `man` page and the `info` page. If you have not got it either, try

```
tar  --help 2>&1 | less
```

If your version of `tar` is v1.11.1 or earlier, consider upgrading to v1.11.8 - This version can call `GNU zip` directly (i.e.: it supports the `-z` option) and has an elaborate help included. Also, it compiles right out of the box on Linux.

<from the Ftape-Howto>

11.13 What block-size should I use with tar ?

When using compression, and in all general, it can be a benefit to specify to `tar`, that it should block the output into chunks. Since *Ftape* cuts things into 29Kbyte blocks, saying `-b58' should be optimum.

"Why 29Kbyte?", I hear you cry. Well, the QIC-80 standard specifies that all data should be protected by an Error Correcting Code (ECC) code. The code specified in the QIC-80 standard is known as a Reed-Solomon (R-S) code. The R-S code takes 29 data bytes and generates 3 parity bytes. To increase the performance of the ECC code, the parity bytes are generated across 29 1Kbyte sectors. Thus, *Ftape* takes 29Kbytes of data, adds 3Kbytes of ECC parity, and writes 32Kbytes to the tape at a time. For this reason, *Ftape* will always read and write 32K byte blocks to be able to detect (and correct) data errors.

If you are curious, and wish to know more, look in the `ecc.c` and `ecc.h` files, for an explanation of the code and a reference to a textbook on Reed-Solomon codes.

<from the Ftape-Howto>

11.14 Where can I find the tar/mt/cpio/dd binaries - sources - manpages?

All of these tools have been developed by the GNU project, and the source (and man page) can be fetched from just-about any ftp site in the world (including `ftp.funet.fi`, `tsx-11.mit.edu`, and `sunsite.unc.edu`). In any case they can be fetched from the official GNU home site: `prep.ai.mit.edu [18.71.0.38]:/pub/gnu`. The latest versions (as of September 12 1996) are:

```
cpio:   2.4.2 (cpio-2.4.2.tar.gz)
dd:     3.13 (fileutils-3.13.tar.gz)
mt:     2.4.2 (cpio-2.4.2.tar.gz)
tar:    1.11.8 (tar-1.11.8.tar.gz)
gzip:   1.2.4 (gzip-1.2.4.tar.gz)
```

They all compile out of the box on Linux `v1.0.4` / `libc v4.5.19` / `gcc v2.5.8`.

<from the Ftape-Howto>

11.15 If I use tapers compression, is it a bad idea to use the compression with zftape, or would it be better to not use tapers compression, and let zftape do it?

It is not bad as such to compress data twice (which would be the case when using tapers compression together with *zftape's* compression) but it doesn't make any sense. You won't gain much further compression, but only waste CPU cycles.

Tapers compression should be quite safe, as taper compresses single files; in contrast to *tar -czf* ... which makes the entire data stream a large compressed block of data, which is really a bad thing with serious backups as a single bad byte at the beginning of the archive can make the entire archive unusable, well, it will be at least quite difficult to recover.

<Answer from Claus Heine>

11.16 How does zftape compression compare to say gzip -9?

TOC

gzip -9 is better (i.e. one gains higher compression). *zftape's* compression is comparable with the Un*x *compress* program, but should be faster, and is faster than *gzip*.

<Answer from Claus Heine>

11.17 I don't trust compression, but hear that the sftape interface is going away. What should I do?

Use the *zftape interface*, but don't load the *zft-compressor module*. The device then becomes `/dev/qft0`.

<answer from Tim Jones>

11.18 Ftape says "This tape has no 'Linux raw format"

You get this complaint if you haven't *erased* your freshly formatted tape. This is because *Ftape* expect a "magic header" on the tape, to be able that it is allowed to interpret the header segment in its own way (eg: file marks). To remove the problem, say

```
mt -f /dev/nftape erase
```

<from the Ftape-Howto>

11.19 Can I exchange tapes with someone using DOS?

No. The DOS software conforms to the QIC-80 specs about the layout of the DOS filesystem, and it should(?) be a small problem to write a program that can read/write the DOS format. In fact, I'd bet that creating a nice user interface would be a bigger problem.

<From the Ftape-Howto>

11.20 How does `mt eom' work when you've started overwriting a tape in the middle?

(EOM is "End Of recorded Media", the position right after all data already recorded to the tape)

One cannot use tape "files" like files on an ordinary file system.

In principle, a tape doesn't allow anything but appending new data at EOM. However, if one positiones just in the middle of the already recorded data AND starts writing, then the driver first deletes all following files (thus moving the EOM to the actual position) and then starts writing.

Thus, the new EOM after finishing the write process, is then after the newly recorded data.

One of the consequences of the above is, of course, that writing to the tape in the middle of the already recorded area, is destructive in the sense, that it not only overwrites the "file" the tape is positioned at, but also deletes all following files.

<from the Ftape-Howto> <Answer from Claus Heine>

11.21 When I made backups before using taper, under the 2.0.29 ftape my drive didn't support fsf, under the new zftape it does, why would this be, and what exactly is fsf ?

TOC

It probably didn't work before because you didn't use a

```
mt -f /dev/rft0 erase
```

before writing data to the cartridge. THIS ISN'T necessary any more.

But, hey, what does *mt fsf*? Tape drives don't store files in the sense that you can use

```
cp somefile /dev/my_what_ever_tape
```

or be able to mount the tape drive like you could mount a harddisk. You can't do nothing with a tape drive but write data to it in a sequential manner.

As this is quite inconvenient, somebody invented something which is known under the name *file mark* or *eof mark* (eof == End Of File). Those marks don't separate files that have been backed up to the tape device, but only separate blocks of data (whatever data that might be).

Normally, the kernel tape device drivers take care of writing file marks when the tape device is closed, i.e.

```
tar -cf /dev/nqft0 /bin
tar -cf /dev/nqft0 /etc
mt -f /dev/nqft0 rewind
```

would result in a backup of all files under */bin* and */etc*. When the first *tar* finishes, the kernel driver will take care of writing a file mark to the tape at the the current tape position, and when the second *tar* process has finished, another file mark is written to the tape cartridge at that position.

Now, the sense of those file marks is, that it is possible to skip between different archives on the tape more quickly than would be possible with reading the data back.

The commands to do that are:

mt fsf

fast skip to the next file mark towards EOT (End Of Tape)

mt bsf

fast skip to the next file marks towards BOT (Begin Of Tape)

Thus, to extract the second archive in the example above, one doesn't need to read the first archive back, but can proceed as follows:

```
mt -f /dev/nqft0 rewind
mt -f /dev/nqft0 fsf
tar -xvf /dev/nqft0
```

<Answer from Claus Heine>

11.22 What exactly is the difference between ftape, and zftape?

TOC

When *Ftape* was young there were two versions of the floppy tape driver, one of them was called *zftape* because of its built-in user-transparent on-the-fly compression. Whether such a thing is a feature or a bug ('cause this needn't be done in kernel space) is another question. However, the ioctl interface and file mark handling provided by *zftape* was much better and had less bugs. And *zftape* allows to use floppy tape cartridges with different OS. Well, you can't exchange data, but *zftape* won't overwrite volumes created by your Windoze program, and vice versa.

Nowadays, *Ftape* is name of the entire floppy tape driver package AND *ftape.o* is the file-name of the kernel module that implements the low-level hardware support. *zftape* has ceased to exist as a separate package, but the new *Ftape* versions (since ftape-3.00) contain a *zftape.o* module that needs to be loaded on top of *ftape.o* (i.e. you need to load BOTH modules to be able to access your floppy tape drive) and implements the file system interface and the advanced (?) features of the previous verions *zftape*.

<Answer from Claus Heine>

11.23 What is the difference between a rewinding, and non rewinding drive?

Well, the rewinding tape devices rewind the tape to BOT (Begin Of Tape) when the device is closed, i.e.

```
tar -cvf /dev/qft0 /bin
```

will rewind the tape cartridge when the tar job has finished. In contrast,

```
tar -cvf /dev/nqft0 /bin
```

will NOT rewind the tape cartridge and leave the tape R/W head at its current position.

Rewinding devices should be used when performing a single backup, non-rewinding devices can be useful when doing multiple backups as one doesn't need to space to EOM (End Of recorded Media) before appending another archive.

Non-rewinding devices MUST be used when sending any of the tape motion command to the tape drive, such as

```
mt -f /dev/nqft0 fsf
```

, because when the *mt* process finishes then the tape device is closed which would result in rewinding the cartridge with the rewinding devices.

<Answer from Claus Heine>

11.24 Can someone tell me how to use mt to rewind my TR-3 drive one record using zftape record, so I can verify it?

Well, it depends. If the tape is still positioned inside the volume just written, "mt bsf 1" (or equivalently "mt bsf") will backspace just to the beginning of that volume (this is how "tar --verify" works). If the tape is already positioned AFTER the filemark that marks the end of the last written volume, then you need to issue "mt bsf 2"

The logic behind this is as follows: "MTBSF count" backspaces over count file marks, stops, and then positions on the `EOT` side of the last skipped file mark. This means, an "mt bsf 2" will position right at the beginning of the previous

volume.

<answer form Claus Heine>

11.25 By non-rewinding, they mean that it doesn't automatically rewind, correct? It doesn't mean that under no circumstances it will rewind, right? I tried using /dev/zqft0, and it instantly rewinds the tape.

You are right: auto-rewind means, the tape is rewound when the tape device is closed, non-rewinding means, the tape isn't automatically rewound when the tape device is closed (but you can, of course, use the tape motion commands BSF/FSF etc. to position the tape head at every position you like).

<answer form Claus Heine>

11.26 What is the difference between what mt considers a record and what it considers a file?

A record is the minimal amount of bytes that will be accepted by the tape in one read/write operation (except in "variable block size mode" where it just should be the amount of data actually written in a single write operation??).

For *zftape* every read and write access has to be a multiple of a fixed block size (fixed, but tunable with `MTSETBLK`). This block size is a "tape record" (as mentioned in the GNU mt man page and defaults to 10kb for *zftape*.

A "file" (in the sense of the mt man page) is a, well, misleading terminus. What is meant is an area of the tape between two file marks. This is not a file like a file on the file system, in the sense that it could have a name, file access modes, could be moved or copied with cp, mv, rm etc.

Instead, It simply is the area of the tape that was recorded in one backup session, its end is marked by a tape file mark, and its beginning is delimited by either `BOT` or the file mark of the previous tape "file". That tape "files" are the things that can be skipped with the `MTBSF/FSF` commands.

<answer form Claus Heine>

11.27 Reusing tapes with zftape without reformatting the tape.

We try to answer the followong questions :

- Is there a good way to erase, as in remove the data or at least the volumes from a tape, without reformating?
- Can you overwrite the last volume on a tape with making a mess out of it?
- Can you overwrite the last several volumes without making a mess?
- Can you delete the last volume?

If you want to "erase" an entire cartridge, then simply do:

```
mt -f /dev/qft0 erase
```

This will erase the volume table (i.e. the "file marks").

TOC

Pre-ftape-3.x releases of zftape and ftape used to allow overwriting of already existing volumes on a cartridge. I hav removed this feature as it was reported that it already has caused data-loss with some backup programs.

If you indeed need to remove some volumes on the tape then you should use the

```
vtblc
```

program that comes with the `ftape-tools` package which can be down-loaded from the same locations as the `ftape` kernel driver package. Please refer to the documentation which is contained in the `ftape-tools` package for more information.

If you simply want to reuse old tapes, then it suffices to do

```
mt rewind
```

If the tape is at BOT (Begin Of Tape) then every write access to the tape will silently erase all file marks and overwrite the data already existing on the tape.

<answer by Claus Heine>

11.28 This script implements a simple contents listing for the zftape package using the "MTIOCVOLINFO" ioctl.

Here is as little perl/bash script that lists the contents of a cartridge using the *zftape* specific "volinfo" ioctl. Hope this shows how to handle this kind of stuff.

What it basically does is the following:

1. Rewind the cartridge
2. Issue the volinfo command:

   ```
   claus@thales:~$ mt volinfo
   file number          = 1
   block size           = 10240
   physical space used  =  522.0 kilobytes
   real size of volume  =  520.0 kilobytes
   ```

   Parse the ouput and place the values in appropriate variables
3. Skip to the next volume with "mt fsf"
4. Exit if this gives an error (EOD), otherwise "goto 2)"

The Perl Script

```
#!/usr/bin/perl
#
#     Copyright (C) 1997 Claus-Justus Heine
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
```

```
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
#   This script implements a simple contents listing for the zftape
#   package using the MTIOCVOLINFO ioctl.
#

$version = <<EOT;
listtape-1.0 -- a perl script to list the contents of a floppy tape cartridge
under Linux using the zftape driver

RCS \$Revision: 1.2 $
RCS \$Date: 1998/08/30 13:44:03 $
EOT

$tapedev = "/dev/tape";
$usage = <<EOT;
Usage: listtape [options ...]

Mandatory or optional arguments to long options are mandatory or optional
for short options too.

-f, --file=FILE       Tape device to use. Default is  "/dev/tape".
-h, --help            Print this help.
-?                    Same as '-h'.
--usage           Same as '-h'.
-V, --version         Print version information.

Author: Claus-Justus Heine <claus\@momo.math.rwth-aachen.de>
EOT

while ($ARGV[0] =~ /^-/) {
$_ = shift;
if (/--file/) {$_ = shift; $tapedev = $_; next;}
if (/-f/) {$_ = shift; $tapedev = $_; next;}
if (/--help/) { print $usage; exit 0; }
if (/-h/) { print $usage; exit 0; }
if (/--usage/) { print $usage; exit 0; }
if (/-\?/) { print $usage; exit 0; }
if (/--version/) { print $version; exit 0; }
if (/-V/) { print $version; exit 0; }
die $usage;
}

&open_tape($tapedev, "status");
while(<FTMT>)
{
```

```
    $online = 1 if (/.*online.*/);
    }

    if (! $online) { die "No cartridge present.\n"; }

    &mtop($tapedev, "rewind");

    printf "%11s%12s%20s%20s\n",
    "file number", "block size", "volume size", "tape space";

    while (1)
    {
    &open_tape($tapedev, "volinfo");
    while (<FTMT>) {
    if (/^file number\s*=\s*([0-9]*)$/) { $filenumber = $1; }
    if (/^block size\s*=\s*([0-9]*)$/) { $blocksize = $1; }
    if (/^physical space used\s*=\s*([[0-9]*.*)/) { $rawsize = $1; }
    if (/^real size of volume\s*=\s*([[0-9]*.*)/) { $size = $1; }
    }
    close(FTMT);
    if (&mtop($tapedev, "fsf 1") != 0) {
    &mtop($tapedev,"rewind");
    print "\nRemaining space: $rawsize\n";
    print "Tape block size: $blocksize\n";
    exit 0;
    }
    printf "%6d          %5d  %20s%20s\n",
        $filenumber, $blocksize, $size, $rawsize;
    }

    sub mtop
    {
    local ($tape, $operation) = @_;
    local ($exitval);
    system "ftmt -f $tape $operation > /dev/null 2>&1";
    }

    sub open_tape
    {
    local ($tape, $operation) = @_;
    local ($command);

    $command = "ftmt -f " . $tape . " " . $operation . " |";
    open(FTMT, $command) || die "Couldn't open $command -- $!\n";
    }
```

The Bash Script

```
#! /bin/bash
#
```

```
#     Copyright (C) 1997 Claus-Justus Heine
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
#   This script implements a simple contents listing for the zftape
#   package using the MTIOCVOLINFO ioctl.
#

#
# insert better option parsing here
#
TAPEDEV=${1-/dev/tape}

if ! echo $TAPEDEV | grep "/dev/n"
then
TAPEDEV=/dev/n$(basename $TAPEDEV)
fi

if ! [ -c $TAPEDEV ]
then
echo $TAPEDEV is not a character device!  1>&2
exit 1
fi

if ! mt -f $TAPEDEV rewind
then
echo Could not rewind $TAPEDEV - no cartridge present?  1>&2
exit 1
fi

echo -e "\nContents of $TAPEDEV:\n"

printf "%11s%12s%20s%20s\n" "file number" "block size" "volume size" "tape space"

trap "rm -f /tmp/$0.$$" exit

while true
do
if ! foo=$(mt -f $TAPEDEV volinfo |cut -f 2 -d =)
then
echo $TAPEDEV doesn\'t seem to be a floppy tape device 1>&2
```

```
exit 1
fi
#
# "echo foo | read foo" will not work as the "read foo" is executed in
# another shell.
#
echo $foo > /tmp/$0.$$
read file blksz used usedunit size sizeunit < /tmp/$0.$$
if ! mt -f $TAPEDEV fsf 1 > /dev/null 2>&1
then
echo -e "\nRemaining space: $used $usedunit"
echo -e "Tape block size: $blksz"
if ! mt -f $TAPEDEV rewind
then
    echo Rewind of $TAPEDEV failed 1>&2
    exit 1
fi
exit 0
fi
printf "%6d          %5d  %20s%20s\n"\
$file $blksz "$size $sizeunit" "$used $usedunit"
done
```

<answer from Claus Heine>

---

12. FAQ: "Tape and Drivers" related questions !

12.1 What are good makers of Travan tapes?

I was the UNIX Product Manager at Archive Corp (Prior to the Conner/Seagate mess) and we performed extensive tests of tape media for compatibility certification, including retentivity, flaking and length consistancy. Based on the results of the tests, we selected the best of these certified manufacturers' products to private label as our own media. Here is the order in which we selected vendors up through 1995 (when I lost contact with the ATI group):

QIC

1. 3M (now known as Imation)
2. QMaxell/Sony (tie)
3. (BTW - Iomega uses Sony private-labelled media)

4MM

1. Fuji
2. Maxell/Sony (tie - is this a trend?)

8MM

1. Fuji/Exabyte - which we believed to be OEM'd Fuji (tie - so much for trend!)
2. Sony
3. Maxell

DLT

1. Maxell
2. Sony

Otherwise, we had entries in our search from other vendors which were generally a private-labelled version of one of the major labels above. The exceptions were Verbatim and DIC. Both of these manufacturer's media had fall-out rates and length discrepancies that were so high that we would not certify them and even warned customers about them indicating that we could not offer any sort of guarantee that they would get a good backup using the media from these manufacturers.

In addition, since coming to EST, I've found that Verbatim media is still not worth the money saved in purchasing it. We have 11 of their TR-Extra and QIC-Extra (QICXL) tapes that were useless after fewer than 20 passes each.

While this is my personal opinion, it is based on over 9 years of experience with this very question, I strongly recommend Imation/3M media for QIC/Travan user, Fuji media 4MM users, Exabyte/Fuji for 8MM and DEC labelled media for DLT users.

<answer from Tim Jones>

TOC

12.2 Where can I obtain the QIC standards?

If you wish to help developing *Ftape*, or add some utility (e.g. a tape formatting program), you will need that appropriate QIC standards. The standard(s) to get is: QIC-80, -117, -3010, and 3020. QIC-117 describes how commands are sent to the tape drive (including timing etc), so you would probably never need it. QIC-80/3010/3020 describes higher level part, such as tape layout, ECC code, standard filesystem. You can get the QIC standards from the following address:

```
Quarter Inch Cartridge Drive Standards, Inc.
311 East Carrillo Street
Santa Barbara, California 93101
Phone: (805) 963-3853
Fax:   (805) 962-1541
```

Note: They are registered as `Freeman Associates, Inc' in the phone book.

<from the Ftape-Howto>

12.3 Is the Iomega Ditto 2GB drive supported?

Yes, if you are using version `ftape-3.x` or later of the *Ftape drivers* from the Ftape Home Page or from ftp://sunsite.unc.edu/pub/Linux/kernel/tapes.

<answer from Tim Jones>

As the Ditto 2GB is a Tr-3 tape (though it can only store 1GB instead of the 1.6GB you get with a regular Tr-3 drive) you need an FDC (FDC means: Floppy Disk Controller) that is capable of at 1Mbit/sec transfer rate. You don't need to worry about this if you have an accellerator card (i.e. the Ditto Dash controller). Otherwise try to purchase an FDC that claims to be capable of driving 2.88Mb floppies because this implies that the FDC is capable of 1Mbit transfer rate.

Ftape prints the maximum data rate of the FDC to the kernel log files like this:

```
ftape-ctl.c (ftape_init_drive) - Highest FDC supported data rate: 500 Kbps.
```

<answer from Claus Heine>

12.4 Is the Iomega Ditto Max drive supported?

Yes, if you are using version `ftape-4.02` or later of the *Ftape drivers* from the Ftape Home Page or from ftp://sunsite.unc.edu/pub/Linux/kernel/tapes.

<answer from Claus Heine>

12.5 Is the Iomega Ditto Max Pro drive supported?

TOC

Yes. But if you want to use the 5GB (10GB with compression) cartridges you don't need it. With `ftape` there doesn't seem to be any difference between the `Ditto Max` and the `Ditto Max Pro`.

<answer from Claus Heine>

---

13. FAQ: Miscellaneous !

13.1 How to subscribe to the Ftape Mailing List?

You can subscribe to that list by sending mail to

```
majordomo@vger.rutgers.edu
```

with the single line

```
subscribe linux-tape
```

in its body. Please store the answer you get from majordomo in a safe place because it contains instructions how to `UNSUBSCRIBE` from the mailing list.

<answer from Claus Heine>

13.2 How to un-subscribe from the Ftape Mailing List?

Send mail to

```
majordomo@vger.rutgers.edu
```

with the single line

```
unsubscribe linux-tape MY@EMAIL.ADDRESS
```

where `MY@EMAIL.ADDRESS` has to be replaced by the the email address that you used when subscribing to the list. Note that you must have received an email with instructions how to unsubscribe from the mailing list at the time you subscribed to it.

<answer from Claus Heine>

13.3 Links to related information.

<http://www.uwsg.indiana.edu/usai/library/backups.html>

More links wanted !!!

14. Debugging the `ftape` driver

14.1 The kernel/`ftape` crashes on me when I do `...' - is that a bug?

No, that is a feature `;‐)`

Seriously, reliable software do not crash. Especially kernels do not or rather **should** not crash. If the kernel crashes upon you when you are running `ftape`, and you can show that it is `ftape` that is messing things up, regard it as a Bug That Should Be Fixed. Mail the details to the maintainer (`<heine@math1.rwth-aachen.de>`) and to the tape list.

14.2 OK, it's a bug ...ehhh... feature - How do I submit a report?

First, make sure you can reproduce the problem. Spurious errors are a pain in the ass, since they are just about impossible to hunt down `:-/` This is a quick check list:

- Kernel version, and patches applied
- `ftape` version
- tape drive model / manufacturer
- Expansion bus type (EISA, ISA, PCI, or VL-bus)
- What you did to expose the problem
- What went wrong on your system.
- Do not delete the kernel and the `ftape.o` file. I might want you run try some patches out or run a different test on your system.

Increase the tracing level to 4 or 5 and run the command that caused problems again (don't do it if your fear that you loose data or damage your hardware, there is absolutely no warranty for neither data loss nor hardware damage caused by `ftape`, remember this). Increasing the trace level beyond 5 probably doesn't make any sense as it affects the timing of the driver in a way that it doesn't work well any more. Get the tracing data from the kernel log or `/proc/kmsg`, depending on where you harvest your error messages. Try to look at what `ftape` spews out at you. It may look in-comprehensible to you at first, but you can get valuable information from the logfile. Most messages have a function name prepended, to make it easier to locate the problem. Look through the source, don't just cry ``WOLF!'', without giving it a try. If your version of the kernel (or `ftape` for that matter), is ``old'', when compared to the newest version of the kernel, try to get a newer (or even the newest) kernel and see if the problem goes away under the new kernel. When you post your problem report, include the information about ftape version, kernel version, expansion bus type (ISA, VL-bus, PCI or EISA), bus speed, floppy controller, and tape

drive. State exactly what you did, and what happened on your system. Some people have experience that `ftape` would not run in a PCI based box, but ran flawlessly in a normal ISA based 386DX mac (see section Getting PCI motherboards to work with <tt/ftape/ on PCI machines above)

Also, please think of the poor souls who actually *pay* the their Internet access (like me): avoid posting a (huge) log from the `ftape` run, without reason. Instead, you could describe the problem, and offer to send the log to the interested parties.

Send your bug report to `<linux-tape@vger.rutgers.edu>`. You might also want to mail the bug to `<heine@math1.rwth-aachen.de>`.

---

15. Contributions

The following is a list of notable folks that have contributed to `ftape`'s HOWTO document. This is a recent addition added by someone coming in midstream. My sincerest apologies if I've inadvertently left someone important off the list. You can view anoterh attempt to collect such kind of information at Ftape's Hall of Fame

`Johan De Wit` <jo@correct.nl>: The maintainer of the Ftape FAQ.

`Kevin Johnson` <kjj@pobox.com>: The previous maintainer of the `Ftape-HOWTO`

`Kai Harrekilde-Petersen` <khp@dolphinics.no>: The previous maintainer of `ftape` and the HOWTO.

`Andrew Martin` <martin@biochemistry.ucl.ac.uk>: Many additions to the HOWTO.

`Bas Laarhoven` <bas@vimec.nl>: The original author of `ftape`.

The Linux GCC HOWTO

Daniel Barlow <dan@detached.demon.co.uk>

v1.17, 28 February 1996

This document covers how to set up the GNU C compiler and development libraries under Linux, and gives an overview of compiling, linking, running and debugging programs under it. Most of the material in it has been taken from Mitch D'Souza's GCC-FAQ, which it replaces, or the ELF-HOWTO, which it will eventually largely replace. This is the first publically released version (despite the version number; that's an artifact of RCS). Feedback is welcomed.

1. Preliminaries

- 1.1 ELF vs. a.out
- 1.2 Administrata
- 1.3 Typography

2. Where to get things

- 2.1 This document
- 2.2 Other documentation
- 2.3 GCC
- 2.4 C library and header files
- 2.5 Associated tools (as, ld, ar, strings etc)

3. GCC installation and setup

- 3.1 GCC versions
- 3.2 Where did it go?
- 3.3 Where are the header files?
- 3.4 Building cross compilers

TOC

4. Porting and Compiling

- 4.1 Automatically defined symbols
- 4.2 Compiler invocation
- 4.3 Portability

5. Debugging and Profiling

- 5.1 Preventative maintenance (lint)
- 5.2 Debugging
- 5.3 Profiling

6. Linking

- 6.1 Shared vs static libraries
- 6.2 Interrogating libraries (`which library is `sin()` in?')
- 6.3 Finding files
- 6.4 Building your own libraries

7. Dynamic Loading

- 7.1 Concepts
- 7.2 Error messages
- 7.3 Controlling the operation of the dynamic loader
- 7.4 Writing programs with dynamic loading

8. Contacting the developers

- 8.1 Bug reports
- 8.2 Helping with development

9. The Remains

- 9.1 The Credits
- 9.2 Translations

- 9.3 Feedback
- 9.4 Legalese

10. Index

1. Preliminaries

1.1 ELF vs. a.out

Linux development is in a state of flux right now. Briefly, there are two formats for the binaries that Linux knows how to execute, and depending on how your system is put together, you may have either. When reading this HOWTO, it helps to know which.

How to tell? Use the `file' utility (eg `file /bin/bash`). For an ELF program it will say something with ELF in, for an a.out program it will say something involving `Linux/i386`.

The differences between ELF and a.out are covered (extensively) later in this document. ELF is the newer format, and generally accepted as better.

1.2 Administrata

The copyright information and like legalese can be found at the *end* of this document, together with the statutory warnings about asking dumb questions on Usenet, revealing your ignorance of the C language by reporting bugs which aren't, and picking your nose while chewing gum.

1.3 Typography

If you're reading this in Postscipt, dvi, or html format, you get to see a little more font variation than people with the plain text version. In particular, filenames, commands, command output and source code excerpts are set in some form of `typewriter` font, whereas `variables' and random things that need emphasizing are *empasized*.

You also get a usable index. In dvi or postscript, the numbers in the index are section numbers. In HTML they're just sequentially assigned numbers that you can click on. In the plain text version, they really are just numbers. Get an upgrade!

The Bourne (rather than C) shell syntax is used in examples. C shell users will want to use

```
% setenv FOO bar
```

where I have written

```
$ FOO=bar; export FOO
```

If the prompt shown is # rather than `$`, the command shown will probably only work as root. Of course, I accept no responsibility for anything that happens to your system as a result of trying these examples. Have a nice day `:-)`

2. Where to get things

2.1 This document

This document is one of the Linux HOWTO series, so is available from all Linux HOWTO repositories, such as http://sunsite.unc.edu/pub/linux/docs/HOWTO/. The HTML version can also be found (possibly in a slightly newer version) from http://ftp.linux.org.uk/~barlow/howto/gcc-howto.html.

2.2 Other documentation

The official documentation for gcc is in the source distribution (see below) as texinfo files, and as `.info` files. If you have a fast network connection, a cdrom, or a reasonable amount of patience, you can just untar it and copy the relevant bits into `/usr/info`. If not, you may find them at tsx-11, but not necessarily always the latest version.

There are two source of documentation for libc. GNU libc comes with info files which describe Linux libc fairly accurately except for stdio. Also, the manpages archive are written for Linux and describe a lot of system calls (section 2) and libc functions (section 3).

2.3 GCC

There are two answers.

(a) The official Linux GCC distribution can always be found in binary (ready-compiled) form at ftp://tsx-11.mit.edu:/pub/linux/packages/GCC/. At the time of writing, 2.7.2 (`gcc-2.7.2.bin.tar.gz`) is the latest version.

(b) The latest source distribution of GCC from the Free Software Foundation can be had from GNU archives. This is not necessarily always the same version as above, though it is just now. The Linux GCC maintainer(s) have made it easy for you to compile the latest version available yourself --- the `configure` script should set it all up for you. Check tsx-11 as well, for patches which you may want to apply.

To compile anything non-trivial (and quite a few trivial things also) you will also need the

2.4 C library and header files

What you want here depends on (i) whether your system is ELF or a.out, and (ii) which you want it to be. If you're upgrading from libc 4 to libc 5, you are recommended to look at the ELF-HOWTO from approximately the same place as you found this document.

TOC

These are available from tsx-11 as above:

`libc-5.2.18.bin.tar.gz`

--- ELF shared library images, static libraries and include files for the C and maths libraries.

`libc-5.2.18.tar.gz`

--- Source for the above. You will also need the `.bin.` package for the header files. If you are deliberating whether to compile the C library yourself or use the binaries, the right answer in nearly all cases is to use the binaries. You will however need to roll your own if you want NYS or shadow password support.

`libc-4.7.5.bin.tar.gz`

--- a.out shared library images and static libraries for version 4.7.5 of the C library and friends. This is designed to coexist with the libc 5 package above, but is only really necessary if you wish to keep using/developing a.out format programs.

2.5 Associated tools (as, ld, ar, strings etc)

From tsx-11, just like everything else so far. The current version is `binutils-2.6.0.2.bin.tar.gz`.

Note that the binutils are only available in ELF, the current libc version is in ELF and the a.out libc is happiest when used in conjunction with an ELF libc. C library development is moving emphatically ELFwards, and unless you have really good reasons for needing a.out things you're encouraged to follow suit.

---

3. GCC installation and setup

3.1 GCC versions

You can find out what GCC version you're running by typing `gcc -v` at the shell prompt. This is also a fairly reliable way to find out whether you are set up for ELF or a.out. On my system it does

```
$ gcc -v
Reading specs from /usr/lib/gcc-lib/i486-box-linux/2.7.2/specs
gcc version 2.7.2
```

The key things to note here are

- `i486`. This indicates that the gcc you are using was built for a 486 processor --- you might have 386 or 586 instead. All of these chips can run code compiled for each of the others; the difference is that the 486 code has added padding in some places so runs faster on a 486. This has no detrimental performance effect on a 386, but does make the binaries slightly larger.
- `box`. This is *not* at all important, and may say something else (such as `slackware` or `debian`) or nothing at all (so that the complete directory name is `i486-linux`). If you build your own gcc, you can set this at build time for cosmetic effect. Just like I did `:-)`
- `linux`. This may instead say `linuxelf` or `linuxaout`, and, confusingly, the meaning of each varies according to the version that you are using.
  - `linux` means ELF if the version is 2.7.0 or newer, a.out otherwise.
  - `linuxaout` means a.out. It was introduced as a target when the definition of `linux` was changed from a.out to ELF, so you won't see any `linuxaout` gcc older than 2.7.0.
  - `linuxelf` is obsolete. It is generally a version of gcc 2.6.3 set to produce ELF executables. Note that gcc 2.6.3 has known bugs when producing code for ELF --- an upgrade is advisable.
- `2.7.2` is the version number.

So, in summary, I have gcc 2.7.2 producing ELF code. Quelle surprise.

3.2 Where did it go?

If you installed gcc without watching, or if you got it as part of a distribution, you may like to find out where it lives in the filesystem. The key bits are

- `/usr/lib/gcc-lib/`*target*`/`*version*`/` (and subdirectories) is where most of the compiler lives. This includes the executable programs that do actual compiling, and some version-specific libraries and include files.
- `/usr/bin/gcc` is the compiler driver --- the bit that you can actually run from the command line. This can be used with multiple versions of gcc provided that you have multiple compiler directories (as above) installed. To find out the default version it will use, type `gcc -v`. To force it to another version, type `gcc -V` *version*. For example

```
# gcc -v
Reading specs from /usr/lib/gcc-lib/i486-box-linux/2.7.2/specs
gcc version 2.7.2
# gcc -V 2.6.3 -v
Reading specs from /usr/lib/gcc-lib/i486-box-linux/2.6.3/specs
```

```
gcc driver version 2.7.2 executing gcc version 2.6.3
```

- `/usr/`*target*`/(bin|lib|include)/`. If you have multiple targets installed (for example, a.out and elf, or a cross-compiler of some sort, the libraries, binutils (`as`, `ld` and so on) and header files for the non-native target(s) can be found here. Even if you only have one kind of gcc installed you might find anyway that various bits for it are kept here. If not, they're in `/usr/(bin|lib|include)`.
- `/lib/`,`/usr/lib` and others are library directories for the native system. You will also need `/lib/cpp` for many applications (X makes quite a lot of use of it) --- either copy it from `/usr/lib/gcc-lib/`*target*`/`*version*`/` or make a symlink pointing there.

3.3 Where are the header files?

Apart from whatever you install yourself under `/usr/local/include`, there are three main sources of header files in Linux:

- Most of `/usr/include/` and its subdirectories are supplied with the libc binary package from H J Lu. I say `most' because you may also have files from other sources (`curses` and `dbm` libraries, for example) in here, especially if you are using the newest libc distribution (which doesn't come with curses or dbm, unlike the older ones).
- `/usr/include/linux` and `/usr/include/asm` (for the files `<linux/*.h>` and `<asm/*.h>`) should be symbolic links to the directories `linux/include/linux` and `linux/include/asm` in the kernel source distribution. You need to install these if you plan to do *any* non-trivial development; they are not just there for compiling the kernel. You might find also that you need to do `make config` in the kernel directory after unpacking the sources. Many files depend on `<linux/autoconf.h>` which otherwise may not exist, and in some kernel versions `asm` is a symbolic link itself and only created at `make config` time. So, if you unpack your kernel sources under `/usr/src/linux`, that's

```
$ cd /usr/src/linux
$ su
# make config
[answer the questions.  Unless you're going to go on and build the kernel
it doesn't matter _too_ much what you say]
# cd /usr/include
# ln -s ../src/linux/include/linux .
# ln -s ../src/linux/include/asm .
```

- Files such as `<float.h>`, `<limits.h>`, `<varargs.h>`, `<stdarg.h>` and `<stddef.h>` vary according to the compiler version, so are found in `/usr/lib/gcc-lib/i486-box-linux/2.7.2/include/` and places of that ilk.

3.4 Building cross compilers

Linux as the target platform

Assuming you have obtained the source code to gcc, usually you can just follow the instructions given in the INSTALL file for GCC. A `configure --target=i486-linux --host=XXX` on platform `XXX` followed by a `make` should do the trick. Note that you will need the Linux includes, the kernel includes, and also to build the cross assembler and cross linker from the sources in ftp://tsx-11.mit.edu/pub/linux/packages/GCC/.

TOC

Linux as the source platform, MSDOS as the target

Ugh. Apparently this is somewhat possible by using the "emx" package or the "go" extender. Please look at ftp://sunsite.unc.edu/pub/Linux/devel/msdos.

I have not tested this and cannot vouch for its abilities.

---

4. Porting and Compiling

4.1 Automatically defined symbols

You can find out what symbols your version of gcc defines automatically by running it with the `-v` switch. For example, mine does:

```
$ echo 'main(){printf("hello world\n");}' | gcc -E -v -
Reading specs from /usr/lib/gcc-lib/i486-box-linux/2.7.2/specs
gcc version 2.7.2
 /usr/lib/gcc-lib/i486-box-linux/2.7.2/cpp -lang-c -v -undef
-D__GNUC__=2 -D__GNUC_MINOR__=7 -D__ELF__ -Dunix -Di386 -Dlinux
-D__ELF__ -D__unix__ -D__i386__ -D__linux__ -D__unix -D__i386
-D__linux -Asystem(unix) -Asystem(posix) -Acpu(i386)
-Amachine(i386) -D__i486__ -
```

If you are writing code that uses Linux-specific features, it is a good idea to enclose the nonportable bits in

```
#ifdef __linux__
/* ... funky stuff ... */
#endif /* linux */
```

Use `__linux__` for this purpose, *not* `linux`. Although the latter is defined, it is not POSIX compliant.

4.2 Compiler invocation

The documentation for compiler switches is the gcc info page (in Emacs, use `C-h i` then select the `gcc' option). Your distributor may not have packed this with your system, or you may have an old version; the best thing to do in this case is to download the gcc source archive from ftp://prep.ai.mit.edu/pub/gnu or one of its mirrors, and copy them out of it.

The gcc manual page (`gcc.1`) is, generally speaking, out of date. It will warn you of this when you try to look at it.

Compiler flags

gcc can be made to optimize its output code by adding `-O`*n* to its command line, where *n* is an optional small integer. Meaningful values of *n*, and their exact effect, vary according to the exact version, but typically it ranges from 0 (no optimization) to 2 (lots) or 3 (lots and lots).

Internally, gcc translates these to a series of `-f` and `-m` options. You can see exactly which `-O` levels map to which options by running gcc with the `-v` flag and the (undocumented) `-Q` flag. For example, for `-O2`, mine says

```
enabled: -fdefer-pop -fcse-follow-jumps -fcse-skip-blocks
-fexpensive-optimizations
         -fthread-jumps -fpeephole -fforce-mem -ffunction-cse -finline
         -fcaller-saves -fpcc-struct-return -frerun-cse-after-loop
         -fcommon -fgnu-linker -m80387 -mhard-float -mno-soft-float
         -mno-386 -m486 -mieee-fp -mfp-ret-in-387
```

Using an optimization level higher than your compiler supports (e.g. `-O6`) will have exactly the same effect as using the highest level that it *does* support. Distributing code which is set to compile this way is a poor idea though --- if further optimisations are incorporated into future versions, you (or your users) may find that they break your code.

Users of gcc 2.7.0 thru 2.7.2 should note that there is a bug in `-O2` on these. Specifically, strength reduction doesn't work. A patch can be had to fix this if you feel like recompiling gcc, otherwise make sure that you always compile with `-fno-strength-reduce`

Processor-specific

There are other `-m` flags which aren't turned on by any variety of `-O` but are nevertheless useful. Chief among these are `-m386` and `-m486`, which tell gcc to favour the 386 or 486 respectively. Code compiled with one of these will still work on the other; 486 code is bigger, but otherwise not slower on the 386.

There is currently no `-mpentium` or `-m586`. Linus suggests using `-m486 -malign-loops=2 -malign-jumps=2 -malign-functions=2`, to get 486 code optimisations but without the big gaps for alignment (which the pentium doesn't need). Michael Meissner (of Cygnus) says

> My hunch is that `-mno-strength-reduce` also results in faster code on the x86 (note, I'm not talking about the strength reduction bug, which is another issue). This is because the x86 is rather register starved (and GCC's method of grouping registers into spill registers vs. other registers doesn't help either). Strength reduction typically results in using additional registers to replace multiplications with addition. I also suspect `-fcaller-saves` may also be a loss.
>
> Another hunch is that `-fomit-frame-pointer` might or might not be a win. On the one hand, it can mean that another register is available for allocation. On the other hand, the way the x86 encodes its instruction set, means that stack relative addresses take more space instead of frame relative addresses, which means slightly less Icache availble to the program. Also, `-fomit-frame-pointer`, means that the compiler has to constantly adjust the stack pointer after calls, while with a frame, it can let the stack accumulate for a few calls.

The final word on this subject is from Linus again:

> Note that if you want to get optimal performance, don't believe me: test. There are lots of gcc compiler switches, and it may be that a particular set gives the best optimizations for you.

`Internal compiler error: cc1 got fatal signal 11`

Signal 11 is SIGSEGV, or `segmentation violation'. Usually it means that the program got its pointers confused and tried to write to memory it didn't own. So, it could be a gcc bug.

gcc is however, a well tested and reliable piece of software, for the most part. It also uses a large number of complex data structures, and an awful lot of pointers. In short, it's the pickiest RAM tester commonly

available. If you *can't duplicate the bug* --- if it doesn't stop in the same place when you restart the compil --- it's almost certainly a problem with your hardware (CPU, memory, motherboard or cache). **Don't** clai a bug because your computer passes the power-on checks or runs Windows ok or whatever; these `tests' are commonly and rightly held to be worthless. And don't claim it's a bug because a kernel compile always stops during ``make zImage`' --- of course it will! `` `make zImage`' is probably compiling over 200 files; we're looking for a slightly *smaller* place than that.

If you can duplicate the bug, and (better) can produce a short program that exhibits it, you can submit it as a bug report to the FSF, or to the linux-gcc mailing list. See the gcc documentation for details of exactly what information they need.

4.3 Portability

It has been said that, these days, if something hasn't been ported to Linux then it is not worth having :-)

Seriously though, in general only minor changes are needed to the sources to get over Linux's 100% POSIX compliance. It is also worthwhile passing back any changes to authors of the code such that in the future only `make' need be called to provide a working executable.

BSDisms (including `bsd_ioctl`, `daemon` and `<sgtty.h>`)

You can compile your program with `-I/usr/include/bsd` and link it with `-lbsd` (i.e. add `-I/usr/include/bsd` to `CFLAGS` and `-lbsd` to the `LDFLAGS` line in your Makefile). There is *no* need to add `-D__USE_BSD_SIGNAL` any more if you want BSD type signal behavior, as you get this automatically when you have `-I/usr/include/bsd` and include `<signal.h>`.

`Missing' signals (`SIGBUS`, `SIGEMT`, `SIGIOT`, `SIGTRAP`, `SIGSYS` etc)

Linux is POSIX compliant. These are not POSIX-defined signals --- ISO/IEC 9945-1:1990 (IEEE Std 1003.1-1990), paragraph B.3.3.1.1 sez:

> ``The signals SIGBUS, SIGEMT, SIGIOT, SIGTRAP, and SIGSYS were omitted from POSIX.1 because their behavior is implementation dependent and could not be adequately categorized. Conforming implementations may deliver these signals, but must document the circumstances under which they are delivered and note any restrictions concerning their delivery."

The cheap and cheesy way to fix this is to redefine these signals to `SIGUNUSED`. The *correct* way is to bracket the code that handles them with appropriate `#ifdef`s:

```
#ifdef SIGSYS
/* ... non-posix SIGSYS code here .... */
#endif
```

K & R Code

GCC is an ANSI compiler; much existing code is not ANSI. There's really not much that can be done about this, except to add `-traditional` to the compiler flags. There is a certain amount of finer-grained control over which varieties of brain damage to emulate; consult the gcc info page.

TOC

Note that `-traditional` has effects beyond just changing the language that gcc accepts. For example, it turns on `-fwritable-strings`, which moves string constants into data space (from text space, where it cannot be written to). This increases the memory footprint of the program.

Preprocessor symbols conflict with prototypes in the code

One of the most frequent problems is that some common functions are defined as macros in Linux's header files and the preprocessor will refuse to parse similar prototype definitions in the code. Common ones are `atoi()` and `atol()`.

`sprintf()`

Something to be aware of, especially when porting from SunOS, is that `sprintf(string, fmt, ...)` returns a pointer to `string` on many unices, whereas Linux (following ANSI) returns the number of characters which were put into the string.

`fcntl` and friends. Where are the definitions of`FD_*` stuff ?

In `<sys/time.h>`. If you are using `fcntl` you probably want to include `<unistd.h>` too, for the actual prototype.

Generally speaking, the manual page for a function lists the necessary `#include`s in its SYNOPSIS section.

The `select()` timeout. Programs start busy-waiting.

Once upon a time, the timeout parameter to `select()` was used read-only. Even then, manual pages warned:

> select() should probably return the time remaining from the original timeout, if any, by modifying the time value in place. This may be implemented in future versions of the system. Thus, it is unwise to assume that the timeout pointer will be unmodified by the select() call.

The future has arrived! At least, it has here. On return from a `select()`, the timeout argument will be set to the remaining time that it would have waited had data not arrived. If no data had arrived, this will be zero, and future calls using the same timeout structure will immediately return.

To fix, put the timeout value into that structure every time you call `select()`. Change code like

```
struct timeval timeout;
timeout.tv_sec = 1; timeout.tv_usec = 0;
while (some_condition)
      select(n,readfds,writefds,exceptfds,&timeout);
```

to, say,

```
struct timeval timeout;
while (some_condition) {
      timeout.tv_sec = 1; timeout.tv_usec = 0;
      select(n,readfds,writefds,exceptfds,&timeout);
}
```

TOC

Some versions of Mosaic were at one time notable for this problem. The speed of the spinning globe anim was inversely related to the speed that the data was coming in from the network at!

Interrupted system calls.

Symptom:

When a program is stopped using Ctrl-Z and then restarted - or in other situations that generate signals: Ctrl-C interruption, termination of a child process etc. - it complains about "interrupted system call" or "write: unknown error" or things like that.

Problem:

POSIX systems check for signals a bit more often than some older unices. Linux may execute signal handlers ---

- asynchronously (at a timer tick)
- on return from any system call
- during the execution of the following system calls: `select()`, `pause()`, `connect()`, `accept()`, `read()` on terminals, sockets, pipes or files in `/proc`, `write()` on terminals, sockets, pipes or the line printer, `open()` on FIFOs, PTYs or serial lines, `ioctl()` on terminals, `fcntl()` with command `F_SETLKW`, `wait4()`, `syslog()`, any TCP or NFS operations.

For other operating systems you may have to include the system calls `creat()`, `close()`, `getmsg()`, `putmsg()`, `msgrcv()`, `msgsnd()`, `recv()`, `send()`, `wait()`, `waitpid()`, `wait3()`, `tcdrain()`, `sigpause()`, `semop()` to this list.

If a signal (that the program has installed a handler for) occurs during a system call, the handler is called. When the handler returns (to the system call) it detects that it was interrupted, and immediately returns with -1 and `errno = EINTR`. The program is not expecting that to happen, so bottles out.

You may choose between two fixes.

(1) For every signal handler that you install, add `SA_RESTART` to the sigaction flags. For example, change

```
signal (sig_nr, my_signal_handler);
```

to

```
signal (sig_nr, my_signal_handler);
{ struct sigaction sa;
  sigaction (sig_nr, (struct sigaction *)0, &sa);
#ifdef SA_RESTART
    sa.sa_flags |= SA_RESTART;
#endif
#ifdef SA_INTERRUPT
    sa.sa_flags &= ~ SA_INTERRUPT;
#endif
    sigaction (sig_nr, &sa, (struct sigaction *)0);
}
```

TOC

Note that while this applies to most system calls, you must still check for `EINTR` yourself on `read()`, `write()`, `ioctl()`, `select()`, `pause()` and `connect()`. See below.

(2) Check for `EINTR` explicitly, yourself:

Here are two examples for `read()` and `ioctl()`,

Original piece of code using `read()`

```
int result;
while (len > 0) {
  result = read(fd,buffer,len);
  if (result < 0) break;
  buffer += result; len -= result;
}
```

becomes

```
int result;
while (len > 0) {
  result = read(fd,buffer,len);
  if (result < 0) { if (errno != EINTR) break; }
  else { buffer += result; len -= result; }
}
```

and a piece of code using `ioctl()`

```
int result;
result = ioctl(fd,cmd,addr);
```

becomes

```
int result;
do { result = ioctl(fd,cmd,addr); }
while ((result == -1) && (errno == EINTR));
```

Note that in some versions of BSD Unix the default behaviour is to restart system calls. To get system calls interrupted you have to use the `SV_INTERRUPT` or `SA_INTERRUPT` flag.

Writable strings (program seg faults randomly)

GCC has an optimistic view of its users, believing that they intend string constants to be exactly that --- constant. Thus, it stores them in the text (code) area of the program, where they can be paged in and out from the program's disk image (instead of taking up swapspace), and any attempt to rewrite them will cause a segmentation fault. This is a feature!

It may cause a problem for old programs that, for example, call `mktemp()` with a string constant as argument. `mktemp()` attempts to rewrite its argument in place.

To fix, either (a) compile with `-fwritable-strings`, to get gcc to put constants in data space, or (b) rewrite the offending parts to allocate a non-constant string and strcpy the data into it before calling.

TOC

Why does the `execl()` call fail?

Because you're calling it wrong. The first argument to `execl` is the program that you want to run. The second and subsequent arguments become the `argv` array of the program you're calling. Remember: `argv[0]` is traditionally set even when a program is run with `no' arguments. So, you should be writing

```
execl("/bin/ls","ls",NULL);
```

not just

```
execl("/bin/ls", NULL);
```

Executing the program with no arguments at all is construed as an invitation to print out its dynamic library dependencies, at least using a.out. ELF does things differently.

(If you want this library information, there are simpler interfaces; see the section on dynamic loading, or the manual page for `ldd`).

---

5. Debugging and Profiling

5.1 Preventative maintenance (lint)

There is no widely-used lint for Linux, as most people are satisfied with the warnings that gcc can generate. Probably the most useful is the `-Wall` switch --- this stands for `Warnings, all' but probably has more mnemonic value if thought of as the thing you bang your head against.

There is a public domain lint available from ftp://larch.lcs.mit.edu/pub/Larch/lclint. I don't know how good it is.

5.2 Debugging

How do I get debugging information into a program ?

You need to compile and link all its bits with the `-g` switch, and without the `-fomit-frame-pointer` switch. Actually, you don't need to recompile all of it, just the bits you're interested in debugging.

On a.out configurations the shared libraries are compiled with `-fomit-frame-pointer`, which gdb won't get on with. Giving the `-g` option when you link should imply static linking; this is why.

If the linker fails with a message about not finding libg.a, you don't have `/usr/lib/libg.a`, which is the special debugging-enabled C library. It may be supplied in the libc binary package, or (in newer C library versions) you may need to get the libc source code and build it yourself. You don't actually *need* it though; you can get enough information for most purposes simply by symlinking it to `/usr/lib/libc.a`

How do I get it out again?

A lot of GNU software comes set up to compile and link with `-g`, causing it to make very big (and often static) executables. This is not really such a hot idea.

If the program has an autoconf generated `configure` script, you can usually turn off debugging information by doing `./configure CFLAGS=` or `./configure CFLAGS=-O2`. Otherwise, check the Makefile. Of course, if you're using ELF, the program is dynamically linked regardless of the `-g` setting, so you can just `strip` it.

TOC

Available software

Most people use **gdb**, which you can get in source form from GNU archive sites, or as a binary from tsx-11 or sunsite. **xxgdb** is an X debugger based on this (i.e. you need gdb installed first). The source may be found at ftp://ftp.x.org/contrib/xxgdb-1.08.tar.gz

Also, the **UPS** debugger has been ported by Rick Sladkey. It runs under X as well, but unlike xxgdb, it is not merely an X front end for a text based debugger. It has quite a number of nice features, and if you spend any time debugging stuff, you probably should check it out. The Linux precompiled version and patches for the stock UPS sources can be found in ftp://sunsite.unc.edu/pub/Linux/devel/debuggers/, and the original source at ftp://ftp.x.org/contrib/ups-2.45.2.tar.Z.

Another tool you might find useful for debugging is `**strace**', which displays the system calls that a process makes. It has a multiplicity of other uses too, including figuring out what pathnames were compiled into binaries that you don't have the source for, exacerbating race conditions in programs that you suspect contain them, and generally learning how things work. The latest version of strace (currently 3.0.8) can be found at ftp://ftp.std.com/pub/jrs/.

Background (daemon) programs

Daemon programs typically execute `fork()` early, and terminate the parent. This makes for a short debugging session.

The simplest way to get around this is to set a breakpoint for `fork`, and when the program stops, force it to return 0.

```
(gdb) list
1       #include <stdio.h>
2
3       main()
4       {
5         if(fork()==0) printf("child\n");
6         else printf("parent\n");
7       }
(gdb) break fork
Breakpoint 1 at 0x80003b8
(gdb) run
Starting program: /home/dan/src/hello/./fork
Breakpoint 1 at 0x400177c4

Breakpoint 1, 0x400177c4 in fork ()
(gdb) return 0
Make selected stack frame return now? (y or n) y
#0  0x80004a8 in main ()
    at fork.c:5
```

```
5         if(fork()==0) printf("child\n");
(gdb) next
Single stepping until exit from function fork,
which has no line number information.
child
7       }
```

Core files

When Linux boots it is usually configured not to produce core files. If you like them, use your shell's builtin command to re-enable them: for C-shell compatibles (e.g. tcsh) this is

```
% limit core unlimited
```

while Bourne-like shells (sh, bash, zsh, pdksh) use

```
$ ulimit -c unlimited
```

If you want a bit more versatility in your core file naming (for example, if you're trying to conduct a post-mortem using a debugger that's buggy itself) you can make a simple mod to your kernel. Look for the code in `fs/binfmt_aout.c` and `fs/binfmt_elf.c` (in newer kernels, you'll have to grep around a little in older ones) that says

```
        memcpy(corefile,"core.",5);
#if 0
        memcpy(corefile+5,current->comm,sizeof(current->comm));
#else
        corefile[4] = '\0';
#endif
```

and change the `0`s to `1`s.

5.3 Profiling

Profiling is a way to examine which bits of a program are called most often or run for longest. It is a good way to optimize code and look at where time is being wasted. You must compile all object files that you require timing information for with `-p`, and to make sense of the output file you will also need `gprof` (from the binutils package). See the `gprof` manual page for details.

---

6. Linking

Between the two incompatible binary formats, the static vs shared library distinction, and the overloading of the verb `link' to mean both `what happens after compilation' and `what happens when a compiled program is invoked' (and, actually, the overloading of the word `load' in a comparable but opposite sense), this section is complicated. Little of it is much more complicated than that sentence, though, so don't worry too much about it.

To alleviate the confusion somewhat, we refer to what happens at runtime as `dynamic loading' and cover it in the next section. You will also see it described as `dynamic linking', but not here. This section, then, is exclusively concerned with the kind of linking that happens at the end of a compilation.

6.1 Shared vs static libraries

The last stage of building a program is to `link' it; to join all the pieces of it together and see what is missing. Obviously there are some things that many programs will want to do --- open files, for example, and the pieces that do these things are provided for you in the form of libraries. On the average Linux system these can be found in `/lib` and `/usr/lib/`, among other places.

When using a static library, the linker finds the bits that the program modules need, and physically copies them into the executable output file that it generates. For shared libraries, it doesn't --- instead it leaves a note in the output saying `when this program is run, it will first have to load this library'. Obviously shared libraries tend to make for smaller executables; they also use less memory and mean that less disk space is used. The default behaviour of Linux is to link shared if it can find the shared libraries, static otherwise. If you're getting static binaries when you want shared, check that the shared library files (`*.sa` for a.out, `*.so` for ELF) are where they should be, and are readable.

On Linux, static libraries have names like `libname.a`, while shared libraries are called `libname.so.x.y.z` where `x.y.z` is some form of version number. Shared libraries often also have links pointing to them, which are important, and (on a.out configurations) associated `.sa` files. The standard libraries come in both shared and static formats.

You can find out what shared libraries a program requires by using `ldd` (List Dynamic Dependencies)

```
$ ldd /usr/bin/lynx
        libncurses.so.1 => /usr/lib/libncurses.so.1.9.6
        libc.so.5 => /lib/libc.so.5.2.18
```

This shows that on my system the WWW browser `lynx' depends on the presence of `libc.so.5` (the C library) and `libncurses.so.1` (used for terminal control). If a program has no dependencies, `ldd` will say `statically linked' or `statically linked (ELF)'.

6.2 Interrogating libraries (`which library is `sin()` in?')

`nm` *libraryname* should list all the symbols that *libraryname* has references to. It works on both static and shared libraries. Suppose that you want to know where `tcgetattr()` is defined: you might do

```
$ nm libncurses.so.1 |grep tcget
         U tcgetattr
```

The `U` stands for `undefined' --- it shows that the ncurses library uses but does not define it. You could also do

```
$ nm libc.so.5 | grep tcget
00010fe8 T __tcgetattr
00010fe8 W tcgetattr
00068718 T tcgetpgrp
```

The `W' stands for `weak', which means that the symbol is defined, but in such a way that it can be overridden by another definition in a different library. A straightforward `normal' definition (such as the one for `tcgetpgrp`) is marked by a `T'

The short answer to the question in the title, by the way, is `libm.(so|a)`. All the functions defined in `<math.h>` are kept in the maths library; thus you need to link with `-lm` when using any of them.

6.3 Finding files

```
ld: Output file requires shared library `libfoo.so.1`
```

The file search strategy of ld and friends varies according to version, but the only default you can reasonably assume is `/usr/lib`. If you want libraries elsewhere to be searched, specify their directories with the `-L` option to gcc or ld.

If that doesn't help, check that you have the right file in that place. For a.out, linking with `-lfoo` makes ld look for `libfoo.sa` (shared stubs), and if unsuccessful then for `libfoo.a` (static). For ELF, it looks for `libfoo.so` then `libfoo.a`. `libfoo.so` is usually a symbolic link to `libfoo.so.x`.

6.4 Building your own libraries

Version control

As any other program, libraries tend to have bugs which get fixed over time. They also may introduce new features, change the effect of existing ones, or remove old ones. This could be a problem for programs using them; what if it was depending on that old feature?

So, we introduce library versioning. We categorise the changes that might be made to a library as `minor' or `major', and we rule that a `minor' change is not allowed to break old programs that are using the library. You can tell the version of a library by looking at its filename (actually, this is, strictly speaking, a lie for ELF; keep reading to find out why) : `libfoo.so.1.2` has major version 1, minor version 2. The minor version number can be more or less anything --- libc puts a `patchlevel' in it, giving library names like `libc.so.5.2.18`, and it's also reasonable to put letters, underscores, or more or less any printable ASCII in it.

One of the major differences between ELF and a.out format is in building shared libraries. We look at ELF first, because it's simpler.

ELF? What is it then, anyway?

ELF (Executable and Linking Format) is a binary format originally developed by USL (UNIX System Laboratories) and currently used in Solaris and System V Release 4. Because of its increased flexibility over the older a.out format that Linux was using, the GCC and C library developers decided last year to move to using ELF as the Linux standard binary format also.

Come again?

This section is from the document '/news-archives/comp.sys.sun.misc'.

> ELF ("Executable Linking Format) is the "new, improved" object file format introduced in SVR4. ELF is much more powerful than straight COFF, in that it *is* user-extensible. ELF views an object-file as an arbitarily long list of sections (rather than an array of fixed size entities), these sections, unlike in COFF, do not HAVE to be in a certain place and do not HAVE to come in any specific order etc. Users can add new sections to object-files if they wish to capture new data. ELF also has a far more powerful debugging format called DWARF (Debugging With Attribute Record Format) - not currently fully supported on linux (but work is underway). A linked list of DWARF DIEs (or Debugging Information Entries) forms the .debug section in ELF. Instead of being a collection of small, fixed-size information records, DWARF DIEs each contain an arbitrarily long list of complex attributes and are written out as a scope-based tree of program data. DIEs can capture a large amount of information that the COFF .debug section simply couldn't (like C++ inheritance graphs etc.).
>
> ELF files are accessed via the SVR4 (Solaris 2.0 ?) ELF access library, which provides an easy and fast interface to the more gory parts of ELF. One of the major boons in using the ELF access library is that you will never need to look at an ELF file qua. UNIX file, it is accessed as an Elf *, after an elf_open() call and from then on, you perform elf_foobar() calls on its components instead of messing about with its actual on-disk image (something many COFFers did with impunity).

The case for/against ELF, and the necessary contortions to upgrade an a.out system to support it, are covered in the ELF-HOWTO and I don't propose to cut/paste them here. The HOWTO should be available in the same place as you found this one.

TOC

ELF shared libraries

To build `libfoo.so` as a shared library, the basic steps look like this:

```
$ gcc -fPIC -c *.c
$ gcc -shared -Wl,-soname,libfoo.so.1 -o libfoo.so.1.0 *.o
$ ln -s libfoo.so.1.0 libfoo.so.1
$ ln -s libfoo.so.1 libfoo.so
$ LD_LIBRARY_PATH=`pwd`:$LD_LIBRARY_PATH ; export LD_LIBRARY_PATH
```

This will generate a shared library called `libfoo.so.1.0`, and the appropriate links for ld (`libfoo.so`) and the dynamic loader (`libfoo.so.1`) to find it. To test, we add the current directory to `LD_LIBRARY_PATH`.

When you're happpy that the library works, you'll have to move it to, say, `/usr/local/lib`, and recreate the appropriate links. The link from `libfoo.so.1` to `libfoo.so.1.0` is kept up to date by `ldconfig`, which on most systems is run as part of the boot process. The `libfoo.so` link must be updated manually. If you are scrupulous about upgrading all the parts of a library (e.g. the header files) at the same time, the simplest thing to do is make `libfoo.so -> libfoo.so.1`, so that ldconfig will keep both links current for you. If you *aren't*, you're setting yourself up to have *all kinds of weird things* happen at a later date. Don't say you weren't warned.

```
$ su
# cp libfoo.so.1.0 /usr/local/lib
# /sbin/ldconfig
# ( cd /usr/local/lib ; ln -s libfoo.so.1 libfoo.so )
```

Version numbering, sonames and symlinks

Each library has a *soname*. When the linker finds one of these in a library it is searching, it embeds the soname into the binary instead of the actual filename it is looking at. At runtime, the dynamic loader will then search for a file with the name of the soname, not the library filename. Thus a library called `libfoo.so` could have a soname `libbar.so`, and all programs linked to it would look for `libbar.so` instead when they started.

This sounds like a pointless feature, but it is key to understanding how multiple versions of the same library can coexist on a system. The de facto naming standard for libraries in Linux is to call the library, say, `libfoo.so.1.2`, and give it a soname of `libfoo.so.1`. If it's added to a `standard' library directory (e.g. `/usr/lib`), `ldconfig` will create a symlink `libfoo.so.1 -> libfoo.so.1.2` so that the appropriate image is found at runtime. You also need a link `libfoo.so -> libfoo.so.1` so that ld will find the right soname to use at link time.

So, when you fix bugs in the library, or add new functions (any changes that won't adversely affect existing programs), you rebuild it, keeping the soname as it was, and changing the filename. When you make changes to the library that would break existing binaries, you simply increment the number in the soname --- in this case, call the new version `libfoo.so.2.0`, and give it a soname of

`libfoo.so.2`. Now switch the `libfoo.so` link to point to the new version and all's well with the world again.

Note that you don't *have* to name libraries this way, but it's a good convention. ELF gives you the flexibility to name libraries in ways that will confuse the pants off people, but that doesn't mean you have to use it.

Executive summary: supposing that you observe the tradition that major upgrades may break compatibility, minor upgrades may not, then link with

```
gcc -shared -Wl,-soname,libfoo.so.major -o libfoo.so.major.minor
```

and everything will be all right.

a.out. Ye olde traditional format

The ease of building shared libraries is a major reason for upgrading to ELF. That said, it's still possible in a.out. Get ftp://tsx-11.mit.edu/pub/linux/packages/GCC/src/tools-2.17.tar.gz and read the 20 page document that you will find after unpacking it. I hate to be so transparently partisan, but it should be clear from context that I never bothered myself :-)

ZMAGIC vs QMAGIC

QMAGIC is an executable format just like the old a.out (also known as ZMAGIC) binaries, but which leaves the first page unmapped. This allows for easier NULL dereference trapping as no mapping exists in the range 0-4096. As a side effect your binaries are nominally smaller as well (by about 1K).

Obsolescent linkers support ZMAGIC only, semi-obsolescent support both formats, and current versions support QMAGIC only. This doesn't actually matter, though, as the kernel can still run both formats.

Your `file' command should be able to identify whether a program is QMAGIC.

File Placement

An a.out (DLL) shared library consists of two real files and a symlink. For the `foo' library used throughout this document as an example, these files would be `libfoo.sa` and `libfoo.so.1.2`; the symlink would be `libfoo.so.1` and would point at the latter of the files. What are these for?

At compile time, `ld` looks for `libfoo.sa`. This is the `stub' file for the library, and contains all exported data and pointers to the functions required for run time linking.

At run time, the dynamic loader looks for `libfoo.so.1`. This is a symlink rather than a real file so that libraries can be updated with newer, bugfixed versions without crashing any application that was using the library at the time. After the new version --- say, `libfoo.so.1.3` --- is completely there, running ldconfig will switch the link to point to it in one atomic operation, leaving any program which had the old version still perfectly happy.

DLL libraries (I know that's a tautology --- so sue me) often appear bigger than their static counterparts.

TOC

They reserve space for future expansion in the form of `holes' which can be made to take no disk spa
A simple `cp` call or using the program `makehole` will achieve this. You can also strip them after building, as the addresses are in fixed locations. **Do not attempt to strip ELF libraries**.

``libc-lite''?

A libc-lite is a light-weight version of the libc library built such that it will fit on a floppy and suffice for all of the most menial of UNIX tasks. It does *not* include curses, dbm, termcap etc code. If your `/lib/libc.so.4` is linked to a lite lib, you are advised to replace it with a full version.

Linking: common problems

Send me your linking problems! I probably won't do anything about them, but I will write them up if I get enough ...

Programs link static when you wanted them shared

> Check that you have the right links for `ld` to find each shared library. For ELF this means a `libfoo.so` symlink to the image, for a.out a `libfoo.sa` file. A lot of people had this problem after moving from ELF binutils 2.5 to 2.6 --- the earlier version searched more `intelligently' for shared libraries, so they hadn't created all the links. The intelligent behaviour was removed for compatibility with other architectures, and because quite often it got its assumptions wrong and caused more trouble than it solved.

The DLL tool `mkimage' fails to find libgcc, or

> As of `libc.so.4.5.x` and above, libgcc is no longer shared. Hence you must replace occurrences of `` `-lgcc' `` on the offending line with `` `gcc -print-libgcc-file-name` `` (complete with the backquotes).
>
> Also, delete all `/usr/lib/libgcc*` files. This is important.

`__NEEDS_SHRLIB_libc_4 multiply defined` messages

> are another consequence of the same problem.

``Assertion failure'' message when rebuilding a DLL ?

> This cryptic message most probably means that one of your jump table slots has overflowed because too little space has been reserved in the original `jump.vars` file. You can locate the culprit(s) by running the `` `getsize' `` command provided in the tools-2.17.tar.gz package. Probably the only solution, though, is to bump the major version number of the library, forcing it to be backward incompatible.

`ld: output file needs shared library libc.so.4`

> This usually happens when you are linking with libraries other than libc (e.g. X libraries), and use the `-g` switch on the link line without also using `-static`.
>
> The `.sa` stubs for the shared libraries usually have an undefined symbol `_NEEDS_SHRLIB_libc_4` which gets resolved from the `libc.sa` stub. However with `-g` you end up linking with `libg.a` or `libc.a` and thus this symbol never gets resolved, leading to the

above error message.

In conclusion, add `-static` when compiling with the `-g` flag, or don't link with `-g`. Quite often you can get enough debugging information by compiling the individual files with `-g`, and linking *without* it.

7. Dynamic Loading

This section is a tad short right now; it will be expanded over time as I gut the ELF howto

7.1 Concepts

Linux has shared libraries, as you will by now be sick of hearing if you read the whole of the last section at a sitting. Some of the matching-names-to-places work which was traditionally done at link time must be deferred to load time.

7.2 Error messages

Send me your link errors! I won't do anything about them, but I might write them up ...

`can't load library: /lib/libxxx.so, Incompatible version`

(a.out only) This means that you don't have the correct major version of the xxx library. No, you can't just make a symlink to another version that you do have; if you are lucky this will cause your program to segfault. Get the new version. A similar situation with ELF will result in a message like

```
ftp: can't load library 'libreadline.so.2'
```

`warning using incompatible library version xxx`

(a.out only) You have an older minor version of the library than the person who compiled the program used. The program will still run. Probably. An upgrade wouldn't hurt, though.

7.3 Controlling the operation of the dynamic loader

There are a range of environment variables that the dynamic loader will respond to. Most of these are more use to `ldd` than they are to the average user, and can most conveniently be set by running ldd with various switches. They include

- `LD_BIND_NOW` --- normally, functions are not `looked up' in libraries until they are called. Setting this flag causes all the lookups to happen when the library is loaded, giving a slower startup time. It's useful when you want to test a program to make sure that everything is linked.
- `LD_PRELOAD` can be set to a file containing `overriding' function definitions. For example, if you were testing memory allocation strategies, and wanted to replace `malloc', you could write your replacement routine, compile it into `malloc.o` and then

```
$ LD_PRELOAD=malloc.o; export LD_PRELOAD
```

```
$ some_test_program
```

LD_ELF_PRELOAD and LD_AOUT_PRELOAD are similar, but only apply to the appropriate type of binary. If LD_*something*_PRELOAD and LD_PRELOAD are set, the more specific one is used.

- LD_LIBRARY_PATH is a colon-separated list of directories in which to look for shared libraries. It does *not* affect ld; it only has effect at runtime. Also, it is disabled for programs that run setuid or setgid. Again, LD_ELF_LIBRARY_PATH and LD_AOUT_LIBRARY_PATH can also be used to direct the search differently for different flavours of binary. LD_LIBRARY_PATH shouldn't be necessary in normal operation; add the directories to /etc/ld.so.conf/ and rerun ldconfig instead.
- LD_NOWARN applies to a.out only. When set (e.g. with LD_NOWARN=true; export LD_NOWARN) it stops the loader from issuing non-fatal warnings (such as minor version incompatibility messages).
- LD_WARN applies to ELF only. When set, it turns the usually fatal ``Can't find library'' messages into warnings. It's not much use in normal operation, but important for ldd.
- LD_TRACE_LOADED_OBJECTS applies to ELF only, and causes programs to think they're being run under ldd:

```
$ LD_TRACE_LOADED_OBJECTS=true /usr/bin/lynx
        libncurses.so.1 => /usr/lib/libncurses.so.1.9.6
        libc.so.5 => /lib/libc.so.5.2.18
```

7.4 Writing programs with dynamic loading

This is very close to the way that Solaris 2.x dynamic loading support works, if you're familiar with that. It is covered extensively in H J Lu's ELF programming document, and the dlopen(3) manual page, which can be found in the ld.so package. Here's a nice simple example though: link it with -ldl

```
#include <dlfcn.h>
#include <stdio.h>

main()
{
  void *libc;
  void (*printf_call)();

  if(libc=dlopen("/lib/libc.so.5",RTLD_LAZY))
  {
    printf_call=dlsym(libc,"printf");
    (*printf_call)("hello, world\n");
  }

}
```

8. Contacting the developers

8.1 Bug reports

Start by **narrowing the problem down**. Is it specific to Linux, or does it happen with gcc on other systems? Is it specific to the kernel version? Library version? Does it go away if you link static? Can you trim the program down to something **short** that demonstrates the bug?

Having done that, you'll know what program(s) the bug is in. For GCC, the bug reporting procedure is explained in the info file. For ld.so or the C or maths libraries, send mail to `linux-gcc@vger.rutgers.edu`. If possible, include a short and self-contained program that exhibits the bug, and a description both of what you want it to do, and what it actually does.

8.2 Helping with development

If you want to help with the development effort for GCC or the C library, the first thing to do is join the `linux-gcc@vger.rutgers.edu` mailing list. If you just want to see what the discussion is about, there are list archives at http://homer.ncm.com/linux-gcc/. The second and subsequent things depend on what you want to do!

9. The Remains

9.1 The Credits

> Only presidents, editors, and people with tapeworms have the right to use the editorial ``we".

(Mark Twain)

This HOWTO is based very closely on Mitchum DSouza's GCC-FAQ; most of the information (not to mention a reasonable amount of the text) in it comes directly from that document. Instances of the first person pronoun in this HOWTO could refer to either of us; generally the ones that say ``I have not tested this; don't blame me if it toasts your hard disk/system/spouse'' apply to both of us.

Contributors to this document have included (in ASCII ordering by first name) Andrew Tefft, Axel Boldt, Bill Metzenthen, Bruce Evans, Bruno Haible, Daniel Barlow, Daniel Quinlan, David Engel, Dirk Hohndel, Eric Youngdale, Fergus Henderson, H.J. Lu, Jens Schweikhardt, Kai Petzke, Michael Meissner, Mitchum DSouza, Olaf Flebbe, Paul Gortmaker, Rik Faith, Steven S. Dick, Tuomas J Lukka, and of course Linus Torvalds, without whom the whole exercise would have been pointless, let alone impossible :-)

Please do not feel offended if your name has not appeared here and you have contributed to this document (either as HOWTO or as FAQ). Email me and I will rectify it.

9.2 Translations

At this time, there are no known translations of this work. If you wish to produce one, please go right ahead, but do tell me about it! The chances are (sadly) several hundred to one against that I speak the language you wish to translate to, but that aside I am happy to help in whatever way I can.

dan@detached.demon.co.uk. My PGP public key (ID 5F263625) is available from my web pages, if you feel the need to be secretive about things.

9.4 Legalese

All trademarks used in this document are acknowledged as being owned by their respective owners.

This document is copyright (C) 1996 Daniel Barlow `<dan@detached.demon.co.uk>` It may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must

be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at `linux-howto@sunsite.unc.edu` via email.

---

10. Index

Entries starting with a non-alphabetical character are listed in ASCII order.

- `-fwritable-strings` 39 56
- /lib/cpp 16
- a.out 1
- `ar` 10
- `as` 8
- <asm/*.h> 19
- `atoi()` 40
- `atol()` 41
- binaries too big 63 65 77
- chewing gum 3
- `cos()` 68
- debugging 59
- `dlopen()` 82
- `dlsym()` 83
- documentation 4
- EINTR 52
- elf 0 71
- `execl()` 57
- `fcntl` 47
- `FD_CLR` 44
- `FD_ISSET` 45
- `FD_SET` 43
- `FD_ZERO` 46
- `file` 2
- <float.h> 20
- gcc 6
- `gcc -fomit-frame-pointer` 61
- `gcc -g` 60

- gcc -v 14
- gcc, bugs 15 28 29 84
- gcc, flags 13 25 26
- gdb 64
- header files 17
- interrupted system calls 51
- `ld` 9
- `LD_*` environment variables 80
- ldd 81
- libc 7
- `libg.a` 62
- libgcc 79
- <limits.h> 21
- lint 58
- <linux/*.h> 18
- manual pages 5
- `<math.h>` 70
- maths 69
- `mktemp()` 55
- optimisation 27
- QMAGIC 76
- segmentation fault 30 54
- segmentation fault, in GCC 33
- select() 50
- `SIGBUS` 34
- `SIGEMT` 35
- `SIGIOT` 36
- SIGSEGV 31 53
- SIGSEGV, in gcc 32
- `SIGSYS` 38
- `SIGTRAP` 37
- `sin()` 67
- soname 73

- `sprintf()` 42
- statically linked binaries, unexpected 66 78
- <stdarg.h> 23
- <stddef.h> 24
- `strings` 11
- `<sys/time.h>` 48
- `<unistd.h>` 49
- <varargs.h> 22
- version numbers 12 74
- weird things 72
- ZMAGIC 75

German HOWTO

Winfried Trümper <winni@xpilot.org>

v1.6, 19 March 1997

Dieses Dokument enthält Tips und Informationen für deutschsprachige Linux-Benutzer. Note to English readers: this document explains the German-specific aspects of Linux.

1. Über das German-HOWTO

- 1.1 Benutzung auf eigene Gefahr
- 1.2 Zielgruppe
- 1.3 Thematischer Umfang
- 1.4 Verfügbarkeit
- 1.5 Rückkopplung
- 1.6 Urheberrechte
- 1.7 Danksagung

2. Rüstzeug und generelle Bemerkungen

- 2.1 Warum sind deutsche Anpassungen notwendig?
- 2.2 Notation
- 2.3 Empfohlene Begleitliteratur
- 2.4 Kurzes Glossar

3. Menschen

- 3.1 Benutzergruppen
- 3.2 Linux-Supporter
- 3.3 Online Chat (IRC)
- 3.4 Newsgroups

TOC

4. Dokumentation

- 4.1 Deutsches Linux HOWTO Projekt (DLHP)
- 4.2 Bücher
- 4.3 Deutschsprachige Manual-Pages
- 4.4 Diverses

5. Das System

- 5.1 Konsole
- 5.2 Zeichensatz für die Textkonsolen
- 5.3 Tastenbelegung von `XFree86'
- 5.4 Zeitzonen

6. Anwendungssoftware

- 6.1 Außer Konkurenz ...
- 6.2 Die Ausreißer
- 6.3 Weiterführende Tips

7. Textdateien

- 7.1 Das Papierformat DIN A4
- 7.2 Konvertierung von DOS- und Mac-Texten nach Linux
- 7.3 Drucken
- 7.4 PostScript

8. Programmierung

- 8.1 National Language Support (NLS)
- 8.2 Übersetzung von GNU-Programmen
- 8.3 ncurses
- 8.4 Erzeugung von PostScript

9. Anhang

- 9.1 Erklärende Übersetzung einiger technischer Begriffe

1. Über das German-HOWTO

1.1 Benutzung auf eigene Gefahr

Text, Abbildungen und Programme in diesem HOWTO wurden mit großer Sorgfalt erarbeitet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden HOWTO enthaltenen Angaben mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Kurz: Benutzung auf eigene Gefahr.

1.2 Zielgruppe

Das German-HOWTO wendet sich an Linux-Benutzer, deren Linux-Installation nicht oder nur teilweise an "deutsche" Begebenheiten angepaßt ist. Die Forderung der Benutzer sollte natürlich sein, daß alle Linux-Distributionen länderspezifische Einstellungen von sich aus berücksichtigen. Schließlich ist dies der Zweck der Distributionen: anhand einfacher Informationen das System weitmöglichst zu konfigurieren, ohne daß der Benutzer erst Guru werden muß.

Kurz: würden alle Distributionen ihre Hausaufgaben machen, wäre diese HOWTO erheblich kürzer.

1.3 Thematischer Umfang

Dieses HOWTO beschränkt sich ausschließlich auf die Beschreibung der für Deutschland spezifischen Konfiguration von Linux und seinen Programmen.

Es wird weder eine allgemeine deutsche Einführung in Linux gegeben, noch wird die vorhandene Original-Dokumentation durch dieses HOWTO überflüssig.

Was ist neu in dieser Version?

Version 1.4

Korrekturen an 1.3.

Version 1.3

Komplette Neufassung des HOWTOs. Jetzt auch ausdrücklich für den gesamten deutschsprachigen Raum.

Version 1.2

?

Version 1.1

Konvertierung in das Format linuxdoc-sgml.

1.4 Verfügbarkeit

Dieses HOWTO ist offizieller Teil des `Linux Documentation Projects' (LDP) und somit wie alle anderen Dokumente des LPD erhältlich. Nähere Informationen findet man unter den URLs

http://www.jura.uni-sb.de/LDP/

http://www.uni-paderborn.de/Linux/mdw/

oder im Verzeichnis `/usr/doc/HOWTO/` auf der eigenen Festplatte.

1.5 Rückkopplung

Korrekturen von Tippfehlern und Verbesserungsvorschläge direkt an mich schicken: `winni@xpilot.org`.

Einstellungen aus MAKEFILEs nehme ich nicht mehr entgegen: sie sind direkt an den Hersteller der Linux-Distribution zu richten. Wo kämen wir hin, wenn das German HOWTO zur allgemeinen Sammelstelle für Bug-Fixes würde?

Man verschone mich mit über das HOWTO hinausgehende Fragen wie z.B. `Warum stürzt ispell ab?'. Ich stehe aus gutem Grund **nicht** auf der Liste der deutschen Linux-Unterstützer und beantworte auch keine Fragen per e-mail.

1.6 Urheberrechte

Copyright (1994,1995,1996,1997) Winfried Trümper `<winni@xpilot.org>`

Das German-HOWTO darf im ganzen oder teilweise vervielfältigt und vertrieben werden, sowohl in gedruckter als auch in elektronischer Form, solange dieses Copyright beibehalten wird. Kommerzieller Vertrieb ist in diesem Sinne ausdrücklich erlaubt, solange höchstens (anteilige) Kosten für die Datenträger berechnet werden.

Note to English readers:

The German-HOWTO may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as the above copyright notice is retained on all copies. Commercial redistribution under the GPL is allowed and encouraged.

1.7 Danksagung

TOC

Folgende Personen haben durch Vorschläge am German-HOWTO mitgewirkt:

Bernhard Aceman <a2607425@athena.rrz.Uni-Koeln.DE>
Andreas Bagge <Andreas_Bagge@h2.maus.de>
Andries Brouwer <Andries.Brouwer@cwi.nl>
Karl Brodowsky <bk1@aixcomp2.urz.uni-heidelberg.de>
Bernhard Buettner <ben@dl6rai.muc.de>
Gert Doering <gert@greenie.muc.de>
Karl Eichwalder <ke@ke.central.de>
David Frey <David.Frey@eos.lugs.ch>
Jochen Hein <jochen.hein@delphi.central.de>
Michael Hinz <michael@fagmed.uit.no>
Ulli Horlacher <framstag@belwue.de>
"Markus G. Kuhn" <kuhn@cs.purdue.edu>
Detlef Lannert <lannert@uni-duesseldorf.de>
Anselm Lingnau <lingnau@tm.informatik.uni-frankfurt.de>
Nat Makarevitch <nat@nataa.frmug.fr.net>
Sebastian Matschke
Frerk Meyer <frerk@educat.hu-berlin.de>
Reiner Miericke <reiner@schildi.xnc.com>
Uwe Münch <muench@ph-cip.uni-koeln.de>
Nils Naumann <naumann@unileoben.ac.at>
Danilo Rene Neuber <daneuber@sbox.tu-graz.ac.at>
Sven Neuhaus <sven%fuzzy@fuzzy.ping.de>
Thomas Ploss <plossts@cip01.mathematik.uni-stuttgart.de>
Roland Rosenfeld <roland@spinnaker.rhein.de>
Sven Rudolph <sr1@irz301.inf.tu-dresden.de>
Dietrich Schmidt <diet@sfb256.iam.uni-bonn.de>
Ingo Schroeck <ioschroe@track.informatik.uni-stuttgart.de>
Juergen Steiner <js@barbar.augusta.de>
Guenter Stueck <guenter@gst0hb.north.de>
Martin Schulze <joey@finlandia.infodrom.north.de>
Jan Vroonhof <vroonhof@math.ethz.ch>

2. Rüstzeug und generelle Bemerkungen

2.1 Warum sind deutsche Anpassungen notwendig?

Linux wird von einer weltweiten Gemeinde von Hackern im Internet entwickelt. Obwohl Linux ursprünglich aus Finnland kommt, ist Englisch gemeinsame Sprache und Ausgangspunkt bei der Programmierung sind meist amerikanische Standards (Papierformat "letter", Zeichensatz ohne Umlaute, usw.).

Gerade weil die Linux-Gemeinde weltweit operiert, sind landesspezifische Einstellungen an vielen Stellen vorgesehen und müssen lediglich aktiviert werden. Das ist schon alles.

2.2 Notation

Tasten werden durch ihre Aufschrift in spitzen Klammern markiert.

2.3 Empfohlene Begleitliteratur

Das Linux Configuration-HOWTO (to appear) erläutert die Handhabung der erwähnten Konfigurationsdateien.

Im Linux Keyboard- und Keystroke-HOWTO findet man alle "Tricks" für die Tastatur.

2.4 Kurzes Glossar

ASCII

ist ein 7-bit Zeichensatz und enthält 2^7 = 2*2*2*2*2*2*2 = 128 Zeichen, was gerade mal für a-z, A-Z, 0-9 und einige wichtige Zeichen ausreicht. Das amerikanische Alphabet halt.

8-bit-clean

ein 8-bit Zeichensatz umfaßt 128*2 = 256 Zeichen und kann einige Sonderzeichen aufnehmen. Ist ein Programm in der Lage mit einen 8-bit Zeichensatz umzugehen, so bezeichnet man es auch als `8-bit clean'. Ist ein Programm nicht 8-bit clean, so erscheinen oft anstelle der Umlaute `äöü' die Zeichen `dv|'.

Man beachte den subtilen Unterschied zwischen `8bit clean' und `localized': Ein 8bit cleanes Programm braucht nichts von seiner Locale (und dem verwendeten Zeichensatz!) zu wissen; es muss ``nur'' alle 256 Zeichen bearbeiten können. Handkehrum kann ein Programm localized sein,

aber trotzdem nicht 8bit clean sein, da es gewisse Zeichen verschluckt (z.B. alle Control-Code

`ISO-latin1' oder `ISO-8859-1'

Internationaler 8-bit Zeichensatz, enthält Umlaute und viele Sonderzeichen für die `westeuropäischen' Sprachen. (Die osteuropäischen und weitere haben andere Endziffern.)

Unicode

Unicode ist ein Standard, der Zeichen aus den unterschiedlichsten Sprachen der Länder dieser Welt enthält. Dazu werden die Zeichenpositionen in einem 32-bit Zeichensatz definiert, was noch längst nicht abgeschlossen ist. Gängigere Zeichensätze (wie die Zeichen der europäischen Sprachfamilien) sind als 16-bit Untergruppen festgelegt, die in kompatibler Weise als einzelne 16-bit Unicode-Zeichensätze verwendet werden können. Wenn hier von einem *Unicode*-Zeichensatz die Rede ist, ist die 16-bit Untergruppe der europäischen Sprachfamilien gemeint.

POSIX

POSIX steht für `Portable Operating System Interface for UniX'. Normierte Schnittstellenbeschreibung zu Kernel- und libc-Funktionen (POSIX.1) resp. zu den Grundutilities (sh, ed, vi, awk, cut, ...) (POSIX.2).

Locale

Ist die formale Beschreibung eines Teils der kulturellen Eigenheiten für ein Land oder einen Spracheraum und die vom jeweiligen Programm benötigten übersetzten Texte.

Internationalization

Veränderung eines Programms zur Unterstützung mehrerer Sprachen.

i18n

Abkürzung für *internationalization*. Da viele Leute zu faul waren dieses lange Wort immer wieder auszuschreiben, wurden einfach der erste und letzte Buchstabe belassen und der Zwischenraum durch die Anzahl fehlender Buchstaben ersetzt.

Localisation

Mit *localisation* wird der Prozeß bezeichnt, einem bereits internationalisierten Programm alle benötigten Informationen zur Verfügung zu stellen, um den nationalen Besonderheiten, bezüglich der Sprache und kulturellen Umfeld zu entsprechen. Kurz gesagt, i18n ist der eigentliche Übersetzungsprozeß.

Der Nutzer kann das lokalisieren eines Programms erreichen, indem er besonderen Umgebungsvariablen geeignete Werte zuweist, welche die zu benutzende `locale' angeben.

l10n

Abkürzung für *localization*, die wie i18n entstanden ist.

3. Menschen

3.1 Benutzergruppen

In vielen Städten haben sich Linux-Benutzer zu sogenannten *Linux User Groups* (LUG) zusammengetan. Diese Benutzergruppen sind offen für jeden und halten regelmäßige Treffen ab. Sie sind der ideale Treffpunkt für all diejenigen, die sich nicht nur im Cyberspace mit anderen Linuxern austauschen wollen. Eine Übersicht befindet sich unter der URL

http://www.schwaben.de/linux/LUG_in_de.html

3.2 Linux-Supporter

Eine Liste mit Ansprechpartnern, die Anfängern in einem begrenzten Rahmen kostenlosen Hilfestellung geben, findet sich unter der URL:

http://localhost.ruhr.de/Linux-Support.html

Ziel ist die Verbreitung von Linux als Betriebssystem zu fördern.

3.3 Online Chat (IRC)

Im IRC-Kanal #LinuxGER halten sich einige deutsche Linux-Aktivisten auf. Nähere Informationen unter der URL:

http://www.infodrom.north.de/Linux/LinuxGER/

3.4 Newsgroups

Früher von mir als kostenlose Hotlines degradiert, habe ich die Newsgroups in dieser Version anders eingeordnet. *Hotline* hat immer den Klang einer Stelle, bei der man auch ohne Lesen des Handbuchs Fragen stellen kann. Obwohl einige Handbücher nun wirklich nicht das gelbe vom Ei sind, sollte man die newgroups mit übermäßigen Fragen verschonen:

de.comp.os.linux.hardware

de.comp.os.linux.misc

de.comp.os.linux.x

de.comp.os.linux.networking

Stattdessen sind die Newsgroup ein prima Forum, in denen man die nicht ganz alltäglichen Dinge diskutieren kann.

4. Dokumentation

4.1 Deutsches Linux HOWTO Projekt (DLHP)

Das Deutsche Linux HOWTO Projekt beschäftigt sich mit der Übersetzung der originalen Linux HOWTOs ins Deutsche. Unter der URL

http://www.tu-harburg.de/~semb2204/dlhp/

kann man eine Liste aller übersetzten HOWTOs und nähere Informationen zum Projekt abrufen.

4.2 Bücher

Mittlerweile gibt es sehr viele deutschsprachige Bücher zu Linux, weshalb ich die Übersicht aus diesem HOWTO herausgenommen habe.

Martin Schulze `<joey@finlandia.infodrom.north.de>` unterhält eine Liste mit deutschen Büchern rund um Linux, die unter der URL:

http://www.infodrom.north.de/Linux/Buecher/

erhältlich ist und die in der Newsgroup de.comp.os.linux.misc regelmäßig gepostet wird.

4.3 Deutschsprachige Manual-Pages

Eine Gruppe von Freiwilligen hat sich die Übersetzung der englischen Manual-Pages vorgenommen. Obwohl bereits ein ansehnlicher Teil übersetzt ist, werden weiterhin Übersetzer gesucht. Details unter der URL:

http://www.Infodrom.North.DE/Linux/manpages-de/

4.4 Diverses

GNU emacs reference card

Eine deutsche Version der äußerst nützlichen Referenzkarte zum Texteditor `emacs' findet man unter der URL:

http://www.inf.tu-dresden.de/~sr1/projects/emacs-refcard/index.html

Crash-Kurs Linux

Wer gerne knappe Bescheibungen mag, sollte einen Blick in den von mir geschriebenen `Crash-Kurs Linux' werfen.

http://www.uni-koeln.de/themen/linux/ckl/

TOC

5. Das System

5.1 Konsole

Die Konsole ist die Tastatur und der Bildschirm eines Linux-Systems; in der Regel also der VGA-Monitor plus Tastatur.

Linux täuscht den Programmen mehr als eine Konsole vor und der Benutzer kann dann mittels Tastenkombination <Alt-Ctrl-F1> - <Alt-Ctrl-F8> zwischen diesen hin- und herblenden. Die vorgetäuschten Konsolen nennt man *virtuelle Konsolen*.

```
 _________________|                              _________________|
|                 |                             |                 |
|  Escape the     |<--  setfont    loadkeys  -->|^1234567890ß  123 |
|  Gates of hell. |                             | qwertzuiopü+ 567 |
|                 |=======+            +=======| asdfghjklöä# 789 |
|  Use Linux.     |       |            |       | <yxcvbnm,.-  0,  |
|                 |       |            |       |_________________|
|_________________|       |            |
        MMM               |            |
     MMMMMMMMM            |            |             +---<---< Internet
                          |            |             |
                  +-------*------------*------+      |
                  |        \          /       |      |      X11 ------+
                  |         \        /        |      |       |        |
                  |  Betriebssystemkern Linux |   telnetd  xterm      |
Alt+Ctrl F1-F8 ----------------/ <->          |       \\   ///        |
als Schalter      |           /               |        \\ ///         |
                  |     /    |          \     +-------+. . .+-------+ |
                  |    /     |           \    | ptyp1 |     | ptyef | |
                  |......+......+ . . . +......+.......+ . . +.......| |
                  | tty0 | tty1 | . . . | tty8 | ttyp1 | . . | ttyef | |
                  +------+------+ - - - +------+-------+ - - +-------+ |
                    |      |              ^             |             |
                bash      emacs           |             less          |
                                          |                           |
                                           `---------------------------'
```

Die nachfolgend beschriebenen Einstellungen betreffen alle virtuellen Konsolen gleichzeitig. Welches die "richtige" Datei für die Kommandos ist, darüber kann nur das Handbuch der eigenen Distribution Auskunft geben. (Tip: das Configuration HOWTO kann in vielen Fällen helfen.)

Weiterführende Literatur: Keystroke-HOWTO (engl.).

Es ist unter Linux übrigens *nicht* notwendig, die virtuellen Bildschirme mit dem Befehl `stty pass8' in den 8-bit Modus zu schalten - das sind sie nämlich von Hause aus.

TOC

Tastenbelegung der Textkonsolen

Für die unterschiedlichen Tastaturen der verschiedenen Länder liegen im Verzeichnis `/usr/lib/kbd/keytables' vorgefertigte Konfigurationsdateien. Mit dem Kommando `loadkeys' kann man eine solche Datei in den Linux-Kernel laden und dadurch den Tasten bestimmte Zeichen zuordnen.

Um Mißverständnissen vorzubeugen: man ordnet den Tasten keine Bedeutungen zu, sondern nur bestimmte Zeichen. Dadurch ist sichergestellt, daß die Tastatur auch beim Arbeiten mit entfernten Rechnern funktioniert, denn Zeichen kann man über jede Art von Netzwerkverbindung übertragen. Dagegen wäre eine direkte Abfrage der Tastatur-Hardware (so wie unter DOS) über ein Netzwerk nicht möglich - es ist ein Widerspruch in sich.

Die Bedeutung eines Zeichens (und damit einer Taste) kann von Programm zu Programm leicht unterschiedlich sein und die Kunst liegt in der effektiven, bzw. komfortablen Konfiguration der Bedeutung. Die länderspezifischen Aspekte werden im kommenden Kapitel behandelt.

Deutschland

Im einfachsten Falle werden Benutzer einer deutschen Tastatur mit der Belegung `de-latin1-nodeadkeys.map' zufrieden sein:

Beispiel:

```
loadkeys   de-latin1-nodeadkeys.map
```

Schweiz

Geeignete Tastenbelegungen für die Schweiz sind `sf-latin1.map' (swiss-french) und `sg-latin1.map' (swiss-german).

In der Deutschschweiz gibt es 2 verschiedene Tastaturlayouts, bei denen wenige Tasten verschieden beschriftet sind (eckige Klammern, @-Zeichen an verschiedenen Orten).

Östereich

?

Länderübergreifende Einstellungen

Die Tastaturtabellen sind Textdateien, die sich mit dem persönlichen Lieblingseditor bearbeiten lassen. Um z.B. die französischen Anführungszeichen auf <AltGr .> und <AltGr -> zu legen, ergänzt man an den entsprechenden Stellen:

```
keycode 51 = comma semicolon guillemotright
keycode 52 = period colon guillemotleft

control keycode 105 = Meta_b
control keycode 106 = Meta_f
```

Die letzten beiden Einträge ermöglichen wortweises Springen auf der Kommandozeile mit Strg-Cursortaste. Als Anregung für eigene Ergänzungen hier eine Tabelle mit allen (?) symbolischen Namen:

```
"  quotedbl     '  apostrophe   `  grave        ~  asciitilde
^  asciicircum  _  underscore   !  exclam       ?  question
```

```
#  numbersign   $  dollar       %  percent      &  ampersand
|  bar          @  at           +  plus         -  minus
*  asterisk     /  slash        \  backslash    .  period
,  comma        :  colon        ;  semicolon    <  less
=  equal        >  greater      (  parenleft    )  parenright
[  bracketleft  ]  bracketright {  braceleft    }  braceright

¡  exclamdown   ¢  cent         £  sterling     ¤  currency
¥  yen          ¦  brokenbar    §  section      "  diaeresis
©  copyright    ª  ordfeminine  <  guillemotleft
¬  notsign      -  hyphen       ®  registered
¯  macron       °  degree       ±  plusminus    ²  twosuperior
³  threesuperior                ´  acute        µ  mu
¶  paragraph    ·  periodcentered               ¸  cedilla
¹  onesuperior  º  masculine    >  guillemotright
¼  onequarter   ½  onehalf      ¾  threequarters
¿  questiondown À  Agrave       Á  Aacute       Â  Acircumflex
Ã  Atilde       Ä  Adiaeresis   °A  Aring        Æ  AE
Ç  Ccedilla     È  Egrave       É  Eacute       ^E  Ecircumflex
Ë  Ediaeresis   Ì  Igrave       Í  Iacute       Î  Icircumflex
Ï  Idiaeresis   Ð  ETH          Ñ  Ntilde       Ò  Ograve
Ó  Oacute       Ô  Ocircumflex  Õ  Otilde       Ö  Odiaeresis
×  multiply     Ø  Ooblique     Ù  Ugrave       Ú  Uacute
^U  Ucircumflex  Ü  Udiaeresis   Ý  Yacute       |O  THORN
ß  ssharp       à  agrave       á  aacute       â  acircumflex
ã  atilde       ä  adiaeresis   å  aring        æ  ae
ç  ccedilla     è  egrave       é  eacute       ê  ecircumflex
ë  ediaeresis   ì  igrave       í  iacute       î  icircumflex
ï  idiaeresis   xo  eth          ñ  ntild        ò  ograve
ó  oacute       ô  ocircumflex  õ  otilde       ö  odiaeresis
÷  division     ø  oslash       ù  ugrave       ú  uacute
û  ucircumflex  ü  udiaeresis   ý  yacute       |o  thorn
"y  ydiaeresis

nul     Tab     Escape  space   BackSpace
Insert  Remove  Home    End     PageUp
Down    Left    Right   Up      PageDown
Macro   Help    Do      Pause   Linefeed
```

Die Tabelle nicht ganz korrekt, da einige Zeichen vom Formatierungswerkzeug nicht verarbeitet werden. Wir arbeiten daran.

Die keytables(5), die dumpkeys(1) und loadkeys(1) Manpages erläutern welche Symbole gültig sind, insbesondere

```
dumpkeys --long-info
```

erzeugt eine Liste aller bekannten Zeichennamen und Modifier.

TOC

Deadkeys

Mit *Deadkey* wird eine Taste bezeichnet, bei deren Betätigung das Zeichen zunächst zurückgehalten wird. Das nachfolgend eingegebene Zeichen wird quasi über das zuerst eingegeben Zeichen gedruckt, falls die Kombination Sinn ergibt. Ansonsten werden beide Zeichen nacheinander ausgegeben.

Beispiel: ist die Tilde als Deadkey konfiguriert, so ergibt <~> <a> ein ã (a mit Tilde darüber).

Compose

Alternativ zu den Deadkeys kann man auch mit der <Compose>-Taste arbeiten: sie druckt zwei nacheinander eingetippte Zeichen übereinander. So ergibt die Tastenfolge <Compose> <~> <a> ein ã, während ein <~> <a> weiterhin ~a anzeigt.

Um <Compose> auf die an vielen neuen Tastaturen vorzufindenden zusätzlichen Tasten zu legen, kann man z.B. konfigurieren:

```
keycode 127 = Compose
```

Bei mir trägt diese Taste das Symbol "Quadrat mit Pfeil in der linken oberen Ecke". Die Taste mit dem Fenster-Symbol (früher: Diamant-Symbol oder *Meta*) hat die Nr. 125.

Unicode

Bei Verwendung eines *Unicode*-Zeichensatz sollte man auch die Tastatur in den Unicode-Modus schalten:

```
kbd_mode -u
```

5.2 Zeichensatz für die Textkonsolen

Damit die länderspezifischen Zeichen überhaupt auf dem Bildschirm dargestellt werden können, muß der entsprechende Zeichensatz für den Bildschirm geladen werden. Akzeptable Ergebnisse erzielt man in der Regel mit `lat1u-16.psf', einem 16-Punkt hohen UNIcode-Zeichensatz. Der zugehörige Befehl zum Laden sieht z.B. so aus:

```
setfont /usr/lib/kbd/consolefonts/lat1u-16.psf
```

Die Details sind wiederum der Dokumentation der eigenen Distribution zu entnehmen.

Unicode

Um den *Unicode*-Zeichensatz für die Anwendungsprogramme freizugeben muß man die Zeichenfolge `ESC ( K` an den virtuellen Bildschirm senden. Dies kann z.B. durch die Befehlsfolge

```
for i in  1 2 3 4 5 6 7 8
do
    echo -n -e "\033(K" > /dev/tty$i
done
```

geschehen. Diese Lösung ist allerdings aus verschiedenen Gründen sub-optimal.

5.3 Tastenbelegung von `XFree86'

TOC

`XFree86' ermöglicht den Betrieb von Programmen, die das `X Window System' benötigen. In Verbindung mit einem sogennanten `Windowmanager', der das Look&Feel bestimmt, stellt es die grafische Benutzeroberfläche unter Linux zur Verfügung.

Oft wird empfohlen, die Tastenbelegung des XFree86 durch die Datei `~/.Xmodmap` oder mit dem Befehl `xmodmap` zu konfigurieren. Dies ist nicht erforderlich, sondern unnötig kompliziert. Tatsächlich ist nur ein Eintrag in der Konfigurationsdatei des XFree86 notwendig, um die <AltGr> Taste zu aktivieren. Die Vorgehensweise unterscheidet sich für die verschiedenen Versionen:

Version 2.1:

Konfigurationsdatei: `/usr/X11/lib/X11/Xconfig`

Mit Ausnahme der in der Tastaturbelegung vereinbarten Deadkeys werden die Tasten genau wie auf der Textkonsole belegt. Die Deadkeys (s.o.) sind unter Version 2.1 tatsächlich `dead' - aber leider ganz, nicht nur `halb'. Die Freigabe der Alt-Tasten muß manuell vorgenommen werden, dazu trägt man im Abschnitt *Keyboard* ein:

```
LeftAlt     Meta
RightAlt    ModeShift
```

Version 3.1

Konfigurationsdateien: `/etc/X11/XF86Config` und `/etc/XF86Config`

Die Tastatur ist genau wie auf der Konsole belegt, auch die Deadkeys und <Compose> funktionieren (falls als solche konfiguriert).

Um allerdings in den Genuß der <Alt >- und <AltGr >-Tasten zu kommen, muß man bei der Erzeugung der Konfigurationsdatei (Programm `xf86config`) folgende Frage mit `y' (yes) beantworten:

```
Do you want to enable these bindings for the Alt keys?
```

Alternativ kann man die entsprechenden Einstellungen auch per Hand im Abschnitt `Section "Keyboard"` vornehmen:

```
LeftAlt     Meta
RightAlt    ModeShift
```

`RightAlt ModeShift' aktiviert die <AltGr>-Taste und `LeftAlt Meta' sorgt für eine Emulation der <Meta>-Taste (benötigen z.B. die Programme `emacs` und `bash`).

Version 3.2 und höher:

Konfigurationsdateien: `/etc/X11/XF86Config` und `/etc/XF86Config`

Hat man eine Konfigurationsdatei der Version 3.1 oder will man es sich besonders einfach machen, dann verfährt man wie oben beschrieben und ergänzt im Abschnitt `Section "Keyboard"` lediglich

```
XkbDisable
```

TOC

Dadurch wird die unter Linux etwas kompliziert anmutende Xkbd-Technik abgeschaltet und das X liest stattdes die Tastaturbelegung der Textkonsolen (auf diese Weise braucht man nur eine Sache zu konfigurieren). Äquiva kann man bei der Erzeugung der Konfigurationsdatei (Programm `xf86config`) folgende Frage mit `n' (no) beantworten:

```
Do you want to use XKB?
```

Beantwortet man diese Frage mit `y' (yes), dann sind in der Konfigurationsdatei folgende Einträge zu entkommentieren (d.h. das `#'-Zeichen ist zu entfernen):

```
#    XkbLayout   "de"
#    XkbVariant  "nodeadkeys"
```

Alle Versionen:

Sondermodelle fertigt man mit dem Befehl

```
xmodmap -pke > ~/.Xmodmap
```

und einem Editor nach Wahl an. Die globale Konfigurationsdatei ist `/etc/X11/Xmodmap`. Leider gibt es keine empfehlenswerte Software zur komfortableren Konfiguration der Tastatur (`xkeycaps` ist katastrophal in Sachen Benutzerführung).

Empfehlenswert für die Xmodmap sind etwa

```
! Freigabe der Compose-Taste
keycode 117 = Multi_key
! Punkt auf das Komma des Ziffern-Blocks
keycode 91 = period
! Was um alles in der Welt ist die Super-Taste?
keycode 115 = Super_L
! Für diejenigen, die ^? als Backspace auf der Konsole konfiguriert haben.
keycode 22 = BackSpace
```

Unicode

?

5.4 Zeitzonen

Die ganze Welt ist in Zeitzonen unterteilt, innerhalb derer eine einheitliche Zeit gilt (unabhängig von der Zeit einer Sonnenuhr an dem betreffenden Ort). Bezugspunkt für die Einteilung ist die *Universal Time*, kurz UTC.

Zu einer Zeitzone gehören bestimmte Regeln, wie z.B. die Umstellung von Sommer- auf Winterzeit. Linux kann diese Regeln automatisch berücksichtigen.

Allerdings ist die automatische Zeitumstellung nur angebracht, wenn Linux das einzige Betriebssystem auf dem Rechner ist oder auch das andere Betriebssystem auf diese Technik beherrscht.

TOC

DOS

Der Treiber `clock.sys` bringt DOS den Umgang mit Zeitzonen bei. Der TZ-Wert für die deutsche Zeitzone in `clock.sys` ist

```
CET-1CEST,M3.5.0/2,M10.5.0/3
```

Der Treiber `clock.sys` ist z.B. erhältlich unter der URL

ftp://ftp.tu-chemnitz.de/pub/simtel/msdos/clock/clk360rs.zip

Linux

Unter Linux gibt es quasi 3 Zeiten:

- Die der auf dem Motherboard installierten CMOS-Uhr,
- die Systemzeit des Linux-Kernels und
- die dem Benutzer von verschiedenen »Uhren« angezeigte Zeit.

```
            »clock -u -w«                »date -u -s«
 CMOS-Uhr  <--------------   Linux    <--------------
(Mainboard)                Systemzeit                 Benutzer-Uhr
           -------------->   (UTC)    -------------->
            »clock -u -s«             »date«, »xclock«
```

Die Programme `clock` und `date` übersetzen zwischen diesen Uhrzeiten.

Zwar liest der Kernel schon beim Booten die CMOS-Uhr aus und interpretiert sie als UTC. Aber um systematische Fehler der CMOS-Uhr zu korrigieren, wird die Uhrzeit vom Skript `/etc/init.d/boot` (oder vergleichbar) ein weiteres Mal gelesen. Dort wird dann der `clock`-Befehl ausgeführt und ist daher für die weiteren Betrachtungen ebenfalls relevant.

Für die Umstellung von Sommer- auf Winterzeit gibt es zwei Möglichkeiten: manuell oder automatisch.

```
Umstellung              manuell          automatisch
-----------------------------------------------------
CMOS-Uhr                Ortzeit          UTC
Option für clock        -keine-          -u
```

Die Zeitzone der CMOS-Uhr setzt man am sichersten während des Bootens im BIOS Setup. "UTC" ist die *Universal Time*: im Winter eine, im Sommer zwei Stunden weniger.

Welches Skript den Befehl `clock` ausführt (und demnach die Option '-u' enthalten kann) muß man dem Handbuch der eigenen Distribution entnehmen.

Die eigentliche Zeitzone wird unabhängig von den obigen Einstellungen durch den Link `/usr/lib/zoneinfo/localtime` festgelegt. In der Regel zeigt dieser Link auf die Hauptstadt des eigenen Landes:

```
Land           Zeitzonen-Datei
                                  |- Zeitzone -|
-----------------------------------------------
Deutschland    /usr/lib/zoneinfo/Europe/Berlin
```

```
Östereich       /usr/lib/zoneinfo/Europe/Vienna
Schweiz         /usr/lib/zoneinfo/Europe/Zurich
```

Um den Link anzulegen kann man das Kommando `ln` verwenden, im Falle von Deutschland lautet die komplette Befehlszeile dann:

```
ln -sf /usr/lib/zoneinfo/Europe/Berlin  /usr/lib/zoneinfo/localtime
```

Die eingestellte Zeitzone kann mit der Umgebungsvariable `TZ` überschrieben werden:

```
export TZ=Asia/Hong_Kong; xclock & export TZ=Africa/Dakar; xclock &
```

Das früher gebräuchliche "MET" sollte unter keinen Umständen weiterverwendet werden. Markus Kuhn schrieb mir:

> Die Datei /usr/lib/zoneinfo/MET existiert nur zwecks Rückwärtskompatibilität. Die Autoren der Zeitzonentabellen (Olson/Eggert/et al.) wollen eigentlich, daß deutsche Benutzer statt MET die Datei Europe/Berlin verwenden. Dann stimmen sogar die historischen Sommerzeiten vor 1945, die Linux auch alle kennt. Außerdem wird dann in der neuesten Version der Zeitzonentabelle "Mitteleuropäische Zeit (MEZ)" endlich korrekt ins Englische mit "Central European Time (CET)" übersetzt, denn "MET" ist ein Übersetzungsfehler (siehe z.B. Langenscheid Englisch; ich habe deswegen sogar beim Physikalisch-Technischen Bundesamt nachgefragt, die für die deutsche Zeit verantwortlich sind). "CEST" ist dann entsprechend "Central European Summer Time" ("MET DST", wie es noch in vielen Linux-Installationen benutzt wird, war eine falsche Uebersetzung von MESZ).

---

6. Anwendungssoftware

Mittlerweile sind viele Programme hinsichtlich der Unterstützung von landesspezifischen Einstellungen so weit genormt, man einen Großteil der Konfiguration durch Setzen der Umgebungsvariable `LANG` auf den Wert `de_DE` erledigen kann. Dies ist ein Standard, der in einem der vielen POSIX-Standards festgelegt wird, der aber nicht alle Bereiche abdeckt. So fehlen z.B. Papiergröße und Zeitzone; allerdings wären beide bei größeren Ländern schlecht mit einer Variable zu erfassen, so daß man sie außen vorgelassen hat.

Der richtige Ort für das Setzen von Umgebungsvariablen sind die Dateien `/etc/profile` oder `~/.bash_profile`:

```
#LANG=de_AT  # in Österreich
#LANG=de_CH  # in der deutschsprachigen Schweiz
LANG=de_DE   # in Deutschland
export LANG
```

Soweit die Theorie. In der Tat ist es etwas nervig, wenn sich deutsche und engliche Fehlermeldungen, Datumsangaben, usw. aufgrund von unvollständiger Internationalisierung mischen:

```
tar: Kann Archiv 'foo.tgz' nicht öffnen: Permission denied
```

In einem solchen Fall überschreibt man lieber diesen Aspekt der LANG-Variable:

```
LC_MESSAGES=C
export LC_MESSAGES
```

Dabei gehe ich davon aus, daß die eigene Linux-Distribution den POSIX-Standard durch die *Locale*-Dateien bereitstellt. Leider sind noch nicht alle Linux-Distributionen auf diesem Stand, so daß die obige Einstellung evtl. zu nervigen Fehlermeldungen führt. Beispiel:

```
Warning: locale not supported by C library, locale unchanged
```

In einem solchen Fall sollte man sich beim Hersteller des Distribution beschweren. Oder auf eigene Faust die locale-Dateien installieren, inkl. Kompilation der libc.

Details zu den *Lokalen* findet man im Linux Locale mini-HOWTO oder in der man-page zu locale. Der Rest dieses Kapitels behandelt die "Ausreißer", die die Umgebungsvariable LANG nicht auswerten.

6.1 Außer Konkurenz ...

Zeichensätze ohne Umlaute

Sowohl für die Textkonsole als auch für X11 existieren Zeichensätze (engl. `fonts'), die nicht alle Zeichen aus *ISO-latin1* enthalten. D.h. die fehlenden Zeichen (z.B. Umlaute) werden als Leerzeichen oder gar nicht auf dem Bildschirm dargestellt. In diesem Fall hilft natürlich die Konfiguration der Programme nicht weiter, sondern es muß ein anderer (vollständiger) Zeichensatz gewählt werden.

Eigene Süppchen

Konfigurationsdateien: `~/.Xresources` und `/etc/X11/Xresources`

Das X-Window-System wartet mit einer eigenen Konfigurationsmöglichkeit für Locale auf. Diese muß man aber nur nutzen, wenn man für einzelne X-Programme die Locale anders setzen will, d.h. wenn man die Einstellungen in den Umgebungsvariablen überschreiben will.

Anders ausgedrückt: durch Setzen der Locale über die X-Resourcen verkompliziert man in den meisten Fällen das Setup der eigenen Linux-Machine (und wer will das schon). Andererseits gibt es natürlich Fälle, in denen das evtl. notwendig ist (aber wer hat die schon).

Gerüchten zufolge bestimmen folgende Resourcen die Locale von X11:

```
!*displayLang:   de_DE
!*timeFormat:    de_DE
!*inputLang:     de_DE
!*basicLocale:   de_DE
!*numeric:       de_DE
```

Achtung: das Kommentarzeichen ist `!', nicht `#'.

6.2 Die Ausreißer

Solange nicht ausdrücklich erwähnt, betreffen die Einstellungen alle Versionen eines Programms. Die vorzunehmenden Einstellungen sind durch Einrückung markiert.

readline und Verwandte (z.B. bash)

Konfigurationsdatei: `~/.inputrc` Versionen: ab bash 1.14.1

```
set meta-flag on
set convert-meta off
set output-meta on
```

Kopiert man die Datei außerdem noch nach `/etc/skel/', so erhält sie jeder neu angelegte Benutzer

automatisch. Eine globale Konfigurationsdatei gibt es leider noch nicht.

Vorsicht: Kommentare und Leerzeilen in dieser Datei können zu Problemen füren.

telnet

Konfigurationsdatei: `~/.telnetrc`

Erscheinen beim `telnet` statt `äöü' die Zeichen `dv|', so muß man für jede Maschine einen Eintrag der Form:

```
localhost
     set binary true
```

anlegen. Denselben Effekt erreicht man mit der Option "-8". Eine globale Konfigurationsdatei für `telnet` existiert nicht.

ytalk

Konfigurationsdateien: `~/.` und `/etc/`

selection

Das Programm selection muß neu compiliert werden, damit es Umlaute korrekt per Cut&Paste transportiert. Die notwendigen Modifikationen (Patch) stammen von Andries Brouwer und liegen unter der URL ftp://ftp.win.tue.nl/pub/linux/util

Gegenüber `gpm` verbraucht `selection` deutlich weniger virtuellen Speicher, daher ist es auch heute noch von Interesse.

dosemu

Konfigurationsdateien: `~/.dosrc` und `/etc/dosemu.conf`

Versionen: ab 0.52pl16

```
keyboard {  layout de-latin1  keybint on  rawkeyboard on  }

X { updatefreq 8 title `DOS in a BOX' icon_name `xdos' keycode }
```

Hinweis: die Einstellung `keycode' funktioniert nur einwandfrei mit XFree86, nicht dagegen mit Xterminals o.ä.

emacs

Versionen: ab 19.22 (GNU)

Hinweis: einige Versionen des emacs haben Probleme mit der Aktivierung der Umlaute. Ab Version 19.29 sollten damit keinerlei Schwierigkeiten mehr auftreten.

Die Haupt-Konfigurationsdateien für den `emacs' sind:

```
/usr/lib/emacs/site-lisp/site-start.el       (global),
~/.emacs                                  (persönlich),
/usr/lib/emacs/site-lisp/default.el          (global),
```

und werden in dieser Reihenfolge abgearbeitet. Insbesondere überschreiben also die Einstellungen in der `default.el' die Einstellungen aus ` /.emacs' und `site-start.el'. Diese Einträge erlauben das Arbeiten mit Umlauten:

```
(set-input-mode (car (current-input-mode))
              (nth 1 (current-input-mode))
                                        0)

(standard-display-european t)
```

Speziellere Informationen findet man in der FAQ zum Thema `emacs und Umlaute' von Karl Brodowsky, erhältlich unter der URL:

ftp://ftp.uni-erlangen.de/pub/doc/ISO/deutsch/emacs-umlaute-faq.gz

less

Konfigurationsdateien: `~/.bash_profile` und `/etc/profile` Versionen: nur ältere Versionen sind betroffen

```
export LESSCHARSET=latin1
```

pine

Konfigurationsdateien: `~/.pinerc` und `/etc/pine.conf`

```
# character-set should reflect the capabilities of the display
# you have. Normal default is US-ASCII.  Typical alternatives
# include ISO-8859-x, where x is a number between 1 and 9.
character-set=ISO-8859-1
```

TOC

joe

Konfigurationsdateien: `~/.joerc` und `/etc/joe/joerc`

```
-asis
```

Achtung, das Minus muß in der 1. Spalte stehen.

elm

Konfigurationsdatei: `~/.elm/elmrc`

```
charset = iso-8859-1
displaycharset = iso-8859-1
textencoding = 8bit
```

nn

Konfigurationsdatei: `~/.nn/init`

```
set data-bits 8
```

Eine globale Konfigurationsdatei existiert nicht.

lynx

Konfigurationsdateien: `~/.lynxrc` und `/etc/lynx.cfg` Versionen: ab 2.4

```
CHARACTER_SET:ISO Latin 1
```

gslp

Kann von Hause aus keine Umlaute. Es ist ein patch erforderlich, dessen Beschreibung die Zielsetzung dieses HOWTO sprengt.

xfig

Kommandozeilenoption:

```
-metric
```

6.3 Weiterführende Tips

TOC

emacs

Der emacs kann in seinen Fähigkeiten durch Lisp-Module erheblich erweitert werden. Er ist dadurch das Schweizer Taschenmesser unter den Texteditoren. "Nur Kaffeekochen kann er nicht".

Die Lisp-Module tragen die Dateiendung `.el` oder `.elc`; letztere ist die compilierte Form einer `.el`-Datei. `emacs` sucht z.B. im Verzeichnis `/usr/lib/emacs/site-lisp` nach solchen Modulen.

Die Laden der Module kann im emacs per Hand mit dem Kommando

```
<Alt x> load-library <return> pc-mode <return>
```

vorgenommen werden. Alternativ kann man durch die Anweisung

```
(load "pc-mode")
```

in einer der Konfigurationsdateien das Modul dauerhaft laden. Nachfolgend die Beschreibung einiger nützlicher Module:

pc-mode

Konfiguriert die Tastaturbelegung so, wie PC-Benutzer es gewohnt sind. Keine Landesspezifischen Abhängigkeiten, aber oft gefragt. In der Konfigurationsdatei eintragen:

```
;;; pc-mode.el --- emulate certain key bindings used on PCs.
(load "pc-mode")
(pc-bindings-mode)
```

pc-select

Stellt Cut&Paste wie unter MacOS oder Win ein.

```
;;; pc-select.el ---  emulate cut&paste like on PC or Mac
(load "pc-select")
(pc-selection-mode)
```

iso-tex und iso-cvt

Zeigen die Umlaute unverändert an, speichern sie aber in dem TeX-Format ab.

konvers

Wie `iso-tex', kann aber nicht nur in beide Richtungen konvertieren, sondern beherrscht auch noch andere Formate wie etwa *MIME*.

iso-acc (iso-accent-mode) und alt-symbol

Erlauben die Eingabe von Umlauten und Akzenten durch vorangestellte Zeichen, d.h. ~a wird

schon während der Eingabe der Zeichen zu einem a mit der Schlange darüber (und auch als solches abgespeichert).

AUCTeX

AUCTeX ist eine leistungsfähige Umgebung zum Verfassen von TeX- und LaTeX-Dokumenten. Die automatische Erkennung von landesspezifischen Besonderheiten wird mit dem Eintrag

```
(setq TeX-parse-self t)
```

in einer der Konfigurationsdateien aktiviert. `\usepackage{german}' wird daraufhin von AUCTeX als Indikator verwendet.

Die erwähnten Pakete sind erhältlich unter der URL:

AUC TeX

ftp://sunsite.auc.dk/packages/auctex/auctex.tar.gz

alt-symbol.el

ftp://ftp.vlsivie.tuwien.ac.at/pub/8bit

iso-tex.el

ftp://ftp.dante.de/pub/tex/support/iso-tex

konvers.el

ftp://ftp.uni-erlangen.de/pub/doc/ISO/charsets/konvers-862.tar.gz

TeX/LaTeX

TeX ist ein professionelles Satzsystem. LaTeX ist eine Sammlung von Macros zum effektiven Einsatz von TeX.

Verarbeitung von Umlauten

LaTeX 2e und höher

Das Paket zur Verwendung des latin1-Zeichensatzes ist mittlerweile fester Bestandteil von LaTeX. Es wird aktiviert mit der Anweisung

```
\usepackage[latin1]{inputenc}
```

im Kopfteil des Dokuments.

plain-TeX, LaTeX2.09

Für diese Formate gibt es entweder keine Pakete oder nur veraltete Style-Dateien. Daher kann man Dokumente, die solche veralteten Pakete enthalten, i.a. nicht an andere versenden. Eine Verwendung der

alten Makros wie `iso.sty`, `xlatin1.sty` und `umlaute.sty` (für den Atari) wird daher nicht empfohlen.

Layout

Folgendes Beispiel kann als simples Grundgerüst für deutsche Texte (hier: Artikel) verwendet werden:

```
\documentclass[a4paper]{article}
    %  Die Option "a4paper" stellt als Papiergröße
    %  DinA4 ein. Alle anderen LaTeX-Pakete verlassen sich auf
    %  diese Einstellung.

\usepackage{german}
    %  stellt Abkürzungen für typische deutsche Besonderheiten
    %  bereit. Die Dokumentation befindet sich auf den
    %  CTAN-Serven unter /pub/tex/language/german/germdoc.tex

%\usepackage[T1]{fontenc}
%    % Erlaubt die Trennung von Wörtern mit Umlauten

\usepackage[latin1]{inputenc}
    % gibt den Zeichensatz des Dokuments an und macht somit die
    % Eingabe von latin1-Zeichen möglich.

\usepackage{a4}
    % stellt den bedruckten Bereich einer Seite, den sogenannten
    % <em/Satzspiegel/, ein. Er wird so berechnet, daß
    % durchschnittlich circa 60 bis 70 Buchstaben in einer Zeile
    % vorkommen, da man sonst beim Lesen die Zeilen verliert.

%\usepackage[german]{babel}
    % Aktiviert die Unterstutzung mehrerer Sprachen gleichzeitig.

\begin{document}
...
\end{document}
```

Weitere Informationen findet man unter der URL http://www.dante.de/dante/dante-faq.html

Eine bessere Alternative zur Satzspiegeleinstellung bietet das Paket "typearea.sty". Dieses Paket findet man bei den Koma-Script-Klassen von Markus Kohm unter ftp://[CTAN]/pub/tex/macros/latex/contrib/supported/koma-script/, die das "typearea.sty"-Paket intern verwenden. Es ist aber auch zusammen mit den Standardklassen (z.B. article) verwendbar.

Die Koma-Script-Klassen beachten auch andere deutsche Typographieregeln und sind daher sehr empfehlenswert. Dokumentation zu den Koma-Script-Klassen und dem "typearea.sty"-Paket befinden

sich unter ftp://[CTAN]/pub/tex/macros/latex/contrib/supported/koma-script/scrguide. ps.

CTAN

Comprehensive TeX Archive Network, bestehend aus (bald wieder) drei Rechnern in Amerika, Großbritannien und Deutschland. Der amerikanische und der deutsche Rechner sind von DANTE e.V (Deutschsprachige Anwendervereinigung TeX) angeschafft und finanziert worden. Der deutsche Rechner hat den Namen `ftp.dante.de`.)

Deutsche Wortrennung

Alle modernen TeX-Distributionen bieten die Möglichkeit, in der zugehörigen Konfigurationsdatei das Kommentarzeichen % vor der Zeile

```
german          ghyph31.tex
```

zu entfernen. Es werden dann Formate mit den deutschen Trennregeln erzeugt.

Mit der Rechtschreibreform ab 1. August 1998 werden andere Trennregeln gültig. Vermutlich werden dann rechtzeitig neue Trenntabellen zur Verfügung stehen.

Rechtschreibprüfung mit deutschem Wörterbuch

Zur Prüfung von Rechtschreibfehlern steht unter Linux das Programm `ispell` zur Verfügung. Die meisten Distributionen bieten dieses zur Installation an, die zugehörige deutsche Wortliste liegt seltener bei.

Die Wortlisten

Einen recht brauchbaren Grundstock für eine eigene Wortliste enthält das Paket "hk-deutsch2" von H. Knutzen. Es umfaßt 22 Dateien, die individuell zu einer eigenen Wortliste zusammengefaßt werden können. Die dazu notwendigen Schritte werden in dem beiliegendem README genau erklärt.

Durch die individuelle Zusammenstellung werden die Wortlisten kleiner und somit die Rechtschreibprüfung beschleunigt.

Es soll aber nicht verschwiegen werden, daß man diesen Listen in der Regel noch eine "eigene" Wörter hinzufügen muß (dies ist mit `ispell` automatisch möglich).

Das Paket ist erhältlich unter der URL:

ftp://ftp.informatik.uni-kiel.de/pub/kiel/dicts/hk2-deutsch.tar.gz

Das Rechtschreibprogramm `ispell`

Da `ispell` nicht jedes Textformat automatisch erkennt, muß man die Optionen `-T latin1` und `-w "äöüÄÖÜß"` verwenden. Kommen weitere nicht-ascii Zeichen im Text vor, so sollte man diese ebenfalls mit der '-w'-Option zulassen.

Ein anderes als das englische Wörterbuch wird mit der Option '-d' ausgewählt, also z.B.

```
ispell -d german -T latin1 -w "äöüÄÖÜß" German-HOWTO.txt
```

für das Deutsche Wörterbuch in der Datei `/usr/dict/german.hash` (Binärformat).

Ruft man `ispell` aus dem Editor `emacs` heraus auf, so kann man obige Parameter automatisch übergeben, indem man folgenden Eintrag in einer der Konfigurationsdateien des emacs vornimmt:

```
(setq ispell-extra-args '("-Tlatin1"
                          "-d" "/usr/lib/german"
                          "-w" "äöüÄÖÜß"))
```

---

7. Textdateien

7.1 Das Papierformat DIN A4

Nähere Informationen zu Papierformaten findet man unter der URL

http://www.ft.uni-erlangen.de/~mskuhn/iso-paper.html

Folgende Programme können auf das Papierformat *DIN A4* eingestellt werden.

xdvi

Konfigurationsdateien: `~/.Xresources` und `/etc/X11/Xresources`

```
XDvi.paper:     a4
```

ghostview

Konfigurationsdateien: `~/.Xresources` und `/etc/X11/Xresources`

```
Ghostview.pageMedia:  A4
```

gs

Kommandozeilenoption

```
-sPAPERSIZE=a4
```

dvips

Konfigurationsdateien: `~/.dvipsrc` und `/usr/lib/texmf/dvips/config.ps`

```
% Paper size information.
%
% If your default is a4 uncomment the following definition
% and comment out the letterSize definition.
%
@ a4size 210mm 297mm
@+ %%PaperSize: a4
```

ImageMagick

Dieses Grafikpaket enthält ein Programm zur Konvertierung beliebiger Grafikformate. Die Kommandozeilenoption

```
-page A4
```

erzwingt das Seitenformat *A4*.

7.2 Konvertierung von DOS- und Mac-Texten nach Linux

Die Textformate von DOS, MACs und UNIX unterscheiden sich in zwei Punkten:

- Codierung der Sonderzeichen (z.B. Umlaute)
- Markierung des Zeilenendes

Die Zeilenenden

Das Ende einer Textzeile wird wie folgt markiert:

```
Linux:    LF ("linefeed" = Zeilenvorschub, "^J", Ascii 10)
DOS:   CR LF
Mac:   CR    ("carriage return" = Wagenvorlauf, "^M", Ascii 13)
```

Die Programme `fromdos`, `todos` und `frommac`, `tomac` wandeln ausschließlich die Zeilenenden um, die Zeichen selbst werden nicht angepaßt. Gleiches gilt für die Optionen `conv=auto', bzw. `conv=text' des mount-Kommandos bzgl. des MS-DOS Dateisystems.

Die Zeichensätze

Die Zeichensätze setzen sich wie folgt zusammen:

```
Linux: ISO-LATIN1
DOS:   IBMPC Charactersets CP 437, CP 850, ...
Mac:   Mac-Eigengebräu
```

Der erste Teil aller dieser Zeichensätze ist der ASCII-Zeichensatz, so daß man Texte ohne Umlaute usw. nicht zu konvertieren braucht. Um sowohl Zeilenenden, als auch die Sonderzeichen zu konvertieren, kann man das Programm `recode` benutzen:

```
recode  -d ibmpc:lat1    hausarb.tex
```

wandelt die DOS-Datei `hausarb.tex' in das Format *ISO-Latin-1* um. Durch die Definition von Aliasen kann man sich das Leben etwas einfacher machen:

```
alias unix2dos='recode lat1:ibmpc'
alias dos2unix='recode ibmpc:lat1'
alias unix2mac='recode lat1:mac'
alias mac2unix='recode mac:lat1'
```

Achtung: diese aliase sind in shell-Skripten nicht verfügbar.

GNU recode ist von jedem GNU-mirror erhältlich, also z.B. von

ftp://ftp.uni-koeln.de/packages/gnu/recode-3.4.tar.gz

7.3 Drucken

```
Der
   Treppenstufen
                -Effekt
                      (engl. staircase)
                                        rührt
                                              von
                                                 den
                                                    oben
beschriebenen
             Unterschieden
                          im
                            Textformat
                                     her.
```

Die meisten Drucker besitzen ein kleines Mäuse-Klavier, an dem man den Drucker vom DOS-Textformat auf das Unix-Textformat umstellen kann. In der Regel wird damit auch automatisch auf die Betriebsart ISO-latin1 geschaltet, so daß auch Umlaute usw. gedruckt werden.

Weitere Informationen findet man im Linux Printing-HOWTO.

7.4 PostScript

Fehlen in einem gedruckten PostScript-Dokument die Umlaute ganz oder werden diese verstümmelt gedruckt, so liegt das nicht am Drucker oder am `ghostscript' (konvertiert PostScript für den Drucker), sondern an der Software: das PostScript wurde nicht korrekt erzeugt.

Besonders geschickt ist Word: es beginnt eine PostScript-Datei mit <Strg d>, dem Zeichen für das Ende einer Datei (unter Linux). Das fällt ebenfalls in die Kategorie "kaputtes PostScript" (wird aber mittlerweile von den meisten Druckerfiltern abgefangen).

---

8. Programmierung

8.1 National Language Support (NLS)

Für Programmierer unbedingt zu empfehlen ist das Paket `GNU gettext'. Es stellt eine Bibliothek zur Verfügung, die den Programmen zur Laufzeit den Zugriff auf länderspezifische Informationen erlaubt.

Welche Schritte im Einzelnen erforderlich sind, um ein Programm zu internationalisieren, erläutert das Manual zu `gettext' (derzeit noch `draft'-Stadium).

`GNU sharutils' ab der Version 4.2 ist ein beispielhaft internationalisiertes Programm-Paket.

8.2 Übersetzung von GNU-Programmen

Wer bei der Übersetzung von GNU-Programmen mithelfen will, sollte den Email-Verteiler `de@li.org` abonnieren. Dahinter steht eine majordomo-mailingliste (sprich: `majordomo@li.org` nimmt die subscribe-Befehle entgegen)

8.3 ncurses

Diese C-Bibliothek zur Bildschirmsteuerung ist seit Version 1.8.2 in der Lage, mit 8-bit Zeichen umzugehen. Ältere Versionen sollten nicht verwendet werden.

8.4 Erzeugung von PostScript

Die auszugebenden Texte können Umlaute enthalten, wenn vorher folgende Funktion aufgerufen wurde:

```
%
% change encoding to ISO8859-1  -  reiner@schildi.xnc.com
%
% <fontname> ISOfindfont => <font>
%
/ISOfindfont {
    dup 100 string cvs (ISO-) exch concatstrings cvn exch
    findfont dup maxlength dict begin
        { 1 index /FID ne {def}{pop pop} ifelse } forall
        /Encoding ISOLatin1Encoding def
        currentdict
    end definefont
} def
```

9. Anhang

9.1 Erklärende Übersetzung einiger technischer Begriffe

Ohne ein gewisses Maß an Englisch-Kenntnissen wird man an Linux nicht viel Freude haben, da sowohl die Dokumentation, als auch die Menüs und Hilfe-Funktionen der Programme vornehmlich in Englisch geschrieben sind.

Dieser Abschnitt soll das Verständnis englischen Orginal-Dokumentation erleichtern, indem der technische Zusammenhang der Vokabeln erläutert wird (wenn nötig).

Im Internet gibt es noch einige interaktive Wörterbücher, z.B. unter der URL:

http://www.chemie.fu-berlin.de/glossar/glossar.html

```
ENGLISH             DEUTSCH
-------------------------------------------------------------------------
abort             Programm oder Verbindung abbrechen
argument          Parameter (Einstellung), der einem Programm übergeben
                    wird
bracket           eckige Klammer
character         Zeichen
comment           Kommentar
device            Gerät; der Zugriff auf Geräte erfolgt unter Linux
                    fast ausschließlich über Dateien, die sogennanten
                    Gerätedateien im Verzeichnis `/dev'; meist bezieht
                    sich <em/device/ auf eine dieser Dateien
directory         Verzeichnis
environment       Umgebung; bezieht sich auf die Einstellungen, die
                    einem Programm mit Umgebungsvariablen (enviroment
                    variables) übergeben wird
file              Datei
file descriptor   Dateizugriffsnummer; programminterne Nummer für
                    eine offene Datei
font              Zeichensatz
host              Maschine, Computer, Rechner (wörtlich: Gastgeber)
I/O               Ein-/Ausgabe
kernel            Betriebssystemkern; der heilige Teil von Linux
key               Taste
keyboard          Tastatur
library           Zusammenfassung von compilierten C-Routinen zu
                    einer Bibliothek, die dann aus jedem anderen
                    Programm heraus benutzt werden kann
link              Verweis; anstatt eine Datei zu kopieren, kann man
                    auch einen Verweis anlegen; dies spart Platz und
                    erlaubt einige Tricks
load              laden, Beladung, Ausnutzung; in Zusammenhang mit
                    der Auslastung einer Maschine gebraucht
lock              Sperre (meist gewollt)
```

TOC

```
mount             montieren, zusammensetzen; stets im Zusammenhang
                    mit dem Verzeichnisbaum, der durch das Kommando
                    `mount ...' zusammengesetzt wird
network           Computernetz(werk)
news
OS                Betriebssystem (Operating System)
overflow          bei einer mathematischen Operation ist ein Ergebnis
                    aufgetreten, daß wegen seiner Größe nicht mehr
                    in die dafür vorgesehene Variable paßt
parenthesis       runde Klammer
path              Pfad, meist als Begriff für den kompletten
                    Verzeichnisnamen zu einer Datei gebraucht
pipe              Röhre, Veranschaulichung der Weiterleitung von Daten
                    von einem zum nächsten Programm; das sogennante
                    `pipe-symbol' ist das Zeichen | (senkr. Strich)
press             Drücken Sie (die Taste)
quote             Anführungszeichen oder Zitat
<!-- quoten = quotes benutzen = zitieren -->
remote            im Sinne von "die Gegenseite der Verbindung"
resolve           Ermittlung (Auflösung) der numerischen IP-Adresse,
                    die zu einem Maschinennamen gehört
server            Computer, der anderen Computern Dateien oder
                    Dienstleistungen zur Verfügung stellt
source            meist Begriff für die Programmtexte (Quellcode,
                    Sourcecode) benutzt; "may the source be with you"
                    UTSL: Use the source, Luke.
space             Leerzeichen
string            Zeichenkette
template          Vorlage, Muster, Schablone
trace             Ablaufverfolgung
type              Tippen Sie ein
underflow         Unterlauf, z.B. "der Unterlauf des Rheins" ;-)
                    Im Ernst: Wenn bei einer Floating-Point Berechnung, die Zahl
                    kleiner als das Maschinen-Epsilon wird, wird dies mit dem
                    Underflow-Flag angezeigt.
wildcard          Joker, meist erfüllt der Stern * diese Funktion,
                    z.B. zur unvollständigen Angabe von Dateinamen
```

Glibc 2 HOWTO

Eric Green, ejg3@cornell.edu

v1.5, 8 February 1998

The glibc 2 HOWTO covers installing and using the GNU C Library version 2 (libc 6) on Linux systems.

1. Introduction.

- 1.1 About glibc 2.
- 1.2 About this document.
- 1.3 Recent changes in this document.

2. Choosing your installation method.

3. Getting the library.

4. Installing as a test library.

- 4.1 Compiling and installing.
- 4.2 Updating the dynamic loader.
- 4.3 Configuring for gcc.
- 4.4 Updating header file links.
- 4.5 Testing your installation.

5. Installing as the primary C library.

- 5.1 Building the library from source.
- 5.2 Preparing for installation.
- 5.3 Installing from the binary package.
- 5.4 Installing from the source.

- 5.5 Updating the gcc specs.
- 5.6 Testing your installation.

6. Compiling with the non-primary libc.

- 6.1 A warning when using non-primary libcs.
- 6.2 Compiling programs with a test glibc.
- 6.3 Compiling programs with libc 5 when glibc is primary library.

7. Compiling C++ programs.

- 7.1 Installing libg++ for a test glibc install.
- 7.2 Installing libg++ for a primary glibc install.
- 7.3 Compiling C++ programs with the non-primary libc.

8. Reporting bugs.

9. Sample specs file.

10. Miscellanea.

- 10.1 Further information.
- 10.2 Credits.
- 10.3 Feedback.
- 10.4 Copyright.

1. Introduction.

1.1 About glibc 2.

Glibc 2 is the latest version of the GNU C Library. It currently runs unmodified on GNU Hurd systems and Linux i386, m68k, and alpha systems. Support for Linux PowerPC, MIPS, Sparc, Sparc 64, and Arm will be in version 2.1. In the future support for other architectures and operating systems will be added.

On Linux, glibc 2 is used as the libc with major version 6, the successor of the Linux libc 5. It is intended by the Linux libc developers to eventually replace libc 5. As of 2.0.6, glibc is considered production quality. Version 2.1 (due out in the near future) will be ready for main stream use along with adding more ports and features.

There are three optional add-ons available for glibc 2:

Crypt

The UFC-crypt package. It is seperate because of export restrictions.

LinuxThreads

An implementation of the Posix 1003.1c "pthread" interface.

Locale data

Contains the data needed to build the locale data files to use the internationalization features of the glibc.

The crypt and LinuxThreads add-ons are strongly recommended... not using them risks to be incompatible with the libraries of other systems. (If you do not wish to use them, you must add the option --disable-sanity-checks when you run configure.)

1.2 About this document.

This HOWTO covers installing the glibc 2 library on an existing Linux system. It is tailored for users of Intel based systems currently using libc 5, but users of other systems and alternate libraries (such as glibc 1) should be able to use this information by substituting the proper filenames and architecture names in the appropriate places.

The latest copy of this HOWTO can be found as part of the Linux Documentation Project or from http://www.imaxx.net/~thrytis/glibc/Glibc2-HOWTO.html.

1.3 Recent changes in this document.

TOC

Differences between version 1.5 and 1.4:

- Indexing added Ed Bailey.
- Changed my email address.

Differences between version 1.4 and 1.3:

- Changed current status from experimental to production.
- Updated list of developmental ports.
- Updated latest version to 2.0.6.

---

2. Choosing your installation method.

There are a few ways to install glibc. You can install the libraries as a test, using the existing libraries as the default but letting you try the new libraries by using different options when compiling your program. Installing in this way also makes it easy to remove glibc in the future (though any program linked with glibc will no longer work after the libraries are removed). Using glibc as a test library requires you to compile the libraries from source. There is no binary distribution for installing libraries this way. This installation is described in Installing as a test library.

The other way described in this document to install is using glibc as your primary library. All new programs that you compile on your system will use glibc, though you can link programs with your old libraries using different options while compiling. You can either install the libraries from binaries, or compile the library yourself. If you want to change optimization or configuration options, or use an add-on which is not distributed as a binary package, you must get the source distribution and compile. This installation procedure is described in Installing as the primary C library.

Frodo Looijaard describes yet another way of installing glibc. His method involves installing glibc as a secondary library and setting up a cross compiler to compile using glibc. The installation procedure for this method is more complicated then the test library install described in this document, but allows for easier compiling when linking to glibc. This method is described in his Installing glibc-2 on Linux document.

If you are currently running Debian 1.3 but do not want to upgrade to the unstable version of Debian to use glibc, the Debian libc5 to libc6 Mini-HOWTO describes how to use Debian packages to upgrade your system.

If you are installing glibc 2 on an important system, you might want to use the test install. Even if there are no bugs, some programs will need to be modified before they will compile due to changes in function prototypes and types.

3. Getting the library.

The glibc 2 consists of the glibc package and three optional add-on packages, LinuxThreads, Locale, and Crypt. The source can be found at

- ftp://prep.ai.mit.edu/pub/gnu/glibc-2.0.6.tar.gz
- ftp://prep.ai.mit.edu/pub/gnu/glibc-linuxthreads-2.0.6.tar.gz
- ftp://prep.ai.mit.edu/pub/gnu/glibc-localedata-2.0.6.tar.gz
- ftp://prep.ai.mit.edu/pub/gnu/glibc-crypt-2.0.6.tar.gz

It will take about 150 MB of disk space for the full compile and install. The basic binary install of just the core library package is about 50 MB.

Binary packages for 2.0.6 are not available. Version 2.0.4 binary packages are available for i386 and m68k, and version 2.0.1 for the alpha can be found at

- Intel x86:
  - ftp://prep.ai.mit.edu/pub/gnu/glibc-2.0.4.bin.i386.tar.gz
  - ftp://prep.ai.mit.edu/pub/gnu/glibc-crypt-2.0.4.bin.i386.tar.gz
- Alpha:
  - ftp://prep.ai.mit.edu/pub/gnu/glibc-2.0.1.bin.alpha-linux.tar.gz
  - ftp://prep.ai.mit.edu/pub/gnu/glibc-crypt-2.0.1.bin.alpha-linux.tar.gz
- m68k:
  - ftp://prep.ai.mit.edu/pub/gnu/glibc-2.0.4-m68k-linux.bin.tar.gz
  - ftp://prep.ai.mit.edu/pub/gnu/glibc-crypt-2.0.4-m68k-linux.bin.tar.gz

There are export restrictions on the crypt add-on. Non-US users should get it from ftp://ftp.ifi.uio.no/pub/gnu.

If you are running a Red Hat distribution, you can get rpms for glibc 2 from ftp://ftp.redhat.com/pub/redhat/. Glibc 2 is the primary C library for the new Red Hat distribution 5.0.

If you are running a Debian distribution, you can get the packages for glibc 2 from ftp://ftp.debian.org/debian/dists/unstable/main/. The files are named libc6. Glibc 2 is now part of the base package of the hamm version of Debian, and will be the primary libc when Debian 2.0 is released.

4. Installing as a test library.

This section covers installing glibc 2 as a test library. Anything you compile will be linked to your existing libraries unless you give some extra parameters to link to the new libraries. It appears that the paths are compiled into quite a few files, so you probably have to install the library from source.

4.1 Compiling and installing.

Prerequisites.

- About 150 MB free disk space
- GNU make 3.75
- gcc >= 2.7.2 (better 2.7.2.1)
- binutils 2.8.1 (for alpha you need a snapshot)
- bash 2.0
- autoconf 2.12 (if you change configure.in)
- texinfo 3.11

On an i586@133 with 64 MB of RAM, it takes about 3 hours to compile with full libraries with add-ons. On a loaded i686@200, it takes about half an hour.

Extracting the source.

You need to extract the source from the archives so you can compile it. The best way to do this is:

```
tar xzf glibc-2.0.6.tar.gz
cd glibc-2.0.6
tar xzf ../glibc-linuxthreads-2.0.6.tar.gz
tar xzf ../glibc-crypt-2.0.6.tar.gz
tar xzf ../glibc-localedata-2.0.6.tar.gz
```

This will put linuxthreads, crypt, and localedata directories in the glibc-2.0.6 directory where configure can find these add-ons.

Configuring.

In the glibc-2.0.6 directory, create a directory named compile, and cd into it. All work will be done in this directory, which will simplify cleaning up. (The developers have not been very concerned with getting 'make clean' perfect yet.)

```
mkdir compile
```

```
cd compile
```

Run `../configure`. To use the add-on packages, you need to specify them with --enable-add-ons, such as --enable-add-ons=linuxthreads,crypt,localedata. You also need to choose a destination directory to install to. /usr/i486-linuxglibc2 is a good choice. The configure line for this would be:

```
../configure --enable-add-ons=linuxthreads,crypt,localedata
--prefix=/usr/i486-linuxglibc2
```

Compiling and installing.

To compile and verify, run:

```
make
make check
```

If the 'make check' succeeds, install the library:

```
make install
```

4.2 Updating the dynamic loader.

1. Create a link from the new `ld.so` to `/lib/ld-linux.so.2`:

   ```
   ln -s /usr/i486-linuxglibc2/lib/ld-linux.so.2 /lib/ld-linux.so.2
   ```

   This is the only library where the location is fixed once a program is linked, and using a link in `/lib` will ease upgrading to glibc as your primary C library when the stable version is released.
2. Edit `/etc/ld.so.conf`. You need to add path to the lib directory the new libraries reside in at the end of the file, which will be `<prefix>/lib`, such as `/usr/i486-linuxglibc2/lib` for the choice above. After you have modified `/etc/ld.so.conf`, run

   ```
   ldconfig -v
   ```

4.3 Configuring for gcc.

The last step of installation is updating `/usr/lib/gcc-lib` so gcc knows how to use the new libraries. First you need to duplicate the existing configuration. To find out which configuration is current, use the -v option of gcc:

```
% gcc -v
```

```
Reading specs from /usr/lib/gcc-lib/i486-unknown-linux/2.7.2.2/sp
gcc version 2.7.2.2
```

TOC

In this case, i486-unknown-linux is the system, and 2.7.2.2 is the version. You need to copy the `/usr/lib/gcc-lib/<system>` to the new test system directory:

```
cd /usr/lib/gcc-lib/
cp -r i486-unknown-linux i486-linuxglibc2
```

Change into your new test system directory and version directory

```
cd /usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2
```

and edit the file `specs` found in this directory. In this file, change `/lib/ld-linux.so.1` to `/lib/ld-linux.so.2`. You also need to remove all expressions `%{...:-lgmon}` in the file, since glibc does not use the gmon library for profiling. A sample specs file can be found in the Sample specs file section.

4.4 Updating header file links.

You need create links in your new include directory to other include directories:

```
cd /usr/i486-linuxglibc2/include
ln -s /usr/src/linux/include/linux
ln -s /usr/src/linux/include/asm
ln -s /usr/X11R6/include/X11
```

You might also have other libraries such as ncurses which need their header files put in this directory. You should copy or link the files from `/usr/include`. (Some libraries may need to be recompiled with glibc2 in order to work with it. In these cases, just compile and install the package to `/usr/i486-linuxglibc2`.)

4.5 Testing your installation.

To test the installation, create the following program in a file glibc.c:

```
#include <stdio.h>

main()
{
    printf("hello world!\n");
}
```

and compile with the options of "-b <base install directory> -nostdinc -I<install directory>/include

-I/usr/lib/gcc-lib/<new system dir>/<gcc version>/include":

```
% gcc -b i486-linuxglibc2 -nostdinc -I/usr/i486-linuxglibc2/include
-I/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include glibc.c -o glibc
```

Use ldd to verify the program was linked with glibc2, and not your old libc:

```
% ldd glibc
libc.so.6 => /usr/i486-linuxglibc2/lib/libc-2.0.6.so (0x4000d000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

If it compiles, the links check out, and it generates "hello world!" when run, the installation succeeded.

---

5. Installing as the primary C library.

This section covers installing glibc 2 as your primary C library. Any new programs you compile will be linked with this library, unless you use special compile options to link with another version.

If you are are using Redhat or Debian and have downloaded the appropriate rpm or deb files, see the Redhat or Debian installion instructions. You can then skip this section.

5.1 Building the library from source.

This section explains how to compile glibc 2 and add-ons from the sources. You must compile the library if you want to change optimization or configuration options or use a package you do not have the binaries for.

Prerequisites.

- About 150 MB free disk space
- GNU make 3.75
- gcc >= 2.7.2 (better 2.7.2.1)
- binutils 2.8.1 (for alpha you need a snapshot)
- bash 2.0
- autoconf 2.12 (if you change configure.in)
- texinfo 3.11

On an i586@133 with 64 MB of RAM, it takes about 3 hours to compile with full libraries with add-ons. On a loaded i686@200, it takes about half an hour.

Extracting the source.

You need to extract the source from the archives so you can compile it. The best way to do this is:

```
tar xzf glibc-2.0.6.tar.gz
cd glibc-2.0.6
tar xzf ../glibc-linuxthreads-2.0.6.tar.gz
tar xzf ../glibc-crypt-2.0.6.tar.gz
tar xzf ../glibc-localedata-2.0.6.tar.gz
```

This will put linuxthreads, crypt, and localedata directories in the glibc-2.0.6 directory where configure can find these add-ons.

Configuring.

In the `glibc-2.0.6` directory, create a directory named compile, and cd into it. All work will be done in this directory, which will simplify cleaning up. (The developers have not been very concerned with getting 'make clean' perfect yet.)

TOC

```
mkdir compile
cd compile
```

Run `../configure`. To use the add-on packages, you need to specify them with --enable-add-ons, such as --enable-add-ons=linuxthreads,crypt,localedata. You probably will also want to specify paths where it will be installed. To match the standard linux distributions, specify --prefix=/usr. (When a prefix of `/usr` is specified on a linux system, configure knows to adjust other paths to place libc.so and other important libraries in `/lib`.) The whole configure line would be:

```
../configure --enable-add-ons=linuxthreads,crypt,localedata --prefix=/usr
```

Compiling.

To compile and verify, run:

```
make
make check
```

5.2 Preparing for installation.

Now you need to move some files around to prepare for the new library, whether you are installing from source or binaries. Any new program compiled will be linked to glibc, but old programs which are not statically linked will still depend on libc 5, so you can not just overwrite the old version.

1. Create a new directory to hold the old files to:

   ```
   mkdir -p /usr/i486-linuxlibc5/lib
   ```

2. The old header files must be evacuated from `/usr/include`:

   ```
   mv /usr/include /usr/i486-linuxlibc5/include
   ```

3. Create a new include directory and set up the links to other include directories:

   ```
   mkdir /usr/include

   ln -s /usr/src/linux/include/linux /usr/include/linux
   ln -s /usr/src/linux/include/asm /usr/include/asm
   ln -s /usr/X11R6/include/X11 /usr/include/X11
   ln -s /usr/lib/g++-include /usr/include/g++
   ```

   The links may need adjusting according to your distribution. At least Slackware puts g++ headers in `/usr/local/g++-include`, while Debian puts the headers in `/usr/include/g++`, and links `/usr/lib/g++-include` to `/usr/include/g++`. In the later case, you probably will want to move the original g++ include directory back to `/usr/include`.

4. Restore any extra header files and links. Some non-standard libraries such as ncurses put files in `/usr/include` or put a link to their include directories in the `/usr/include`. These files and links need to be restored in order to use the extra libraries properly.

TOC

5. Add your new library directory (such as `/usr/i486-linuxlibc5/lib`) *at the top* of your `/etc/ld.so.conf` file. You should have ld.so 1.8.8 or better installed to avoid getting strange messa once glibc is installed.
6. Move/copy all the old C libraries into the new directory.

```
mv /usr/lib/libbsd.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libc.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libgmon.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libm.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libmcheck.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libc.so /usr/i486-linuxlibc5/lib
mv /usr/lib/libm.so /usr/i486-linuxlibc5/lib
cp /lib/libm.so.5.* /usr/i486-linuxlibc5/lib
cp /lib/libc.so.5.* /usr/i486-linuxlibc5/lib
```

   `libm.so.5` and `libc.so.5` should be copied and not moved if `/usr` is a seperate partition from `/`, because they are required by programs used to start linux and must be located on the root drive partition.
7. Move the `/usr/lib/*.o` files into the new directory.

```
mv /usr/lib/crt1.o /usr/i486-linuxlibc5/lib
mv /usr/lib/crti.o /usr/i486-linuxlibc5/lib
mv /usr/lib/crtn.o /usr/i486-linuxlibc5/lib
mv /usr/lib/gcrt1.o /usr/i486-linuxlibc5/lib
```

8. Update your library cache after your libraries are moved.

```
ldconfig -v
```

5.3 Installing from the binary package.

If you are installing glibc from precompiled binaries, you must:

```
cd /
gzip -dc glibc-2.0.bin.i386.tar.gz | tar tvvf -
gzip -dc glibc-crypt-2.0.bin.i386.tar.gz | tar tvvf -
ldconfig -v
```

If you have a different architecture or version, substitute the proper file names.

5.4 Installing from the source.

To install the library from source, run:

```
make install
ldconfig -v
```

TOC

5.5 Updating the gcc specs.

The final step of the installation (for both binary and source installs) is to update the gcc `specs` file so you can link your programs properly. To determine which specs file is the one used by gcc, use:

```
% gcc -v
reading specs from /usr/lib/gcc-lib/i486-unknown-linux/2.7.2.2/specs
gcc version 2.7.2.2
```

In this case, i486-unknown-linux is the system, and 2.7.2.2 is the version. You need to copy the `/usr/lib/gcc-lib/<system>` to the old system directory:

```
cd /usr/lib/gcc-lib/
cp -r i486-unknown-linux i486-linuxlibc5
```

Change into the original directory and version directory

```
cd /usr/lib/gcc-lib/i486-unknown-linux/2.7.2.2
```

and edit the file `specs` found in this directory. In this file, change `/lib/ld-linux.so.1` to `/lib/ld-linux.so.2`. You also need to remove all expressions `%{...:-lgmon}` in the file, since glibc does not use the gmon library for profiling. A sample specs file can be found in the Sample specs file section.

5.6 Testing your installation.

To test the installation, create the following program in a file glibc.c:

```
#include <stdio.h>

main()
{
    printf("hello world!\n");
}
```

and compile the program.

```
% gcc glibc.c -o glibc
```

Use ldd to verify the program was linked with glibc2, and not your old libc:

```
% ldd glibc
libc.so.6 => /lib/libc.so.6 (0x4000e000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

If this compiles and generates "hello world!" when run, the installation was successful.

6. Compiling with the non-primary libc.

There are times you will want to use an alternate library to compile your programs with. This section explains how to accomplish this, using the directories and installation names used in the examples in the previous two sections. Remember to change the names to fit your setup.

6.1 A warning when using non-primary libcs.

Before compiling any programs which is used in the system boot process, remember that if the program is dynamically linked and is used before the non-root partitions are mounted, all linked libraries must be on the root partition. Following the installation process in the previous section for installing glibc as your primary C library, the old libc is left in `/lib`, which will be on your root partition. This means all of your programs will still work during booting. However, if `/usr` is on a different partition and you install glibc as a test library in `/usr/i486-linuxglibc2`, any new programs you compile with glibc will not work until your `/usr` partition is mounted.

6.2 Compiling programs with a test glibc.

To compile a program with a test-install glibc, you need to reset the include paths to point to the glibc includes. Specifying "-nostdinc" will negate the normal paths, and "-I/usr/i486-linuxglibc2/include" will point to the glibc includes. You will also need to specify the gcc includes, which are found in `/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include` (assuming you installed the test lib in i486-linuxglibc2 with gcc version 2.7.2.2).

To link a program with a test-install glibc, you need to specify the gcc setup. This is done by using the option "-b i486-linuxglibc2".

For most programs, you can specify these new options by adding them to the `$CFLAGS` and `$LDFLAGS` makefile options:

```
 CFLAGS = -nostdinc -I/usr/i486-linuxglibc2/include
-I/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include -b i486-linuxglibc2
 LDFLAGS = -b i486-linuxglibc2
```

If you are using a configure script, define the `$CFLAGS` and `$LDFLAGS` shell variables (by using env/setenv for csh/tcsh, or set/export for sh/bash/etc) before running configure. The makefiles generated by this should contain the proper `$CFLAGS` and `$LDFLAGS`. Not all configure scripts will pick up the variables, so you should check after running configure and edit the makefiles by hand if necessary.

If the programs you are compiling only call gcc (and not cpp or binutils directly), you can use the following script to save having to specify all of the options each time:

```
#!/bin/bash
/usr/bin/gcc -b i486-linuxglibc2 -nostdinc \
```

```
                -I/usr/i486-linuxglibc2/include \
                -I/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include "$
```


You can then use this script instead of "gcc" when compiling.

6.3 Compiling programs with libc 5 when glibc is primary library.

To compile a program with your old libraries when you have installed glibc as your main library, you need to reset the include paths to the old includes. Specifying "-nostdinc" will negate the normal paths, and "-I/usr/i486-linuxlibc5/include" will point to the glibc includes. You must also specify "-I/usr/lib/gcc-lib/i486-linuxlibc5/2.7.2.2/include" to include the gcc specific includes. Remember to adjust these paths based on the what you named the new directories and your gcc version.

To link a program with your old libc, you need to specify the gcc setup. This is done by using the option "-b i486-linuxlibc5".

For most programs, you can specify these new options by appending them to the `$CFLAGS` and `$LDFLAGS` makefile options:

```
 CFLAGS = -nostdinc -I/usr/i486-linuxlibc5/include
-I/usr/lib/gcc-lib/i486-linuxlibc5/2.7.2.2/include -b i486-linuxlibc5
 LDFLAGS = -b i486-linuxlibc5
```

If you are using a configure script, define the `$CFLAGS` and `$LDFLAGS` shell variables (by using env/setenv for csh/tcsh, or set/export for sh/bash/etc) before running configure. The makefiles generated by this should contain the proper `$CFLAGS` and `$LDFLAGS`. Not all configure scripts will pick up the variables, so you should check after running configure and edit the makefiles by hand if necessary.

If the programs you are compiling only call gcc (and not cpp or binutils directly), you can use the following script to save having to specify all of the options each time:

```
#!/bin/bash
/usr/bin/gcc -b i486-linuxlibc5 -nostdinc \
             -I/usr/i486-linuxlibc5/include \
             -I/usr/lib/gcc-lib/i486-linuxlibc5/2.7.2.2/include "$@"
```

You can then use this script instead of "gcc" when compiling.

---

7. Compiling C++ programs.

Libg++ uses parts of the math library, so is link to libm. Since your existing libg++ will be compiled with your old library, you will have to recompile libg++ with glibc or get a binary copy. The latest source for libg++ along with a binary linked with glibc (for x86) can be found at ftp://ftp.yggdrasil.com/private/hjl/.

7.1 Installing libg++ for a test glibc install.

If you have installed glibc as a test library, you need to install the files into the directory you installed glibc into (such as `/usr/i486-linuxglibc2` for the example in the previous sections). If you are installing from the binary package (which i would recommend, since i never had any luck compiling libg++ this way), you need to extract the files into a temporary directory and move all the `usr/lib/` files into the `<install directory>/lib/` directory, the `usr/include/` files into the `<install directory>/include/` directory (remember to delete your `include/g++` link first!), and the `usr/bin/` files into the `<install directory>/bin/` directory.

7.2 Installing libg++ for a primary glibc install.

If you have installed glibc as the primary library, you first need to move your old libg++ files into your old libc directory if you still want to be able to compile g++ programs with your old libc. Probably the easiest way to do this is by installing a new copy of the libg++ compiled with libc 5 as in the previous section, and then installing the glibc version normally.

7.3 Compiling C++ programs with the non-primary libc.

If you are trying to compile a C++ program with a non-primary libc, you will need to include the g++ include dir, which in the examples above would be `/usr/i486-linuxglibc2/include/g++` for a test glibc install or `/usr/i486-linuxlibc5/include/g++` for a primary glibc install. This can usually be done by appending the `$CXXFLAGS` variable:

```
 CXXFLAGS = -nostdinc -I/usr/i486-linuxglibc2/include
-I/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include
-I/usr/i486-linuxlibc5/include/g++ -b i486-linuxglibc2
```

8. Reporting bugs.

If you think the lib is buggy, please read first the FAQ. It might be that others had the same problem and there's an easy solution. You should also check the section "Recommended Tools to Install the GNU C Library" in the `INSTALL` file since some bugs are bugs of the tools and not of glibc.

Once you've found a bug, make sure it's really a bug. A good way to do this is to see if the GNU C library behaves the same way some other C library does. If so, probably you are wrong and the libraries are right (but not necessarily). If not, one of the libraries is probably wrong.

Next, go to http://www-gnats.gnu.org:8080/cgi-bin/wwwgnats.pl, and look through the bug database. Check here to verify the problem has not already be reported. You should also look at the file `BUGS` (distributed with libc) to check for known bugs.

Once you're sure you've found a new bug, try to narrow it down to the smallest test case that reproduces the problem. In the case of a C library, you really only need to narrow it down to one library function call, if possible. This should not be too difficult.

The final step when you have a simple test case is to report the bug. When reporting a bug, send your test case, the results you got, the results you expected, what you think the problem might be (if you've thought of anything), your system type, the versions of the GNU C library, the GNU CC compiler, and the GNU Binutils which you are using. Also include the files `config.status` and `config.make` which are created by running `configure`; they will be in whatever directory was current when you ran `configure`.

All bug reports for the GNU C library should be sent using the `glibcbug` shell script which comes with the GNU libc to bugs@gnu.org (the older address bugs@gnu.ai.mit.edu is still working), or submitted through the GNATS web interface at http://www-gnats.gnu.org:8080/cgi-bin/wwwgnats.pl.

Suggestions and questions should be sent to the mailing list at bugs-glibc@prep.ai.mit.edu. If you don't read the gnewsgroup gnu.bug.glibc, you can subscribe to the list by asking bug-glibc-request@prep.ai.mit.edu.

Please DO NOT send bug report for the GNU C library to <bug-gcc@prep.ai.mit.edu>. That list is for bug reports for GNU CC. GNU CC and the GNU C library are separate entities maintained by separate people.

9. Sample specs file.

Included here is a sample `specs` file for glibc 2 which is used by gcc for compiling and linking. It should be found in the directory `/usr/lib/gcc-lib/<new system dir>/<gcc version>`. If you are running an x86 system, you probably can copy this section to the file exactly.

```
 *asm:
 %{V} %{v:%{!V:-V}} %{Qy:} %{!Qn:-Qy} %{n} %{T} %{Ym,*} %{Yd,*} %{Wa,*:%*}

 *asm_final:
 %{pipe:-}

 *cpp:
 %{fPIC:-D__PIC__ -D__pic__} %{fpic:-D__PIC__ -D__pic__} %{!m386:-D__i486__}
%{posix:-D_POSIX_SOURCE} %{pthread:-D_REENTRANT}

 *cc1:
 %{profile:-p}

 *cc1plus:


 *endfile:
 %{!shared:crtend.o%s} %{shared:crtendS.o%s} crtn.o%s

 *link:
 -m elf_i386 %{shared:-shared}   %{!shared:     %{!ibcs:       %{!static:
%{rdynamic:-export-dynamic}     %{!dynamic-linker:-dynamic-linker
/lib/ld-linux.so.2}}  %{static:-static}}}

 *lib:
 %{!shared: %{pthread:-lpthread}        %{profile:-lc_p} %{!profile: -lc}}

 *libgcc:
 -lgcc

 *startfile:
 %{!shared:      %{pg:gcrt1.o%s} %{!pg:%{p:gcrt1.o%s}
%{!p:%{profile:gcrt1.o%s}                        %{!profile:crt1.o%s}}}}    crti.o%s
%{!shared:crtbegin.o%s} %{shared:crtbeginS.o%s}

 *switches_need_spaces:


 *signed_char:
 %{funsigned-char:-D__CHAR_UNSIGNED__}

 *predefines:
 -D__ELF__ -Dunix -Di386 -Dlinux -Asystem(unix) -Asystem(posix) -Acpu(i386)
-Amachine(i386)

 *cross_compile:
 0
```

```
*multilib:
. ;
```

10. Miscellanea.

10.1 Further information.

Web pages.

- FSF's GNU C Library Home Page
- Using GNU Libc 2 with Linux
- Installing glibc-2 on Linux.
- Debian libc5 to libc6 Mini-HOWTO.

Newgroups.

- comp.os.linux.development.system
- comp.os.linux.development.apps
- linux.dev.kernel
- gnu.bugs.glibc

Mailing lists.

Glibc 2 Linux discussion list.

This list is intended for discussion among Linux users who have installed glibc2, the new GNU C libraries. Topics might include compatibility issues and questions about the compilation of code in a Linux/glibc setting. To subscribe, send mail to Majordomo@ricardo.ecn.wfu.edu with a body of "subscribe glibc-linux <your email address>.

10.2 Credits.

Most of this information was stolen from the GNU Libc web page and from Ulrich Drepper's <drepper@gnu.ai.mit.edu> glibc 2 announcement and his comments. Andreas Jaeger <aj@arthur.rhein-neckar.de> provided some of the Reporting bugs section.

The following people have provided information and feedback for this document:

- Allex <allex@ms2.accmail.com.tw>
- Mark Brown <M.A.Brown-4@sms.ed.ac.uk>
- Ulrich Drepper <drepper@gnu.ai.mit.edu>

- Scott K. Ellis <ellis@valueweb.net>
- Aron Griffis <agriffis@coat.com>
- Andreas Jaeger <aj@arthur.rhein-neckar.de>
- Frodo Looijaard <frodol@dds.nl>
- Ryan McGuire <rmcguire@freenet.columbus.oh.us>
- Shaya Potter <spotter@capaccess.org>
- Les Schaffer <godzilla@futuris.net>
- Andy Sewell <puck@pookhill.demon.co.uk>
- Gary Shea <shea@gtsdesign.com>
- Stephane <sr@adb.fr>
- Jan Vandenbos <jan@imaxx.net>

Translations of this document are being done by:

- Chinese: Allex <allex@ms2.accmail.com.tw>
- French: Olivier Tharan <tharan@int-evry.fr>
- Japanese: Kazuyuki Okamoto <ikko-@pacific.rim.or.jp>

10.3 Feedback.

Besides writing this HOWTO, maintaining the glibc 2 for Linux page, and using it on my machine, I have nothing to do with the glibc project. I am far from knowledgeable on this topic, though I try to help with problems mailed to me. I welcome any feedback, corrections, or suggestions you have to offer. Please send them to ejg3@cornell.edu.

10.4 Copyright.

Copyright (c) 1997 by Eric Green. This document may be distributed under the terms set forth in the LDP license.

Linux HAM-HOWTO, Amateur Radio.

Terry Dawson, VK2KTJ, terry@perf.no.itg.telstra.com.au

v2.3, 1 April 1997

It is hoped that this document will assist Amateur Radio operators in finding and trying the various amateur radio software that has been written for or ported to Linux. It is also hoped that as a consequence of this information being available that more amateur radio operators will choose Linux as the platform of choice for their experimentation, and that software developers will choose Linux as the platform for their software development, further expanding the role of operating systems like Linux in the Amateur Radio field.

1. Introduction.

- 1.1 Changes from the previous version

2. Where to obtain new versions of this document.

3. Satellite Software.

- 3.1 MicroSat Ground Station Software
- 3.2 SatTrack - Satellite tracking program
- 3.3 Predict
- 3.4 UO11
- 3.5 Dove
- 3.6 Kepgen

4. Shack Automation Software.

- 4.1 FT-890 remote control

5. Packet Radio

- 5.1 F6FBB Packet BBS
- 5.2 DX Cluster Watcher
- 5.3 Digiinfo
- 5.4 BayBox BBS
- 5.5 JNET
- 5.6 XNet
- 5.7 monax25
- 5.8 splitscreen
- 5.9 talk-ax25
- 5.10 7Plus for Linux
- 5.11 TNOS
- 5.12 N0ARY Packet BBS for UN*X
- 5.13 LBBS - Linux BBS message gateway
- 5.14 MBL/RLI message to NNTP and email converter.
- 5.15 CLX Packet DX Cluster Program
- 5.16 DPTNT Terminal and DPBOX BBS package.
- 5.17 IPIP encapsulation daemon.
- 5.18 AXIP encapsulation daemon.
- 5.19 Ping-Pong Convers Server
- 5.20 RSPF Daemon
- 5.21 Michael Westfall's TTYLINK Daemon
- 5.22 Craig Small's TTYLINK Daemon

6. Morse Code

- 6.1 Pileup
- 6.2 bip
- 6.3 cw2hex
- 6.4 SoundCard CW
- 6.5 GW4PTS Morse trainer.
- 6.6 morse (aka superiormorse)

TOC

7. AMTOR Software.

8. PACTOR Software.

9. Slow Scan Television Software.

10. Facsimile Software.

11. Design and Construction Software.

- 11.1 oscope - Enhanced Software Oscillosope
- 11.2 Software Oscilloscope
- 11.3 Printed Circuit Board design tool.
- 11.4 Chipmunk circuit design and simulation tool
- 11.5 irsim
- 11.6 Spice vers. 3f4
- 11.7 svgafft - Spectrum analyser
- 11.8 Audio Spectrum Analyser
- 11.9 ObjectProDSP
- 11.10 bpf - bandpass filter calculator

12. Training/Educational Software.

13. Miscellaneous Software.

- 13.1 twclock - World Time Clock for Hams
- 13.2 Chirp - Contest Logger
- 13.3 Contest - Contest Logging Program
- 13.4 Linux for HAMS CD-ROM
- 13.5 SunClock
- 13.6 Xearth

14. How to contribute or update an entry.

15. Discussion relating to Amateur Radio and Linux.

16. Copyright.

---

1. Introduction.

This list was prompted by comments that had been expressed in the various Linux newsgroups about the number of amateur radio operators that were involved with Linux. It is now clear that Linux is becoming a popular platform for Amateur Radio development work.

This list was originally called the RADIOLINUX list, but has been moved into the Linux Documentation Project HOWTO collection, so it has been renamed the HAM-HOWTO. I make no apologies for the name.

1.1 Changes from the previous version

```
Additions:
        Added entry based last updated field.
        bpf - filter calculator
        BayBox - Packet BBS
        digiinfo - digipeater information database
        Contest logging program
        Chirp contest logger
        bip
        pileup
        twclock
        DX Cluster watcher
        F6FBB BBS

Corrections/Updates:
        Changed all ftp: url's to point to the directory rather than the file.
        CLX, and just about everything else.
        Updated location of John Gotts mirror.

ToDo:
        find a way of automating this, or find an alternate means of
        presenting this information.
```

2. Where to obtain new versions of this document.

The Linux Documentation Project runs a Web Server and this document appears there as The HAM-HOWTO.

Dennis Boylan N4ZMZ `dennis@nanovx.atl.ga.us` makes it available at the following three locations:

www.com, www.hboc.com and www.lan.com.

John Gotts N8QDW `jgotts@engin.umich.edu` makes it available at: www-personal.engin.umich.edu.

Dan Todd `dantodd@fusilli.ucdavis.edu` makes it available at the following two locations: wheel.dcn.davis.ca, and fusilli.ucdavis.edu.

Alan Hargreaves VK2KVF `alan@dap.CSIRO.AU` makes it available in Australia at: www.dap.csiro.au.

The Tuebingen Radio Club has it available from their home page at: DARC Home Page

Please let me know if you'd like to make it available somewhere too. I'd like to see it on some Web Servers that are accessible from radio.

3. Satellite Software.

The following software is for use in experimentation with Satellite communication.

3.1 MicroSat Ground Station Software

Author

John Melton, G0ORX/N6LYT, `g0orx@amsat.org` and Jonathan Naylor G4KLX, `g4klx@amsat.org`

Description

Microsat Ground Station software.

Status

BETA. Version 0.91 released.

Updated

01 Apr 97

System requirements

Alan Cox's kernel based AX.25 support ver 1.1.12 or better. X-Windows. The programs make use of the Athena Widgets and look much better with the 3D libraries.

Detail

This software allows you to use of a KISS tnc to directly communicate with the Microsat series of satellites. It provides an Athena Widgets based X-Windows interface, and allows you a comprehensive range of means of interacting with the satellite. The software should work with any window manager.

The software provides the following programs:

xpb:

broadcast monitor

xpg:

ftl0 file upload program, message upload program

xtlm:

telemetry display program

downloaded:

downloaded file list viewer

directory:

directory list viewer

message:

message preparation application

viewtext:

uncompressed ASCII text file viewer

viewlog:

display the contents of some log files

xweber:

special program for downloading webersat images

phs:

general purpose PACSAT header stripper

Where and How to obtain it.

John's software is available from: ftp.cs.nott.ac.uk or ftp.amsat.org or ftp.funet.fi. The version current at the time of writing was `microsat-0.91.tar.gz`. Please check for new versions.

Licensing/Copyright

GNU Public License. Freely redistributable, No warranty.

Contributed by:

John Melton, G0ORX/N6LYT, Alan Cox, GW4PTS, Jonathon Naylor, G4KLX

3.2 SatTrack - Satellite tracking program

Name

SatTrack

Author

Manfred Bester, DL5KR, `manfred@ssl.berkeley.edu`, (510) 849-9922

Description

Satellite realtime tracking and orbit prediction program with X11 color graphics.

Status

Version 4.0 is a fully commercial release. Version 3.1.5 is the last freely available release.

Updated

01 Apr 97

System requirements

A vt100 terminal and or X11 server. A Maths Coprocessor is useful, but not essential, for good performance.

Detail

SatTrack is a satellite tracking program that has been written in 'C' language on a UNIX workstation. It provides two different live displays, for single or multiple satellites, and two different orbit prediction facilities. Cursor controls used in both live displays are compatible with the VT100 standard, which allows the program to be run basically from any terminal. An optional X Window graphics display shows the typical world map tracking chart. The orbit prediction can be run either interactively or in a batch mode. In the latter case all necessary parameters are specified on the command line. The program uses the NORAD/NASA two-line Keplerian element (TLE) sets directly.

The current version of SatTrack can track a single satellite and/or display multiple satellites for a single ground station. It also can control suitable ground station hardware, like antennas and radio equipment, and has an autotrack mode in which it switches automatically between a number of specified satellites. For calibration purposes tracking of the Sun and the Moon are also provided. Future releases will provide more options. SatTrack requires about 5MB of disk space in the full distribution with eight different world maps (2 styles, with 4 different sizes each, to fit everyone's screen and taste) and less than 1.5 MB of memory at run time. More information can be obtained from the SatTrack WWW Home Page.

Where and How to obtain it.

The commerical version of SatTrack can be found at: SatTrack WWW Home Page.

The free version of SatTrack may be found at: ftp.jvnc.net or ftp.amsat.org.

Licensing/Copyright

The software is Copyright of Manfred Bester. It can be used without special permission for non-profit, non-commercial use. For commercial applications a license from the author is required. Version 4.0 is a fully commercially supported release.

Contributed by:

Manfred Bester, DL5KR

3.3 Predict

Author

John A. Magliacane, KD2BD, `kd2bd@amsat.org`

Description

A satellite orbital prediction program that also tracks the sun's position to predict visible satellite passes.

Status

Complete and working.

Updated

01 Apr 97

System requirements

predict is a screen-oriented program that utilizes the ncurses-1.8.5 (or later) library. A color terminal is a plus, but not a necessity.

Detail

Features of predict include azimuth and elevation headings of satellite passes, orbital phase, sub-satellite point locations, slant ranges, orbit numbers, and sunlight visibility as a function of date and time. The program is simple, fast, and easy to use. Predict maintains an orbital database for 21 satellites that can be updated by the user through the keyboard or by using files containing NASA 2-line orbital element data. Since this program operates under a multiuser environment, each user on the system owns and maintains his or her own unique orbital data base.

This program has been compiled using the highest level of compiler optimization for high speed execution, and uses an a.out binary format for greatest compatibility.

To install the package use the following:

```
# cd /
# tar xvfz predict.tgz
```

Where and How to obtain it.

You can obtain `predict.tgz` from: amsat.org

Licensing/Copyright

This software is available free of charge and must be used for non-commercial purposes only.

Contributed by

John A. Magliacane, KD2BD

3.4 UO11

Author

John A. Magliacane, KD2BD, `kd2bd@amsat.org`

Description

A simple UoSAT-OSCAR-11 satellite telemetry decoding program.

Status

Complete and working.

Updated

01 Apr 97

System requirements

A 1200 baud UoSAT-OSCAR-11 FSK demodulator (Bell-202 is suitable), a VHF-FM receiver that can tune to 145.825 MHz, a small yagi antenna, and a communications program such as Minicom that can save 1200 baud ASCII telemetry data to a text file.

Detail

uo11 decodes ASCII telemetry data that has been captured through a terminal program, error checks the received data, and applies calibration equations to the received data to produce a report that can be easily understood and analyzed. uo11 comes pre-compiled (a.out format) with a man page entry for ease of use.

To install you should use the following commands:

```
# cd /
# tar xvfz uo11.tgz
```

Where and How to obtain it.

You can obtain `uo11.tgz` from: amsat.org

Licensing/Copyright

This software is available free of charge and must be used for non-commercial purposes only.

Contributed by

John A. Magliacane, KD2BD

3.5 Dove

Author

John A. Magliacane, KD2BD, `kd2bd@amsat.org`

Description

A simple DOVE-OSCAR-17 satellite telemetry decoding program.

Status

Complete and working.

Updated

01 Apr 97

System requirements

A terminal node controller (TNC), a VHF-FM receiver that can tune to 145.825 MHz, a small yagi antenna, and a communications program such as Minicom that can save 1200 baud ASCII telemetry data to a text file.

Detail

dove decodes ASCII telemetry data that has been captured through a terminal program and applies calibration equations to the received data to produce a report that can be easily understood and analyzed. dove comes pre-compiled (a.out format) with a man page entry for ease of use.

To install Dove you should use the following commands:

```
# cd /
# tar xvfz dove.tgz
```

Where and How to obtain it.

You can obtain `dove.tgz` from: amsat.org

Licensing/Copyright

This software is available free of charge and must be used for non-commercial purposes only.

Contributed by

John A. Magliacane, KD2BD

3.6 Kepgen

Author

John A. Magliacane, KD2BD, `kd2bd@amsat.org`

Description

A simple, yet useful utility that generates Keplerian data files in the NASA 2-line format from data entered through the keyboard.

Status

Complete.

Updated

01 Apr 97

System requirements

kepgen is a screen-oriented program that utilizes the ncurses-1.8.5 (or later) library. A color terminal is a plus, but not a necessity.

Detail

kepgen allows the user to create checksummed Keplerian data files in the NASA 2-line format using orbital data entered via the keyboard. kepgen was created out of the need for generating 2-line orbital files from orbital data copied from WA3NAN audio transmissions during Space Shuttle missions. The files created can be read by virtually any satellite tracking or orbital prediction program. kepgen comes pre-compiled (a.out format) with a man page entry for ease of use.

To install kepgen you should use the following commands:

```
# cd /
# tar xvfz kepgen.tgz
```

Where and How to obtain it.

You can obtain `kepgen.tgz` from: amsat.org

Licensing/Copyright

This software is available free of charge and must be used for non-commercial purposes only.

Contributed by

John A. Magliacane, KD2BD

---

4. Shack Automation Software.

Software for simplifying tasks in the shack. Examples might include software for controlling the newer breed of radios, logging programs, QSL database, or antenna rotation.

4.1 FT-890 remote control

Author

Emarit Ranu, KG0CQ, `drranu@holly.ColoState.EDU`

Description

A simple program to control a Yaesu FT-890 transciever via its CAT port from linux.

Status

Working

Updated

01 Apr 97

System requirements

Yaesu FT-890, Yaesu FIF-232C interface converter or homebuilt equivalent (Feb 1993 "QST", page 37). Linux, serial cable.

Detail

The program is started with one parameter specifying the com port to use. Once started the program is menu driven. Future versions will support an X-Windows interface.

Where and How to obtain it.

The software is currently only available from the author by email.

Licensing/Copyright

You may copy this program millions of times, the only restrictions are:

- You may not charge ANYONE for the program itself.
- You may not distribute modified versions of the program.
- If you distribute the program, it MUST be the ORIGINAL tar file as received by the author. The README file has to be included.

Contributed by:

Emarit Ranu, KG0CQ

5. Packet Radio

Software for use in conjunction with, or for facilitating packet radio.

5.1 F6FBB Packet BBS

Author

Jean-Paul Roubelat, F6FBB, `jpr@f6fbb.org`

Description

Fully Features Packet Radio BBS system with X11 Windows interface

Status

Version 7.00b45 is Beta.

Updated

01 Apr 97

System Requirements

Kernel based AX.25/NetRom/Rose network configuration. X11 windowing system for the X11 version.

Detail

This is a port of the well known F6FBB Packet Bulletin Board system to Linux. The BBS supports connections from AX.25, NetRom, Rose and TCP/IP and offers all of the features you'd expect from the product. The optional X11 interface provides a nicely presented sysop interface.

A mailing list for F6FBB is available. To subscribe send a mail message to: `majordomo@f6fbb.org` with the line `subscribe xfbb' in the body of the message. The mailing list is archived at zone.pspt.fi.

Where and how to obtain it

You can obtain the F6FBB package from its home site at: F6FBB Home Page.

Licensing/Copyright etc.

The F6FBB software is frelly available for non-commercial use.

5.2 DX Cluster Watcher

Author

Pierluigi Guerzoni, IZ4AKO, `iz4ako@ing49.unife.it`

Description

Displays DX information passively received from a DX Cluster Node.

Status

Version 4, stable.

Updated

01 Apr 97

System Requirements

Requires Kernel based AX.25 configuration.

Detail

The DX Cluster watchers listens to the DX Cluster traffic on your frequency, captures it and displays it in a coloured text window. It conserves traffic on the frequency by not requiring you to actually connect to the cluster node.

Where and how to obtain it

You can obtain the DX Cluster Watcher from the: DX Cluster Watcher Home Page.

Licensing/Copyright etc.

GNU GPL

Contributed by

Pierluigi Guerzoni, IZ4AKO, `iz4ako@ing49.unife.it`

5.3 Digiinfo

Author

Ported by Mario Lorenz, DL5MLO, `dl5mlo@vdazone.org`

Description

A database of packet radio network information.

Status

Version 2.5.1 is stable.

Updated

01 Apr 97

TOC

System Requirements

No special requirements.

Detail

digiinfo was a program initially written for the Atari by Karsten DC7OS. Digiinfo is a database program for displaying information about the packet radio networks.

Where and how to obtain it

You can obtain the `digiinfo` package and database from its home site at: www.vdazone.org.

Licensing/Copyright etc.

ALAS, Amateur only license

Contributed by

Mario Lorenz, DL5MLO, `dl5mlo@vdazone.org`

5.4 BayBox BBS

Author

Flori Radlherr, DL8MBT, `flori@t-online.de`

Description

A Packet Bulleting Board System for Linux.

Status

Version 1.38, stable.

Updated

01 Apr 97

System Requirements

Recommended hardware is 80486/33Mhz with 8MB RAM

Detail

Unfortunately the documentation is in German and I'm unable to understand most of it. I get the impression though that this is a BBS that supports all of the usual features on a packet radio bbs, forwarding etc, but the documentation seems to suggest that the Linux version supports telnet access only. This isn't really a big constraint, as you can easily configure a session under the Linux *ax25d* program to accept AX.25/NetRom/Rose connections and connect the BBS via telnet.

If you're able to help by giving a better description, I'd appreciate it.

Where and how to obtain it

You can obtain the `BayBOX` from its home sites at: members.aol.com

Licensing/Copyright etc.

TOC

The BayCom-Mailbox software is intended for usage in amateur radio context only. Any usage outside of the amateur radio AX.25 network, especially commercial usage or distribution is prohibited. Within the scope of this limitation, usage and copying of the BayCom-Mailbox without fee is hereby granted. In any case of usage, the software is provided "as is" and comes with absolutely no warranty, neither implied nor expressed.

Contributed by

Iztok Saje, S52D, `s52d@s55tcp.ampr.org`

5.5 JNET

Author

Takeshi Shoji, JF2CEX, `taka@myk.ilc.or.jp`

Description

An enhanced version of KA9Q NOS for Linux and other platforms.

Status

Version 961210, Stable, released.

Updated

01 Apr 97

System Requirements

No special system requirements, requires KISS TNC.

Detail

This another actively maintained derivative of the KA9Q NOS package. This version has been ported to a wide range of alternative platforms and offers some enhancements not available in other NOS derivatives. A small catch is that most of the documentation is in Japanese.

Where and how to obtain it

You can obtain the latest version of JNET from its home site at: www.myk.ilc.or.jp.

Licensing/Copyright etc.

I'm actually not sure. I suspect though that as a NOS derivative it is freely usable by amateur radio operators.

Contributed by

Isao SEKI, http://www.prug.or.jp/people/jm1wbb

5.6 XNet

Author

Richard R. Parry, W9IF, `rparry@qualcomm.com`

Description

A traffic analyser for AX.25 networks.

Status

Stable, released.

Updated

01 Apr 97

System Requirements

TCL-7.5/Tk-4.1, KISS TNC.

Detail

The XNet software provides a variety of tools that an AX.25 lan manager could use to analyze and manage traffic. The tools present a graphical view of all of the traffic received at the monitoring node allowing the observer to easily see where the traffic flows normally and other information such as:

- Packet counts
- Node statistics
- Network Utilisation figures

The network is displayed with icons representing hosts and lines between them representing the traffic flow. The graphical display provides an instantly absorbable view of the traffic on the network being monitored. Richard has more information and sample displays at: www.qualcomm.com.

Where and how to obtain it

You can obtain the software from: www.qualcomm.com

Licensing/Copyright etc.

XNET is a freeware program. It may be freely distributed.

Contributed by

Richard Parry, W9IF

5.7 monax25

Author

Skip Hansen, WB6YMH and Harold Price, NK6K.

Maintained by

Jonathon Naylor, `jsn@cs.nott.ac.uk`

Description

The *monax25* provides a number of utilities for collecting statistics on AX.25 channel usage.

Status

Version 1.11, Stable

Updated

01 Apr 97

System Requirements

KISS TNC.

Detail

The original *ax25mon* programs were written for MSDOS. This package is a port of that software to Linux.

Where and how to obtain it

The `monax25' package may be obtained from: sunsite.unc.edu ham applications directory.

Licensing/Copyright etc.

Freely copyable, not GPL though.

5.8 splitscreen

Author

Dave Brown, N2RJT, `dcb@vectorbd.com`

Description

Split screen `ttylink' style program with scroll back and log to file.

Status

Version 1.4.1, stable.

Updated

01 Apr 97

System Requirements

Requires the `*ncurses*' package.

Detail

The *splitscreen* is a unix implementation of the NOS *ttylink* command. It provides a splitscreen display that makes user to user chats much easier than the usual telnet interface. The scroll back

buffer and log to file features are useful when you use the *splitscreen* program to access servic like *convers*.

Where and how to obtain it

The `splitscreen' package is available from: sunsite.unc.edu ham applications directory

Licensing/Copyright etc.

Public Domain

5.9 talk-ax25

Author

Dave Brown, N2RJT, `dcb@vectorbd.com`

Description

Berkeley `talk' client modified to support Linux AX.25

Status

Version 960426, working.

System Requirements

Requires the `*ncurses*' package.

Detail

The `talk-ax25' program is a modifed version of the conventional Berkeley `talk' program to support Linux AX.25 operation.

Where and how to obtain it

The `talk-ax25' program is available from: sunsite.unc.edu ham applications directory or from: ftp.ucsd.edu.

Licensing/Copyright etc.

Berkeley style. Freely distributable, copyright messages must be retained.

5.10 7Plus for Linux

Author

Axel Bauda, DG1BBQ, `DG1BBQ@DB0CL.#HB.DEU.EU`

Description

Uuencode-like file coder for BBS Send and Forward of binary files.

System Requirements

none special

Detail

Provides the ability to 7Plus encode binary files to allow them to be transmitted within AX.25 BBS messages.

Where and how to obtain it

The `7pl217sr' encoder package is available from: sunsite.unc.edu ham applications directory.

Licensing/Copyright etc.

Hamware - Free for use by amateur radio use.

5.11 TNOS

Author

Brian A. Lantz, `brian@lantz.com`

Description

TNOS is a NOS derivative that is being actively supported and which is available for both DOS and Linux. It features many enhancements not found in other NOS versions.

Status

Version 2.21. Stable.

Updated

01 Apr 97

System Requirements

Kernel networking slip support if you want to link the normal Linux network facilities with TNOS. A kiss tnc.

Detail

Brian maintains a World Wide Web server which contains up to date information on TNOS, you can find it at www.lantz.com.

Where and how to obtain it

The current version of TNOS is always available from: ftp.lantz.com, sunsite.unc.edu, or ftp.ucsd.edu.

Licensing/Copyright etc.

Free for use by amateur radio operators and educational oraganisations

5.12 N0ARY Packet BBS for UN*X

TOC

Author

Bob Arasmith, N0ARY, ported to Linux (and others) by Bob Proulx, KF0UW, rwp@fc.hp.com

Description

A packet bbs implemented under UN*X.

Status

ALPHA. Bob is running version ARY-0.9 on both Linux and HP-UX. The next version from N0ARY which is called 4.0 is in alpha test/development release. Bob has not been able to bring this online yet and is still running 0.9+kf0uw mods. Bob has about half it working with Alan Cox's AX.25 kernel which would be the desirable combination.

Updated

01 Apr 97

System Requirements.

Linux installation, C Compiler, plus HAM radio TNC hardware.

Detail

This bbs has an excellent packet user interface. It has a compatible set of commands with the RLI bbs so users will be familiar with it immediately. It then extends the command set to be a very nice natural language style interface (e.g "list at allus about KPC-3"). Many csh style bang commands are supported. Also included is a mail interface to provide a packet to internet gateway.

Currently you need to customize the source code for your installation so you need some C programming proficiency.

Where and How to obtain it.

For a Linux version or for Linux information send mail to `rwp@fc.hp.com`. For a SunOS version contact `bob@arasmith.com`. This code is not packaged for distribution yet since it is not past the alpha stage of development.

A version of the software is available from ftp.funet.fi

Licensing/Copyright

Copyright by Bob Arasmith, N0ARY, but freely redistributable.

Contributed by:

Bob Proulx, KF0UW

5.13 LBBS - Linux BBS message gateway

TOC

Author

Daryl L Miles, G7LED, `dlm@frink.demon.co.uk`

Description

A BBS package designed to operate with either the native AX.25 kernel code or with the Wampes package.

Status

In development, working.

Updated

01 Apr 97

System requirements

Kernel AX.25 support or the Wampes package. sendmail, CNEWS.

Detail

Daryl's aim was to build a platform on which amateurs could build a Packet BBS to smtp gateway for Linux. I'm not sure what the current status is as the last version I could find was dated 30 Jun 1995. The latest binary version supports:

- MBL, FBB batching, FBB compressed, FBB type 1 compressed forwarding with restart facility implemented, deferring, and rejection.
- various Linux transport mechanisms including TCP kernel, AX.25 kernel, Wampes NetROM, AX25 or TCP.
- Simple dialer, to allow connections through AX.25 or NetROM nodes.
- White Pages
- Transient handling of type A (ACK messages).
- SMTP/NNTP<>BBS gatewaying implemented heavily to the extent that the user front end will be accessing the NetNews spool to retrieve articles.
- Message filter mechanism
- Field swapping (like SWAPP.SYS in FBB).
- Multiuser/connect.

To install, I did the following:

```
# cd /usr/src
# mkdir LBBS
# tar xvfz LBBS-0.1.6A.tar.gz
# cd LBBS
# ./Install.sh
```

This created a `/usr/lbbs` directory. The `README.Setup` file suggest an order in which to

configure the BBS.

Where and How to obtain it.

You can obtain the `LBBS' package from: ftp.ucsd.edu.

Licensing/Copyright

Freely distributable for any non-commercial Amateur Radio use only. No warranty. Will be changed to GPL when software is stable.

5.14 MBL/RLI message to NNTP and email converter.

Author

SM0OHI, `pme@it.kth.se`

Description

Software that will convert incoming MBL/RLI messages into either NNTP or RFC-822 formatted mail messages.

Status

Development, not yet released.

Updated

01 Apr 97

System requirements

Unknown.

Detail

This software would be ideally suited to those who want to establish a mail and news gateway between conventional tcp/ip networks and the amateur radio mail network.

Where and How to obtain it.

Not yet availale.

Licensing/Copyright

Unknown.

Contributed by:

SM0OHI

5.15 CLX Packet DX Cluster Program

TOC

Author

Franta Bendl, DJ0ZY, and Bernhard ("Ben") Buettner, DL6RAI, `root@dl6rai.muc.de`

Description

A PacketCluster like system running on Linux.

Status

Current version 2.07

Updated

01 Apr 97

System Requirements

Linux Kernel version 2.0 or greater, TCP/IP networking, SysV IPC. Also requires Perl and optionally Expect.

Detail

PacketCluster nodes have been available for about seven years. The original software was written by Dick Newell, AK1A, and is running under DOS. CLX is a system which clones a PacketCluster node. To the outside user commands and features are mostly identical. Remote PacketCluster nodes, which can be networked, don't see the difference between a generic PacketCluster node and CLX. So it fits well into an established network of PacketCluster nodes.

CLX is not a user application, it is network node software. As such it is not of much use for the individual amateur. SysOPs of PacketCluster nodes with some experience on both PacketCluster and Linux will find this application interesting.

Here are some of the features of CLX:

- It is fully compatible with the internal PCxx protocol and from a users point of view. It allows multitasking and different priorities.
- The software detects looping DX spots without generating endless messages. CLX filters out such duplicate information. We have implemented a concept which allows us to connect the network at multiple points, which would formerly have been called a "Cluster Loop". CLX knows of so-called "active" and "passive" links. The outcome is that more DX information is seen by CLX users.
- An interface for extending CLX with your own programs and applications.
- Extended connectivity through the use of connect scripts. CLX now can build up and maintain virtually any connection using TCP/IP, UUCP or any other communication programs.
- CLX uses modern programming concepts like shared memory for common data, lex and yacc to implement command language, shared libraries to keep binaries small, remote procedure calls for interprocess communication. It is strictly modularized with its own

dedicated process for every major task.

- Comprises of several modules running as separate processes.

Where and How to obtain it

You can obtain the software from ftp.funet.fi

There are five files that make up the package:

```
00Index
clx_206.tgz
clx.txt
README
```

Licensing/Copyright

CLX (c) by Franta Bendl (DJ0ZY) and is available for non-commercial use, but the authors want to keep track of who is using their product, so before you can actually use the software you have to ask for an encrypted callsign string to be placed in the configuration file. You must send mail to `clx_us@dl6rai.muc.de` to organise this.

Contributed by

Bernhard ("Ben") Buettner, DL6RAI

5.16 DPTNT Terminal and DPBOX BBS package.

Author

Mark Wahl, DL4YBG, `DL4YBG@DB0GR.#BLN.DEU.EU`, `wahlm@berlin.snafu.de` and Joachim Schurig, DL8HBS, `DL8HBS@DB0GR.#BLN.DEU.EU`, `h0187akk@rz.hu-berlin.de`

Description

A suite of programs that provide BBS and Terminal programs for Linux.

Status

Stable and working.

Updated

01 Apr 97

System requirements

TNC supporting WA8DED hostmode or KISS protocols.

Detail

TNT is a full featured hostmode terminal program. Compiling it on Linux is as difficult as untarring the source and typing `make'. It provides both a `dumb' terminal and X11 version. It comes with some comprehensive documentation which describes its features in some detail. Its

main features are:

Multiple sessions:

TNT supports multiple packet radio connections on virtual screens that you can switch between. Each session window provides split screen (seperate transmit and receive text) operation with a status line. Commands can be entered in either a command session, or in any terminal session using a command key. Each of the virtual screens can be larger than the physical screen and can be scrolled around.

Data capture, file transfer and logging:

A number of options are available for logging received text to files. You can log transmit, receive or both to a log file. You can overwrite or append received data to existing files. You can use the `autobin' protocol to transmit or receive binary files.

Remote Shell operation:

You can provide a shell to remote users so that they can access other programs on your computer. You can also run a program and direct its input/output to a channel so that users can use it.

Redirection of devices to a channel:

TNT allows you to redirect input/output from a channel to a device, a modem for example.

Umlaut conversion:

TNT provides umlaut conversion if necessary.

Remote mode:

Remote users can issue command themselves if allowed to do so.

Socket mode:

You can configure tnt so that it opens a socket for users to telnet to that allows them operation just as you have from the console. This is useful where your linux machine is on an ethernet and you want other terminal on the net to have use of your radio.

New versions will support the Linux kernel AX.25 code.

DPBOX is a full featured BBS supporting the following features:

- F6FBB, W0RLI, TheBox style forwarding protocols.
- 8-bit transparent handling of messages using AutoBIN protocol.
- BBS data compressed to save space on disk, forwarding and downloading.
- "Live message capture" off BBS frequency, saves network traffic.
- PACSAT style bulletin broadcast transmitter and receiver.
- up to 200 simultaneous users.

Where and How to obtain it.

The ``dptnt'' package may be found at: ftp.funet.fi, or ftp.tapr.org, or ftp.ucsd.edu.

The Home Pages of the developers are at: DL4YBG and DL8HBS

Licensing/Copyright

The TNT and DPBOXT programs are covered by the GNU Public License. The TFKISS program is covered by ALAS (a german license for Amateur Radio use), and the DPBOX program is freely distributable for Amateur Radio use.

5.17 IPIP encapsulation daemon.

Name

Mike Westerhof's IPIP encapsulation daemon.

Author

Mike Westerhof KA9WSB (original code), then Bdale Garbee N3EUA (port to BSD), then Ron Atkinson N8FOW and John Paul Morrison (port to Linux), then Bdale rolled it all back together again.

Description

A daemon that will allow you to use your linux machine as an IPIP encapsulating gateway. *Note, this probably isn't necessary any more in most installations because recent kernel support a kernel device to perform encapsulation.*

Status

Ron is running this code himself, and the other version have been run quite successfully.

Updated

01 Apr 97

System requirements

KISS TNC, any version of Linux supporting RAW sockets.

Detail

If you've ever used an internet gateway from your amateur packet radio you have probably been connected by an encapsulation gateway of some descpription. Most run KA9Q's NOS and DOS, but others run Unix and this daemon. It allows you to encapsulate IP within IP, so that you can 'tunnel' IP connections over the Internet. The software allows you to connect a KISS TNC to your linux machine and to have all datagrams received on it carried across the Internet to a similar gateway. Bdale has installation instructions in the package.

Where and How to obtain it.

This software is available from the col.hp.com ftp site.

Licensing/Copyright etc.

Freely distributable, though Bdale asks that if you use the code and like it you might send him a QSL card or a postcard.

Contributed by:

Ron Atkinson, N8FOW, and Bdale Garbee, N3EUA

Note:

You may want to consider using kernel based tunnel daemon now instead.

5.18 AXIP encapsulation daemon.

Name

Mike Westerhof's AXIP encapsulation daemon.

Author

Mike Westerhof KA9WSB, ported to Linux by Ron Atkinson N8FOW.

Description

A daemon that will allow you to use your linux machine as an AXIP encapsulating gateway.

Status

Not properly tested, but assumed to work ok.

Updated

01 Apr 97

System requirements

KISS TNC, any version of Linux supporting RAW sockets.

Detail

This daemon is the partner to the IPIP encapsulation daemon. It allows you to encapsulate AX.25 frames in IP to carry them across the Internet. This is useful for linking AX.25 networks in remote areas. Ron has supplied a `Makefile` for linux, and with it the software compiled without errors on my system running a recent version kernel. Ron has also written some `README` files which give you the detail necessary to compile the system and the original documentation describes how to configure it.

Where and How to obtain it.

This software is available in the sunsite.unc.edu ham apps directory. An enhanced version of this software is included in recent versions of the Linux AX.25 utilities. Refer to the AX25-HOWTO for more information.

Licensing/Copyright etc.

Freely distributable so long as the original copyright notice is not removed.

Contributed by:

Ron Atkinson, N8FOW

5.19 Ping-Pong Convers Server

Name

Fred Baumgartens Convers Server for Linux.

Author

Fred Baumgarten, DC6IQ, `dc6iq@insu1.etec.uni-karlsruhe.de`, and Brian Lantz `brian@lantz.com`.

Description

This is a version of the convers server that allows multiple users to simultaneously chat with each other in a round-table style conversation. It is compatible with the NOS convers servers, but provides extra facilities such as saving/restoring of Personal Text and Chanel Topics. Brian Lantz has further enhanced the software with additional features.

Status

In use of a number of popular convers servers, appears very stable.

Updated

01 Apr 97

System requirements.

Linux, GNU make, C compiler.

Detail.

Complete installation instructions are included in the `INSTALL` file that is included in the distribution.

Where and How to obtain it.

A distribution of this software is available at sunsite.unc.edu. The home of the software is at insu1.etec.uni-karlsruhe.de ftp server

Licensing/Copyright etc.

Presumably copyright Fred Baumgarten (not specifically stated), but some portions appear to be free for non-commercial use and copying provided the copyright notices stay intact.

5.20 RSPF Daemon

Name

RSPF - Radio Shortest Path First routing daemon for Linux

Author

Craig Small vk2xlz `csmall@triode.apana.org.au`

Description

An implementation of the Radio Shortest path First routing protocol for linux. The daemon supports version 2.2 of the protocol which corrects a lot of bugs and problems that were present in version 2.1.

Status

Alpha - running, needs testing and bug reports. Current version is 0.08

Updated

01 Apr 97

System requirements.

Requires Alan Cox's kernel based AX.25 code.

Detail

This program will allow Linux to become an RSPF router. It addresses some of the quirks of RSPF v2.1, the version in NOS which means that it is incompatible with the NOS version. It will also allows remote/local queries via a tcp session for debugging and diagnostic purposes.

Where and How to obtain it.

Current versions are always available from sunsite.unc.edu or ftp.ucsd.edu or from the author.

Licensing/Copyright

GNU Public License vers 2.0

Contributed by:

Craig Small, VK2XLZ

5.21 Michael Westfall's TTYLINK Daemon

Name

ttylinkd - a ttylink daemon for Linux.

Author

Michael Westfall, N6KUY, `mwestfal@.csci.csusb.edu`

Description

A split screen terminal program that listens on the ttylink port and accepts incoming connection requests.

Status

Seems to work fine.

Updated

01 Apr 97

System requirements

none special.

Detail

This is a standalone program that does require you to change the configuration of your system at all. If you have root access on your machine then you run the program in a VT or in an XTerm and it will clear the screen and wait for an incoming connection on port 87 (ttylink). When a connection is received it will provide you with a split screen display with which to communicate with the calling person.

Where and How to obtain it.

You can obtain ttylinkd from: sunsite.unc.edu or ftp.ucsd.edu.

Licensing/Copyright

Unknown, presumed freely distributable.

5.22 Craig Small's TTYLINK Daemon

Name

ttylinkd - a ttylink daemon for Linux.

Author

Craig Small, VK2XLZ, `csmall@triode.apana.org.au`

Description

A daemon to accept a ttylink (port 87) tcp/ip connection and convert it into a talk request for a user on the host machine. This may also be configured for use with the *ax25d* program so that AX.25 users may also make use of it to connect to you.

Status

Alpha, but working.

Updated

01 Apr 97

System requirements.

Kernel networking support with AX.25 support if desired.

Detail

Because the `talk` protocol is a non-standard, that is it isn't actually defined anywhere (except the source), the `ttylink` protocol from NOS has become the defacto standard, at least for amateurs, to provide keyboard to keyboard communication over tcp/ip. This daemon fills the need for something to answer ttylink requests to linux machines.

Where and How to obtain it.

You can obtain the latest version of ttylinkd from: sunsite.unc.edu or ftp.ucsd.edu.

An slightly enhanced version of this software is now included in the Linux AX25 utilities pac

Licensing/Copyright

GNU Public License vers 2.0

6. Morse Code

Software for use in conjunction with, or for facilitating Morse communication.

6.1 Pileup

Author

Richard Everitt G4ZFE `richard@babbage.demon.co.uk`. Based on *sccw* by Steve Merrifield.

Description

CW pileup training program

Status

v1.0. Stable but extra development planned.

Updated

01 Apr 97

System requirements

Linux supported sound card

Detail

CW pileup trainer (similar to PED) based on *sccw*. Up to 9 stations calling at once with variable CW speed and tone/volume to simulate a pileup. Callsigns are taken from the 1996 CQ WW CW contest entry of M6A to add some realism.

I fed /usr/dict/words to the program and it was quite a chuckle. So even if you aren't into Morse it is worth trying out just for the giggle.

Where and How to obtain it.

You can obtain the `pileup` package from its home site at: Pileup Home Page or sunsite.unc.edu

Licensing/Copyright

GNU GPL

Contributed by

Richard Everitt G4ZFE `richard@babbage.demon.co.uk`.

6.2 bip

Author

Marcin Skubiszewski, `Marcin.Skubiszewski@inria.fr`. 24 rue de l'Oasis, 92800 Puteaux, France

Description

A program to learn Morse code (for various Unix-like platforms)

Status

Production release

Updated

01 Apr 97

System requirements

Works on many Unix-like systems, including Linux. On linux it works with or without X11 or a Soundcard.

Detail

bip offers a very wide range of options that make it very suitable as a code practice program. It can send text from text file or random character groups but it has a range of other features. Some of these are

- Morse compliant with ARRL specification
- adjustable volume and tone
- adjustable speed, including Farenheit method
- Jitter feature to simulate hand sent morse

Where and How to obtain it.

Not available online, but anyone can write to `Marcin.Skubiszewski@inria.fr` for a copy of the source code.

Licensing/Copyright

Shareware $10.

Contributed by

Marcin Skubiszewski, `Marcin.Skubiszewski@inria.fr`.

6.3 cw2hex

Author

G. Forrest Cook, WB0RIO, `cook@stout.atd.ucar.edu`

Description

TOC

`cw2hex` converts ascii text into Hex files for making keyer EPROMS.

Status

version 1.1 is tested and ready to go.

Updated

01 Apr 97

System requirements

standalone textual program, requires gcc to compile

Detail

This program takes a textual input and converts it to morse encoded characters in the form of a standard Intel HEX file. The Hex file can be used to program an EPROM which can then be connected to a simple circuit that generates canned messages. Possible uses are CW calling machines and CW beacon transmitter keyers. There is an accompanying schematic for a keyer circuit that works with this software.

Where and How to obtain it.

You can obtain `cw2hex` from www.atd.ucar.edu

Licensing/Copyright

GNU Copyleft

Contributed by:

G. Forrest Cook, WB0RIO

6.4 SoundCard CW

Author

Steve Merrifield, VK3ESM, `sjm@ee.latrobe.edu.au`

Description

Morse Code practice for Linux, using a soundcard

Status

Completed

Updated

01 Apr 97

System requirements

Linux supported soundcard with FM support.

Detail

This program generates morse code characters using the soundcard. It does not rely on the Lin virtual console sound features and may therefore be used either with X11 or from a virtual cor Its features include:

- adjustable frequency
- adjustable volume
- adjustable speed
- adjustable delay
- adjustable group count
- generation of random groups
- able to decode and play text files

Where and How to obtain it.

The software may be obtained from its home site at: livewire.ee.latrobe.edu.au.

Copyright/Licensing

Berkeley style copyright. The software may be freely used so long as credit is given to the original author.

Contributed by:

Steve Merrifield, VK3ESM

6.5 GW4PTS Morse trainer.

Author

Alan Cox, GW4PTS, `alan@lxorguk.ukuu.org.uk`

Description

A Morse Code trainer that uses the PC internal speaker

Status

stable, works quite well, unfinished.

Updated

01 Apr 97

System requirements

Linux, any version

Detail

Alan wrote this small program in only an hour. It is quite neat, and allows you to play morse at a range of speeds and frequencies through the PC internal speaker. You can specify the text to be played either from the command line, from a file, or the program is capable of generating random character groups. As it stands you must invoke the program from a Linux Virtual Console, as it

relies on certain kernel calls to produce the sound, and these don't work as easily from an XTe As it sounds each character it lists the character in verbal form (Di, Dit, Dah etc.) to the screen Alan is hoping that someone will take the code and enhance it with the features he has listed in the comments at the head of the source file.

Where and How to obtain it.

I've had a large number of responses from people seeking this code, so I've obtained Alan's permission to make it available. You can obtain it from sunsite.unc.edu.

Copyright/Licensing

GNU Public License 2, freely redistributable, no warranty.

6.6 morse (aka superiormorse)

Author

Joe Dellinger, `joe@sep.stanford.edu`, `jdellinger@trc.amoco.com`, `joe@soest.hawaii.edu`

Description

A morse-code practice program for UNIX workstations.

Status

Quite stable

Updated

01 Apr 97

System Requirements

Currently supports X11, Sun4, HP, Indigo, and Linux

Detail

Lots of features and options. Includes a utility for generating random content QSOs, similar to those used in the FCC exams.

Where and How to obtain it:

`morse.tar.gz` can be obtained from: sepftp.standford.edu.

Licensing/Copyright

Freely Redistributable

Contributed by:

Brian Suggs, AC6GV, and John Gotts

7. AMTOR Software.

Software for use in conjunction with, or for facilitating AMTOR.

8. PACTOR Software.

Software for use in conjunction with, or for facilitating PACTOR.

9. Slow Scan Television Software.

Software for use in conjunction with, or for facilitating Slow Scan Television.

10. Facsimile Software.

Software for use in conjunction with, or for facilitating Facsimile.

11. Design and Construction Software.

Software to assist in the design and construction of amateur radio related things. Antenna, Circuit Board, Filter, and QSL card design packages are all good candidates for this section.

11.1 oscope - Enhanced Software Oscillosope

Author

Tim Witham, `twitham@pcocd2.intel.com`

Description

`oscope' is a digital 44kHz oscilloscope for /dev/dsp. It is based on Jeff Tranter's original work described below but has been significantly enhanced.

Status

First production release.

Updated

01 Apr 97

System requirements

Requires a kernel supported sound card that can do 8800Hz - 44kHz, the `*libsx*' package to build the *xoscope* X11 client, the `*svgalib*' package to build console-based *oscope*, the `*g3vga*' package to add console text, and a mixing program to select input source and level.

Detail

Includes oscope for SVGA console display and xoscope, an X11 client. Features include 8-bit sampling of 2 channels at 44kHz, 5 us/div to 5 ms/div time scale, 5 automatic measurements, built-in and external math functions including sum, diff, avg, and FFT, 26 memory buffers, 8 simultaneous signal displays, and file save/load.

Where and How to obtain it.

The `oscope` package may be obtained from either www.bobsplace.com or sunsite.unc.edu.

Licensing/Copyright

GNU Public License, Freely redistributable, No warranty.

Contributed by:

Tim Witham

11.2 Software Oscilloscope

Author

Jeff Tranter, `Jeff_Tranter@Mitel.COM`

Description

Scope is a simple software emulation of an oscilloscope. It graphically displays voltage as a function of time.

Status

ALPHA. First release.

Updated

01 Apr 97

System requirements

Sound card with input capability supported by the kernel sound driver. SVGALIB is used to do the display work.

Detail

Scope uses the `/dev/dsp` device to take audio in from the soundcard and displays it on the screen in a manner similar to an oscilloscope. Jeff claims Scope was written more for amusement value than for any serious purpose.

Where and How to obtain it.

You can obtain source, makefile and man page for Scope in the `scope` package from: sunsite.unc.edu

Licensing/Copyright

GNU Public License, Freely redistributable, No warranty.

11.3 Printed Circuit Board design tool.

Author

Thomas Nau, `Thomas.Nau@rz.uni-ulm.de`

Description

An X11 based interactive printed circuit board design tool.

Status

Vers 1.4.1, stable.

Updated

01 Apr 97

TOC

System requirements

X11 Windowing system and a recent flex program.

Detail

The pcb package comes with good instructions on how to compile and install the software. I compiled it without error under X11R6. It comes with package layouts for a range of popular components and the example layout nicely demonstrates the capabilities of the package.

Where and How to obtain it.

You can obtain the `pcb` package from: ftp.uni-ulm.de or from: sunsite.unc.edu

Licensing/Copyright

GNU Public License

11.4 Chipmunk circuit design and simulation tool

Author

Dave Gillespie `daveg@synaptics.com` and John Lazzaro `lazzaro@cs.berkeley.edu`

Description

Schematic Entry and Digital or Analog Simulation tool suite.

Status

vers 5.10, stable.

Updated

01 Apr 97

System requirements

X11 windowing system and older (XFree86-2.x) libs, 8 bpp (256 color) only

Detail

I haven't tried this one out yet. It is a suite of tools for circuit design and simulation.

Where and How to obtain it.

You can obtain the Chipmunk package from its home site at www.pcmp.caltech.edu.

Licensing/Copyright

Modified GNU Public License, Caltech specifically disclaimed from liability.

TOC

11.5 irsim

Author

Dmitry Teytelman, `dim@leland.stanford.edu`

Description

An event-driven logic-level simulator for MOS circuits

Status

Version 9.2, production.

Updated

01 Apr 97

System Requirements

X-Windows.

Detail

irsim is an X11 based simulator for MOS circuits. It has two simulation modes, either *switch* where each transistor is modelled as a voltage controlled switch, or *linear* where each transistor is modelled as a resistor in series with a voltage controlled switch, and each node has a capacitance.

Where and How to obtain it.

You can obtain the `irsim` package from: sunsite.unc.edu.

Licensing/Copyright

Freely Redistributable

11.6 Spice vers. 3f4

Author

University of California, Berkeley, port by `Jeff@RyeHam.EE.Ryerson.Ca`

Description

Spice is an analog circuit emulator.

Status

Stable release.

Updated

01 Apr 97

System requirements

Unknown.

Detail

Spice allows you to design and test circuits in a computer modelled environment to see how they will behave without having to touch a soldering iron, or solder. A `readme` file comes with the package. To install I did the following:

```
# cd /usr/src
# gzip -dc spice3f4.tar.gz | tar xvf -
# cd spice3f4
# ./utils/build linux
# ./utils/build linux install
```

Where and How to obtain it.

You can obtain version 3f4 of the `spice` package from: sunsite.unc.edu

Licensing/Copyright

Copyright held by University California, Berkeley. Freely redistributable provided you are not unfriendly to the U.S.

11.7 svgafft - Spectrum analyser

Author

Andrew Veliath, `veliaa@rpi.edu`

Description

svgafft is a spectrum analyser for Linux.

Status

Alpha, but usable.

Updated

01 Apr 97

System requirements

F77, Linux supported sound card, svgalib, a 486DX33 or better.

Detail

svgafft uses the `/dev/dsp` device and currently supports 16-bit and 8-bit sample resolution on supported Linux soundcards. It's display is something akin to a high-end spectrum analyzer with falling peaks. Andrew is working on an X-Motif version to be released soon.

Where and How to obtain it:

You can find the `svgafft` package at tsx-11.mit.edu and sunsite.unc.edu.

Licensing/Copyright

GNU Public License.

Contributed by:

Andrew Veliath

11.8 Audio Spectrum Analyser

Author

Philip VanBaren, `phillipv@eecs.umich.edu`

Description

An audio spectrum analyser for Linux SVGALIB or X-Windows.

Status

Alpha but usable

Updated

01 Apr 97

System requirements

Linux supported sound card, svgalib or X-Windows.

Detail

freq5a1 uses the `/dev/dsp` device, samples the incoming audio, performs a fast fourier transform on the data and displays the output in either 640x480 VGA or an X-Window.

Where and How to obtain it:

You can obtain `freq5a1.tar.gz` from: sunsite.unc.edu or by mail from the author.

Licensing/Copyright

GNU Public License vers 2.0.

11.9 ObjectProDSP

Author

Paul Budnik, Mountain Math Software, `support@mtnmath.com`

Description

ObjectProDSP is an object oriented tool for design, development and implementation of Digital Signal Processing. It allows you to design your DSP network, runs simulations with oscilloscope style plots and listings at any point in the network.

Status

Alpha version 0.2.

TOC

Updated

01 Apr 97

System requirements

XFree86 3.1.1 or better, g++ from gcc 2.6.3 or better, 32Mb of total memory (real+swap) recommended, 6Mb of disk space for minimum configuration, 40Mb disk for full installation.

Detail

There are a number of files that comprise the package. There is a binary only package for Linux which I recommend you start with. There is comprehensive documentation included from which you can obtain installation and operation instructions.

Where and How to obtain it.

You can obtain ObjectProDSP from: sunsite.unc.edu, tsx-11.mit.edu, or ftp.funet.fi. There are a number of files that comprise the package, some of the quite large so be prepared.

Licensing/Copyright

GNU Public License vers 2.0., Commercial licensing available.

11.10 bpf - bandpass filter calculator

Author

G. Forrest Cook, WB0RIO, `cook@stout.atd.ucar.edu`

Description

`bpf` assists in the design of two pole bandpass filters.

Status

Stable.

Updated

01 Apr 97

System requirements

standalone textual program, requires gcc to compile

Detail

bpf is a simple text interface program that offers you a menu of options of different ways of inputting design data. After you've entered the data it presents a text based circuit diagram and a list of the appropriate component values. It sounds simple, and it is, but it's also quite cute, and bound to be useful.

Where and How to obtain it.

You can obtain `bpf.tar.Z` from www.atd.ucar.edu

Licensing/Copyright

GNU Copyleft

12. Training/Educational Software.

Software to assist in education or training for amateur radio. Morse Code tutorials, technical examination database, Computer Based Training software, and the like are listed here.

13. Miscellaneous Software.

Software that didn't neatly fit into any other category.

13.1 twclock - World Time Clock for Hams

Author

Ted Williams, `ted@bluestone.com`

Description

A world clock designed for ham radio operators

Status

Version 1.0, stable.

Updated

01 Apr 97

System requirements

Motif libraries if you want to recompile it. Just X11 if you want to run the precompiled binaries.

Detail

Anyone who needs the time of day someplace else in the world, or anyone tired of the same old look of their clock will find this software of interest. It is implemented with Motif widgets, and will displays the current time in major cities around the world.

Where and How to obtain it.

You can obtain the `twclock` package from sunsite.unc.edu.

Licensing/Copyright

GNU GPL

13.2 Chirp - Contest Logger

Author

Kenneth E. Harker, N1PVB, `kharker@cs.utexas.edu`

Description

Contest logging program

Status

TOC

Active development

Updated

01 Apr 97

System requirements

Requires ncurses library and header files to compile.

Detail

A linux based contest logging program. Kenneth is writing it so he can use it on his laptop while working VHF contests. He is also writing it as an excuse to learn ncurses programming. His goal is a general purpose contest logger that a user can specify new contests in as they come about. It supports all bands and modes.

Where and How to obtain it.

When version 1.0 is available, you should be able to find it from the Chirp Home Page

Licensing/Copyright

Probably GNU GPL.

Contributed by

Kenneth E. Harker, N1PVB, `kharker@cs.utexas.edu`

13.3 Contest - Contest Logging Program

Author

Carsten Wilderotter, `carsten.wilderotter@student.uni-ulm.de`

Description

A logfile program for Contest operation

Status

Version 0.0.1, Alpha

Updated

01 Apr 97

System requirements

No special requirements.

Detail

Linux contest logging program.

Where and How to obtain it.

You can obtain the `contest` package from: sunsite.unc.edu

Licensing/Copyright

GNU GPL

13.4 Linux for HAMS CD-ROM

Author

Bruce Perens, `bruce@pixar.com`

Description

A complete linux distribution on CDROM specifically designed for Amateur Radio operators.

Status

Not yet available.

Updated

01 Apr 97

System requirements.

Linux compatible hardware, some software will obviously require specialised hardware to operate. No additional software should be required.

Detail

A complete Linux distribution specifically catering for Amateur Radio operators. It will include a fully featured Linux installationplus a range of amateur radio specific software such as those listed in this document. Bruce has detail and updated status report available at the LinuxForHams WWW page. Bruce has been extremely busy with the Debian project and this has reduced his available time to just about zero. He does still intend to pursue this. Why don't you help out by maintaining one of the software packages as Debian packages ? More information from the Debian Web Site.

Where and how to obtain it:

It isn't yet available, details will be made available when it is released.

Licensing/Copyright

GNU Public License.

Contributed by:

Bruce Perens

13.5 SunClock

Author

John Mackin, `john@cs.su.oz.AU`

Description

TOC

A clock that will show you instantly what parts of the globe are exposed by sunlight and what parts aren't.

Status

Released.

Updated

01 Apr 97

System requirements.

X-Windows.

Detail

sunclock is another of those desktop gadgets that most people think look nice but really don't have a lot of use for. I use sunclock to obtain an at-a-glance indication of the time anywhere in the world. In its iconic form it sits in a small Mercator projection. When maximised it produces the same image but obviously larger with slightly more detail. It also displays the date, local time and UTC. sunclock actually calculates mathematically what parts of globe are sunlit and which aren't, it seems quite accurate, so long as you assume the earth has no atmosphere.

Where and How to obtain it.

I haven't seen any precompiled sunclock binaries for Linux about, so check your nearest *archie* server. sunclock compiled straight out of the box for me.

Licensing/Copyright

Public Domain and may be freely copied as long as the notices at the top of sunclock.c remain intact.

13.6 Xearth

Author

Kirk Johnson, `tuna@cag.lcs.mit.edu`

Description

A rotating earth for X-windows root window. It has real life shading, and options for geostatic view, and non geostatic view. A prettier and more modern version of *sunclock* but requires a bit more processing power.

Status

Version 1.0.

Updated

01 Apr 97

System requirements.

X-Windows.

Detail

xearth is a sophisticated program which draws a colour picture of the earth onto your X Windows display. It offers either an orthographic view as you would see from space or the map like mercator projection. You have a number of options in determining the behaviour of the view. The details are available from the xearth Home Page and in the documentation which is supplied with the package.

Where and How to obtain it.

The source can be obtained from its home site at cag.lcs.mit.edu or just about any X11/contrib directory.

Licensing/Copyright

Copyright (C) 1989, 1990, 1993 by Kirk Lauritz Johnson. The copyright notice included states that *xearth* is freely redistributable so long as the copyright notice is left intact, and be included in documentation.

Contributed by:

Kirk Johnson

---

14. How to contribute or update an entry.

I'd like for this list to be as complete and up-to-date as possible. So I'm keen to hear about any developments or products that I don't already know about, or that the entry is obselete or outdated for.

What I'd like as a minimum set of requirement would be something like the following:

Name

The name of the software in question.

Author

Who wrote, or ported the software. An email address, or some other means of contacting them is also essential.

Description

A single line description of what the software does.

Status

An indication of the software's status. Is it still in testing? Is it a production release? Is it still in the design stage?

System requirements

What does the software require to run? Does it require X-Windows? Does it need a soundcard? Does it need a certain version of kernel? Does it need other software to support it?

Detail

I'm not keen on including a large amount of detail on each piece of software as this would consume a lot of time reading and trying to keep up to date. So instead what I'd like to list is what makes this software unique, anything special about it. Perhaps its most outstanding features, that sort of thing.

Where and How to obtain it.

If the software is freely distributable then ftp details would be great. If it is commercial software then the name of the company distributing the software, and an address or telephone number. If it is available only by some other means, say mail order, then details on where and how to obtain it.

Licensing/Copyright

Is the software Copyleft? Copyright? Shareware? Public Domain? Restricted in use in any way?

Don't worry if you don't know all of these details, just send me what you do know and I'll list what I can. I'd rather have an incomplete listing than no listing at all.

Please mail any contributions to:

terry@perf.no.itg.telstra.com.au

I'd list a packet radio address too but I'm still not properly operational again yet after moving house.

TOC

15. Discussion relating to Amateur Radio and Linux.

There are various places that discussion relating to Amateur Radio and Linux take place. They take place in the `comp.os.linux.*` newsgroups, they also take place on the `linux-hams` list on `vger.rutgers.edu`. Other places where they are held include the `tcp-group` mailing list at `ucsd.edu` (the home of amateur radio tcp/ip discussions).

To join the Linux **linux-hams** mail list, send mail to:

```
Majordomo@vger.rutgers.edu
```

with the line:

```
subscribe linux-hams
```

in the message body. The subject line is ignored.

To join the `tcp-group` send mail to:

```
listserver@ucsd.edu
```

with the line:

```
subscribe tcp-group
```

in the body of the text.

Note: Please remember that the `tcp-group` is primarily for discussion of the use of advanced protocols, of which tcp/ip is one, in Amateur Radio. *Linux specific questions should not ordinarily go there.*

16. Copyright.

The HAM-HOWTO, an introduction to Amateur Radio software available for Linux and information on how to configure some of it. Copyright (c) 1995,1996 Terry Dawson.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the:

Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

HOWTO HOWTO

Mark F. Komarinski <markk@cgipc.com>

v0.12, 2 September 1999

Getting a new LDP author up and running with tools, ideas, and conventions used by the LDP

1. Introduction

- 1.1 History
- 1.2 New versions
- 1.3 Copyrights and Trademarks
- 1.4 Acknowledgements and Thanks

2. Background on the LDP and SGML

- 2.1 The LDP
- 2.2 SGML
- 2.3 The tools

3. Getting Started

- 3.1 Mailing lists
- 3.2 Downloading and installing the tools
- 3.3 Writing SGML by hand
- 3.4 Writing SGML using other tools
- 3.5 CVS basics
- 3.6 Distributing your documentation

TOC

4. Style guides

5. FAQs about the LDP

- 5.1 I want to help the LDP. How can I do this?
- 5.2 I want to publish a collection of LDP documents in a book. How is the LDP
- 5.3 I found an error in an LDP document. Can I fix it?

---

1. Introduction

1.1 History

This document was started on Aug 26, 1999 by Mark F. Komarinski markk@cgipc.com after two day's worth of frustration getting tools to work. If even one LDP author is helped by this, then I did my job.

1.2 New versions

The newest version of this can be found on my homepage http://www.cgipc.com/~markk in its SGML source. Other versions may be found in different formats at the LDP homepage http://www.linuxdoc.org/.

Version History

v0.12 (Sep 2, 1999)

- Completed most sections
- Integrated changes from ldp-discuss list

v0.10 (Aug 27, 1999)

- Got up to section 3.4 written
- Added to the outline some
- Changed location of LDP mailing list to lists.debian.org from thepuffingroup.com.

v0.01 (Aug 27, 1999)

- First pass, got web page up, simple outline written.
- Take some of what I wrote with a grain of salt. Some things need to be verified.

1.3 Copyrights and Trademarks

(c) 1999 Mark F. Komarinski

This document may be distributed under the terms set forth in the LDP license at http://www.linuxdoc.org/COPYRIGHT.html.

1.4 Acknowledgements and Thanks

TOC

Thanks to everyone that gave comments as I was writing this. This includes Deb Richardson and Daniel Barlow and other members of the ldp-discuss list.

Some sections I got from the HOWTO Index (available at many LDP locations) and the sgmltools documentation. There are pointers to sgmltools and the LDP elsewhere in this document.

---

2. Background on the LDP and SGML

2.1 The LDP

The Linux Documentation Project (LDP) was started to provide new users a way of getting information quickly about a particular subject. It not only contains a series of books on administration, networking, and programming, but has a large number of smaller works on individual subjects, written by those who have used it. If you want to find out about printing, you get the Printing HOWTO. If you want to do some networking, grab the Ethernet HOWTO, and so on.

At first, many of these works were in text or HTML. As time went on, there had to be a better way of managing these documents. One that would let you read it from a web page, a text file on a CD-ROM, or even your handheld PDA. The answer, as it turns out, is SGML.

2.2 SGML

The Standard Generalized Markup Language (SGML) is a language that is based on marking up text. In this way, its similar to Tex or groff, or HTML. The power of SGML is that unlike WYSIWYG (What You See Is What You Get), you don't define things like colors, or font sizes, or even some kinds of formatting. Instead, you define elements (paragraph, section, numbered list) and let the SGML processor and the end program worry about placement, colors, fonts, and so on. HTML does the same thing, and is actually a subset of SGML.

SGML has really two parts that make it up. First is the Structure, which is what is commonly called the DTD, or Document Type Definition. The DTD defines the relationship between each of the elements. The LinuxDoc DTD, used to create this document, is an example of this. The DTD gives a common look and feel to each document that's created using the DTD. Second is the Content, which is what gets rendered by the SGML processor and is eventually seen by the user. This paragraph is content, but so would a graphic image, table, numbered list, and so on. Content is surrounded by tags to separate out each different element.

Over time, the LinuxDoc DTD is going to change over to the DocBook DTD, used by others and giving the LDP a consistent look and feel to other SGML documentation. As this happens, we'll keep you updated via this HOWTO or on the mailing lists. The biggest difference between LinuxDoc and DocBook is that DocBook assigns tags to different types of content (such as commands, file names, directories, and so on) while LinuxDoc assigns tags based on the way the text should look (you can assign emphasized or typewriter for example)

TOC

Why SGML instead of HTML or other formats?

SGML provides for more than just formatting. You can automatically build indexes, table of contents, and links within the document or to outside. The sgmltools package also lets you export (I'll call it render from here on) SGML to LaTeX, info, text, HTML, and RTF. From these basic formats, you can then create other formats (DOC, PostScript, and so on). SGML doesn't suffer from some of the bloating seen in HTML of late. I don't think you'll be seeing a <blink> tag in SGML anytime soon. This makes the code that comes out not only easy to render, but easy to write as well. Programs like LyX (right now my WYSIWYM editor of choice) allow you to write in TeX format, then export it as SGML and render from SGML to whatever you chose.

In the end, SGML is more concerned about the way elements work instead of the way they look. A big distinction, and one that will let you write faster, since you don't have to worry about placement of paragraphs, font sizes, font types, and so on.

2.3 The tools

In this section, I'll go over some of the tools that you'll need or want to use to create your own LDP documentation. I'll describe them here, and better define them later on, along with how to install them. If you use some other tool to assist in writing LDP, please let me know and I'll add a blurb here for it.

sgmltools

Required

The sgmltools package contains the SGML tools needed to render SGML as any of the file formats listed above. It also contains the LinuxDoc DTD, needed to make LDP documentation. To create only SGML documentation, this is all you need. If you want to render to formats like TeX, you'll need to get those packages as well. The sgmltools package is available either with your distribution of choice, or via http://www.sgmltools.com/

TeX

Optional

TeX (rhymes with blech!) is the markup language of choice for many, including those in the mathematics world. I still remember many Calculus exams that were actually written in TeX. It is also one of the first markup languages that is still around (the other being the *roff formats used in man pages). TeX actually follows some of the same concepts that SGML does. However, TeX renders its files into DVI (Device Independent) that can then be rendered into another format. Unfortunately, DVI can't be easily converted into anything other than printer languages (PostScript, PCL), making it hard to use to generate HTML. TeX is installed or is available with most Linux distributions. TeX is available on almost all distributions as LaTeX or TeTeX. Either should work for you.

LyX

Optional

The LyX program is a graphical WYSIWYM (What You See Is What You Mean) and provides a much-needed link between an easy-to-use graphical app and renderer and the sometimes-complex rules of SGML. LyX was really used to write TeX documentation, and many of the TeX rules apply in LyX. For example, while sections are automatically numbered, you can't insert whitespace (spaces and tabs) easily. It's against what TeX was designed to do. As it is, SGML often ignore the same whitespace. The LyX program can read the LinuxDoc DTD and provide a template document for you to write (or edit) your LDP documentation in a way that you're familiar with, without having to use vi and remember what the tags are for itemizing a list. LyX is available at http://www.lyx.org/.

3. Getting Started

This section shows how to get involved in writing your own LDP documentation. Getting and setting up the tools, making contact with the LDP in general, and distributing what you know to all the Linux users out there.

3.1 Mailing lists

There are a few mailing lists to subscribe to so you can take part in how the LDP works. First is ldp-discuss@lists.linuxdoc.org, which is the main discussion group of the LDP. To subscribe, send a message with the subject reading "subscribe" to ldp-discuss-request@lists.linuxdoc.org. To unsubscribe, send an e-mail with the subject of "unsubscribe" to ldp-discuss-request@lists.linuxdoc.org.

3.2 Downloading and installing the tools

sgmltools

Download the sgmltools package from http://www.sgmltools.org/, or directly from your distribution. The source files from sgmltools.org is in source code format, so you will have to compile the source code for your machine. Using a pre-built package for your distribution is easier, as you don't have to compile it and potentially run into compilation issues (that is, if you're not a coder).

With RedHat, the sgmltools is included with the distribution. If not, you can download it from ftp.redhat.com or any of its mirrors as part of the main distribution.

If you're using Debian, it too has sgmltools in the standard distribution. If you don't have the package installed, you can use the apt-get command to download and install the package for you:

```
# apt-get install sgml-tools
```

For more information on the Debian package, you can look at http://www.debian.org/Packages/stable/text/sgml-tools.html

If compiling from source, all you need to do is:

```
# tar -zxvf sgmltools-x.x.x.tar.gz
# cd sgmltools-x.x.x
# ./configure
```

```
# make
# make install
```


Replace `sgmltools-x.x.x` with the actual version of the sgmltools package you're using. The current version as of this writing that supports LinuxDoc is 1.0.9. The version that supports DocBook is 2.0.2. Both are available at the above web site.

Once the tools are installed, you have a number of commands available to you.

`sgmlcheck file.sgml`- Checks the syntax of a given document.

`sgml2html file.sgml`- Converts an SGML file into HTML. Creates a `file.html` file that contains the Table Of Contents, then creates `file-x.html` files where `x` is the section number.

`sgml2rtf file.sgml`- Converts an SGML file into Rich Text Format (RTF). Creates two files, the first being `file.rtf` that contains the TOC, and a `file-0.rtf` that contains all the sections.

`sgml2txt file.sgml`- Converts an SGML file into ASCII text. The TOC and all sections are all put into `file.txt`.

`sgml2info file.sgml`- Blah SGML blah INFO, used by the info command. All output is sent to `file.info`.

`sgml2latex file.sgml`- Blah SGML blah TeX.

`sgml2lyx file.sgml`- SGML yadda LyX graphical editor. This is great if you have pre-generated SGML files and want to convert them for use in LyX.

3.3 Writing SGML by hand

Much like HTML, you can write SGML by hand, once you know all the markup codes you want to use. This section will go over as many of these codes as possible, along with practical examples of each. A nice place to start would be the SGML source for this document, which is available at the web site in the Introduction. As the SGML may be processed differently depending on the file format you go to, I'll try to list some things to know about as you're writing.

Starting out

To start a new document, create a new file in your favorite ASCII editor and start with this:

```
<!doctype linuxdoc system>
```

This defines the document type (LinuxDoc in our case) that the SGML processor will use when it renders the file in an output format. Nothing is rendered from this tag.

TOC

Next you need to enclose the rest of your work in `<article>` and `</article>` tags. This signifi the start of the content (or article, eh?). If you're familiar with HTML, this is similar to enclosing all content with `<html>` and `</html>`.

Header information

The first part of the content should contain general information about the rest of the content. This would be similar to the first few pages of a book, where you have a title page (title of the work, author, date of publication, table of contents, and so on).

The title of the content is enclosed in `<title>` and `</title>` tags. The author is specified in `<author>` and `</author>` tags. The date uses `<date>` and `</date>`.

The two remaining sections are the `<abstract>` and `</abstract>` tags, which provide an executive summary of what the content is about, and the `<toc>` tag, which specifies the location of the table of contents. The TOC is automatically generated by the SGML processor. We'll get into sections later on.

Now, how does it all look together? Taking a nice bit of SGML code (that is, what was used to create this document) you'll see:

```
<!doctype linuxdoc system>
<!-- LinuxDoc file was created by LyX 1.0 (C) 1995-1999 by <markk>
 Fri Aug 27 09:42:28 1999 -->
<article>
<title>HOWTO HOWTO
</title>
<author>Mark F. Komarinski
</author>
<date>Aug 27, 1999
</date>
<abstract>Getting a new LDP author up and running with tools, ideas,
 and conventions used by the LDP
</abstract>
<toc>
```

This bit of content created the main page you see when you look at this document in RTF or HTML format, listing all the information on one page.

Sections

In order to build the Table of Contents, you need to have something to build with. Sections in the case of SGML is the same as chapters in traditional publishing. You have multiple sections, and each section can have a subsection, and each of those can have a subsection and so on.

Starting your document with sections is great as it lets you create an outline of the major topics you want to cover. You can then break down these major sections into gradually smaller sections, until you have a

nugget of information you can write about in a few short paragraphs. In writing this document, I actu started this way.

Sections are one of the few sets of SGML tags that don't require to be closed. That is, there is no `</sect>` tag. Nor do you have to worry about numbering. The SGML processor will handle it all when you render the SGML into something else.

Sections are started with `<sect>` tags. A new section is started with each `<sect>` tag. The first section is numbered 1.

Creating subsections (like 1.1) is done with the `<sect1>` tag. It also starts with 1.

Subsubsections (1.1.1) is done with the `<sect2>` tag, and also starts with 1.

When the SGML processor comes across the `<toc>` tag, it runs through the rest of the document and builds the Table Of Contents based on the number of section tags within it. Sections are numbered and listed in the TOC and then used in the rest of the document. Subsubsections (1.1.1) do not show up in the TOC, but are put in emphasized text if possible.

Normal paragraphs

Writing paragraphs of content is just like in HTML. Use a `<p>` tag to specify a new line, and start writing. SGML will ignore whitespace such as tabs, multiple spaces, and newlines. When SGML comes across a `<p>` tag, it starts a new paragraph. Proper SGML has you put in a `</p>` to end the paragraph.

Enhanced Text

Every now and then you need a touch of text to stand out from the others. Either to highlight code or to `list a command name.` The first (emphasizing text) is done with `<em>` and `</em>` tags. Typewriter text (the second example) is done with `<tt>` and `</tt>` tags.

Lists

There are two forms of doing lists under SGML. First is an enumerated list, where each item in the list is numbered (like sections) starting with 1.

1. This is the first entry in the enumerated list.
2. This is the second.
3. Third.

The code for the above list looks like this:

```
<enum>
<item>This is the first entry in the enumerated list.
<item>This is the second.
<item>Third.
</enum>
```

TOC

The `<enum>` tag specifies that the following items are going to be enumerated.

The other method of writing lists is itemized, where each item merely has a star, or circle, or dot, or some other method of itemizing each item.

- This is the first entry in the itemized list
- This is the second
- Third

The above code looks like this in raw SGML:

```
<itemize>
<item>This is the first entry in the itemized list
<item>This is the second.
<item>Third.
</itemize>
```

As you can see, the `<item>` tag is the same for enumerated and itemized lists.

A third form of lists is the description lists. This has a term being described, and the phrase that describes it.

LDP

The Linux Documentation Project

SGML

Standard Generalized Markup Language

The code to create the above descriptions is:

```
<descrip>
<tag>LDP</tag>The Linux Documentation Project
<tag>SGML</tag>Standard
 Generalized Markup Language
</descrip>
```

This isn't quite the same as itemized or enumerated lists, but you have the entire list surrounded by a tag (`<descrip>` and `</descrip>`) and each item in the line that is a word being defined is enclosed in `<tag>` and `</tag>`. The remainder of the line is taken to be the definition of the word.

Verbatim text

Sometimes you just need to print some text the way you write it. For this, you can use the `<verb>` and `</verb>` tags to enclose a paragraph in verbatim mode. Spaces, carriage returns, and other literal text (including special characters) are preserved until the `</verb>`.

```
The following is verbatim text    .
```

URLs

Also in SGML is the ability to handle Universal Resource Locators (URL) of any kind. Note that this would only work when exported to HTML mode, but you'll get some use out of this tag in other formats (does RTF use it too?).

A URL doesn't have an end tag, but puts its information within the `<url>` tag itself. Here is a URL that points to the LDP homepage: http://www.linuxdoc.org/. And here's the code to create it:

```
<url url="http://www.linuxdoc.org/" name="http://www.linuxdoc.org/">
```

The `url="http://www.linuxdoc.org/"` tells the browser where to go, while the contents of the `name="http://www.linuxdoc.org/"` tells the browser what to print out to the screen. In this case, the two are similar, but I could create a URL tag that looks like this:

```
<url url="http://www.linuxdoc.org/" name="LDP">
```

And then looks on the page like this: LDP.

References

While URLs are great for linking to content outside the LDP document you're working on, it's not that great for linking within the content itself. For this, you use the `<label>` and `<ref>` tags. The `<label>` tag creates a point in the document where you want to refer back to later on, almost like a bookmark. Creating the `<label>` is easy. Find the point where you want to refer back to later on, and insert the following:

```
<label id="Introduction">
```

You have now created a point in the content that you can refer to later on as "Introduction". This label actually appears in this SGML work at the front of the document. When you want to refer back to that point later on (say here), you insert the following SGML:

```
<ref id="Introduction" name="here">
```


and the SGML will know to put in a link called "here" (see above) that links back to the location of the Introduction section.

The other part of references is indexing. Since LDP documentation is usually published on paper as a large collection of documents, there needs to be a way of building the index at the back of the book, based on words and subjects.

Special characters

Much like HTML, you will need to escape many non-alphanumeric characters to prevent the SGML processor from interpreting them as SGML code. Here's a list of the SGML codes used. More are listed in the sgmltools User's Guide located at http://www.sgmltools.org/guide/guide.html

- Use & for the ampersand (&)
- Use < for a left bracket (<)
- Use > for a right bracket (>)
- Use &etago; for a left bracket with a slash (</)
- Use $ for a dollar sign ($)
- Use # for a hash (#)
- Use % for a percent (%)
- Use ˜ for a tilde (~)
- Use `` and '' for quotes, or use &dquot for "
- Use for a soft hyphen (that is, an indication that this is a good place to break a word for horizontal justification).

3.4 Writing SGML using other tools

LyX

I'm still gushing aboutLyX. Okay, so I'm a bit biased towards this application because I really like it. It provides the power of writing SGML with the ease-of-use of a regular word processor. It's not a WYSIWYG program, but more WYSIWYM (What You Get Is What You Mean) application, since what you see on the screen isn't necessarily what happens after the SGML processor is done with it.

To create a LinuxDoc document with LyX, download and install the application. Make sure you have TeX and sgmltools installed first (see Installing the Tools for more information on this). Once complete, start up LyX and select "file->new from template..." Select "Templates" then click on `linuxdoctemplate.lyx` and you'll have a template document set up, with most of the header information that an LDP document should have. Change the data to suit your need (that is, fill in the Title, Author, Date, Abstract, and so on) and then start writing. The pull down menu in the upper left hand corner can be used to select types of content (standard, itemized and enumerated lists, sections).

The exclamation point is used to emphasize text, and you can either click it and begin typing in emphasized mode, or highlight text with the mouse and click on it to emphasize the highlighted text. Many other features of SGML can be found under the Insert menu bar. You can insert URL locations, cross references, index entries, and other kinds of data. When complete with your documentation, you can save it in LyX format, then export to LinuxDoc and have the file saved with a .sgml extension. That file is then ready to be checked with sgmlcheck and rendered to the formats you want.

Emacs

I have this thing about Emacs. I don't use it, and it doesn't get me peeved. Anyone with more Emacs experience would be very helpful.

Other SGML tools

If there are other SGML tools out there, or even commercial ones that the LinuxDoc DTD can be used with to create LDP documentation, please let me know.

3.5 CVS basics

At this time, the LDP does not have a shared repository for you to store your content online. Hopefully this will change. There are a few good reasons for using CVS:

1. CVS will keep an off-site backup of your documents. In the event that you hand over a document to another author, they can just retrieve the document from CVS and continue on. In the event you need to go back to a previous version of a document, you can retrieve it as well.
2. It's great if you have many people working on the same document. You can have CVS tell you what changes were made while you were editing your copy by another author, and integrate those changes in.
3. Keeps a log of what changes were made. These logs (and a date stamp) can be placed automatically inside the document when you use some special tags that get processed before the SGML processor.
4. Can provide for a way for a program to automatically update the LDP web site with new documentation as it's written and submitted.

3.6 Distributing your documentation

Before you distribute

Before you distribute your code to millions of potential readers there are a few things you should do.

First, be sure to spell-check your document. Nothing says "Hi, I'm stupid!" faster in the Internet-land than misspellings. Most utilities that you would use to write SGML (emacs, LyX, other text editors) have plug-ins to perform a spell check. If not, there's always the ispell program, installed in just about every distribution. Also use the sgmlcheck command with sgmltools to verify you have correct SGML tags.

TOC

Second, get someone to review your documentation for comments and factual correctness. The documentation that is published by the LDP needs to be as factually correct as possible, as there are millions of Linux users that may be reading it. If you're part of a larger mailing list talking about the subject, ask others from the list to help you out.

Third, create a web site where you can distribute your documentation. This isn't required, but is helpful for people to find the original location of your document.

Submission to LDP

Once your LDP document has been reviewed by a few people and you took into account their comments, you can release a first draft (or a final one) to the LDP in general. Send an e-mail to ldp-submit@lists.linuxdoc.org with your SGML source code. Within 24 hours you should find out if it was accepted and posted to the main LDP site.

---

4. Style guides

This isn't a hard and fast guide on writing SGML (yet), but consider it a bunch of hunts to help you along as you write.

- Be clear. Everyone needs to know what you're talking about.
- Use examples where possible. It lets everyone see what you're talking about.
- Organize. Don't jump between unrelated topics in the same section.

You can get many more hints from the LDP style guide located at http://www.linuxdoc.org/LDP/HOWTO/LDP-Style-Guide.html.

TOC

5. FAQs about the LDP

5.1 I want to help the LDP. How can I do this?

The easiest way is to find something and document it. Also check the unmaintained HOWTOs and see if there is a subject there that you know about and can continue documenting.

5.2 I want to publish a collection of LDP documents in a book. How is the LDP content licensed?

Please see http://www.linuxdoc.org/COPYRIGHT.html.

5.3 I found an error in an LDP document. Can I fix it?

Contact the author of the document, or the LDP coordinator (e-mail?) and mention the problem and how you think it needs to be fixed.

Linux Hardware Compatibility HOWTO

Patrick Reijnen, < antispam.hardware_howto@antispam.reijnen.nl.com (remove both "antispam.")>

v99.2, 20 March 1999

This document lists most of the hardware supported by Linux and helps you locate any necessary drivers.

1. Introduction

- 1.1 Welcome
- 1.2 Copyright
- 1.3 System architectures

2. Computers/Motherboards/BIOS

- 2.1 Specific system/motherboard/BIOS
- 2.2 Unsupported

3. Laptops

- 3.1 Specific laptops
- 3.2 PCMCIA

4. CPU/FPU

5. Memory

6. Video cards

- 6.1 Diamond video cards
- 6.2 SVGALIB (graphics for console)
- 6.3 XFree86 3.3.2
- 6.4 S.u.S.E. X-Server
- 6.5 Commercial X servers

TOC

7. Controllers (hard drive)

- 7.1 Alpha, Beta drivers

8. Controllers (hard drive RAID)

9. Controllers (SCSI)

- 9.1 Supported
- 9.2 Alpha, Beta drivers
- 9.3 Unsupported

10. Controllers (I/O)

11. Controllers (multiport)

- 11.1 Non-intelligent cards
- 11.2 Intelligent cards

12. Network adapters

- 12.1 Supported
- 12.2 Alpha, Beta drivers
- 12.3 Unsupported

13. Sound cards

- 13.1 Supported
- 13.2 Alpha, Beta drivers
- 13.3 Unsupported

14. Hard drives

- 14.1 Unsupported

15. Tape drives

- 15.1 Supported
- 15.2 Alpha, Beta drivers
- 15.3 Unsupported

TOC

16. CD-ROM drives

- 16.1 Supported
- 16.2 Alpha, Beta drivers
- 16.3 Notes

17. CD-Writers

18. Removable drives

19. Mice

- 19.1 Supported
- 19.2 Alpha, Beta drivers
- 19.3 Notes

20. Modems

21. Printers/Plotters

- 21.1 Ghostscript

22. Scanners

- 22.1 Supported
- 22.2 Alpha, Beta drivers
- 22.3 Unsupported

23. Other hardware

- 23.1 Amateur Radio
- 23.2 VESA Power Savings Protocol (DPMS) monitors
- 23.3 Touch screens
- 23.4 Terminals on serial port
- 23.5 Joysticks
- 23.6 Video capture boards / Frame Grabbers / TV tuner
- 23.7 Digital Camera
- 23.8 UPS
- 23.9 Multifunction boards

- 23.10 Data acquisition
- 23.11 Watchdog timer interfaces
- 23.12 Miscellaneous

24. Related sources of information

25. Acknowledgments

26. Appendix A. S3 cards supported by XFree86 3.3.x.

27. Appendix B. Supported PCMCIA cards

- 27.1 Ethernet cards
- 27.2 Fast Ethernet (10/100baseT) adapters
- 27.3 Token-ring adapters
- 27.4 Wireless network adapters
- 27.5 ISDN
- 27.6 Modem and serial cards
- 27.7 Memory cards
- 27.8 SCSI adapters
- 27.9 ATA/IDE CD-ROM adapters
- 27.10 Multifunction cards
- 27.11 ATA/IDE card drives
- 27.12 Miscellaneous cards
- 27.13 Cards with separately distributed drivers
- 27.14 Working on ...
- 27.15 Unsupported

28. Appendix C. Supported Parallel Port devices

- 28.1 Ethernet
- 28.2 Hard drives
- 28.3 Tape drives
- 28.4 CD-ROM drives
- 28.5 Removable drives
- 28.6 IDE Adapter
- 28.7 SCSI Adapters
- 28.8 Digital Camera

TOC

29. Appendix D. Plug and Play devices

30. Appendix E. Linux incompatible Hardware

31. Glossary

---

1. Introduction

NOTE: USB is not yet supported by Linux.

1.1 Welcome

Welcome to the Linux Hardware Compatibility HOWTO. This document lists most of the hardware components (not computers with components build in) supported by Linux, so reading through this document you can choose the components for your own Linux computer. As the list of components supported by Linux is growing rapidly, this document will never be complete. So, when components are not mentioned in this HOWTO, the only reason will be that I don't know they are supported. I simply have not found support for the component and/or nobody has told me about support.

Subsections titled 'Alpha, Beta drivers' list hardware with alpha or beta drivers in varying degrees of usability. Note that some drivers only exist in alpha kernels, so if you see something listed as supported but isn't in your version of the Linux kernel, upgrade.

The latest version of this document can be found on http://users.bart.nl/~patrickr/hardware-howto/Hardware-HOWTO.html, SunSite and all the usual mirror sites. Translations of this and other Linux HOWTO's can be found at http://sunsite.unc.edu/pub/Linux/docs/HOWTO/translations and ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/translations.

If you know of any Linux hardware (in)compatibilities not listed here please let me know, just send mail.

Still need some help selecting components after reading this document? Check the "Build Your Own PC" site at http://www.verinet.com/pc/.

Want to have a preconfigured Linux system? Have a look at http://www.linuxresources.com/web/.

1.2 Copyright

Copyright 1997, 1998, 1999 Patrick Reijnen

This HOWTO is free documentation; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free software Foundation; either version 2 of the license, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details. You can obtain a copy of the GNU General Public License by writing to the Free Software Foundation,, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

If you use this or any other Linux HOWTO's in a commercial distribution, it would be nice to send the authors a complimentary copy of your product.

1.3 System architectures

This document only deals with Linux for Intel platforms, for other platforms check the following:

- ARM Linux
  http://www.arm.uk.linux.org/
- Linux/68k
  http://www.clark.net/pub/lawrencc/linux/index.html
- Linux/8086 (The Embeddable Linux Kernel Subset)
  http://www.linux.org.uk/ELKS-Home/index.html
- Linux/Alpha
  http://www.azstarnet.com/~axplinux/
- Linux/MIPS
  http://www.fnet.fr/linux-mips/
- Linux/PowerPC
  http://www.linuxppc.org/
- Linux for Acorn
  http://www.ph.kcl.ac.uk/~amb/linux.html
- Linux for PowerMac
  http://ftp.sunet.se/pub/os/Linux/mklinux/mkarchive/info/index.html

---

2. Computers/Motherboards/BIOS

ISA, VLB, EISA, and PCI buses are all supported.

PS/2 and Microchannel (MCA) is supported in the standard kernel 2.0.7. There is support for MCA in kernel 2.1.16 and newer, but this code is still a little buggy. For more information you can always look at the Micro Channel Linux Home Page (Where is this page gone to ??)

2.1 Specific system/motherboard/BIOS

- IBM PS/2 MCA systems
- EFA E5TX-AT motherboard has a solvable problem with RedHat Linux 5.0 and possibly other versions of Linux. It spontaneously reboots while probing hardware. To solve, update BIOS to version 1.01. Get the BIOS update at http://www.efacorp.com/download/Motherboard/e5tx101.exe.
- The Edom MP080 motherboard needs a BIOS flash for Linux to work. Without the BIOS flash Linux will reboot during the hardware scan. for the BIOS flash check http://www.edom.com/tech/tech.htm and http://www.edom.com/download

Many new PCI boards are causing a couple of failure messages during boot time when "Probing PCI Hardware". The procedure presents the folowing message

```
Warning : Unknown PCI device (8086:7100).  Please read include/linux/pci.h
```

It tells you to read the pci.h file. From this file is the following quote

```
     PROCEDURE TO REPORT NEW PCI DEVICES
We are trying to collect information on new PCI devices, using
the standard PCI identification procedure. If some warning is
displayed at boot time, please report
     - /proc/pci
     - your exact hardware description. Try to find out
       which device is unknown. It may be you mainboard chipset.
       PCI-CPU bridge or PCI-ISA bridge.
     - If you can't find the actual information in your hardware
       booklet, try to read the references of the chip on the board.
     - Send all that to linux-pcisupport@cao-vlsi.ibp.fr,
       and I'll add your device to the list as soon as possible

BEFORE you send a mail, please check the latest linux releases
to be sure it has not been recently added.

       Thanks
             Frederic Potter.
```

Normally spoken you motherboard and the unknown PCI devices will function correctly.

2.2 Unsupported

- Supermicro P5MMA with BIOS versions 1.36, 1.37 and 1.4. Linux will not boot on this motherboard. A new (beta) release of the BIOS which makes Linux boot, is available at ftp.supermicro.com/mma9051.zip
- Supermicro P5MMA98. Linux will not boot on this motherboard. A new (beta) release of the BIOS which makes Linux boot, is available at ftp.supermicro.com/a98905.zip?
- DataExpert Corp. ExpertColor TX531 V1.0 motherboard with chipset ACER M1531 (Date: 9729, TS6) and ACER M1543 (Date: 9732 TS6) seems not reproducible segmentations faults, kernel oops and kernel hangs under heavy load and tape access. The problem seems to be the PCI-bus, respectively the ACER chipset.

---

3. Laptops

For more information about Linux and laptops, the following site is a good starting point.

- Linux Laptop Homepage
  http://www.cs.utexas.edu/users/kharker/linux-laptop/

Other information related to laptops can be found at the following sites:

- Avanced Power Management
  ftp://ftp.cs.unc.edu/pub/users/faith/linux/
- Notebook battery status
  ftp://sunsite.unc.edu/pub/Linux/system/power/
- non-blinking cursor
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/console/noblink-1.7.tar.gz
- other general info
  ftp://tsx-11.mit.edu/pub/linux/packages/laptops/

3.1 Specific laptops

- Compaq Concerto (pen driver)
  http://www.cs.nmsu.edu/~pfeiffer/
- Compaq Contura Aero
  http://domen.uninett.no/~hta/linux/aero-faq.html
- IBM ThinkPad
  http://peipa.essex.ac.uk/tp-linux/tp-linux.html
- IBM Thinkpad 770 series
  http://resources.inw.com/linux/thinkpad770
- NEC Versa M and P
  http://www.santafe.edu:80/~nelson/versa-linux/
- Tadpole P1000
  http://www.tadpole.com/support/linux.html
- Tadpole P1000 (another one)
- TI TravelMate 4000M
  ftp://ftp.biomath.jussieu.fr/pub/linux/TM4000M-mini-HOWTO.txt.Z
- TI TravelMate 5100
- Toshiba Satellite Pro 400CDT
  http://terra.mpikg-teltow.mpg.de/~burger/T400CDT-Linux.html

3.2 PCMCIA

- PCMCIA
  http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html

PCMCIA drivers currently support all common PCMCIA controllers, including Databook TCIC/2, Intel i82365SL, Cirrus PD67xx, and Vadem VG-468 chipsets. Motorola 6AHC05GA controller used in some Hyundai laptops is not supported. See Appendix B for a list of supported PCMCIA cards.

---

4. CPU/FPU

Intel/AMD/Cyrix 386SX/DX/SL/DXL/SLC, 486SX/DX/SL/SX2/DX2/DX4 are supported. Intel Pentium, Pentium Pro and Pentium II (basically it's a Pentium Pro with MMX) also work. AMD K5 and K6 work good, although older versions of K6 should be avoided as they are buggy. Setting "internal cache" disabled in bios setup can be a workaround.

Also IDT Winchip C6-PSME2006A processors are supported under Linux (http://www.winchip.com).

Linux has built-in FPU emulation if you don't have a math coprocessor.

Experimental SMP (multiple CPU) support is included in kernel 1.3.31 and newer. Check the Linux/SMP Project page for details and updates.

- Linux/SMP Project
  http://www.linux.org.uk/SMP/title.html

A few very early AMD 486DX's may hang in some special situations. All current chips should be okay and getting a chip swap for old CPU's should not be a problem.

ULSI Math*Co series has a bug in the FSAVE and FRSTOR instructions that causes problems with all protected mode operating systems. Some older IIT and Cyrix chips may also have this problem.

There are problems with TLB flushing in UMC U5S chips in very old kernels. (1.1.x)

- enable cache on Cyrix processors
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/CxPatch030.tar.z
- Cyrix software cache control
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/linux.cxpatch
- Cyrix 5x86 CPU register settings
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/cx5x86mod_1.0c.tgz

5. Memory

All memory like DRAM, EDO and SDRAM can be used with Linux. There is one thing you have to look at: normally the kernel is not supporting more than 64 Mb of memory. When you add more than 64 Mb of memory you have to add the following line to your LILO configuration file.

```
append="mem=<number of Mb>M"
```

So, when you have 96 Mb of memory this should become

```
append="mem=96M"
```

Don't type a number higher than the number Mb you really have. This can present unpredictable crashes.

6. Video cards

Linux will work with all video cards in text mode, VGA cards not listed below probably will still work with mono VGA and/or standard VGA drivers.

If you're looking into buying a cheap video card to run X, keep in mind that accelerated cards (ATI Mach, ET4000/W32p, S3) are MUCH faster than unaccelerated or partially accelerated (Cirrus, WD) cards.

``32 bpp'' is actually 24 bit color aligned on 32 bit boundaries. It does NOT mean the cards are capable of 32 bit color, they still display 24 bit color (16,777,216 colors). 24 bit packed pixels modes are not supported in XFree86, so cards that can do 24 bit modes to get higher resolutions in other OS's are not able to do this in X using XFree86. These cards include Mach32, Cirrus 542x, S3 801/805/868/968, ET4000, and others.

AGP (Accelerated Graphics Port) support is growing fast. Most of the X-servers (both freely available and commercial versions) have more or less support for AGP.

6.1 Diamond video cards

Most currently available Diamond cards ARE supported by the current release of XFree86. Early Diamond cards may not be officially supported by XFree86, but there are ways of getting them to work. Diamond is now actively supporting the XFree86 Project.

6.2 SVGALIB (graphics for console)

- VGA
- EGA
- ARK Logic ARK1000PV/2000PV
- ATI VGA Wonder
- ATI Mach32
- Cirrus 542x, 543x
- OAK OTI-037/67/77/87
- S3 (limited support)
- Trident TVGA8900/9000
- Tseng ET3000/ET4000/W32

6.3 XFree86 3.3.2

TOC

Accelerated

- ARK Logic ARK1000PV/VL, ARK2000PV/MT
- ATI Mach8
- ATI Mach32 (16 bpp supported for cards with RAMDAC ATI68875, AT&T20C49x, BT481 and 2Mb video ram)
- ATI Mach64 (16/32 bpp supported for cards with RAMDAC ATI68860, ATI68875, CH8398, STG1702, STG1703, AT&T20C408, 3D Rage II, internal, IBM RGB514)
- Chips & Technologies 64200, 64300, 65520, 65525, 65530, 65535, 65540, 65545, 65546, 65548, 65550, 65554
- Cirrus Logic 5420, 542x/5430 (16 bpp), 5434 (16/32 bpp), 5436, 544x, 546x, 5480, 62x5, 754x
- Diamond Viper 330
- Gemini P1 (ET6000 chip)
- IBM 8514/A
- IBM XGA-I, XGA-II
- IIT AGX-010/014/015/016 (16 bpp)
- Matrox MGA2064W (Millennium)
- Matrox MGA1064SG (Mystique)
- Number Nine Imagine I128
- Oak OTI-087
- S3 732 (Trio32), 764 (Trio64), Trio64V+, 801, 805, 864, 866, 868, 86C325 (ViRGE), 86C375 (ViRGE/DX), 86C385 (ViRGE/GX), 86C988 (ViRGE/VX), 911, 924, 928, 964, 968
    - see Appendix A for list of supported S3 cards
- SiS 86c201, 86c202, 86c205
- Trident 9440, 96xx, Cyber938x
- Tseng ET4000/W32/W32i/W32p, ET6000
- Weitek P9000 (16/32 bpp)
    - Diamond Viper VLB/PCI
    - Orchid P9000
- Western Digital WD90C24/24A/24A2/31/33

Unaccelerated

- Alliance AP6422, AT24
- ATI VGA Wonder series
- Avance Logic AL2101/2228/2301/2302/2308/2401

TOC

- Cirrus Logic 6420/6440, 7555
- Compaq AVGA
- DEC 21030
- Genoa GVGA
- MCGA (320x200)
- MX MX68000/MX68010
- NCR 77C22, 77C22E, 77C22E+
- NVidia NV1
- Oak OTI-037C, OTI-067, OTI-077
- RealTek RTG3106
- SGS-Thomson STG2000
- Trident 8800CS, 8200LX, 8900x, 9000, 9000i, 9100B, 9200CXr, 9320LCD, 9400CXi, 9420, 9420DGi, 9430DGi
- Tseng ET3000, ET4000AX
- VGA (standard VGA, 4 bit, slow)
- Video 7 / Headland Technologies HT216-32
- Western Digital/Paradise PVGA1, WD90C00/10/11/30

Monochrome

- Hercules mono
- Hyundai HGC-1280
- Sigma LaserView PLUS
- VGA mono

Alpha, Beta drivers

- EGA (ancient, from c. 1992)
  ftp://ftp.funet.fi/pub/Linux/BETA/Xega/

6.4 S.u.S.E. X-Server

S.u.S.E. is building a serie of X-servers based on the XFree-86 code. These X-servers support new video cards and are bug fixe releases for XFree86 X-servers. S.u.S.E is building these X-servers together with The XFree86 Project, Inc. These X-Servers will be in the next XFree86 version. These X-servers can be found at http://www.suse.de/index.html. At this moment S.u.S.E. X-Servers are available for the following video cards.

- XSuSE Elsa GLoria X-Server
  - ELSA GLoria L, GLoria L/MX, Gloria S

TOC

- Video cards with the Alliance Semiconductor AT3D (also AT25) Chip
  - Hercules Stingray 128 3D
- XSuSE NVidia X-Server (PCI and AGP support, NV1 chipset and Riva128)
  - ASUS 3Dexplorer
  - Diamond Viper 330
  - ELSA VICTORY Erazor
  - STB Velocity 128
- XSuSE Matrox. Support for Mystique, Millennium, Millennium II and Millennium II AGP
- XSuSE Trident. Support for the 9685 (including ClearTV) and the latest Cyber chipset
- XSuSE Tseng. W32, W32i ET6100 and ET6300 support.

6.5 Commercial X servers

Commercial X servers provide support for cards not supported by XFree86, and might give better performances for cards that are supported by XFree86. In general they support many more cards than XFree86, so I'll onlys list cards that aren't supported by XFree86 here. Contact the vendors directly or check the Commercial HOWTO for more info.

Xi Graphics, Inc

Xi Graphics, Inc http://www.xig.com (formerly known as X Inside, Inc) is selling three X server products (cards supported are sorted by manufacturer):

- Accelerated-X Display Server
  - 3Dlabs
    - 300SX
    - 500TX Glint
    - 500MX Glint
    - Permedia 4MB/8MB
    - Permedia II 4MB/8MB
  - Actix
    - GE32plus 1MB/2MB
    - GE32ultra 2MB
    - GraphicsENGINE 64 1MB/2MB
    - ProSTAR 64 1MB/2MB
  - Alliance
    - ProMotion-3210 1MB/2MB
    - ProMotion-6410 1MB/2MB
    - ProMotion-6422 1MB/2MB

TOC

- ARK Logic
  - ARK1000PV 1MB/2MB
  - ARK1000VL 1MB/2MB
  - ARK2000PV 1MB/2MB
- AST
  - Manhattan 5090P (GD5424) 512KB
- ATI
  - 3D Xpression 1MB/2MB
  - 3D Pro Turbo PC2TV 4MB/8MB
  - 3D Pro Turbo PC2TV 6144
  - 3D Xpression+ PC2TV 2MB/4MB
  - 3D Xpression+ 2MB/4MB
  - ALL-IN-WONDER 4MB/8MB
  - ALL-IN-WONDER PRO 4MB/8MB
  - Graphics Ultra (Mach8) 1MB
  - Graphics Pro Turbo (Mach64/VRAM) 2MB/4MB
  - Graphics Pro Turbo 1600 (Mach64/VRAM) 2MB/4MB
  - Graphics Ultra Plus (Mach32) 2MB
  - 8514/Ultra (Mach8) 1MB
  - Graphics Ultra Pro (Mach32) 1MB2MB
  - Graphics Vantage (Mach8) 1MB
  - VGA Wonder Plus 512KB
  - VGA Wonder XL 1MB
  - Video Xpression 1MB
  - XPERT@Play 4MB/6MB/8MB
  - XPERT@Work 4MB/6Mb/8MB
  - Video Xpression 2MB
  - WinBoost (Mach64/DRAM) 2MB
  - WinTurbo (Mach64/VRAM) 2MB
  - Graphics Wonder (Mach32) 1MB
  - Graphics Xpression 1MB/2MB
  - Rage II (SGRAM) 2MB/4MB/8MB
  - Rage II+ (SGRAM) 2MB/4MB/8MB
  - Rage Pro 2MB/4MB/8MB
- Avance Logic
  - ALG2101 1MB
  - ALG2228 1MB/2MB

TOC

        - ALG2301 1MB/2MB
    - Boca
        - Voyager 1MB/2MB
        - Vortek-VL 1MB/2MB
    - Colorgraphic
        - Dual Lightning 2MB
        - Pro Lightning Accelerator 2MB
        - Quad Pro Lightning Accelerator 2MB
        - Twin Turbo Accelerator 1MB/2MB
    - Chips & Technology
        - 64300 1MB/2MB
        - 64310 1MB/2MB
        - 65510 512KB
        - 65520 1MB
        - 65530 1MB
        - 65535 1MB
        - 65540 1MB
        - 65545 1MB
        - 65550 2MB
        - 82C450 512KB
        - 82C451 256KB
        - 82C452 512KB
        - 82C453 1MB
        - 82C480 1MB/2MB
        - 82C481 1MB/2MB
    - Cirrus Logic
        - GD5402 512KB
        - GD5420 1MB
        - GD5422 1MB
        - GD5424 1MB
        - GD5426 1MB/2MB
        - GD5428 1MB/2MB
        - GD5429 1MB/2MB
        - GD5430 1MB/2MB
        - GD5434 1MB/2MB
        - GD5436 1MB/2MB
        - GD5440 1MB/2MB

TOC

    - GD5446 1MB/2MB
    - GD5462 2MB/4MB PCI and AGP
    - GD5464 2MB/4MB PCI and AGP
    - GD5465 2MB/4MB PCI and AGP
    - GD54M30 1MB/2MB
    - GD54M40 1MB/2MB
  - Compaq
    - ProLiant Series 512KB
    - ProSignia Series 512KB
    - QVision 1024 1MB
    - QVision 1280 1MB/2MB
    - QVision 2000+ 2MB
    - QVision 2000 2MB
  - DEC
    - DECpc XL 590 (GD5428) 512KB
  - Dell
    - 466/M & 466/ME (S3 805) 1MB
    - OnBoard ET4000 1MB
    - DGX (JAWS) 2MB
    - OptiPlex XMT 590 (Vision864) 2MB
  - Diamond
    - Fire GL 1000 Pro 4MB/8MB
    - Fire GL 1000 4MB/8Mb
    - Stealth 3D 2000 2MB/4MB
    - Stealth 3D 3000XL 2MB/4MB
    - Stealth 64 Graphics 2001 1MB/2MB
    - Stealth 64 Graphics 2121XL 1MB/2MB
    - Stealth 64 Graphics 2201XL 2MB
    - SpeedStar 1MB
    - SpeedStar 64 Graphics 2000 1MB/2MB
    - SpeedStar 24 1MB
    - SpeedStar 24X 1MB
    - SpeedStar 64 1MB/2MB
    - SpeedStar Hicolor 1MB
    - SpeedStar PCI 1MB
    - SpeedStar Pro 1MB
    - SpeedStar Pro SE 1MB/2MB

TOC

    - Stealth 1MB
    - Stealth 24 1MB
    - Stealth 32 1MB/2MB
    - Stealth 64 VRAM 2MB/4MB
    - Stealth 64 DRAM 1MB/2MB
    - Stealth 64 Video VRAM (175MHz) 2MB/4MB
    - Stealth 64 Video DRAM 1MB/2MB
    - Stealth 64 Video VRAM (220MHz) 2MB/4MB
    - Stealth Hicolor 1MB
    - Stealth Pro 1MB/2MB
    - Stealth SE 1MB/2MB
    - Stealth 64 Video 2001TV 2MB
    - Stealth 64 Video 2121 1MB/2MB
    - Stealth 64 Video 2121TV 1MB/2MB
    - Stealth 64 Video 2201 2MB
    - Stealth 64 Video 2201TV 2MB
    - Stealth 64 Video 3200 2MB
    - Stealth 64 Video 3240 2MB/4MB
    - Stealth 64 Video 3400 4MB
    - Viper 1MB/2MB
    - Viper Pro 2MB
    - Viper Pro Video 2MB/4MB
    - Viper SE 2MB/4MB
  - ELSA
    - VICTORY 3D 2MB/4MB
    - WINNER 1000 1MB/2MB
    - WINNER 1000AVI 1MB/2MB
    - WINNER 1000ISA 1MB/2MB
    - WINNER 1000PRO 1MB/2MB
    - WINNER 1000TRIO 1MB/2MB
    - WINNER 1000TRIO/V 1MB/2MB
    - WINNER 100VL 1MB
    - WINNER 2000 2MB/4MB
    - WINNER 2000AVI 2MB/4MB
    - WINNER 2000AVI/3D 2MB/4MB
    - WINNER 2000PRO 2MB/4MB
    - WINNER 2000PRO/X 2MB/4MB/8MB

TOC

        - WINNER 3000-L 4MB
        - WINNER 3000-M 2MB
        - WINNER 3000-S 2MB
        - WINNER 1024 1MB
        - WINNER 1280, TLC34075 Palette 2MB
        - WINNER 1280, TLC34076 Palette 2MB
        - Gloria-XL
        - Gloria-MX
        - Gloria-L
        - Synergy
    - Everex
        - ViewPoint 64P 1MB/2MB
        - VGA Trio 64P 1MB/2MB
    - Gateway
        - Mach64 Accelerator (Mach64/VRAM) 2MB
    - Genoa
        - 5400 512KB
        - 8500/8500VL 1MB
        - Phantom 32i 8900 2MB
        - Phantom 64 2MB
    - Hercules
        - Dynamite 1MB
        - Dynamite Pro 1MB/2MB
        - Dynamite Power 2MB
        - Dynamite 3D / GL
        - Graphite 1MB
        - Stingray 64 1MB/2MB
        - Stingray Pro 1MB/2MB
        - Stringray 1MB
        - Terminator 3D 2MB/4MB
        - Terminator 64/Video 2MB
        - Graphite Terminator Pro 2MB/4MB
    - HP
        - NetServer LF/LC/LE (TVGA9000i) 512KB
        - Vectra VL2 (GD5428) 1MB
        - Vectra XM2i (Vision864) 1MB/2MB
        - Vectra XU (Vision864) 1MB/2MB

TOC

- IBM
  - 8514/A 1MB
  - PC 300 Series (GD5430) 1MB
  - PC 300 Series (Vision864) 1MB/2MB
  - PC 700 Series (Vision864) 1MB/2MB
  - PS/ValuePoint Performance Series (Vision864) 1MB/2MB
  - VC550 1MB
  - VGA 256KB
  - XGA-NI 1MB
  - XGA 1MB
- IIT
  - AGX014 1MB
  - AGX015 1MB/2MB
- Integral
  - FlashPoint 1MB/2MB
- Leadtek
  - WinFast L2300 4MB/8MB
- Matrox
  - Comet 2MB
  - Marvel II 2MB
  - Impression (MGA-IMP/3/A/H, MGA-IMP/3/V/H, MGA-IMP/3/M/H) 3MB
  - Impression Lite (MGA-IMP+/LTE/P) 2MB
  - Impression Plus Lite (MGA-IMP+/LTE/V) 2MB
  - Millennium (MGA-MIL) 2MB/4MB/8MB
  - Millennium 220 (MGA-MIL) 2MB/4Mb/8MB
  - Millennium PowerDoc (WRAM) 2MB/4MB/8MB
  - Millennium II (WRAM) 2MB/4MB/8MB PCI and AGP
  - Mystique (MGA-MYS) 2MB/4MB
  - Mystique 220
  - Matrox (con.t)
  - Impression Plus (MGA-IMP+/P, MGA-IMP+/A) 2MB/4MB
  - Impression Plus 220 (MGA-IMP+/P/H, MGA-IMP+/A/H) 2MB/4MB
  - Impression Pro (MGA-PRO/4.5/V) 4.5MB
  - Ultima Plus (MGA-PCI/2+, MGA-VLB/2+) 2MB/4MB
  - Ultima (MGA-ULT/2/A, MGA-PCI/2, MGA-VLB/2) 2MB
  - Ultima (MGA-ULT/2/A/H, MGA-ULT_2/M/H) 2MB
  - Ultima Plus 200 (MGA-PCI/4/200, MGA-VLB/4/200) 4MB

TOC

- MaxVision
  - VideoMax 2000 2MB/4MB
- Metheus
  - Premier 801 1MB
  - Premier 928-1M 1MB
  - Premier 928-2M 2MB
  - Premier 928-4M 4MB
- Micronics
  - Mpower 4 Plus (Mach64) 1MB
- MIRO
  - miroCRYSTAL 10AD 1MB
  - miroCRYSTAL 12SD 1MB
  - miroCRYSTAL 12SD 2MB
  - miroCRYSTAL 20PV 2MB
  - miroCRYSTAL 20SD 2MB
  - miroCRYSTAL 20SV 2MB
  - miroCRYSTAL 22SD 2MB
  - miroCRYSTAL 40SV 4MB
  - miroCRYSTAL VR2000 2MB/4MB
  - miroMAGIC 40PV 4MB
  - miroMAGIC plus 2MB
  - miroVIDEO 12PD 1MB/2MB
  - miroVIDEO 20SD 2MB
  - miroVIDEO 20SV 2MB
  - miroVIDEO 20TD 2MB
  - miroVIDEO 22SD 2MB
  - miroVIDEO 40SV 4MB
- NEC
  - Versa P Series 1MB
- Nth Graphics
  - Engine/150 2MB
  - Engine/250 2MB
- Number Nine
  - GXE Level 10, AT&T 20C491 Palette 1MB
  - GXE Level 10, Bt485 or AT&T20C505 Palette 1MB
  - GXE Level 11 2MB
  - GXE Level 12 3MB

TOC

    - GXE Level 14 4MB
    - GXE Level 16 4MB
    - GXE64 1MB/2MB
    - GXE64pro 2MB/4Mb
    - GXE64pro (-1600) 2MB/4MB
    - Imagine 128 2MB
    - Image 128 (-1280) 4MB
    - Image 128 Series 2 (DRAM) 2MB/4Mb
    - Image 128 Pro (-1600) 4MB/8MB
    - Image 128 Series 2 (VRAM) 2MB/4MB/8MB
    - Image 128 Series III (Revolution 3D) (WRAM) 8MB/16MB PCI and AGP
    - Revolution 3D "Ticket to Ride" (WRAM) 8MB/16MB PCI and AGP
    - 9FX Motion331 1MB/2MB
    - 9FX Motion531 1MB/2MB
    - 9FX Motion771 2MB/4MB
    - 9FX Reality332 2MB
    - 9FX Reality772 2MB/4MB
    - 9FX Reality 334 PCI and AGP
    - 9FX Vision330 1MB/2MB
  - Oak Technology
    - OTI-067 512KB
    - OTI-077 1MB
    - OTI-087 1MB
    - OTI-107 1MB/2MB
    - OTI-111 1MB/2MB
  - Orchid
    - Farenheit 1280 Plus, ATT20C491 Palette 1MB
    - Farenheit 1280 1MB
    - Farenheit 1280 Plus, SC15025 Palette 1MB
    - Farenheit ProVideo 64 2MB/4MB
    - Farenheit Video 3D 2MB
    - Kelvin 64 1MB/2MB
    - Kelvin Video64 1MB/2MB
    - P9000 2MB
  - Packard Bell
    - Series 5000 Motherboard 1MB
  - Paradise

TOC

        - 8514/A 1MB
        - Accelerator 24 1MB
        - Accelerator Value card 1MB
        - Bahamas 64 1MB/2MB
        - Bali 32 1MB/2MB
        - VGA 1024 512KB
        - VGA Professional 512KB
    - Pixelworks
        - WhrilWIN WL1280 (110MHz) 2MB
        - WhrilWIN WL1280 (135MHz) 2MB
        - WhirlWIN WW1280 (110MHz) 2MB
        - WhirlWIN WW1280 (135MHz) 2MB
        - WhrilWIN WW1600 1MB
    - Radius
        - XGA-2 1MB
    - Reveal
        - VC200 1MB
        - VC300 1MB
        - VC700 1MB
    - S3
        - ViRGE 2MB/4MB
        - ViRGE/DX 2MB/4MB
        - ViRGE/GX 2MB/4MB
        - ViRGE/GX /2 2MB/4MB
        - ViRGE/VX 2MB/4MB
        - Trio32 1MB/2MB
        - Trio64 1MB/2MB
        - Trio64V+ 1MB/2MB
        - Trio64V2/DX 1MB/2MB
        - Trio64V2/GX 1MB/2MB
        - 801 1MB/2MB
        - 805 1MB/2MB
        - Vision864 1MB/2MB
        - Vision866 1MB/2MB
        - Vision868 1MB/2MB
        - 911 1MB
        - 924 1MB

TOC

        - 928 1MB
        - 928 2MB/4MB
    - Sierra
        - Falcon/64 1MB/2MB
    - Sigma
        - Legend 1MB
    - SPEA/V7
        - Mercury P64 2MB
        - Storm Pro 4MB
        - ShowTime Plus 2MB
        - STB
        - Evolution VGA 1MB
        - Horizon Plus 1MB
        - Horizon VGA 1MB
        - Horizon 64 1MB/2MB
        - Horizon 64 Video 1MB/2MB
        - Horizon Video 1MB
        - LightSpeed 2MB
        - LightSpeed 128 2MB
        - Nitro 3D 2MB/4MB
        - Nitro 64 1MB/2MB
        - Nitro 64 Video 1MB/2MB
        - PowerGraph VL-24 1MB
        - PowerGraph X-24 1MB
        - PowerGraph 64 3D 2MB
        - PowerGraph 64 1MB/2MB
        - PowerGraph 64 Video 1MB/2MB
        - PowerGraph Pro 2MB
        - Velocity 3D 4MB
        - Velocity 64V 2MB/4MB
    - Toshiba
        - T4900CT 1MB
    - Trident
        - TGUI9400CXi 1MB/2MB
        - TGUI9420DGi 1MB/2MB
        - TGUI9440 1MB/2MB
        - TGUI9660 1MB/2MB

TOC

        - TGUI9680 1MB/2MB
        - TVGA8900B 1MB
        - TVGA8900C 1MB
        - TVGA8900CL 1MB
        - TVGA8900D 1MB
        - TVGA9000 512KB
        - TVGA9000i 512KB
        - TVGA9200CXr 1MB/2MB
    - Tseng Labs
        - ET3000 512KB
        - ET4000 1MB
        - ET6000 2MB/4MB
        - VGA/16 (ISA) 1MB
        - VGA/16 (VLB) 1MB/2MB
        - VGA/32 1MB/2MB
        - ET4000/W32 1MB
        - ET4000/W32i 1MB/2MB
        - ET4000/W32p 1MB/2MB
    - VLSI
        - VL82C975 (AT&T RAMDAC) 2MB
        - VL82C975 (BrookTree RAMDAC) 2MB
        - VL82C976 (Internal RAMDAC) 2MB
    - Western Digital
        - WD90C00 512KB
        - WD90C11 512KB
        - WD90C24 1MB
        - WD90C26 512KB
        - WD90C30 1MB
        - WD90C31 1MB
        - WD90C33 1MB
        - WD9510-AT 1MB
    - Weitek
        - P9100 2MB
        - P9000 2MB
        - W5186 1MB
        - W5286 1MB
- Laptop Accelerated-X Display Server

TOC

- Broadax
  - NP8700 (Cyber 9385)
- Chips & Technology
  - 65510 512KB
  - 65520 1MB
  - 65530 1MB
  - 65535 1MB
  - 65540 1MB
  - 65545 1MB
  - 65554 2MB/4MB
  - 65555 2MB
- Cirrus Logic
  - GD7541 1MB/2MB
  - GD7543 1MB/2MB
  - GD7548 2MB
- Compaq
  - LTE 5400 (Cirrus Logic CL5478)
  - Presario 1090ES (NM 2093)
- Dell
  - Latitude XPi 896 (NeoMagic 2070)
  - Latitude XPi (NM 2070)
  - Latitude XPi CD 1MB (NM 2090)
  - Latitude LM (NM 2160)
  - Latitude CP (NM 2160)
  - Inspiron 3000 (NM 2160)
- Digital (DEC)
  - HiNote VP (NeoMagic 2090)
  - Fujitsu
  - Lifebook 435DX (NeoMagic 2093)
- Gateway 2000
  - Solo 2300 (NeoMagic 2160)
  - Solo 2300 SE (NM 2160)
  - Solo 9100 (C&T 65554)
  - Solo 9100XL (C&T 65555)
- Hewlett Packard
  - OmniBook 800 (NM 2093)
- Hitachi

TOC

        - Notebook E133T (NeoMagic 2070)
    - IBM
        - VGA 256KB
        - Thinkpad 380D (NeoMagic 2090)*
        - Thinkpad 385ED (NeoMagic 2090)*
        - Thinkpad 560E (Cyber 9382)
        - Thinkpad 760XD (Cyber 9385)
        - Thinkpad 770 (Cyber 9397)
    - Micron
        - TransPort XKE (NeoMagic 2160)
        - Millenia Transport (Cirrus Logic GD7548)
    - NEC
        - Versa P Series 1MB
        - Versa 6230 2MB (NeoMagic 2160)
    - NeoMagic
        - MagicGraph128 / NM2070 896
        - MagicGraph128 / NM2070
        - MagicGraph128V / NM2090
        - MagicGraph128V+ / NM2097
        - MagicGraph128ZV / NM2093
        - MagicGraph128XD / NM2160
    - Sony
        - VAIO PCG-505 (NeoMagic 2097)
    - Toshiba
        - T4900CT 1MB
        - Tecra 740CDT (C&T 65554)
    - Trident
        - Cyber 9397
        - Cyber 9385
        - Cyber 9382
    - Twinhead
        - Slimnote 9166TH (Cyber 9385)

    * Numerous XiG customers have comfirmed support.

- Multi-head Accelerated-X Display Server

Metro-X 4.3.0

Metro Link < sales@metrolink.com>

Supported Cards:

Graphics	Card Chipset
ATI 3D RAGE	3D RAGE
ATI 3D RAGE II	3D RAGE II
ATI ALL-IN-WONDER PRO AGP	3D RAGE PRO
ATI ALL-IN-WONDER PRO PCI	3D RAGE PRO
ATI Graphics Pro Turbo	Mach64
ATI Graphics Ultra	Mach8
ATI Graphics Xpression	Mach64
ATI Mach32	Mach32
ATI Mach64	Mach64
ATI VGA STEREO-F/X	ATI 28800
ATI Winturbo PCI	Mach64
ATI XPERT@Play	3D RAGE PRO
ATI XPERT@Play AGP	3D RAGE PRO
ATI XPERT@Work	3D RAGE PRO
ATI XPERT@Work AGP	3D RAGE PRO
Diamond Fire GL 1000Pro	PERMEDIA 2
Diamond SpeedStar 24X	Western Digital 90C31
Diamond SpeedStar Pro SE	Cirrus 5430
Diamond Stealth 24	S3 801
Diamond Stealth 32	ET4000/W32p
Diamond Stealth 3D 2000	S3 ViRGE
Diamond Stealth 64	S3 964, Bt485KPJ135
Diamond Stealth 64 DRAM	S3 Trio64
Diamond Stealth 64 DRAM (SDAC)	S3 864, S3 SDAC
Diamond Stealth 64 Graphics 2000 Series	S3 864, S3 SDAC
Diamond Stealth 64 Graphics 2200	S3 Trio64
Diamond Stealth 64 VRAM	S3 968, IBM RGB526CF22
Diamond Stealth 64 Video 3000 Series	S3 968, TI 3026-175
Diamond Stealth 64 Video VRAM	S3 968, TI 3026-175

TOC

Diamond Stealth Video (SDAC)	S3 868, S3 SDAC
Diamond Stealth Video 2000 Series	S3 868, S3 SDAC
Diamond Viper (110 MHz RAMDAC)	P9000
Diamond Viper (135 MHz RAMDAC)	P9000
ELSA GLoria Synergy	PERMEDIA 2
ELSA Victory 3D	S3 ViRGE
ELSA WINNER 2000 Office AGP	PERMEDIA 2
ELSA Winner 1000 TRIO/V	S3 Trio64V+
ELSA Winner 2000 AVI	S3 968, TI 3026-175
ELSA Winner 2000 PRO/X-2, -4	S3 968, TI 3026-220
ELSA Winner 2000 PRO/X-8	S3 968, IBM RGB528CF25
EPS Apex L-200	C&T 65550
Generic	ATI 28800
Generic	Alliance ProMotion
Generic	Ark 2000
Generic	Avance Logic 22xx/23xx/24xx
Generic	Chips & Technologies
Generic	Cirrus 5420
Generic	Cirrus 5422/5424
Generic	Cirrus 5426/5428
Generic	Cirrus 5429
Generic	Cirrus 5430
Generic	Cirrus 5434
Generic	Cirrus 5436
Generic	Cirrus 5446
Generic	Cirrus 5462
Generic	Cirrus 5462/5465
Generic	Cirrus 5480
Generic	Cirrus 62x5
Generic	Cirrus 6410/6412/6420/6440
Generic	Cirrus 754x
Generic	ET3000
Generic	ET4000/W32P
Generic	ET4000AX
Generic	ET6000

TOC

Generic	Mach32
Generic	Mach64
Generic	Mach8
Generic	P9000
Generic	PERMEDIA 2
Generic	S3 864/868/924/928/964
Generic	S3 968
Generic	S3 Trio64
Generic	S3 Trio64V+
Generic	S3 ViRGE
Generic	S3 ViRGE/GX/DX
Generic	SiS 86c201/86c202/86c205
Generic Trident	TGUI9440
Generic Trident	TGUI96xx
Generic	Trident8900
Generic	VGA
Generic	Western Digital SVGA
Genoa Phantom 64	S3 Trio64V+
Genoa WindowsVGA 8500VL	Cirrus 5426
Hercules Dynamite 128/Video	ET6000
Hercules Dynamite 3D/GL	PERMEDIA 2
Hercules Dynamite 3D/GL AGP	PERMEDIA 2
Hercules Stingray	Avance Logic 2301
Hercules Stingray 128/3D	Alliance ProMotion AT3D
Hercules Stingray 64	Ark 2000
Hercules Terminator 3D	S3 ViRGE/DX
IBM VGA	
Matrox Marvel	ET4000
Matrox Marvel II	ET4000
Matrox Millennium	MGA Storm
Matrox Millennium II AGP	MGA 2164, TI 3026-250
Matrox Millennium II PCI (220 MHz)	MGA 2164, TI 3026-220
Matrox Millennium II PCI (250 MHz)	MGA 2164, TI 3026-250
Matrox Mystique	MGA 1064
Matrox Mystique 220	MGA 1164

TOC

Number Nine GXE64	S3 864
Number Nine Imagine 128	Imagine 128
Number Nine Imagine 128 Series 2	Imagine 128 Series 2
Number Nine Motion 531	S3 868
Number Nine Motion 771	S3 968
Number Nine Revolution 3D	Ticket to Ride
Number Nine Vision 330	S3 Trio64
Orchid Kelvin 64	Cirrus 5434
SPEA Mirage Video	S3 Trio64V+
STB NITRO 3D	S3 ViRGE/GX
STB/Symmetric GLyder MAX-2	PERMEDIA 2
Sigma Designs VGA Legend	ET4000
Tech Source Raptor	Imagine 128 Series 2
Trident 64-Bit Providia 9685	
Trident 8900	Trident 8900
Trident 9440	TGUI9440-2
V PCI-53	Cirrus 5434

7. Controllers (hard drive)

Linux will work with standard IDE, MFM and RLL controllers. When using MFM/RLL controllers it is important to use ext2fs and the bad block checking options when formatting the disk.

Enhanced IDE (EIDE) interfaces are supported. With up to two IDE interfaces and up to four hard drives and/or CD-ROM drives. Linux will detect these EIDE interfaces:

- CMD-640 (Support for buggy interfaces in kernel 2.2)
- DTC 2278D
- FGI/Holtek HT-6560B VLB (Support for secondary interface in kernel 2.2)
- RZ1000 (Support for buggy interfaces in kernel 2.2)
- Triton I (82371FB) (with busmaster DMA)
- Triton II (82371SB) (with busmaster DMA)

ESDI controllers that emulate the ST-506 (MFM/RLL/IDE) interface will also work. The bad block checking comment also applies to these controllers.

Generic 8 bit XT controllers also work.

Starting with pre-patch-2.0.31-3 IDE/ATAPI is provided.

7.1 Alpha, Beta drivers

- UMC 8672 interfaces (experimental in kernel 2.2)
- Promise DC4030VL caching interface card (experimental support in kernel 2.2)

8. Controllers (hard drive RAID)

- Tekram D690CD IDE PCI Cache Controller (with RAID level 1 Mirroring and caching)
- ARCO Inc. DupliDisk IDE disk mirroring controller
  Support for ATA, IDE, E-IDE and UDMA drive. Controllers available can be plugged into ISA and PCI slots, and directly into the IDE controller. Furthermore, 3.5-inch and 5.25-inch Bay Mount units are available that fit into the respective drive bays. More information at http://www.arcoide.com

9. Controllers (SCSI)

It is important to pick a SCSI controller carefully. Many cheap ISA SCSI controllers are designed to drive CD-ROM's rather than anything else. Such low end SCSI controllers are no better than IDE. See the SCSI HOWTO and look at performance figures before buying a SCSI card.

9.1 Supported

- AMI Fast Disk (**VLB/EISA**) (BusLogic compatible)
- Adaptec AVA-1502E (**ISA/VLB**) (AIC-6360) (*AHA1520*)
- Adaptec AVA-1505/1515 (**ISA**) (Adaptec AHA-152x compatible)
- Adaptec AHA-1510/152x (**ISA/VLB**) (AIC-6260/6360)
- Adaptec AHA-154x (**ISA**) (all models)
- Adaptec AHA-174x (**EISA**) (in enhanced mode)
- Adaptec AHA-274x/274xT (**EISA**) (AIC-7771). The 274xT is supported since kernel series 2.1.x (*AHA2740*)
- Adaptec AHA-284x (**VLB**) (AIC-7770) (*AHA2740*)
- Adaptec AHA-2910B (**PCI**) (since kernel series 2.1.x)
- Adaptec AHA-2920 (**PCI**). Use the Future Domain driver. LILO parameters are needed when used for hard disks.
- Adaptec AHA-2940/U/W/AU/UW/U2W (**PCI**) (AIC-7861, AIC-7871, AIC-7844, AIC-7881, AIC-7884). Some of these are only supported since kernel series 2.1.x (*AHA2740*)
- Adaptec AHA-2944D/WD/UD/UWD (**PCI**). Some of these are only supported since kernel series 2.1.x (*AHA2740*)
- Adaptec AHA-3940/U/W/UW (**PCI**) (AIC-7872, AIC-7882) (since 1.3.6). Some of these are only supported since kernel series 2.1.x
- Adaptec AHA-398x/U/W/UW (**PCI**) (AIC-7873, AIC-7883). Some of these are only supported since kernel series 2.1.x
- Adaptec **PCI** controllers with AIC-7850, AIC-7855, AIC-7860
- Adaptec on board controllers with AIC-777x (**EISA**), AIC-785x, AIC-786x, AIC-787x (**PCI**), AIC-788x (**PCI**), AIC-7895. AIC-786x and AIC-7895 are supported since kernel series 2.1.x
- AdvanSys ABP510/5150 Bus-Master (**ISA**)
- AdvanSys ABP5140 Bus-Master (**ISA**) PnP
- AdvanSys ABP5142 Bus-Master (**ISA**) PnP with floppy
- AdvanSys ABP920 Bus-Master (**PCI**)
- AdvanSys ABP930/U Bus-Master (**PCI/Ultra**>)

TOC

- AdvanSys ABP960/U Bus-Master (**PCI/ULTRA**) MAC/PC
- AdvanSys ABP542 Bus-Master (**ISA**) with floppy (single channel)
- AdvanSys ABP742 Bus-Master (**EISA**) (single channel)
- AdvanSys ABP842 Bus-Master (**VL**) (single channel)
- AdvanSys ABP940/U Bus-Master (**PCI/Ultra**) (single channel)
- AdvanSys ABP970/U Bus-Master (**PCI/Ultra**) MAC/PC (single channel)
- AdvanSys ABP752 Dual Channel Bus-Master (**EISA**) (dual channel)
- AdvanSys ABP852 Dual Channel Bus-Master (**VL**) (dual channel)
- AdvanSys ABP950 Dual Channel Bus-Master (**PCI**) (dual channel)
- Always IN2000
- AMD AM53C974
- BusLogic FlashPoint LT/DL/LW/DW (BT-930(R), BT-920, BT-932(R), BT-950(R), BT-952(R)) http://www.dandelion.com/Linux/
- Initio Corp. INI-9090U INI-9100, INI-9100W/A/UW, INI-9200U/UW, INI-9400U/UW, INI-9520U/UW, INI-A100U2W
- Initio Corp. INIC-950
- Mylex (formerly BusLogic) W Series (**PCI**) (BT-948, BT-958, BT-958D)
- Mylex (formerly BusLogic) C Series (**ISA/EISA/VLB/PCI**) (BT-946C, BT-956C, BT-956CD, BT-445, BT-747C, BT-757C, BT-757CD, BT-545C, BT-540CF)
- Mylex (formerly Buslogic) S Series (**ISA/EISA/VLB**) (BT-445S, BT-747S, BT-747D, BT-757S, BT-757D, BT-545S, BT-542D, BT-742A, BT-542B)
- Mylex (formerly BusLogic) A Series (**ISA/EISA**) (BT-742A, BT-542B)
- DPT PM2001, PM2012A (EATA-PIO)
- DPT Smartcache/SmartRAID Plus,III,IV families (**ISA/EISA/PCI**)
  Take a look at http://www.uni-mainz.de/~neuffer/scsi/dpt/(EATA-DMA)
  Cards in these families are PM2011, PM2021, PM2041, PM3021, PM2012B, PM2022, PM2122, PM2322, PM2042, PM3122, PM3222, PM3332, PM2024, PM2124, PM2044, PM2144, PM3224, PM3334
- DTC 3180/3280
- DTC 329x (**EISA**) (Adaptec 154x compatible)
- Future Domain TMC-16x0, TMC-3260 (PCI)
- Future Domain TMC-8xx, TMC-950
- Future Domain chips TMC-1800, TMC-18C50, TMC-18C30, TMC-36C70
- ICP-Vortex PCI-SCSI Disk Array Controllers (many RAID levels supported)
  Patches for Linux 1.2.13 and 2.0.29 are available at ftp://icp-vortex.com/download/linux/. The controllers GDT6111RP, GDT6121RP, GDT6117RP, GDT6127RP, GDT6511RP, GDT6521RP, GDT6517RP, GDT6527RP, GDT6537RP and GDT6557RP are supported. You can also use pre-patch-2.0.31-4 to pre-patch-2.0.31-9.
- ICP-Vortex EISA-SCSI Controllers (many RAID levels supported)

TOC

Patches for Linux 1.2.13 and 2.0.29 are available at ftp://icp-vortex.com/download/linux/. The controllers GDT3000B, GDT3000A, GDT3010A, GDT3020A and GDT3050A are supported. You can also use pre-patch-2.0.31-4 to pre-patch-2.0.31-9.

- Iomega PPA3 parallel port SCSI Host Bus Adapter embedded in ZIP drive
- Media Vision Pro Audio Spectrum 16 SCSI (**ISA**)
- NCR 5380 generic cards
- NCR 53C400 (Trantor T130B) (use generic NCR 5380 SCSI support)
- NCR 53C406a (Acculogic ISApport / Media Vision Premium 3D SCSI)
- NCR chips 53C7x0 (the 53C710 is only supported in PCI variant)
- NCR chips 53C810(A), 53C815, 53C820, 53C825(A), 53C860, 53C875, 53C895 (53C895 supported 'on paper')
- Qlogic / Control Concepts SCSI/IDE (FAS408) (**ISA/VLB**)
- Qlogic FASXXX/FASXX family of chips (**ISA/VLB**)
- QLogic IQ-PCI, IQ-PCI-10, IQ-PCI-D (**PCI**) (ISP1020 chip)
- Quantum ISA-200S, ISA-250MG
- Seagate ST-01/ST-02 (**ISA**)
- SIIG Ultrawide SCSI Pro (Initio chipset). Drivers and kernel patch to be found at http://www.initio.com/suse.htm
- SoundBlaster 16 SCSI-2 (Adaptec 152x compatible) (**ISA**)
- Tekram DC-390, DC-390W/U/F
- Trantor T128/T128F/T228 (**ISA**)
- UltraStor 14F (**ISA**), 24F (**EISA**), 34F (**VLB**)
- Western Digital WD7000 SCSI

9.2 Alpha, Beta drivers

- AMD AM79C974 (**PCI**) (Compaq, HP, Zeos onboard SCSI)
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/scsi/AM53C974-0.3.tgz
- Adaptec ACB-40xx SCSI-MFM/RLL bridgeboard
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/scsi/adaptec-40XX.tar.gz
- Always Technologies AL-500
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/scsi/al500-0.2.tar.gz
- Iomega PC2/2B
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/scsi/iomega_pc2-1.1.x.tar.gz
- Ricoh GSI-8
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scsi/gsi8.tar.gz

9.3 Unsupported

- NCR chip 53C710 (**ISA**) (old obsolete chip, but still used in some Compaq models)
- Non Adaptec compatible DTC boards (327x, 328x)

10. Controllers (I/O)

Any standard serial/parallel/joystick/combo cards. Linux supports 8250, 16450, 16550, and 16550A UART's. Cards that support non-standard IRQ's (IRQ > 9) can be used.

See National Semiconductor's ``Application Note AN-493'' by Martin S. Michael. Section 5.0 describes in detail the differences between the NS16550 and NS16550A. Briefly, the NS16550 had bugs in the FIFO circuits, but the NS16550A (and later) chips fixed those. However, there were very few NS16550's produced by National, long ago, so these should be very rare. And many of the ``16550'' parts in actual modern boards are from the many manufacturers of compatible parts, which may not use the National ``A'' suffix. Also, some multiport boards will use 16552 or 16554 or various other multiport or multifunction chips from National or other suppliers (generally in a dense package soldered to the board, not a 40 pin DIP). Mostly, don't worry about it unless you encounter a very old 40 pin DIP National ``NS16550'' (no A) chip loose or in an old board, in which case treat it as a 16450 (no FIFO) rather than a 16550A. - Zhahai Stewart < zstewart@hisys.com>

11. Controllers (multiport)

11.1 Non-intelligent cards

Supported

- AST FourPort and clones (4 port)
- Accent Async-4 (4 port)
- Arnet Multiport-8 (8 port)
- Bell Technologies HUB6 (6 port)
- Boca BB-1004, 1008 (4, 8 port) - no DTR, DSR, and CD
- Boca BB-2016 (16 port)
- Boca IO/AT66 (6 port)
- Boca IO 2by4 (4 serial / 2 parallel, uses 5 IRQ's)
- Computone ValuePort (4, 6, 8 port) (AST FourPort compatible)
- DigiBoard PC/X, PC/Xem, PCI/Xem, EISA/Xem, PCI/Xr (4, 8, 16 port)
- Comtrol Hostess 550 (4, 8 port)
- PC-COMM 4-port (4 port)
- SIIG I/O Expander 4S (4 port, uses 4 IRQ's)
- STB 4-COM (4 port)
- Twincom ACI/550
- Usenet Serial Board II (4 port)

Non-intelligent cards usually come in two varieties, one using standard com port addresses and use 4 IRQ's, and another that's AST FourPort compatible and uses a selectable block of addresses and a single IRQ. (Addresses and IRQ's are set using `setserial`.) If you're getting one of these cards, be sure to check which standard it conforms to, prices are no indication.

11.2 Intelligent cards

Supported

- Computone IntelliPort II (4/8/16 port)
  ftp://ftp.computone.com/pub/bbs/beta/ip2linux-1.0.2.tar.gz
- Cyclades Cyclom-8Yo/8Ys/16Y/Ye/YeP (8, 16 port) (ISA/PCI)
  http://www.cyclades.com/

- Cyclades Cyclades-8zo/Ze
  http://www.cyclades.com/
- DigiBoard PC/Xe (ISA), PC/Xi (EISA) and PC/Xeve
  ftp://ftp.digibd.com/drivers/linux/
- Equinox SST Intelligent serial I/O cards
  http://www.equinox.com
- Hayes ESP 1, 2 and 8 port versions
  Included in kernel since 2.1.15. The driver for kernel versions 2.0.x can be found at http://www.nyx.net/~arobinso
- Stallion EasyIO (ISA) / EasyConnection 8/32 (ISA/MCA) / EasyConnection 8/64 (PCI)
  For DIP switch settings and configuration files check http://www.stallion.com
- Stallion EasyConnection 8/64 (ISA/EISA) / ONboard (ISA/EISA/MCA) / Brumby (ISA)
  The latest driver can be found at ftp://ftp.stallion.com/drivers/ata5/Linux/v544.tar.gz

Alpha, Beta drivers

- Comtrol RocketPort (8/16/32 port)
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/serial/comtrol-1.04.tar.gz (kernels 1.2.x). A driver for kernels 2.x can be found at http://ftp.leidenuniv.nl/linux/tsx-11/packages/comtrol/
- DigiBoard COM/Xi
  Contact Simon Park (si@wimpol.demon.co.uk) or Mark Hatle (fray@krypton.mankato.msus.edu). NOTE: both email addresses seem not to exist any longer.
- Moxa C102, C104, C168, C218 (8 port), C320 (8/16/24/32 expandable) and C320T
  ftp://ftp.moxa.com.tw/drivers/linux/
- RISCom/8
- Specialix SIO/XIO (modular, 4 to 32 ports)
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/serial/sidrv.taz
- Specialix IO8+
  Contact devices@BitWizard.nl

---

12. Network adapters

Ethernet adapters vary greatly in performance. In general the newer the design the better. Some very old cards like the 3Com 3c501 are only useful because they can be found in junk heaps for $5 a time. Be careful with clones, not all are good clones and bad clones often cause erratic lockups under Linux. Read the Ethernet HOWTO, http://sunsite.unc.edu/LDP/HOWTO/, for detailed descriptions of various cards.

12.1 Supported

Ethernet

For ethernet cards with the DECchip DC21x4x family the "Tulip" driver is available. More information on this driver can be found at http://cesdis.gsfc.nasa.gov/linux/drivers/tulip.html.

- 3Com 3c501 - ``avoid like the plague'' (3c501 driver)
- 3Com 3c503 (3c503 driver), 3c505 (3c505 driver), 3c507 (3c507 driver), 3c509/3c509B (ISA) / 3c579 (EISA)
- 3Com Etherlink III Vortex Ethercards (3c590, 3c592, 3c595, 3c597) (PCI), 3Com Etherlink XL Boomerang (3c900, 3c905) (PCI) and Cyclone (3c905B, 3c980) Ethercards (3c59x driver) and 3Com Fast EtherLink Ethercard (3c515) (ISA) (3c515 driver)
  Newer versions of this drivers are available at http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html
  Avoid the 3c900 card when possible as the driver is not functioning well for this card.
- 3Com 3ccfe575 Cyclone Cardbus (3c59x driver)
- 3Com 3c575 series Cardbus (3c59x driver) (ALL PCMCIA ??)
- AMD LANCE (79C960) / PCnet-ISA/PCI (AT1500, HP J2405A, NE1500/NE2100)
- AT&T GIS WaveLAN
- Allied Telesis AT1700
- Allied Telesis LA100PCI-T
- Allied Telesyn AT2400T/BT ("ne" module)
- Ansel Communications AC3200 EISA
- Apricot Xen-II / 82596
- Cabletron E21xx
- Cogent EM110
- Crystal Lan CS8920, Cs8900
  http://www.cirrus.com/private/drivers/ethernet/edrivers.html
- Danpex EN-9400
- DEC DE425 (EISA) / DE434/DE435 (PCI) / DE450/DE500 (DE4x5 driver)

TOC

- DEC DE450/DE500-XA (dc21x4x) (Tulip driver)
- DEC DEPCA and EtherWORKS
- DEC EtherWORKS 3 (DE203, DE204, DE205)
- DEC QSilver's (Tulip driver)
- Digi International RightSwitch
- Fujitsu FMV-181/182/183/184
- HP PCLAN (27245 and 27xxx series)
- HP PCLAN PLUS (27247B and 27252A)
- HP 10/100VG PCLAN (J2577, J2573, 27248B, J2585) (ISA/EISA/PCI)
  More information at http://cesdis1.gsfc.nasa.gov:80/linux/drivers/100vg.html
- ICL EtherTeam 16i / 32 EISA
- Intel EtherExpress
- Intel EtherExpress Pro
- KTI ET16/P-D2, ET16/P-DC ISA (work jumperless and with hardware-configuration options)
- Macromate MN-220P (PnP or NE2000 mode)
- NCR WaveLAN
- NE2000/NE1000 (be careful with clones)
- Netgear FA-310TX (Tulip chip)
- New Media Ethernet
- PureData PDUC8028, PDI8023
- SEEQ 8005
- SMC Ultra / EtherEZ (ISA)
- SMC 9000 series
- SMC PCI EtherPower 10/100 (Tulip driver)
- SMC EtherPower II (epic100.c driver)
- Sun LANCE adapters (kernel 2.2 and newer)
- Sun Intel adapters (kernel 2.2 and newer)
- Schneider & Koch G16
- Western Digital WD80x3
- Zenith Z-Note / IBM ThinkPad 300 built-in adapter
- Znyx 312 etherarray (Tulip driver)

ISDN

- Linux ISDN WWW page
  This page seems to be gone ??
- ISDN4Linux tools are available from ftp://ftp.franken.de/pub/isdn4linux/v2.0
- 3Com Sonix Arpeggio

TOC

ftp://sunsite.unc.edu/pub/Linux/kernel/patches/network/sonix.tgz

- ASUSCOM Network Inc. ISDNLink 128K PC adapter (HiSax)
- AVM A1 (HiSax)
- Combinet EVERYWARE 1000 ISDN
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/network/combinet1000isdn-1.02.tar.gz
- Compaq ISDN S0 (ISA) (HiSax)
- Creatix PnP S0 (HiSax)
- Dr. Neuhaus Niccy PnP/PCI (HiSax)
- Dynalink IS64PH (HiSax)
- Eicon.Diehl Diva 2.0 ISA and PCI (S0 and U interface, no PRO version) (HiSax)
- Eicon.Diehl Diva Piccola (HiSax)
- Elsa Microlink PCC-16, PCF, PCF-Pro, PCC-8 (HiSax)
- ELSA QuickStep 1000/1000PCI/3000 (HiSax)
- HFC-2BS0 based cards (HiSax)
- IBM Active 2000 (ISA) (act2000)
- ICN ISDN cards (icn)
- Ith Kommunikationstechnik GmbH MIC 16 (ISA) (HiSax)
- ITK ix1-micro Rev.2 (HiSax)
- Octal PCBIT (pcbit)
- Sedlbauer Speed Card (HiSax)
- Teles SO-8/SO-16.0/SO-16.3/SO-16.3c/SO-16.4 and compatible ones (HiSax)
- Traverse Technologie NETjet PCI S0 (HiSax)
- USR Sportster internal TA (HiSax)

ISDN cards that emulate standard modems or common Ethernet adapters don't need any special drivers to work.

Frame Relay

- Emerging Technologies Inc Synchronous Adapters (http://www.etinc.com)
  - ET/5025 (1 port, 8-bit ISA)
  - ET/5025-16 (2 ports, 16-bit ISA)
  - ET/5025-25 (2 ports, 16-bit ISA)
  - ET/5025pq (4 ports, PCI)

Wireless

- ZCOM WL2420 ISA
  Product information can be found at http://www.zcom.com.tw. Object file kernel drivers are

available at www.boerde.de/~matthias/airnet/zcom_v12

X25

- Emerging Technologies Inc Synchronous Adapters (http://www.etinc.com)
  - ET/5025 (1 port, 8-bit ISA)
  - ET/5025-16 (2 ports, 16-bit ISA)
  - ET/5025-25 (2 ports, 16-bit ISA)
  - ET/5025pq (4 ports, PCI)

Pocket and portable adapters

- For more information on Linux and use of the parallel port, go to the Linux Parallel Port Home Page http://www.torque.net/linux-pp.html or http://www.torque.net/parport/. Check Appendix C for a complete list of supported parallel port devices (excluding printers)

Slotless

- SLIP/CSLIP/PPP (serial port)
- EQL (serial IP load balancing)
- PLIP (parallel port) - using ``LapLink cable'' or bi-directional cable

ARCnet

- Works with all ARCnet cards

TokenRing

- Any IBM tokenring card not using DMA
- IBM Tropic chipset cards
- Madge TokenRing OCI 16/4 Mk2

FDDI

- DEC DEFEA (EISA) / DEFPA (PCI) (kernel 2.0.24 and later)

Amateur radio (AX.25)

- Gracilis PackeTwin
- Ottawa PI/PI2
- Most generic 8530 based HDLC boards

TOC

PCMCIA cards

- See Appendix B for complete list or the webpages of David Hinds at http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html

12.2 Alpha, Beta drivers

Ethernet

- Racal-Interlan NI5210 (i82586 Ethernet chip). Improved support in kernel 2.2 and newer
- Racal-Interlan NI6510 (am7990 lance chip). Starting with kernel 1.3.66 more than 16Mb Ram is supported.
- Racal-Interlan PCI card (AMD PC net chip 97c970) ??

ISDN

- SpellCaster's Datacomute/BRI, Telecomute/BRI (ISA) (sc)

ATM

- Efficient Networks ENI155P-MF 155 Mbps ATM adapter (PCI) http://lrcwww.epfl.ch/linux-atm/

Frame Relay

- Sangoma S502 56K Frame Relay card ftp://ftp.sovereign.org/pub/wan/fr/

Wireless

- Proxim RangeLan2 7100 (ISA) / 630x (OEM mini-ISA) http://www.komacke.com/distribution.html

12.3 Unsupported

- IBM PCI Token Ring cards (all of them) ??
- Sysconnect / Schneider & Koch Token Ring cards (all of them)

---

13. Sound cards

More information on sound drivers and sound cards can be found on http://www.opensound.com/ossfree or http://www.opensound.com/oss.html.

13.1 Supported

- 4Front Technology Virtual Mixer (includes SoftOSS)
- 4Front Technology Virtual Synth (SoftOSS)
- 6850 UART MIDI
- A-Plus Sound of Music (OPL3-SA)
- A-Trend Harmony 3Ds751 (PCI)
- AcerMagic S23
- Adlib FM synthesizer card
- Adlib MSC 16 PnP (CS4235)
- AMD Interwave reference card
- ARC Probook
- Audio Excell DSP16
- AW32 Pro (R2.2-W2)
- AW35 (CS4237)
- AW37 Pro (CS4235)
- Aztech Sound Galaxy NX Pro, NX Pro 16, WaveRider 32+
- Aztech Washington
- BTC Mozart Sound System
- BTC-1831 Sound Card (Opti 1688)
- Bravo Sound Card (Opti 82C930)
- Bull PowerPc builtin audio
- CDR4235-6/-8
- CS32-3DI
- Compaq Deskpro XL integrated Business Audio
- Creative EMU8000 add on (PnP)
- Creative Phone Blaster 28.8/33.6
- Creative Sound Blaster 1.0 to 1.5
- Creative Sound Blaster Pro

TOC

- Creative Sound Blaster 16 ASP
- Creative Sound Blaster 16 PnP (type-1 upto type-10)
- Creative Sound Blaster 16 Vibra
- Creative Sound Blaster 2.x
- Creative Sound Blaster 32/AWE (Configure like Sound Blaster 16)
- Creative Sound Blaster 32/AWE PnP (type-1 upto type-10)
- Creative Sound Blaster AWE64 (type-1 upto type-7)
- Creative Sound Blaster AWE64 Gold (type-1 and type-2)
- Creative Sound Blaster PCI64/128
- Creative Sound Blaster AWE64/Gold and 16/32/AWE PnP cards need to be activated using isapnptools
- Creative ViBRA16C/CL/S (type-1 and type-2) PnP
- Creative ViBRA16X PnP (half duplex only)
- CrystaLake Crystal Clear Series 100
- Crystal Audio (CS4235)
- Crystal CRD4236B-1E
- Crystal CRD4237B-5/-8
- Crystal CSC0B35 (CS4236B)
- Crystal CX4237B-SIDE
- Crystal Onboard PnP Audio (CS4235)
- Dell Latidude builtin audio
- Diamond Crystal MM PC/104
- Digital AXP builtin audio
- ECHO-PSS cards (Orchid SoundWave32, Cardinal DSP16)
- ESS 1868, 1869 (type-1 and type-2), 1878, 1879, 1968 PnP AudioDrive
- Ensoniq AudioPCI (ES1371)
- Ensoniq AudioPCI / SoundBlaster PCI (ES1370)
- Ensoniq Soundscape Elite
- Ensoniq Soundscape PnP (model 1 and 2)
- Ensoniq Soundscape S-2000
- Ensoniq Soundscape VIVO, VIVO90
- Ensoniq ActionNote 880 C/CX
- Generic AD1815, AD1816 based soundcard (PnP)
- Generic CMI8330 based soundcard (PnP)
- Generic Crystal CS4232 based soundcard or motherboard (non PnP)
- Generic Crystal CS4232 by Acer (PnP mode)
- Generic Crystal CS4232 type-1 upto type-3 (PnP mode)

TOC

- Generic Crystal CS4235 type-1
- Generic Crystal CS4236 (type-1 upto type-3)
- Generic Crystal CS4236 based soundcard or motherboard (non PnP)
- Generic Crystal CS4236A (type-1 and type-2), CS4236B
- Generic Crystal CS4237 based soundcard or motherboard (non PnP)
- Generic Crystal CS4237B (type-1 and type-2)
- Generic Crystal CS4238 based soundcard or motherboard (non PnP)
- Generic ESS ES1688, ES1788, ES1887, ES1888, ES688 based soundcard or motherboard
- Generic Jazz16 based soundcard
- Generic MAD16 (OPTi 82C928), MAD16 Pro, MAD16 Pro (duplex) (OPTi 82C929)
- Generic Mozart soundcard (OAK OTI-601 chip)
- Generic OPTi 82C924, 82C925 based sound card (PnP)
- Generic OPTi 82C924 soundcard (non PnP mode). Use the MSS driver and the isapnp tools
- Generic OPTi 82C930
- Generic OPTi 82C931
  See http://spoke.nols.com/~drees/opti931.html
- Generic Soundscape based soundcard
- Generic Windows Sound System compatible
- Generic Yamaha OPL3-SA (YMF701) based soundcard
- Generic Yamaha OPL3-SA2 based soundcard (type-1, type-3, type-4)
- Generic Yamaha OPL3-SA3 based soundcard
- Generic Yamaha OPL3-SAx (YMF715/YMF719) non-PnP
- Gravis Ultrasound
- Gravis Ultrasound Extreme
- Gravis Ultrasound 16-bit sampling daughterboard
- Gravis Ultrasound MAX
- Gravis Ultrasound ACE
- Gravis Ultrasound PnP (with RAM), PnP Pro
- HP OmniBook 2100 (CS4236)
- Home Studio 64 (analog audio only)
- IBM Audio Feature (CS423x)
- Logitech SoundMan Games (SBPro, 44kHz stereo support)
- Logitech SoundMan Wave (Jazz16/OPL4)
- Logitech SoundMan 16 (PAS-16 compatible)
- MED3201 audio card
- Maxi Sound 32 PnP (analog audio only)
- Maxi Sound 64 Dynamic 3D (analog audio only)

TOC

- Media Sound SW/32 (non PnP mode)
- MediaTriX AudioTriX Pro, 3D XG
- Media Vision Premium 3D (Jazz16)
- Media Vision Pro Sonic 16 (Jazz)
- Media Vision Pro Audio Spectrum 16 (PAS-16)
- Media Vision Pro Audio Studio 16
- Media Vision Thunderboard
- Microsoft Windows Sound System board (AD1848)
- MultiWave AudioWave Green 16
- Music Quest MIDI connector card (MCC)
- Music Quest MQX-16, MQX-16S MIDI adapter
- Music Quest MQX-32, MQX-32M MIDI adapter
- Music Quest PC MIDI card
- NEC Harmony
- Orchid SoundDrive 16EZ
- Pine PT201
- Primax SoundStorm FM 16, SoundStorm Wave
- RME Digi32, Digi32 Pro, Digi32/8
- Reveal SC300
- Reveal WaveExtreme Pro (with RAM)
- Roland MPU IPC-T MIDI adapter
- S3 SonicVibes
- Shark Mako
- Sharp PC8800
- Shuttle Sound System 48
- Spacewalker HOT-255 PCI 3D (PCI)
- TerraTec Maestro 32/96
- Terratec EWS64XL (audio only)
- Terratec Sound System Base 1 (AD1816)
- Terratec Sound System Base 64 (AD1816)
- Tomato Sound System (OPTi 82C930)
- Trust Sound Expert De Luxe Wave 32
- Turtle Beach Maui
- Turtle Beach Monte Carlo 928, Monte Carlo 929
- Turtle Beach Tropez, Tropez Plus (audio only)
- Turtle Beach Daytona (audio only)
- Wearnes Classic 16

TOC

- Yamaha Sound Edge SW20-PC
- Zefiro Acoustics ZA2 (NOT RECOMMENDED)
- Zenith Z-Player
- AWE32/64 supports is started in kernel series 2.1.x (check the SoundBlaster AWE mini-HOWTO by Marcus Brinkmann for installation details)
- MPU-401 MIDI Intelligent mode (don't enable blindly)
  - MPU IPC-T
  - MQX-32M
- MPU-401 MIDI UART only dumb port (don't enable blindly)
- Yamaha FM synthesizers (OPL2, OPL3, OPL3-SAx (since kernel series 2.1.x) and OPL4)

OSS supports all MIDI daughter cards including Wave Blaster, TB Rio and Yamaha DB50XG. The only requirement is that the "host" card is supported by OSS. Note that only the "host" card needs to be configured using soundconf. The daughter card will be automatically accessible through the MIDI of the "host" card.

13.2 Alpha, Beta drivers

- 4Front Tech. Waveloop loopback audio device
- Aztech AZT1008, AZT2320, AZT3000
- PC speaker / Parallel port DAC
  ftp://ftp.informatik.hu-berlin.de/pub/os/linux/hu-sound/
- Rockwell WaveArtist chipset
- Sonorus STUDI/O
- Turtle Beach Malibu
- Turtle Beach MultiSound/Tahiti/Monterey
  ftp://ftp.cs.colorado.edu/users/mccreary/archive/tbeach/multisound/
- Turtle Beach Tahiti

13.3 Unsupported

- A-Trend Harmony 3DS724 PCI
- Actech PCI 388-A3D
- Adaptec AME-1570
- Aureal Vortex PCI
- Cardinal DSP 16
- Contributed lowlevel drivers
- Crystal CS4614 PCI
- Cyrix MediaGX builtin audio

- Diamond Monster Sound MX300
- Diamond Sonic Impact
- Dream 94PnP Home Studio
- EON Bach SP901 (A3D)
- ESS PCI
- ESS Maestro-1 PCI, Maestro-2 PCI
- ESS Solo-1 PCI
- Echo Personal Sound System
- Generic ALS007, ALS100 based soundcard
- Orchid NuSound 3D
- Orchid SoundWave 32
- Paradise DSP-16
- Quicknet Internet LineJACK
- Sound Blaster Live
- Terratec EWS64S
- Terratec XLerate (A3D)
- Turtle Beach Fiji
- Turtle Beach Montego
- Turtle Beach Pinnacle
- Turtle Beach TBS-2000
- Videologic SonicStorm
- Wearnes Beethoven ADSP-16
- Western Digital Paradise DSP-16
- Yamaha YMF724 PCI

The ASP chip on Sound Blaster 16 series is not supported. AWE32's onboard E-mu MIDI synthesizer is not supported.

Nathan Laredo < laredo@gnu.ai.mit.edu> is willing to write AWE32 drivers if you send him a complimentary card. He is also willing to write drivers for almost any hardware if you send him free samples of your hardware.

Sound Blaster 16's with DSP 4.11 and 4.12 have a hardware bug that causes hung/stuck notes when playing MIDI and digital audio at the same time. The problem can happen with either Wave Blaster daughterboards or MIDI devices attached to the MIDI port. There is no known fix.

---

14. Hard drives

All hard drives should work if the controller is supported.

(From the SCSI HOWTO) All direct access SCSI devices with a block size of 256, 512, or 1024 bytes should work. Other block sizes will not work (Note that this can often be fixed by changing the block and/or sector sizes using the MODE SELECT SCSI command).

Large IDE (EIDE) drives work fine with newer kernels. The boot partition must lie in the first 1024 cylinders due to PC BIOS limitations.

Some Conner CFP1060S drives may have problems with Linux and ext2fs. The symptoms are inode errors during `e2fsck` and corrupt file systems. Conner has released a firmware upgrade to fix this problem, contact Conner at 1-800-4CONNER (US) or +44-1294-315333 (Europe). Have the microcode version (found on the drive label, 9WA1.6x) handy when you call.

Many Maxtor and Western Digital IDE drives are reported to not happily co-exist on the same IDE cable with the other manufacturers drive. Usually one of the drives will fail during operation. Solution is to put them on different IDE cables.

Certain Micropolis drives have problems with Adaptec and BusLogic cards, contact the drive manufacturers for firmware upgrades if you suspect problems.

- Multiple device driver (RAID-0, RAID-1)
  ftp://sweet-smoke.ufr-info-p7.ibp.fr/public/Linux/

14.1 Unsupported

The following hard drives are mentioned as not supported by Linux. Read the bug report available.

- NEC D3817, D3825, D3827, D3847
  "These drives are slightly non-SCSI-2 compliant in the values reported in Mode Sense Page 3. In Mode Sense Page 3 all NEC D38x7 drives report their sector size as zero. The NEC drives are the first brand of drive we have ever encountered that reported the sector size as zero. Unfortunately, that field in Mode Sense Page 3 is not modifiable and there is no way to update the firmware on the D38x7 drives to correct this problem." Problems are mentioned for D3825 and D3827 (both revision 0407). Revision 0410 of these two hard drives seems to solve this problem.

15. Tape drives

15.1 Supported

- SCSI tape drives
  (From the SCSI HOWTO) Drives using both fixed and variable length blocks smaller than the driver buffer length (set to 32k in the distribution sources) are supported. Virtually all drives should work. (Send mail if you know of any incompatible drives.)
  - Seagate Sidewinder 50 AIT (on ICP 6527 RAID-controller)
- QIC-02 drives
- Iomega Ditto internal (ftape 3.04c and newer)

15.2 Alpha, Beta drivers

- QIC-117, QIC-40/80, QIC-3010/3020 (QIC-WIDE) drives
  Most tape drives using the floppy controller should work. Various dedicated controllers (Colorado FC-10/FC-20, Mountain Mach-2, Iomega Tape Controller II) are also supported
  ftp://sunsite.unc.edu/pub/Linux/kernel/tapes
- ATAPI tape drives
  For these an alpha driver (ide-tape.c) is available in the kernel. ATAPI tape drives supported are
  - Seagate TapeStor 8000
  - Conner CTMA 4000 IDE ATAPI Streaming tape drive

15.3 Unsupported

- Emerald and Tecmar QIC-02 tape controller cards - Chris Ulrich < insom@math.ucr.edu>
- Drives that connect to the parallel port (eg: Colorado Trakker)
- Some high speed tape controllers (Colorado TC-15)
- Irwin AX250L/Accutrak 250 (not QIC-80)
- IBM Internal Tape Backup Unit (not QIC-80)
- COREtape Light

16. CD-ROM drives

For more information on CD-ROM drives check the CDROM-HOWTO at http://sunsite.unc.edu/LDP/HOWTO/.

16.1 Supported

Common CD-ROM drives

- SCSI CD-ROM drives
  (From the CD-ROM HOWTO) Any SCSI CD-ROM drive with a block size of 512 or 2048 bytes should work under Linux; this includes the vast majority of CD-ROM drives on the market.
- EIDE (ATAPI) CD-ROM drives (IDECD)
  Almost all double, quad and six speed drives are supported, including
  - Mitsumi FX400
  - Nec-260
  - Sony 55E

Proprietary CD-ROM drives

- Aztech CDA268-01A, Orchid CDS-3110, Okano/Wearnes CDD-110, Conrad TXC, CyCDROM CR520ie/CR540ie/CR940ie (AZTCD)
- Creative Labs CD-200(F) (SBPCD)
- Funai E2550UA/MK4015 (SBPCD)
- GoldStar R420 (GSCD)
- IBM External ISA (SBPCD)
- Kotobuki (SBPCD)
- Lasermate CR328A (OPTCD)
- LMS Philips CM 206 (CM206)
- Longshine LCS-7260 (SBPCD)
- Matsushita/Panasonic CR-521/522/523/562/563 (SBPCD)
- MicroSolutions Backpack parallel portdrive (BPCD)
- Mitsumi CR DC LU05S (MCD/MCDX)
- Mitsumi FX001D/F (MCD/MCDX)
- Optics Storage Dolphin 8000AT (OPTCD)
- Sanyo CDR-H94A (SJCD)
- Sony CDU31A/CDU33A (CDU31A)
- Sony CDU-510/CDU-515 (SOMYCD535)

- Sony CDU-535/CDU-531 (SONYCD535)
- Teac CD-55A SuperQuad (SBPCD)

16.2 Alpha, Beta drivers

- LMS/Philips CM 205/225/202
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/cdrom/lmscd0.4.tar.gz
- NEC CDR-35D (old)
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/cdrom/linux-neccdr35d.patch
- Sony SCSI multisession CD-XA
  ftp://tsx-11.mit.edu/pub/linux/patches/sony-multi-0.00.tar.gz
- Parallel Port Driver
  http://www.torque.net/linux-pp.html

16.3 Notes

All CD-ROM drives should work similarly for reading data. There are various compatibility problems with audio CD playing utilities. (Especially with newer low-end NEC drives.) Some alpha drivers may not have audio support yet.

Early (single speed) NEC CD-ROM drives may have trouble with currently available SCSI controllers.

PhotoCD (XA) is supported. The hpcdtoppm program by Hadmut Danisch converts PhotoCD files to the portable pixmap format. The program can be obtained from ftp://ftp.gwdg.de/pub/linux/hpcdtoppm or as part of the PBM utilities.

Also, reading video CD is supported in kernel series 2.1.3x and later. A patch is available for kernel 2.0.30.

Finally, most IDE CD-ROM Changers are supported.

---

17. CD-Writers

Many CD-Writers are supported by Linux now. For an up to date list of CD-Writers supported check the CD-Writing mini-HOWTO at http://metalab.unc.edu/LDP/HOWTO/CD-Writing-HOWTO.html, check http://www.shop.de/cgi-bin/winni/lsc.pl or check http://www.guug.de:8080/cgi-bin/winni/lsc.pl. Cdwrite ftp://sunsite.unc.edu/pub/Linux/utils/disk-management/ and cdrecord http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html can be used for writing CD's. The X-CD-Roast package for Linux is a graphical front-end for using CD writers. The package can be found at ftp://sunsite.unc.edu/pub/Linux/utils/disk-management/xcdroast-0.96d.tar.gz.

- Grundig CDR 100 IPW
- HP CD-Writer+ 7100
- HP SureStore 4020i
- HP SureStore 6020es/i
- JVC XR-W2010
- Kodak PCD 225
- Mitsubishi CDRW-226
- Mitsumi CR-2600TE
- Olympus CDS 620E
- Philips CDD-521/10,522,2000,2600,3610
- Pinnacle Micro RCD-5020/5040
- Plextor CDR PX-24CS
- Ricoh MP 1420C
- Ricoh MP 6200S/6201S
- Sanyo CRD-R24S
- Smart and Friendly Internal 2006 Plus 2.05
- Sony CDU 920S/924/926S
- Taiyo Yuden EW-50
- TEAC CD-R50S
- WPI(Wearnes) CDR-632P
- WPI(Wearnes) CDRW-622
- Yamaha CDR-100
- Yamaha CDR-200/200t/200tx
- Yamaha CDR-400t/400tx

18. Removable drives

All SCSI drives should work if the controller is supported, including optical (MO), WORM, floptical, Bernoulli, Zip, Jaz, SyQuest, PD, and others.

- Panasonic MO (combines a CD-ROM drive and an optical removable disk). You have to set a switch when configuring the kernel to get both part work at the same time.
- Parallel port Zip drives
  ftp://gear.torque.net/pub/
- Parallel port Avatar Shark-250
  http://www.torque.net/parport/

Removable drives work like hard disks and floppies, just `fdisk`/`mkfs` and mount the disks. Linux provides drive locking if your drives support it. `mtools` can also be used if the disks are in MS-DOS format.

CD-R drives require special software to work. Read the CD-R Mini-HOWTO.

Linux supports both 512 and 1024 bytes/sector disks. Starting with kernel 2.1.32 Linux also supports 2048 bytes/sector. A patch to kernel 2.0.30 is available at http://liniere.gen.u-tokyo.ac.jp/2048.html.

The 2048 bytes/sector support is needed for

- Fujitsu magneto-optical disk drives M2513

Starting with pre-patch-2.0.31-3 IDE/ATAPI internal Zip drives, flopticals and PD's are supported.

- LS-120 floptical
- PD-CD

19. Mice

19.1 Supported

- Microsoft serial mouse
- Mouse Systems serial mouse
- Logitech Mouseman serial mouse
- Logitech serial mouse
- ATI XL Inport busmouse
- C&T 82C710 (QuickPort) (Toshiba, TI Travelmate)
- Microsoft busmouse
- Logitech busmouse
- PS/2 (auxiliary device) mouse

19.2 Alpha, Beta drivers

- Sejin J-mouse
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/console/jmouse.1.1.70-jmouse.tar.gz
- MultiMouse - use multiple mouse devices as single mouse
  ftp://sunsite.unc.edu/pub/Linux/system/misc/MultiMouse-1.0.tgz
- Microsoft Intellimouse

19.3 Notes

Touchpad devices like Alps Glidepoint also work, so long they're compatible with another mouse protocol.

Newer Logitech mice (except the Mouseman) use the Microsoft protocol and all three buttons do work. Eventhough Microsoft's mice have only two buttons, the protocol allows three buttons.

The mouse port on the ATI Graphics Ultra and Ultra Pro use the Logitech busmouse protocol. (See the Busmouse HOWTO for details.)

20. Modems

All internal modems or external modems connected to the serial port should work. Alas, some manufactures have created Windows 95 only modems. Check Appendix E for Linux incompatible hardware. Furthermore, many flash upgradable modems only have flash programs for Win95/NT. These modems cannot be upgraded under Linux.

A small number of modems come with DOS software that downloads the control program at runtime. These can normally be used by loading the program under DOS and doing a warm boot. Such modems are probably best avoided as you won't be able to use them with non PC hardware in the future.

All PCMCIA modems should work with the PCMCIA drivers.

Fax modems need appropriated fax software to operate. Also be sure that the fax part of the modem supports Class 2 or Class 2.0. It seems to be generally true for any fax software on unix that support for Class 1.0 is not available.

- Digicom Connection 96+/14.4+ - DSP code downloading program
  ftp://sunsite.unc.edu/pub/Linux/apps/serialcomm/smdl-linux.1.02.tar.gz
- Motorola ModemSURFR internal 56K. Add a couple of line to RC.SERIAL to account for IRQ and ports if they are non-standard.
- ZyXEL U-1496 series - ZyXEL 1.4, modem/fax/voice control program
  http://www.pe1chl.demon.nl/ZyXEL/ZyXEL-1.6.tar.gz
- ZyXEL Elite 2864 series - modem/fax/voice control program
  http://www.pe1chl.demon.nl/ZyXEL/ZyXEL-1.6.tar.gz
- ZyXEL Omni TA 128 - modem/fax/voice control program
  http://www.pe1chl.demon.nl/ZyXEL/ZyXEL-1.6.tar.gz

Also multimodem cards are supported by Linux.

- Moreton Bay RAStel multimodem card
  Check http://www.moreton.com.au/linux.htm for Linux drivers.

21. Printers/Plotters

All printers and plotters connected to the parallel or serial port should work. Alas, some manufacturers have created Windows 95 only printers. Check Appendix E for Linux incompatible hardware.

- HP LaserJet 4 series - free-lj4, printing modes control program
  ftp://sunsite.unc.edu/pub/Linux/system/printing/free-lj4-1.1p1.tar.gz
- BiTronics parallel port interface
  ftp://sunsite.unc.edu/pub/Linux/kernel/patches/misc/bt-ALPHA-0.0.1.module.patch.gz
- Epson Stylus Color 850. Use Magicfilter with either of the filters 'stylus800-filter', 'stylus_color_360dpi-filter' or 'stylus_color_720dpi-filter'.

21.1 Ghostscript

Many Linux programs output PostScript files. Non-PostScript printers can emulate PostScript Level 2 using Ghostscript.

- Ghostscript
  ftp://ftp.cs.wisc.edu/pub/ghost/aladdin/

Ghostscript 5.1 supported printers

- Apple Imagewriter
- Apple Dot Matrix printer
- Apple StyleWriter 2x00 (*bjc600*)
- Brother HL-660 (*ljet4*)
- C. Itoh M8510
- Canon BubbleJet BJ10e, BJ20 (*bj10e*)
- Canon BubbleJet BJ100, BJ200, BJC-210 (B/W only), BJC-240 (B/W only), BJC-250 (B/W only), BJC-70 (B/W only) (*bj200*)
- Canon BubbleJet BJC-600, BJC-610, BJC-4000, BJC-4100 (B/W only), BJC-4200, BJC-4300, BJC-4400, BJC-4550, BJC-210, BJC-450, MultiPASS C2500, BJC-240, BJC-70 (*bjc600*)
- Canon BubbleJet BJC-800, BJC-7000 (*bjc800*)
- Canon Bubblejet BJC-610 (*uniprint*)
- Canon LBP-8II, LIPS III
- DEC LA50/70/75/75plus
- DEC LN03, LJ250 (*decl250*)
- Epson 9 pin, 24 pin, LQ series, AP3250

TOC

- Epson Stylus Color/Color II/400/500/600/800 (*stcolor*)
- Epson Stylus Color/Color II/500/600/800/1520 (*uniprint*)
- Fujitsu 3400,2400,1200
- HP 2563B
- HP DesignJet 650C
- HP DeskJet, Deskjet Plus (*deskjet*)
- HP Deskjet 500, Deskjet Portable (*djet500*)
- HP Deskjet 500C (*cdeskjet*)
- HP Deskjet 550C (*uniprint*)
- HP DeskJet 400/500C/520C/540C/690C/693C (*cdj500*)
- HP DeskJet 550C/560C/600/660C/660Cse/682C/683C/693C/694C/695C/850/870Cse (*cdj550*)
- HP DeskJet 850/855/870Cse/870Cxi/890C/672C/680/1100C (*cdj850*)
- HP DeskJet 500C/510/520/5540C/693C printing black only (*cdjmono*)
- HP DeskJet 600 (*lj4dith*)
- HP DeskJet 600/870Cse, LaserJet 5/5L/6L (*ljet4*)
- HP Deskjet 600/1200C/1600C (*pjxl300*)
- HP Deskjet 500/500C/510/520/540/550C/560C/850C/855C and other PCL3 printers
  ftp:ftp.pdb.sni.de/pub/utilities/misc/hpdj-2.1.tar.gz
- HP Deskjet 720, 820 and 1000 series
  http://www.rpi.edu/~normat/technical/ppa
- HP Paintjet (*pjtest*)
- HP Paintjet XL (*pjxltest*)
- HP PaintJet XL300 (*pjxl300*)
- HP LaserJet/Plus/II/III/4/5/6
- IBM 3853 Jetprinter color
- IBM Proprinter
- Imagen ImPress
- Lexmark Optra E+ (*ljet4*)
- Mitsubishi CP50 color
- NEC P6/P6+/P60
- NEC Pinwriter P2X (*uniprint*)
- NEC SuperScript 860 (*ljetplus*)
- Oki OL410ex LED (*ljet4*)
- Okidata MicroLine 182
- Ricoh 4081/6000 (*r4081*)
- SPARCprinter
- StarJet 48 inkjet printer

TOC

- Tektronix 4693d color 2/4/8 bit
- Tektronix 4695/4696 inkjet plotter
- Xerox XES printers (2700, 3700, 4045, etc.)

Alpha, Beta drivers

- Epson Stylus Color 440

---

22. Scanners

For scanner support there is the package SANE (Scanner Access Now Easy). Information can be found at http://www.mostang.com/sane/. It can be downloaded from ftp://ftp.mostang.com/pub/sane/. This is a universal scanner interface. It comes complete with documentation and several frontends and backends.

More information on handheld scanners can be found at
http://www.angelfire.com/de/willamowius/scanner.html

Many scanners also have their own, scanner specific, software packages which include drivers.

22.1 Supported

- A4 Tech AC 4096 / AS 8000P
  ftp://ftp.informatik.hu-berlin.de/pub/local/linux/a4scan/a4scan.tgz
- Adara Image Star I
  http://fb4-1112.uni-muenster.de/ffwd/
  ftp://fb4-1112.uni-muenster.de/pub/ffwd/mtekscan-0.2.tar.gz
- Conrad Personal Scanner 64, P105 handheld scanners
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/scan-driver-0.1.8.tar.gz
- Epson GT-5500 (*SANE epson*)
- Epson GT-6000
  ftp://sunsite.unc.edu/pub/Linux/apps/graphics/capture/ppic0.5.tar.gz
- Escom Image Scanner 256 (*SANE umax*)
- Fujitsu SCSI-2 scanners
  contact Dr. G.W. Wettstein < `greg%wind.UUCP@plains.nodak.edu`>
- Genius ColorPage-SP2
  http://fb4-1112.uni-muenster.de/ffwd/
  ftp://fb4-1112.uni-muenster.de/pub/ffwd/mtekscan-0.2.tar.gz
- Genius GS-B105G handheld scanner
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/gs105-0.0.1.tar.gz
- Genius GeniScan GS4500, GS4500A handheld scanners
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/gs4500-2.0.tar.gz
- HighScreen Greyscan 256 handheld scanner
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/gs4500-2.0.tar.gz
- HP ScanJet II series SCSI
  ftp://sunsite.unc.edu/pub/Linux/apps/graphics/capture/hpscanpbm-0.3a.tar.gz

TOC

- HP ScanJet IIc, IIcx, IIp, 3c, 4c, 4p, 5p, 5pse, plus
  http://www.tummy.com/xvscan/
- Linotype Hell Jade, Jade2 (*SANE umax*)
- Logitech Scanman+, Scanman 32, Scanman 256 handheld scanners
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/logiscan-0.0.4.tar.gz
- Microtek ScanMaker E3, E6, II, IIXE, III and 35t models
  http://fb4-1112.uni-muenster.de/ffwd/
  ftp://fb4-1112.uni-muenster.de/pub/ffwd/mtekscan-0.2.tar.gz
  E3 and E6 scanners are also supported by http://www.tummy.com/xvscan/
- Mustek M105 handheld scanner
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/scan-driver-0.1.8.tar.gz
- Mustek HT800 Turbo, Matador 105, Matador 256 handheld scanners
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/scan-driver-0.1.8.tar.gz
- Mustek Paragon 6000CX
  ftp://sunsite.unc.edu/pub/Linux/apps/graphics/capture/muscan-2.0.6.taz
- Nikon Coolscan SCSI 35mm film scanner
  ftp://sunsite.unc.edu/pub/Linux/apps/graphics/capture/coolscan-0.2.tgz
- Nikon AX-210 (*SANE umax*)
- Pearl 256 handheld scanner
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/scan-driver-0.1.8.tar.gz
- Polaroid DMC (*SANE dmc*)
- Vobis/Highscreen Scanboostar Premium (*SANE umax*)
- UMAX SCSI scanners
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/umax-0.5.5.tar.gz
- UMAX Vista S6, S6E, T630, Supervista S-12 (*SANE umax*)
- UMAX S-6E, S-6EG, S-12, S-12G (*SANE umax*)
- UMAX Astra 600S, 610S, 1200S, 1220S (*SANE umax*)
- UMAX UC 630, 840, 1200S, 1200SE (*SANE umax*)
- UMAX UG 80, 630 (*SANE umax*)
- UMAX PSD, Gemini D-16 (*SANE umax*)

The Mustek drivers work only with GI1904 interface cards. Eric Chang eric.chang@chrysalis.org has created a patch to use them with IF960 interface cards.

22.2 Alpha, Beta drivers

- Abaton Scan 300/S (*SANE abaton*)
- Abaton Scan 300/GS (*SANE abaton*)
- Agfa Focus, Focus II (*SANE agfafocus*)

TOC

- Agfa Focus Color, Focus Color Plus (*SANE agfafocus*)
- Agfa Focus Lineart (*SANE agfafocus*)
- Agfa Arcus II (*SANE microtek*)
- Agfa StudioScan II, IIsi (*SANE microtek*)
- Agfa SnapScan 300, 310, 600 (*SANE snapscan*)
- Apple Scanner, OneScanner, ColorOneScanner (*SANE apple*)
- Artec/Ultima AT3, AT6, AT12 (*SANE artec*)
- Artec A6000C+ (*SANE artec*)
- Canon CanoScan 300, CanoScan 600, CanoScan 2700F (*SANE canon*)
- Genius GS-4000, ScanMate/32, ScanMate/GS handheld scanners
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/gs4500-2.0.tar.gz
- HP ScanJet IIc, IIp, IIcx, 3c, 4c, 3p, 4p, 5p, 6100c, 6200c (*SANE hp*)
- HP PhotoSmart PhotoScanner (*SANE hp*)
- Kodak DC210 (*SANE dc210*)
- Kodak DC20, DC25 (*SANE dc25*)
- Microtek Scanmaker E2, E3, E6, II, IIG, IIHR, IISP, III, 35t+, 600Z(S), 600G(S) (*SANE microtek*)
- Microtek ScanMaker E3plus, 330, 630, 636, X6 (*SANE microtek2*)
- Microtek Phantom 636 (*SANE microtek2*)
- Mustek MFC-600S, MFC-600CD, MFC-800S (*SANE mustek*)
- Mustek MFS-6000CX, MFS-6000SP, MFS-8000SP, MFS-1200SP, MFS-12000CX (*SANE mustek*)
- Mustek SE-6000SP, SE-12000SP (*SANE mustek*)
- Mustek HT105, M800 handheld scanners
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/scan-driver-0.1.8.tar.gz
- Nikon LS-20, LS-30, LS-1000 (*SANE Coolscan*)
- Plustek 4830, 9630, OpticPro 600 (*SANE plustek*)
- Siemens S9036 (*SANE agfafocus*)
- Tamarack Artiscan 6000C, 8000C, 12000C (*SANE tamarack*)
- UMAX Vista-S8, UC-1260, Mirage IIse, PL-II (*SANE umax*)
- Vobis HighScan (*SANE microtek2*)
- Voelkner Personal Scanner 64 handheld scanner
  ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/scan-driver-0.1.8.tar.gz
- Vuego 310S (*SANE snapscan*)

22.3 Unsupported

TOC

- Acer scanners. Acer is not releasing any programming information.
- Escom 256 (Primax Lector Premier 256) handheld scanner
- Genius ScanMate/256, EasyScan handheld scanners
- Mustek CG8000 handheld scanner
- All Primax scanners. Primax has no interest in Linux and they do not release data-protocols for scanners.
- Trust Ami Scan handheld scanner
- UMAX parallel scanners

---

23. Other hardware

23.1 Amateur Radio

The following cards etc. are supported:

- KISS based Terminal Node Controllers
- Ottawa PI card
- Gracilis PacketTwin card
- Other Z8530 SCC based cards
- Parallel and serial port Baycom modems
- Soundblaster cards
- Soundcards based on the Crystal chipset

23.2 VESA Power Savings Protocol (DPMS) monitors

Support for power savings is included in the Linux kernel. Use `setterm` to enable support in the Linux console, and `xset` to enable support under X.

23.3 Touch screens

The Metro-X X-server is supporting the following touch screen controllers:

- Carrol Touch serial touch screen
  http://www.carrolltouch.com
- EloGraphics
- Lucas Deeco
- MicroTouch

23.4 Terminals on serial port

Old terminals can easily be used under Linux by connecting them to the serial port of your system. At least the following terminals will be supported:

- VT52
- VT100
- VT220

- VT320
- VT420

23.5 Joysticks

Joystick support is in the latest XFree86 distributions (3.3.x) and in kernel versions 2.1.xx and 2.2.xx. For older kernels the links below are usefull.

- Joystick driver
  For information check http://atrey.karlin.mff.cuni.cz/~vojtech/joystick. An FTP archive can be found at ftp://atrey.karlin.mff.cuni.cz/pub/linux/joystick.

Currently supported joysticks are:

- Amiga joysticks on Amiga
- CH Flightstick Pro compatibles with additional two hats and two buttons
- DirectPad Pro parallel port joystick interfaces (http://www.ziplabel.com/dpadpro/index.html)
- FP Gaming Assasin 3D (http://www.fpgaming.com/portfolio/assn3d.html)
- Gamepads with 6 and 8 buttons
- Genius Flight2000 Digital F-23 (http://www.genius.kye.de/english/product/game.html)
- Gravis Blackhawk Digital (http://www.gravis.com/products/js_blackhawkdigital.html)
- Gravis GamePad Pro (http://www.gravis.com/products/gp_gamepadpro.html)
- Gravis Xterminator GamePad (http://www.gravis.com/products/xterminator.html)
- Logitech CyberMan 2 (http://www.cyberman2.de)
- Logitech ThunderPad Digital (http://www.logitech.ch/Game+Controllers/ThunderPadDigital.html)
- Logitech WingMan Extreme Digital (http://www.logitech.ch/Game+Controllers/WingManExtremeDigital.html)
- MadCatz Panther (http://www.fpgaming.com/portfolio/panther.html)
- MadCatz Panther XL (http://www.fpgaming.com/portfolio/panthxl.html)
- Microsoft SideWinder 3D Pro (http://www.eu.microsoft.com/products/hardware/sidewinder/3Dpro/default.htm)
- Microsoft SideWinder Force Feedback Pro (http://www.eu.microsoft.com/products/hardware/sidewinder/force-feedback/default.htm)
- Microsoft SideWinder GamePad (http://www.eu.microsoft.com/products/hardware/sidewinder/gamepad/default.htm)
- Microsoft SideWinder Precision Pro http://www.eu.microsoft.com/products/hardware/sidewinder/precision-pro/default.htm)
- Multisystem joysticks (Atari, Amiga, Commodore, Amstrad)
- Multisystem joysticks using 0.8.0.2 hw interface

- Nintendo Entertainment System (and clone - SVI, Pegasus ...) gamepads
- PDPI Lightning L4 gamecard (http://www.pdpi.net)
- Sega Genesis (MegaDrive) gamepads
- Sega Master System gamepads
- Sega Saturn gamepads
- SNESKey parallel port joystick interfaces
- Sony PlayStation gamepads
- Standard joysticks with 2, 3 or 4 axes, and up to 4 buttons
- Super Nintendo Entertainment System gamepads
- ThrustMaster FCS compatibles with additional hat
- ThrustMaster Millenium 3D Inceptor (http://www.thrustmaster.com/products/millennium.htm)
- ThrustMaster Rage 3D (http://www.thrustmaster.com/products/rage3d.htm)
- TurboGraFX parallel port joystick interface (http://www2.burg-halle.de/~schwenke/parport.html)

23.6 Video capture boards / Frame Grabbers / TV tuner

A couple of programs are available that support TV tuners. These are:

- BTTV http://www.thp.Uni-Koeln.DE/~rjkm/linux/bttv.html
- Xawtv
- Xtvscreen
- Adlink 7200 Digital I/O device
  ftp://sunsite.unc.edu/pub/Linux/science/lab/adl7200-0.60.tar.gz
- Adlink 7300A Digital I/O device
  ftp://ftp.systemy.it/pub/develop/adl7300-0.04.tar.gz
- CMOS Video Conferencing Kit. The video capture card has a BT849 chipset. It comes with a CCD camera.
- Data Translation DT2803
- Data Translation DT2851 Frame Grabber
  ftp://sunsite.unc.edu/pub/Linux/apps/video/dt2851-2.01.tar.gz
- Data Translation DT3155
  http://krusty.eecs.umich.edu/people/ncowan/linux/welcome.html
- Diamond DTV2000 (based on BT848)
- Dipix XPG1000/FPG/PPMAPA (based on TI C40 DSP). Most add-on cards are supported.
  http://www.thp.Uni-Koeln.DE/~rjkm/linux/bttv.html
- Epix SVM

TOC

- Epix Silicon Video MUX series of video frame grabbing boards
  http://www.ssc.com/lj/issue13/npc13c.html
- FAST Screen Machine II
  ftp://sunsite.unc.edu/pub/Linux/apps/video/ScreenMachineII.2.0.tgz
- Hauppage Wincast TV PCI (based on BT848)
  http://www.thp.Uni-Koeln.DE/~rjkm/linux/bttv.html
- Imaging Technology ITI/IC-PCI
  ftp://ftp.gom-online.de/pub/IC-PCI/icpci-0.3.2.tar.gz
- ImageNation Cortex I
  ftp://sunsite.unc.edu/pub/Linux/apps/video/cortex.drv.1.1.tgz
- ImageNation CX100
  ftp://sunsite.unc.edu/pub/Linux/apps/video/cxdrv-0.86.tar.gz
- ImageNation PX500
  ftp://ftp.systemy.it/pub/develop
- ImageNation PXC200
  ftp://ftp.systemy.it/pub/develop
- Imaging Technology Inc. IC-PCI frame grabber board
  ftp://gandalf.expmech.ing.tu-bs.de/pub/driver/icpci-0.2.0.tar.gz
- Matrox Meteor
  ftp://sunsite.unc.edu/pub/Linux/apps/video/meteor-1.4a.tgz
- Matrox PIP-1024
  http://www.powerup.com.au/~sobeyp/pip_tar.gz
- MaxiTV/PCI (based on ZR36120)
  ftp://sunsite.unc.edu/pub/Linux/kernel/misc-cards/zr36120-971127.tgz
- Miro PCTV (based on BT848)
  http://www.thp.Uni-Koeln.DE/~rjkm/linux/bttv.html
- MuTech MV1000 PCI
  ftp://sunsite.unc.edu/pub/Linux/apps/video/mv1000drv-0.33.tgz
- MuTech MV200
  http://www.powerup.com.au/~sobeyp/mu_tar.gz
- Philips PCA10TV (not in production anymore)
  ftp://ftp.il.ft.hse.nl/pub/tv1000/pctv1000.02.tgz
- Pro Movie Studio
  ftp://sunsite.unc.edu/pub/Linux/apps/video/PMS-grabber.3.0.tgz
- Quanta WinVision B&W video capture card
  ftp://sunsite.unc.edu/pub/Linux/apps/video/fgrabber-1.0.tgz
- Quickcam
  ftp://sunsite.unc.edu/pub/Linux/apps/video/qcam-0.7c-5.tar.gz

TOC

- Sensus 700
  http://www.robots.com/s700.htm
- Smart Video Recoder III (based on BT848)
  http://www.thp.Uni-Koeln.DE/~rjkm/linux/bttv.html
- STB TV PCI Television Tuner (based on BT848)
  http://www.thp.Uni-Koeln.DE/~rjkm/linux/bttv.html
- Tekram C210 (based on ZR36120)
  ftp://sunsite.unc.edu/pub/Linux/kernel/misc-cards/zr36120-971127.tgz
- Video Blaster, Rombo Media Pro+
  ftp://sunsite.unc.edu/pub/Linux/apps/video/vid_src-0.7.tgz
- VT1500 TV cards
  ftp://sunsite.unc.edu/pub/Linux/apps/video/vt1500-1.0.9.tar.gz

23.7 Digital Camera

Currently there are three programs which can be used in combination with digital cameras.

- Camediaplay (http://www.itojun.org/itojun.html)
  You can download it from ftp://ftp.itojun.org/pub/digicam/C-400L/unix/
- Photopc (http://www.average.org/digicam/)
  It can be downloaded from ftp://ftp.average.org/pub/photopc/
- Qvplay (http://www.asahi-net.or.jp/~XG2K-HYS/index-e.html) It can be downloaded from
  http://www.asahi-net.or.jp/~XG2K-HYS/qvplay-0.93.tar.gz

Photopc can be extended with a graphical Tk frontend. This can be found at http://www.mediacity.com/~pwhite/Phototk/phototk.html. Also Qvplay cab be extended with a graphical Tk frontend, which can be found at http://www.bekkoame.or.jp/~tormato/pub/qvplaytk-0.73c.tar.gz

Supported

- Agfa ePhoto line of cameras (photopc, camediaplay)
  http://www.agfahome.com/ephoto/
- Casio QV10, QV-10A, QV-11, QV-30, QV-70, QV-100, QV-200, QV-300, QV-700, QV-770 (qvplay)
  http://www.casio.com/digitalimaging/digital-results.cfm?ObjectGroup_ID=171
- Chinon ES-1000 (same hardware, protocol and image format as Kodak DC20
  http://home.t-online.de/home/oliver.hartmann/dc20secr.htm
- Epson "Colorio Photo" CP-100 (PhotoPC) (photopc, camediaplay)
  http://www.epson.com/graphicarts/cameras/
- Epson "Colorio Photo" CP-200 (PhotoPC 500) (photopc, camediaplay)
  http://www.epson.com/graphicarts/cameras/

TOC

- Epson "Colorio Photo" CP-500 (PhotoPC 600) (photopc, camediaplay)
  http://www.epson.com/graphicarts/cameras/
- Epson PhotoPC 550 (photopc, camediaplay)
  http://www.epson.com/graphicarts/cameras/
- HP Photo Smart Digital Camera
  ftp://ftp.itojun.org/pub/digi-cam/
- Kodak DC20 and DC25
  http://home.t-online.de/home/oliver.hartmann/dc20secr.htm
- Olympus "Camedia" C-400L (D-200L) (photopc, camediaplay)
  http://www.olympusamerica.com/digital/dhome.html
- Olympus "Camedia" C-820L (D-320L) (photopc, camediaplay)
  http://www.olympusamerica.com/digital/dhome.html
- Sanyo VPC-G200/G200EX (photopc, camediaplay)
  http://www.sanyo.co.jp/AV/products/VPCG200fe.html
- Sanyo DSC-V1 (VPC-G200E) (photopc, camediaplay)
- Sanyo DSC-X300 (photopc, camediaplay)
  http://www.sanyo.co.jp/AV/products/VPCX300.html
- Nikon Coolpix 600/900 (Coolpix 600 untested) (photopc)
  http://www.nikonusa.com/products/products.taf?id=128 and
  http://www.nikonusa.com/products/products.taf?id=129
- Sierra Imaging SD640 (photopc)
  http://www.sierraimaging.com/sd640.html
- Toshiba PDR-2 (not sure: photopc)
  http://www.toshiba.com/taisisd/dsc/indexj.htm

Unsupported

- Casio QV-120, QV-5000SX, QV-7000SX
- Kodak DC40, DC50, DC120

23.8 UPS

Various other UPS's are supported, read the UPS HOWTO

- APC SmartUPS
  http://www.dyer.vanderbilt.edu/server/apcupsd
- APC-BackUPS 400/600, APC-SmartUPS SU700/1400RM
  http://www.dyer.vanderbilt.edu/server/apcupsd
- Fenton PowerPal
  ftp://megatec.com.tw/Rups2/UNIX/v3.0.1 for downloads and manuals. Web site information can

be found at http://www.fentonups.com/index2.htm

- Fenton Tele-UPS
  ftp://megatec.com.tw/Rups2/UNIX/v3.0.1 for downloads and manuals. Web site information can be found at http://www.fentonups.com/index2.htm
- Fenton PowerOn
  ftp://megatec.com.tw/Rups2/UNIX/v3.0.1 for downloads and manuals. Web site information can be found at http://www.fentonups.com/index2.htm
- UPS's with RS-232 monitoring port (genpower package)
  ftp://sunsite.unc.edu/pub/Linux/system/ups/genpower-1.0.1.tgz
- MGE UPS's
  http://www.mgeups.com/download/softlib.htm and
  http://www.mgeups.com/download/software/linux/upsp.tgz
- A daemon to shut down and up computers connected to ups's. It's network aware and allows server- and client-mode
  ftp://sunsite.unc.edu/pub/Linux/system/ups/powerd-2.0.tar.gz

23.9 Multifunction boards

- Pro Audio Spectrum 16 SCSI / Sound interface card

23.10 Data acquisition

The Linux Lab Project site collects drivers for hardware dealing with data acquisition, they also maintain some mailing lists dealing with the subject. I have no experience with data acquisition so please check the site for more details.

- Linux Lab Project
  http://www.llp.fu-berlin.de/
- CED 1401
- DBCC CAMAC
- IEEE-488 (GPIB, HPIB) boards
- Keithley DAS-1200
- National Instruments AT-MIO-16F / Lab-PC+
- Analog Devices RTI-800/815 ADC/DAC board
  contact Paul Gortmaker < gpg109@anu.edu.au>

23.11 Watchdog timer interfaces

- Berkshire Products PC Watchdog Card (ISA cards rev. A and C)
  Check ftp://ftp.bitgate.com/pub/bitgate/pcwd for the PC Watchdog program. A driver is included

TOC

in recent kernels. More information on this product can be found at http://www.berkprod.com/wdog.htm

- ICS WDT500-P
  http://www.indcompsrc.com/products/data/html/wdt500-p.html
- ICS WDT501-P (with and without fan tachometer)
  http://www.indcompsrc.com/products/data/html/wdt500-p.html
- Outsource Engineering & Manufacturing Inc. Basic Watchdog Timer Board (ISA)
  Information can be found at http://www.ctec.net/basicwdt. Drivers currently running on 2.0.29, 2.0.33 and 2.0.36 kernels

23.12 Miscellaneous

- Mattel Powerglove
- AIMS Labs RadioTrack FM radio card
  ftp://sunsite.unc.edu/pub/Linux/apps/sound/radio/radiotrack-1.1.tgz
- Reveal FM Radio card
  ftp://magoo.uwsuper.edu/docs/radio.html
- Videotext cards
  ftp://sunsite.unc.edu/pub/Linux/apps/video/videoteXt-0.6.tar.gz

---

24. Related sources of information

- Cameron Spitzer's hardware FAQ archive (??)
  ftp://ftp.rahul.net/pub/cameron/PC-info/
- Guide to Computer Vendors
  http://guide.sbanetweb.com/
- System Optimization Information
  http://www.dfw.net/~sdw/

25. Acknowledgments

Thanks to all the authors and contributors of other HOWTO's, many things here are shamelessly stolen from their works; to FRiC, Zane Healy and Ed Carp, the original authors of this HOWTO; and to everyone else who sent in updates and feedbacks. Special thanks to Eric Boerner and lilo (the person, not the program) for the sanity checks. And thanks to Dan Quinlan for the original SGML conversion.

26. Appendix A. S3 cards supported by XFree86 3.3.x.

CHIPSET	RAMDAC	CLOCKCHIP	BPP	CARD
801/805	AT&T 20C490		16	Actix GE 32 / 32+ 2Mb
				Orchid Fahrenheit 1280(+)
801/805	AT&T 20C490	ICD2061A	16	STB PowerGraph X.24
801/805				Del S3 805
				Miro Crystal 8S
				Orchid Fahrenheit VA
				VL-41
805	S3 GENDAC		16	Miro 10SD VLB/PCI
				SPEA Mirage VLB
801/805	SS2410	ICD2061A	8	Diamond Stealth 24 VLB/ISA
801/805	AT&T 20C490	Ch8391	16	JAX 8231/8241, SPEA Mirage
801/805	S3 GENDAC			Miro Crystal 10SD
805i				Actix GE 32i
				ELSA Winner 1000 ISA
928	AT&T 20C490		16	Actix Ultra
928	Sierra SC15025	ICD2061A	32	ELSA Winner 1000 ISA/VLB/EISA
928	Bt485	ICD2061A	32	STB Pegasus VL
928	Bt485	SC11412	16	SPEA(/V7) Mercury VLB
928	Bt485	ICD2061A	32	#9 GXE Level 10/11/12
928	Ti3020	ICD2061A	32	#9 GXE Level 14/16
928				928Movie
				Diamond Stealth Pro
				ELSA Winner 1000TwinBus
				ELSA Winner 1000VL
				ELSA Winner 2000
				Miro Crystal 16S
864		ICD2061A		Miro Crystal 20SD (BIOS 2.xx)

TOC

864	AT&T 20C498	ICS2494	32	Miro (Crystal) 20SD (BIOS 1.xx)
864	AT&T 20C498/	ICD2061A/	32	ELSA Winner 1000 PRO VLB/PCI
864	STG1700	ICS9161		MIRO 20SD (BIOS 2.x)
				ELAS Winner 1000 PRO
864	STG1700	ICD2061A	32	Actix GE 64 VLB
864	AT&T 20C498/	ICS2595	16	SPEA(/V7) Mirage P64 DRAM (BIOS 3.x)
	AT&T 21C498			
864	S3 86C716 SDAC		32	ELSA Winner 1000 PRO
				Miro 20SD (BIOS 3.x)
				SPEA Mirage P64 DRAM (BIOS 4.x)
				Diamond Stealth 64 DRAM
				Genoa Phantom 64i
				Miro Crystal 20SD VLB (BIOS 3.xx)
864	ICS5342	ICS5342	32	Diamond Stealth 64 DRAM (some)
864	SDAC			Diamond Stealth 64 Graphics 2001
864	AT&T 20C498-13	ICD2061A	32	#9 GXE64 PCI
864				ASUS Video Magic PCI V864
				VidTech FastMax P20

CHIPSET	RAMDAC	CLOCKCHIP	BPP	CARD
964				ELSA Winner 2000 PRO-2,4
				spider Tarantula 64
964	AT&T 20C505	ICD2061A	32	Miro Crystal 20SV PCI/40SV
964	Bt485	ICD2061A	32	Diamond Stealth 64
964	Bt9485	ICS9161A	32	SPEA Mercury 64
964	Ti3020	ICD2061A	8	ELSA Winner 2000 PRO PCI
964	Ti3025	Ti3025	32	#9 GXE64 Pro VLB/PCI
				Miro Crystal 40SV
964	IBM RGB		32	Hercules Graphite Terminator 64
868	S3 86C716 SDAC		32	ELSA Winner 1000AVI
				Miro Crystal 20SD PCI
868	AT&T 29C409			ELSA Winner 1000AVI
868				Diamond Stealth Video DRAM
				Diamond Stealth 64 Video 2120/2200

TOC

			ELSA Winner 1000PRO/X
			#9 FX Motion 531
			VideoLogic GrafixStar 500
968			Diamond Stealth 64 Video 3200
			ELSA Gloria-4/8
			ELSA Winner 2000AVI
			ELSA Winner 2000PRO/X-2/X-4/X-8
			Genoa VideoBlitz III AV
			Hercules Graphite Terminator Pro 64
			LeadTek WinFast S430
			LeadTek WinFast S510
			Miro Crystal 80SV
			Miro Crystal 20SV
			#9 FX Motion 771
			VideoLogic GrafixStar 700
			WinFast S430/S510
968	TVP3026	32	ELSA Winner 2000PRO/X
			Diamond Stealth 64 Video VRAM
968	IBM RGB	32	Genoa VideoBlitz III AVI
			Hercules Terminator Pro 64
			STB Velocity 64 Video
			#9 FX Motion 771
			Diamond Stealth 64 Video 3240/3400
968	TI RAMDAC		Diamond Stealth 64 Video 3240/3400
732	(Trio32)	32	Diamond Stealth 64 DRAM SE
			(all Trio32 based cards)
764	(Trio64)	32	SPEA Mirage P64 (BIOS 5.x)
			Diamond Stealth 64 DRAM
			Diamond Stealth 64 Graphics 2xx0
			#9 FX Vision 330
			STB PowerGraph 64
			(all Trio64 based cards)

CHIPSET	RAMDAC	CLOCKCHIP	BPP	CARD

TOC

(Trio64V+)	DSV3326
	Diamond Stealth 64 Video 2001
	DataExpert DSV3365
	ExpertColor DSV3365
	MAXColor S3 Trio64V+
	ELSA Winner 1000TRIO/V
	Hercules Terminator 64/Video
	#9 FX Motion 331
	STB Powergraph 64 Video
	VideoLogic GrafixStar 400
(Trio64V2)	ELSA Winner 1000/T2D
(ViRGE)	Canopus Co. Power Window 3DV
	DSV3325
	DataExpert DSV3325
	Diamond Multimedia Stealth 3D 2000
	Diamond Multimedia Stealth 3D 2000 PRO
	Diamond Stealth 3D 2000
	Diamond Stealth 3D 2000 PRO
	Diamond Stealth 3D 3000
	ELSA Victory 3D
	ELSA Victory 3DX
	ELSA Winner 3000-S
	Expertcolor DSV3325
	Hercules Terminator 64/3D
	LeadTek WinFast 3D S600
	MELCO WGP-VG4S
	#9 FX Motion 332
	Orchid Tech. Fahrenheit Video 3D
	STB systems Powergraph 3D
	WinFast 3D S600
(ViRGE/DX)	Hercules Terminator 3D/DX
(ViRGE/GX)	STB Nitro 3D
(ViRGE/VX)	ELSA Winner 2000AVI/3D
	ELSA Winner 3000
	ELSA Winner 3000-L-42/-M-22

		MELCO WGP-VX8
		STB Systems Velocity 3D
911/924		Diamond Stealth VRAM
924	SC1148 DAC	

NOTE: for the ViRGE/VX,DX,GX,GX2 chipsets you need XFree86 3.3.1 or newer. You should use the XF86_SVGA server.

---

27. Appendix B. Supported PCMCIA cards

These cards are supported by David Hinds' PCMCIA package and this list is taken from his web page (http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html).

27.1 Ethernet cards

- SMC, Megahertz and Ositech cards use the smc91c92_cs driver
- 3Com and Farallon cards use the 3c589_cs driver
- Fujitsu, TDK, RATOC, CONTEC, Eagle and Nextcom cards use the fmvj18x_cs driver

All other cards use the pcnet_cs driver. Other NE2000-compatible cards that are not on the list are also likely to work with pcnet_cs.

- 3Com 3c589, 3c589B, 3c589C, 3c589D, 3CXE589D (3c589_cs)
- Accton EN2212, EN2216 EtherCard (pcnet_cs)
- Addtron Ethernet (pcnet_cs)
- Allied Telesis CentreCOM CE6001, LA-PCM (pcnet_cs)
- AmbiCom AMB8002 (pcnet_cs)
- Apollo RE450CT (pcnet_cs)
- Asante FriendlyNet (pcnet_cs)
- AST 1082 Ethernet (pcnet_cs)
- Billionton LNT-10TB (pcnet_cs)
- Califormia Access LAN Adapter
- CeLAN EPCMCIA (pcnet_cs)
- CNet CN30BC, CN40BC Ethernet (pcnet_cs)
- Compaq Ethernet Adapter (xirc2ps_cs)
- Compex/ReadyLINK Ethernet Combo (pcnet_cs)
- Compex Linkport Ethernet (pcnet_cs)
- Connectware LANdingGear Adapter (pcnet_cs)
- CONTEC C-NET(PC)C (fmvj18x_cs)
- Danpex EN-6200P2 Ethernet (pcnet_cs)
- Datatrek NetCard (pcnet_cs)
- Dayna Communications CommuniCard E (pcnet_cs)
- Digital DEPCM-AA, PCP78-AC Ethernet (pcnet_cs)
- Digital EtherWORKS Turbo Ethernet (pcnet_cs)

TOC

- D-Link DE-650, DE-660 (pcnet_cs)
- DynaLink L10C Ethernet (pcnet_cs)
- Eagle NE200 Ethernet (fmvj18x_cs)
- Edimax Technology Ethernet Combo (pcnet_cs)
- EFA InfoExpress 205, 207 Combo (pcnet_cs)
- Eiger Labs EPX-ET10T2 Combo (pcnet_cs)
- Eiger Labs EPX-10BT, EPX-ET 10BT (fmvj18x_cs)
- ELECOM Laneed LD-CDWA, LD-CDX, LD-CDNIA, LD-CDY (pcnet_cs)
- EP-210 Ethernet (pcnet_cs)
- Epson Ethernet (pcnet_cs)
- EtherPRIME Ethernet (pcnet_cs)
- Explorer NE-10000 Ethernet (pcnet_cs)
- EZLink 4109 Ethernet (pcnet_cs)
- Farallon Etherwave, EtherMac (3c589_cs)
- Fiberline FL-4680 (pcnet_cs)
- Fujitsu FMV-J181, FMV-J182, FMV-J182A (fmvj18x_cs)
- Fujitsu Towa LA501 (fmvj18x_cs)
- Gateway 2000 Ethernet (pcnet_cs)
- Genius ME3000II Ethernet (pcnet_cs)
- Grey Cell Ethernet (pcnet_cs)
- GVC NIC-2000P Ethernet Combo (pcnet_cs)
- Hitachi HT-4840-11 EtherCard (fmvj18x_cs)
- Hypertec HyperEnet (pcnet_cs)
- IBM CreditCard Ethernet Adapter (pcnet_cs)
- IC-Card Ethernet (pcnet_cs)
- Infotel IN650ct Ethernet (pcnet_cs)
- I-O Data PCLA/T (pcnet_cs)
- Katron PE-520 Ethernet (pcnet_cs)
- KingMax Technology EN10-T2 Ethernet (pcnet_cs)
- Kingston KNE-PCM/M, KNE-PC2 (pcnet_cs)
- KTI PE-520 Plus (pcnet_cs)
- LANEED Ethernet (pcnet_cs)
- LanPro EP4000A (pcnet_cs)
- Lantech Ethernet (pcnet_cs)
- Linksys EtherCard (pcnet_cs)
- Logitec LPM-LN10T, LPM-LN10BA, LPM-LN20T Ethernet (pcnet_cs)
- Longshine ShineNet LCS-8534TB Ethernet (pcnet_cs)

TOC

- Macnica ME-1 Ethernet (pcnet_cs)
- Maxtech PCN2000 Ethernet (pcnet_cs)
- Megahertz XJ10BT, XJ10BC, CC10BT Ethernet (smc91c92_cs)
- Melco LPC-TJ, LPC-TS (pcnet_cs)
- Micronet Etherfast Adapter (pcnet_cs)
- NDC Instant-Link (pcnet_cs)
- Network General ``Sniffer'' (pcnet_cs)
- New Media EthernetLAN (nmclan_cs)
- New Media LiveWire (NOT the LiveWire+) (nmclan_cs)
- New Media BASICS Ethernet (smc91c92_cs)
- NextCom NC5310 (fmvj18x_cs)
- Novell/National NE4100 InfoMover (pcnet_cs)
- Ositech Four of Diamonds (smc91c92_cs)
- OvisLink Ethernet (pcnet_cs)
- Panasonic CF-VEL211P-B (pcnet_cs)
- Planet SmartCom 2000, 3500 (pcnet_cs)
- Pretec Ethernet (pcnet_cs)
- PreMax PE-200 Ethernet (pcnet_cs)
- Proteon Ethernet (pcnet_cs)
- Ratoc REX-9822, REX-5588A/W (fmvj18x_cs)
- Relia RE2408T Ethernet (pcnet_cs)
- Reliasys 2400A Ethernet (pcnet_cs)
- RPTI EP400, EP401 Ethernet (pcnet_cs)
- SCM Ethernet (pcnet_cs)
- Sky Link Express (pcnet_cs)
- SMC 8020BT EtherEZ (not the EliteCard) (smc91c92_cs)
- Socket Communications EA LAN Adapter (pcnet_cs)
- Socket Communications LPE Low Power Ethernet (pcnet_cs)
- SuperSocket RE450T (pcnet_cs)
- Surecom Ethernet (pcnet_cs)
- SVEC PN605C (pcnet_cs)
- TDK LAC-CD02x, LAK-CD021, LAK-CD022A, LAK-CD021AX Ethernet (fmvj18x_cs)
- Thomas-Conrad Ethernet (pcnet_cs)
- Trust Ethernet Combo (pcnet_cs)
- Volktek NPL-402CT Ethernet (pcnet_cs)
- Xircom CreditCard CE2 (xirc2ps_cs)

TOC

27.2 Fast Ethernet (10/100baseT) adapters

- 3Com 3c574TX (3c574_cs, buggy)
- 3Com 3c575TX CardBus (3c575_cb, performance problem)
- Compaq Netelligent 10/100 (xirc2ps_cs)
- D-Link DFE-650 (pcnet_cs)
- Intel EtherExpress PRO/100 (xirc2ps_cs)
- Level One EPC-0100TB (pcnet_cs)
- Linksys EtherFast 10/100 (pcnet_cs)
- Micronet EtherFast Adapter (pcnet_cs)
- NetGear FA410TXC (pcnet_cs)
- Xircom CreditCard CE3 (xirc2ps_cs)

27.3 Token-ring adapters

You should at least have kernel 1.3.72

- 3Com 3c389 Tokenlink Velocity (ibmtr_cs)
- 3Com 3c689 TokenLink III (ibmtr_cs)
- IBM Token ring Adapter (ibmtr_cs)
- IBM Turbo 16/4 Token Ring (ibmtr_cs)

27.4 Wireless network adapters

- AT&T / NCR / Lucent WaveLAN version 2.0 (wavelan_cs)
- DEC RoamAbout/DS (wavelan_cs)
- Xircom CreditCard Netwave (netwave_cs)

27.5 ISDN

- ELSA PCMCIA
- Sedlbauer Speed Star
- Teles S0/PCMCIA

27.6 Modem and serial cards

Virtually all modem cards, simple serial port cards, and digital cellular modems should work. Also ISDN modems that emulate a standard UART are supported.

- Advantech COMpad-32/85 dual serial

- Omega Engineering QSP-100
- Quatech, IOTech dual RS-232 cards
- Quatech quad RS-232 card
- Socket Communications dual RS-232 card
- Trimble Mobile GPS

27.7 Memory cards

All SRAM cards should work. Unsupported flash cards can be read but not written.

- Epson 2MB SRAM
- IBM 8MB Flash
- Intel Series 2, Series 2+ and Value Series 100 Flash
- Maxtor MobileMax 16MB Flash
- New Media SRAM
- TDK Flash Memory SFM20W/C 20MB

27.8 SCSI adapters

Be careful. Many vendors, particularly CD-ROM vendors, seem to switch controller chips at will. Generally, They will use a different product code, but not always: older (supported) New Media Bus Toaster cards are not easily distinguishable from the current (unsupported) Bus Toaster cards.

- Adaptec APA-1450A, APA-1460, APA-1460A, APA-1460B SlimSCSI (aha152x_cs)
- Adaptec SlimSCSI 1480 Cardbus (apa1480_cb, experimental, requires kernel 2.1.90 or later)
- Digital SCSI II adapter
- Eiger Labs SCSI (qlogic_cs)
- Future Domain SCSI2GO (fdomain_cs)
- IBM SCSI (fdomain_cs)
- Iomega ZIP and JAZ Cards (PPA3) (aha152x_cs)
- IO-DATA PCSC-II, PCSC-II-L
- IO-DATA CDG-PX44/PCSC CD-ROM
- Logitec LPM-SCSI2
- Logitec LCD-601 CD-ROM
- MACNICA mPS110, mPS110-LP SCSI (qlogic_cs)
- Melco IFC-SC2, IFC-DC
- NEC PC-9801N-J03R (qlogic_cs)
- New Media Bus Toaster SCSI (older cards only) (aha152x_cs)
- New Media Toast 'n Jam (SCSI only) (aha152x_cs)
- Noteworthy Bus Toaster SCSI (aha152x_cs)

TOC

- Panasonic KXL-D740, KXL-DN740A, KXL-DN740A-NB 4X CD-ROM
- Pioneer PCP-PR1W CD-ROM
- Qlogic FastSCSI (qlogic_cs)
- Raven CD-Note 4X (qlogic_cs)
- RATOC REX-9530 SCSI-2 (qlogic_cs)
- Simple Technologies SCSI (fdomain_cs)
- Sony CD-ROM Discman PRD-250
- Taxan ICD-400PN
- Toshiba NWB0107ABK, SCSC200B (qlogic_cs)

27.9 ATA/IDE CD-ROM adapters

You should at least have kernel 1.3.72

- Argosy EIDE CD-ROM
- Caravelle CD-36N
- Creative Technology CD-ROM
- Digital Mobile Media CD-ROM
- EXP CD940 CD-ROM (Some work, some do not)
- EXP Traveler 620 CD-ROM
- H45 Technologies Quick 2x CD-ROM
- H45 Technologies QuickCD 16X
- IO-DATA CDP-TX4/PCIDE, CDP-TX6/PCIDE, CDP-TX10/PCIDE, CDV-HDN6/PCIDE, MOP-230/PCIDE
- Sony PCGA-CD5 CD-ROM
- TEAC IDE Card/II

27.10 Multifunction cards

You should at least have kernel 1.3.73

- 3Com 3c562, 3c562B/C/D, 3c563B/C/D (3c589_cs)
- 3Com 3CCEM556, 3CXEM556 (3c589_cs)
- 3Com 3CCFEM556 (3c574_cs)
- Accton UE2218 (pcnet_cs)
- ActionTex ComNet (pcnet_cs)
- Dayna Communicard (pcnet_cs)
- D-Link DME336T (pcnet_cs)
- Gateway Telepath Combo (smc91c92_cs)

- Grey Cell GCS3400 (pcnet_cs)
- IBM Home and Away, Home and Away 28.8 (pcnet_cs)
- Intel EtherExpress PRO/100 LAN/Modem (xirc2ps_cs)
- Linksys LANmodem 28.8, 33.6 (pcnet_cs)
- Megahertz/U.S. Robotics EM1144, EM3288, EM3336 (smc91c92_cs)
- Motorola Mariner (smc91c92_cs)
- Motorola Marquis (3c589_cs)
- Ositech Jack of Diamonds (smc91c92_cs)
- PREMAX LAN modem (pcnet_cs)
- Psion V.34 Gold Card (not the 56K) (pcnet_cs)
- Rover ComboCard 33.6 (pcnet_cs)
- Xircom CreditCard CEM28, CEM33, CEM56 (xirc2ps_cs)
- Xircom RealPort

27.11 ATA/IDE card drives

These card drives are supported starting with kernel 1.3.72. Both Flash-ATA cards and rotating-media cards are supported. The very old Western Digital 40MB drives are not supported, because they do not conform to the PCMCIA ATA specifications.

27.12 Miscellaneous cards

- Trimble Mobile GPS (uses serial/modem driver)

27.13 Cards with separately distributed drivers

- GemPlus GPR400 Smart Card Reader http://www.linuxnet.com/smartcard/code.html (Wolf Geldmacher wgeldmacher@paus.ch)
- IBM Smart Capture (iscc_cs) (Koji Okamura oka@nanotsu.kobe-u.ac.jp)
- IBM Etherjet (cs89x0_cs) (Danilo Beuche danili@cs.tu-berlin.de)
- Elsa MicroLink ISDN adapter (elsa_cs) (Klaus Lichtenwalder Klaus.Lichtenwalder@WebForum.DE)
- MPS ISLINEnote ISDN adapter (mpsuni_cs) (Detlef Glaschick glaschick@mps-software.de)
- National Instruments DAQcard700 ftp://ftp.rtlinux.org/pub/rtlinux/sources/DAQCard700/ (Oleg Subbotin)
- Netwave AirSurfer Plus (asplus_cs) http://ipoint.vlsi.uiuc.edu/wireless/asplus.html (Jay Moorman jrmoorma@uiuc.edu)
- New Media Bus Toaster SCSI (new version) (sym53c500_cs) (Tim Corner tcorner@via.at)

TOC

- New Media BASICS SCSI (sym53c500_cs) (Tim Corner tcorner@via.at)
- RATOC REX-9590 (iscc_cs) (Koji Okamura oka@nanotsu.kobe-u.ac.jp)
- Sedlbauer Speed Star ISDN adapter (sedl_cs) (Marcus Niemann niemann@www-bib.fh-bielefeld.de)
- Teles ISDN adapter (teles_cs) (Christof Petig ea0141@uni-wuppertal.de)
- Y-E Data FlashBuster floppy drive adapter (floppy_cs) (David Bateman dbateman@eng.uts.edu.au)

27.14 Working on ...

People are working on the following cards:

- Roland SCP-55 MIDI (Toshiaki Nakatsu ir9k-nkt@asahi.net.or.jp)
- CyberRom CD-ROM (David Rowntree rowntree@dircon.co.uk)
- IO DATA PCSC-II (Katayama Nobuhiro kata-n@po.iijnet.or.jp)
- Macnica mPS-1x0 (Katayama Nobuhiro kata-n@po.iijnet.or.jp)
- Harris PRISM/AM79C930 (Mark Mathews mark@mail.absoval.com)
- TView Preso (Brenden Tuck friar@zendragon.com)

27.15 Unsupported

- Adaptec/Trantor APA-460 SlimSCSI
- CanonCompaq PCMCIA floppy drive
- New Media .WAVjammer and all other sound cards
- Nikon CoolPix100
- Panasonic KXL-D720, KXL-D745, KXL-D783
- SMC 8016 EliteCard
- Telxon/Aironet wireless adapter
- WaveLAN/IEEE
- Xircom CE II Ethernet/Modem
- Xircom CE-10BT Ethernet

---

28. Appendix C. Supported Parallel Port devices

More and more, the parallel port is used to connect other devices than printers. To support this parallel port drivers are written for the devicesto work. This appendix presents devices for which parallel port support is written.

To be clear: printers are not presented in this appendix as they are not supported by parallel port support projects.

Also, check the Linux Parallel Port support pages for more information http://www.torque.net/parport. Here you can find

- paride subsystem for parallel port IDE devices (http://www.torque.net/parport/paride.html)
- support for parallel port SCSI devices (http://www.torque.net/parport/parscsi.html)

28.1 Ethernet

- Accton EtherPocket adapter
- AT-Lan-Tec/RealTek parallel port ethernet adapter
- D-Link DE600/DE620 parallel port ethernet adapter

28.2 Hard drives

- H45 Quick HD
- KingByte IDE/ATAPI disks
- KT Technologies PHd portable hard disk
- MicroSolutions backpack hard-drives
- SyQuest EZ-135
- SyQuest EZ-230
- SyQuest SparQ
- ValueStor external hard-drive

28.3 Tape drives

- Hewlett-Packard Colorado Tracker 250 tape drive (all except the T1000e)
- Hewlett-Packard HP Colorado 5GB tape drive
- Iomega Ditto tape drive
- MicroSolutions backpack 8000t, 8000td tape drives

TOC

28.4 CD-ROM drives

- Freecom Power CD
- Freecom Traveller CD-ROM
- H45 Quick CD
- Hewlett-Packard HP 7100e/7200e CD-R
- KingByte IDE/ATAPI CD-ROMs
- MicroSolutions backpack CD-ROM. Models 163550 and later are supported by the paride driver. For models 160550 and 162550 seperate drivers are availavble.
- MicroSolutions backpack PD/CD drive
- SyQuest SyJet

28.5 Removable drives

- Avatar Shark 250
- Imation Superdisk
- Iomega ZIP, ZIP Plus drives

28.6 IDE Adapter

- Arista ParaDrive products
- DataStor Commuter disks
- Fidelity International Technologies TransDisk products
- Freecom IQ Cable Parallel
- Shuttle Technology EPAT/EPEZ parallel port IDE adapter
- Shuttle Technology EPIA parallel port IDE adapter

28.7 SCSI Adapters

- Adaptec APA-348 mini-SCSI plus adapter cable
  Driver available at http://www.torque.net/parpart/parscsi.html
- Adaptec APA-358 mini-SCSI EPP adapter cable
  Driver available at http://www.torque.net/parpart/parscsi.html
- Shuttle Technology EPSA-2 parallel port SCSI adapter
  Driver available at http://www.torque.net/parpart/parscsi.html
- Shuttle Technology EPST parallel port SCSI adapter
  Driver available at http://www.torque.net/parpart/parscsi.html

28.8 Digital Camera

- Connectix QuickCam

29. Appendix D. Plug and Play devices

For people having trouble getting Plug and Play devices to work, the ISA PnP utilities written by Peter Fox are available. Quote from the README:

```
These programs allow ISA Plug-And-Play devices to be configured
on a Linux machine.

This program is suitable for all systems, whether or not they
include a PnP BIOS.
```

Commands have been taken from the Plug and Play ISA specification Version 1.0a.

More information on ISA PnP utilities can be found on the website of Peter Fox: http://www.roestock.demon.co.uk/isapnptools/

Please let me know about hardware (not normally supported under Linux) which can be put to work with the aid of these utilities. A list of this hardware will be put in this appendix.

30. Appendix E. Linux incompatible Hardware

Some hardware manufacturers have created devices which are compatible with MS-Dos and Windows 95 only. They seem to emulate part of the normally available hardware in the devices by software packages sold together with the device. Specification on these devices are not presented to the world so it is almost impossible to write drivers for these devices. Below a list of devices reported as being Linux incompatible will be given.

Simply put, it is best to avoid hardware which states things like "Needs Windows" or "Windows only".

- Canon LBP-465 printer
- HP Deskjet 710
- HP Deskjet 720, 820 and 1000 series printers (although Windows only) are supported under Linux. Have a look at http://www.rpi.edu/~normat/technical/ppa. Be warned: Support is still in early development.
- Lexmark 1000 inkjet printer
- Lexmark 5000 printer
- Lexmark CJ5000 (ColorJet) printer
- OkiData OkiPage 4w
- Sharp JX-9210 printer
- 3Com 3CXM356, 3CCM356 PCMCIA
- AT&T/Lucent winmodem
- Boca Research 28.8 internal modem (model MV34AI)
- Boca Research 33.6 internal modem (model MV34)(Joe Harjung has succeeded in configuring the modem under Win95 and then soft booting into Linux with the modem still working. Filippo is using this modem under Linux directly without any problems and without soft booting from Windows. I definitely need more info on these Boca Research modems.)
  The Boca Research 33.6 modem (model M336I) is mentioned to work with Linux. The only thing that needed to be done was diabling Plug and Play. Here are the specs of the modem
  - Three stickers saying "MC2920A-3.3", "E6030D 4035-01" and "1721 8011 A"
  - Chips etc on the board
    - Cirrus Logic CL-MD3450D-SC-B
    - Cirrus Logic MD1724-11VC-D
    - Datatronic VLM301-1??
    - Omron G5V-1 (2 of them)
    - AST (?) M628032-20E1
    - Cirrus Logic CL-MD4450C-SC-A
    - Abracon 23-040-20

        - two empy places for additional chips, one of which might be a Cirrus Logic CL-MD1724D
    - 4 jumpers for COM port selection
    - 10 jumpers for IRQ selection
    - other unknown jumpers
- Compaq 192 PCMCIA modem/serial card
- HP Fastmodem D4810B
- IBM Mwave ("Dolphin") card. This card is a combination of sound, modem, fax, voice control and dictation. Software is replacing part of hardware functionality so this software should be loaded to get things working.
- Multiwave Innovation CommWave V.34 modem (http://www.multiwave.com/)
- Megahertz XJ/CC2560 PCMCIA
- New Media Winsurfer PCMCIA modem/serial card
- Rockwell SoftK56
- US Robotics WinModem series
- Zoltrix 33.6 Win HSP Voice/Speaker Phone modem

---

31. Glossary

AGP

Accelerated Graphics Port. A bus interconnect mechanism designed to improve performance of 3D graphics applications. AGP is a dedicated bus from the graphics subsystem to the core-logic chipset. http://www.euro.dell.com/intl/euro/r+d/r+dnews/vectors/vect_2-1/v2-1_agp.htm

ATAPI

AT Attachment Packet Interface. A new protocol for controlling mass storage devices similar to SCSI protocols. It builds on the ATA (AT Attachment) interface, the official ANSI Standard name for the IDE interface developed for hard disk drives. ATAPI is commonly used for hard disks, CD-ROM drives, tape drives, and other devices.

ATM

Asynchronous Transfer Mode

CDDA

Compact Disk Digital Audio. Capability of CD-ROM/Writer to read out audio tracks.

DMA

Direct Memory Access

EGA

Enhanced Graphics Adapter

EIDE

Enhanced IDE

EISA

Extended Industry System Architecture

FDDI

Fiber Distributed Data Interface. High-speed ring local area network.

IDE

Integrated Drive Electronics. Each drive has a built-in controller.

ISA

Industry System Architecture

ISDN

Integrated Services Digital Network

TOC

MCA

MicroChannel Architecture

MFM

Modified Frequency Modulation

MMX

Multimedia Extensions. Added to the newest generation of Intel Pentium Processors. It offers better audio and video quality

PCI

Pheripheral Component Interconnect. 32-bit bus designed by Intel. Information can be found on http://www.pcisig.com and http://infoserver.ee.siue.edu/~jbutter/EE580_1.html.

RAID

Redudant Arrays of Inexpensive Disks. The basic idea of RAID is to combine multiple small, inexpensive disk drives into an array of disk drives which yields performance exceeding that of a single large expensive drive. There are five types of redundant array Architectures; RAID-1 through RAID-5. A non-redudant array of disk drives is referred to as RAID-0. Some RAID systems can mix formats.
http://www.uni-mainz.de/~neuffer/scsi/what_is_raid.html

PPA

Printing Performance Architecture. Protocol developed by Hewlett Packard for their series of Deskjet printers. In essence, the protocol moves the low-level processing of the data to the host computer rather than the printer. This allows for a low-cost printer with a small amount of memory and computing power and a flexible driver. However, this comes at the price of compatibility. HP's decision was to develop drivers only for Windows 95 for this printer.

RLL

Run Length Limited

SCSI

Small Computer Systems Interface. A standerd interface defined for all devices in a computer. It make it possible to use a single adapter for all devices.
http://www.uni-mainz.de/~neuffer/scsi/what_is_scsi.html

SVGA

Super Video Graphics Adapter

UART

Universal Asynchronous Receiver Transmitter

USB

Universal Serial Bus. Not yet supported by Linux.

VGA

TOC

Video Graphics Adapter

VLB

VESA Local Bus. Used in some 486 PC's.

WORM

Write Once Read Many

The Hebrew HOWTO

Maintained by Yair G. Rajwan, yair@hobbes.jct.ac.il

v0.4, 12 September 1995

This `Frequently Asked Questions' (FAQ) / HOWTO document describes how to configure your Linux machine to use Hebrew characters on X-Windows and Virtual Consoles. The most up-to-date version of the Hebrew-HOWTO may be obtained from my Web page or from ftp://hobbes.jct.ac.il.

1. Introduction

- 1.1 Changes.
- 1.2 Thanks

2. Standards for representation of Hebrew characters

- 2.1 ASCII
- 2.2 DOS Hebrew
- 2.3 ISO Hebrew
- 2.4 OLD PC Hebrew
- 2.5 Conversions

3. Virtual Consoles (VCs)

4. X Windows setup - XFree86 3.1

- 4.1 Hebrew fonts.
- 4.2 Installing fonts
- 4.3 Making an X application to use Hebrew fonts.

- 4.4 Mapping the keyboard.
- 4.5 Integrating all the above, examples.

5. Shells setup.

- 5.1 bash
- 5.2 tcsh

6. Applications

- 6.1 Vim
- 6.2 Hebrew pine and pico
- 6.3 Some emacs Hebrew ports.
- 6.4 Dosemu
- 6.5 XHTerm
- 6.6 TeX--XeT - Hebrew Tex.

7. Printer setup

8. Commercial products.

- 8.1 El-Mar software.

9. Hebrew around the Internet.

- 9.1 WWW
- 9.2 Gopher
- 9.3 Ftp

1. Introduction

Any language setup, other than the original American English, has two issues:

1. Displaying the right characters (fonts) - for Hebrew it's `ISO-8859-8` standard.
2. Mapping the keyboard.

There is much more to Hebrew than that (like right to left, geometry in X-Windows,etc), but this HOWTO (at least for the first draft) deals only with the basic issues.

More information can be found in the various "national" HOWTOs (German, Danish, etc.) and in the ISO 8859-1 HOWTO (`ftp://ftp.vlsivie.tuwien.ac.at/pub/8bit` FAQ-ISO-8859-1).

1.1 Changes.

- FIRST DRAFT to 0.2.

  Most of this file is taken from the first draft by Vlad Moseanu.
- 0.2 to 0.3Beta.

  Added excerpts from documents from the archive e-brew.zip from ftp://ftp.jer1.co.il/pub/software/msdos/communication, and some bug fixes with the help of JCT Linux-il group members.
- 0.3Beta to 0.4.

  After the first release of the Hebrew-HOWTO to the Linux-il it contain all the E-mail send to me regarding spelling/grammer and Tex-Xet, Mule and Vim info.

1.2 Thanks

This `HOWTO` prepared by the help of all the group: Linux-il - 'The Israeli Linux users group' and especially by:

The Linux-il group (Linux-il@hagiga.jct.ac.il)

Vlad Moseanu (vlad@actcom.co.il)

Gili Granot (gil@csc.cs.technion.ac.il)

Harvey J. Stein (hjstein@math.huji.ac.il)

Dovie Adler (dadler@hobbes.jct.ac.il)

Gavrie Philipson (gavrie@shekel.jct.ac.il)

2. Standards for representation of Hebrew characters

2.1 ASCII

To make one thing clear, for once and forever: There is no such thing as 8-bit ASCII. ASCII is only 7 bits. Any 8-bit code is not ASCII, but that doesn't mean it's not standard. ISO-8859-8 is standard, but not ASCII. Thanks!

2.2 DOS Hebrew

The Hebrew encoding starts at 128d for Aleph. Therefore, encoding requires 8 bits. This is what you have on the Video card EPROM hardware fonts, all of the Hebrew DOS based editors use this table (Qtext, HED, etc.).

2.3 ISO Hebrew

The Hebrew encoding starts at 224 for Aleph. This is the Internet standard, international standard and basically the standard for Ms-Windows and for Macintoshes (Dagesh, etc...).

2.4 OLD PC Hebrew

This is 7-bit, and obsolete, as it occupies essentially the same ASCII range as English lowercase letters. So, it is best avoided. However, when ISO Hebrew gets its eighth bit stripped off by some ignorant Unix mail program (so you get a jumble of English letters for the Hebrew part of your message and the regular English, reversed or not, mixed in), you will get this, and will need to transform it to PC or ISO. If there was English mixed in with the Hebrew, this will be a sad situation, as you will either get Hebrew plus jumble, or English plus jumble...

2.5 Conversions

Here are some simple scripts to convert from each standard to the other:

```
DOS - ISO:      tr '\200-\232' '\340-\372' < {dos_file} > {iso_file}
ISO - DOS:      tr '\340-\372' '\200-\232' < {iso_file} > {dos_file}
OLD - DOS:      tr -z '\200-\232' < {old_Hebrew_file} > {dos_file}
```

NOTE: The numbers use by `tr` are in octal!

3. Virtual Consoles (VCs)

Every distribution of Slackware comes with kbd; the package is called keytbls under Slackware (a4 in 2.3.0 - kbd 0.90). Joel Hoffman has contributed Hebrew fonts and keymaps from his original codepage.tar.Z file. Look under /usr/lib/kbd for iso08.* files. It follows ISO 8859-8 and the Hebrew keytables and maps.

Put the following lines in /etc/rc.d/rc.local:

```
-----
#!/bin/sh
#       Put any local setup commands in here
#
INITTY=/dev/tty[1-6]
PATH=/sbin:/etc:/bin:/usr/sbin:/usr/bin
#
#       kbd - Set the the console font and keyboard
#       set numlock and set metabit mode on tty1 .. tty8
for tty in $INITTY
do
#       setleds -D +num < $tty > /dev/null
        setmetamode metabit < $tty > /dev/null
done
#       Latin8(Hebrew) keyboard/console
setfont iso08.f16
mapscrn trivial
loadkeys Hebrew
#       enable mapping
for tty in $INITTY
do
        echo -n -e "\\033(K" >$tty
done
-----
```

NOTE: If you are using X Windows be careful with "setleds", it may hang the X server.

The above setup works fine with the Hebrew version of pico (pine) and displays correctly ISO 8859-8 Hebrew (X Windows, MS Windows).

4. X Windows setup - XFree86 3.1

4.1 Hebrew fonts.

XFree86 3.1 comes with two Hebrew fonts: heb6x13, heb8x13. Additional Hebrew fonts can be found on the Net:

- The web Type1 fonts (Helvetica/David style (proportional) and Courier/Shalom Stick style (fixed space)) from the snunit-project archive at ftp://snunit.huji.ac.il/pub/fonts/, it's good for netscape Hebrew pages.
- Avner Lottem, (lottem@techUnix.technion.ac.il) put some Hebrew-ISO 8859-8 fonts on archive at ftp://sunsite.unc.edu/pub/Linux/X11/fonts/hebxfonts-0.1.tgz, it has a font that's good for dosemu under X-Windows (read his README file).

4.2 Installing fonts

- Fonts exaptable: pcf (Portable Compiled Format), bdf (Bitmap Distribution Format), pfb (Type1 fonts).
- Move the fonts to some existing directory (/usr/lib/X11/fonts/misc) or create a new one (/usr/lib/X11/fonts/Hebrew). `compress` (to *.Z) the fonts to save space (NOT GZIP!!!).
- Run the `mkfontdir` to create/re-create the `fonts.dir` and edit `fonts.alias` (optional) to define new aliases.
- For Type1 fonts, `mkfontdir` does nothing. You have to add these fonts to `fonts.dir` manually.
- Make sure that the directory is in the X server path. Edit the XF86Config and add the appropriate path -- FontPath "/usr/X11R6/lib/X11/fonts/...".

4.3 Making an X application to use Hebrew fonts.

In short you need to set the appropriate resource.

Xterm

Put the following line in the $HOME/.Xresources:

```
xterm*font: heb8x13
```

or simply start xterm with `xterm -fn heb8x13` The above font is way too small, so search for a better one ... See the comments/examples on starting X11.

Netscape

Usaly you can use the hebrew fonts from snunit - webfonts, Install it as described, and then put the next defaults in your local `.Xdefaults` or in the app-defaults/Netscape.

```
----
*documentFonts.latin1.variable.italic*slant:          r
*documentFonts.latin1.variable.boldItalic*slant:      r
*documentFonts.latin1.variable*family:                web
*documentFonts.latin1.fixed*family:                   webmono
```

```
*documentFonts.latin1*registry:                          iso8859
*documentFonts.latin1*encoding:                          8
----
```

In general you can put any fonts insted of the webfonts files as long as its supported by X11 as described.

4.4 Mapping the keyboard.

For some reason the X server doesn't inherit the keymap from the previous paragraph, and anyway I would like to define ALT Left and ALT Right and Scroll Lock. When pressing ALT together with some key it will generate a Hebrew character, Scroll Lock will lock in Hebrew mode.

To do that we need to use `xmodmap`. Following is a Xmodmap which also corrects the bugs with the "Num Lock":

```
-----
! Hebrew key mapping for XFree86 (for US/Hebrew keyboards).
! By Vlad Moseanu
!
keysym Alt_L = Mode_switch
keysym Alt_R = Mode_switch
!clear Mod1
clear Mod2
!add Mod1 = Alt_L
add Mod2 = Mode_switch
!
! Set the mapping for each key
!
keycode   8 =
keycode   9 = Escape
keycode  10 = 1 exclam
keycode  11 = 2 at
keycode  12 = 3 numbersign
keycode  13 = 4 dollar
keycode  14 = 5 percent
keycode  15 = 6 asciicircum
keycode  16 = 7 ampersand
keycode  17 = 8 asterisk
keycode  18 = 9 parenleft
keycode  19 = 0 parenright
keycode  20 = minus underscore
keycode  21 = equal plus
keycode  22 = Delete
keycode  23 = Tab
keycode  24 = q Q slash Q
keycode  25 = w W apostrophe W
keycode  26 = e E 0x00f7 E
keycode  27 = r R 0x00f8 R
keycode  28 = t T 0x00e0 T
keycode  29 = y Y 0x00e8 Y
keycode  30 = u U 0x00e5 U
keycode  31 = i I 0x00ef I
keycode  32 = o O 0x00ed O
keycode  33 = p P 0x00f4 P
```

```
keycode  34 = bracketleft braceleft
keycode  35 = bracketright braceright
keycode  36 = Return
keycode  37 = Control_L
keycode  38 = a A 0x00f9 A
keycode  39 = s S 0x00e3 S
keycode  40 = d D 0x00e2 D
keycode  41 = f F 0x00eb F
keycode  42 = g G 0x00f2 G
keycode  43 = h H 0x00e9 H
keycode  44 = j J 0x00e7 J
keycode  45 = k K 0x00ec K
keycode  46 = l L 0x00ea L
keycode  47 = semicolon colon 0x00f3 colon
keycode  48 = apostrophe quotedbl comma quotedbl
keycode  49 = grave asciitilde semicolon asciitilde
keycode  50 = Shift_L
keycode  51 = backslash bar
keycode  52 = z Z 0x00e6 Z
keycode  53 = x X 0x00f1 X
keycode  54 = c C 0x00e1 C
keycode  55 = v V 0x00e4 V
keycode  56 = b B 0x00f0 B
keycode  57 = n N 0x00ee N
keycode  58 = m M 0x00f6 M
keycode  59 = comma less 0x00fa less
keycode  60 = period greater 0x00f5 greater
keycode  61 = slash question period question
keycode  62 = Shift_R
keycode  63 = KP_Multiply
!keycode  64 = Alt_L Meta_L
keycode  65 = space
keycode  66 = Caps_Lock
keycode  67 = F1
keycode  68 = F2
keycode  69 = F3
keycode  70 = F4
keycode  71 = F5
keycode  72 = F6
keycode  73 = F7
keycode  74 = F8
keycode  75 = Escape
keycode  76 = F10
keycode  77 = Num_Lock
keycode  78 = Scroll_Lock
keycode  79 = KP_7
keycode  80 = KP_8
keycode  81 = KP_9
keycode  82 = KP_Subtract
keycode  83 = KP_4
keycode  84 = KP_5
keycode  85 = KP_6
keycode  86 = KP_Add
```

```
keycode  87 = KP_1
keycode  88 = KP_2
keycode  89 = KP_3
keycode  90 = KP_0
keycode  91 = KP_Decimal
keycode  92 = Sys_Req
keycode  93 =
keycode  94 =
keycode  95 = F11
keycode  96 = F12
keycode  97 = Home
keycode  98 = Up
keycode  99 = Prior
keycode 100 = Left
keycode 101 = Begin
keycode 102 = Right
keycode 103 = End
keycode 104 = Down
keycode 105 = Next
keycode 106 = Insert
keycode 107 = Delete
keycode 108 = KP_Enter
keycode 109 = Control_R
keycode 110 = Pause
keycode 111 = Print
keycode 112 = KP_Divide
!keycode 113 = Alt_R Meta_R
keycode 114 = Break
!
! This xmodmap file can be use to set the correct numerical keypad mapping
! when "ServerNumLock" is set in the XF86Config file.  In this case the
! Xserver takes care of the Num Lock processing.
!
!
keycode  136 = KP_7
keycode  137 = KP_8
keycode  138 = KP_9
keycode  139 = KP_4
keycode  140 = KP_5
keycode  141 = KP_6
keycode  142 = KP_1
keycode  143 = KP_2
keycode  144 = KP_3
keycode  145 = KP_0
keycode  146 = KP_Decimal
keycode  147 = Home
keycode  148 = Up
keycode  149 = Prior
keycode  150 = Left
keycode  151 = Begin
keycode  152 = Right
keycode  153 = End
```

```
keycode  154 = Down
keycode  155 = Next
keycode  156 = Insert
keycode  157 = Delete
-----
```

To use the Xmodmap above define "Scroll-Lock Mode-Lock" in the XF86Config.

4.5 Integrating all the above, examples.

If you are using xdm a $HOME/.xsession should look like the following:

```
-----
#!/bin/sh
# $XConsortium: Xsession,v 1.9 92/08/29 16:24:57 gildea Exp $
#
# General defs
#
export OPENWINHOME=/usr/openwin
export MANPATH=/usr/local/man:/usr/man/preformat:/usr/man:/usr/X11R6/man
#export HOSTNAME="`cat /etc/HOSTNAME`"
export PATH="/bin: /usr/bin: /usr/X11/bin: /usr/X386/bin: /usr/TeX/bini:
/usr/local/bin: /usr/games:."
LESS=-MM
if [ -z $XAPPLRESDIR ]; then
        XAPPLRESDIR=/usr/lib/X11/app-defaults:/usr/local/lib/X11/app-defaults
else
        XAPPLRESDIR=$XAPPLRESDIR:/usr/lib/X11/app-defaults
fi
export XAPPLRESDIR
#
sysresources=/usr/lib/X11/Xresources
sysmodmap=/usr/lib/X11/Xmodmap
resources=$HOME/.Xresources
xmodmap=$HOME/.Xmodmap
if [ -f $sysresources ]; then
        xrdb -merge $sysresources
fi
if [ -f $sysmodmap ]; then
        xmodmap $sysmodmap
fi
if [ -f $resources ]; then
        xrdb -merge $resources
fi
if [ -f $xmodmap ]; then
        xmodmap $xmodmap
fi
#
# Start applications
#
# xterm -ls -sb &
xhost +                 # look out !!!
```

```
exec fvwm
-----
```

If you prefer `startx` use the above as an example for `.xinitrc`.

5. Shells setup.

For more details read the ISO 8859-1 HOWTO.

5.1 bash

Create a $HOME/.inputrc contain the following:

```
-----
set meta-flag On
set convert-meta Off
set output-meta On
-----
```

5.2 tcsh

Define the following in the $HOME/.login or /etc/csh.login: `setenv LANG iw_IL.ISO8859-8 (or iw_IL)` Actually because the binary version of tcsh is complied without nls the LANG can be set to anything and it will still work (no need for /usr/lib/nls...). The lang. name also shows my Digital bias ...

6. Applications

6.1 Vim

- The Vim is a Vi IMproved editor with some enhanced commands and the hebrew support was made bu Dov Grobgeld (HED developer).
- Another Vim patch announced by Avner Lottem, lottem@techunix.technion.ac.il and can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/editors/vi/vim3.0-rlh0.1.tgz.
- For more info, you can look at http://www.cs.technion.ac.il/~gil/var.html

6.2 Hebrew pine and pico

The pine and it's additional editor pico had been changed by Helen Zommer from CC-huji and has a bug-report mail: pineh-bug@horizon.cc.huji.ac.il. It can be down-loaded from ftp://horizon.cc.huji.ac.il/pub.

6.3 Some emacs Hebrew ports.

- Hebrew package by Joseph Friedman. It includes some Hebrew fonts in BDF format, patch for emacs 18.58 and an elisp package. It is fine, but nobody uses emacs 18.* anymore. It can be obtained from: ftp://archive.cis.ohio-state.edu/pub/gnu/emacs/elisp-archive/misc/Hebrew.tar.Z.
- A very simple Hebrew package. Includes only right-to-left cursor movement support and right-to-left sorting. Works without any patches with FSF emacs 19. Can be obtained from ftp://archive.cis.ohio-state.edu/pub/gnu/emacs/elisp-archive/misc/Hebrew.el.Z.
- One of emacs branches - MULE (Multi Lingual Emacs) Supports a lot of languages including Hebrew. It compiles and runs under Linux with no problem. It is full Emacs, with Hebrew support and double-direction handling. It can be obtained from: ftp://kelim.jct.ac.il/pub/Hebrew

6.4 Dosemu

For a VC dosemu you can use your Hebrew from the Video card EPROM, and if you don't have it there are plenty of Hebrew dos fonts from EGA support to the VGA Hebrew support.

For X-Windows support you should download the file: ftp://sunsite.unc.edu/pub/Linux/X11/fonts/hebxfonts-0.1.tgz it's contain some fonts include one called vgah.pcf that you should install it on your fonts directory as describe above the fonts are:

6.5 XHTerm

There is a main port of the regular X-Term program for use with a Hebrew fonts - XHTerm = xterm + Hebrew support. The port for a sun machine was made avalible by the help of Danny `danny@cs.huji.ac.il`. Evgeny has some patch for use this port under Linux. His version should come with a pre-compiled XHTerm for both X11R5 and X11R6. You should use xhterm with the option `-fn` and a Hebrew font as described! Danny's port (for SUN) can be obtained from: ftp://ftp.huji.ac.il/pub/local/xhterm and the patched version of Evgeny Stambulchik is on: ftp://plasma-gate.weizmann.ac.il/pub/software/linux Get it from there and you'll get 5 fonts with it: `[heb10x20.pcf, heb6x13.bdf, heb6x13.pcf, heb8x13.bdf, heb8x13.pcf]`

6.6 TeX--XeT - Hebrew Tex.

The bigest problem with Tex with Hebrew is that the charecters should go backwards relative to Visual look (i.e. pico inserts the charecters from right to left), so the best thing is to get XHterm with a regular emacs and write the Hebrew left to right, backwards as well.

The newer NTeX distribution on sunsite (v1.5) includes everything, including TeX--XeT, precompiled for Linux. It can be obtained from ftp://sunsite.unc.edu.gz/pub/Linux/apps/tex/ntex. An older version of TeX--XeT can be obtained from ftp://noa.huji.ac.il/tex. This older version, however, has to be recompiled (not recommended).

These TeX distributions are fine if you use LaTeX2.09. If you want to use LaTeX2e (the current de facto standard) you have a problem. Alon Ziv (alonz@csa.cs.technion.ac.il) is currently working in support for LaTeX2e with Hebrew, using the Babel languages system. I don't know the current status of his work -- ask him!

---

7. Printer setup

Mainly there is not to say, if you have a regular ASCII line printer (who does, these days?) there is a good chance that there are Hebrew fonts in it on the `EPROM` chip.

If you use PostScript, you should download soft fonts to the printer (you can always use the earlier mentioned Web fonts for that. These fonts are also useable with Ghostscript).

If you have a PCL printer (LaserJet etc.), you can either use font cartridges or use Ghostscript.

8. Commercial products.

8.1 El-Mar software.

The Hebrew Support for X-Windows & Motif, is a product of El-Mar Software, which adds Hebrew functionality to many of the parts and layers of X-Windows and Motif, including Xlib, all of the widgets of Motif, hterm (Hebrew xterm), demos and simple useful applications (e.g. bi-lingual Motif-based editor), fonts (including scalable Type1), keyboard-manager in order to allow Hebrew and push-mode for non-Motif applications, etc.

Despite allowing many new features and variations for Motif widgets, the support doesn't have any modification to internal data-structures of Motif, so existing applications which were compiled and linked under non-Hebrew environment and libraries, can be relinked (without compilation!) and run with Hebrew (you can replace shared-libraries, so even the relink is not needed!)

By using another tool of us, Motif/Xplorer, you can take commercial applications (without their source) and translate them to Hebrew. This was the way of giving Hebrew support for Oracle Forms 4, Intellicorp's Kappa and OMW, CA-Unicenter, and many other leading UNIX tools sold in Israel. This product was purchased and adopted by most of the workstation vendors (9 of them, including the biggest: Sun, HP, SGI), and many other software houses. There are Makefiles for more than 30 platforms and operating systems.

We believe only in open software, so all the customers get the compelete source code. We have good relations with the leading forces in this industry, including the technical staff of X-Consortium and the technical staff of COSE.

```
Eli Marmor
El-Mar Software Ltd.
Voice: 050-237338
FAX: 09-984279
```

marmor@sunshine.cs.biu.ac.il

P.S.: The announcement of the Arabic Support for X-Windows & Motif, is expected in January. English, Hebrew, and Arabic will be handled by 8 bits (!), including the full set of Arabic glyphes.

9. Hebrew around the Internet.

9.1 WWW

- Jerusalem 1 - has many program and FAQ files about Hebrew on Unix and other platforms http://www.jer1.co.il.
- Gili Granot's Hebrew archive page - sumerize of all Hebrew related issues around the Web (include all kind of files) http://www.cs.technion.ac.il/~gil.
- Gavrie has some info about Hebrew on his ftp site: ftp://kelim.jct.ac.il

9.2 Gopher

- A one word testing for Hebrew-gopher can be found on gopher://shekel.jct.ac.il

9.3 Ftp

- Some Tex-Xet programs and the main FTP site for Tex Hebrew support for PC and Unix is at ftp://noa.huji.ac.il/tex.
- Horizon site as said allready contains the main site of pine/pico Hebrew support - ftp://horizon.huji.ac.il/pub.
- Gili Granot's Hebrew archive page ftp site is at ftp://ssl.cs.technion.ac.il/pub.

Linux Information Sheet

Michael K. Johnson <johnsonm@redhat.com>

v4.14, 1 September 1998

This document provides basic information about the Linux operating system, including an explanation of Linux, a list of features, some requirements, and some resources.

1. Introduction to Linux

2. Linux Features

3. Hardware Issues

- 3.1 Minimal configuration
- 3.2 Usable configuration
- 3.3 Supported hardware

4. An Incomplete List of Ported Programs and Other Software

5. Who uses Linux?

6. Getting Linux

- 6.1 Anonymous FTP
- 6.2 CDROM
- 6.3 Other methods of obtaining Linux

7. Legal Status of Linux

8. News About Linux

9. The Future

10. This document

11. Legalese

1. Introduction to Linux

Linux is a completely free reimplementation of the POSIX specification, with SYSV and BSD extensions (which means it looks like Unix, but does not come from the same source code base), which is available in both source code and binary form. Its copyright is owned by Linus Torvalds <torvalds@transmeta.com> and other contributors, and is freely redistributable under the terms of the GNU General Public License (GPL). A copy of the GPL is included with the Linux source; you can also get a copy from ftp://prep.ai.mit.edu/pub/gnu/COPYING

Linux, per se, is only the kernel of the operating system, the part that controls hardware, manages files, separates processes, and so forth. There are several combinations of Linux with sets of utilities and applications to form a complete operating system. Each of these combinations is called a **distribution** of Linux. The word **Linux**, though it in its strictest form refers specifically to the kernel, is also widely and correctly to refer to an entire operating system built around the Linux kernel. For a list and brief discription of various distributions, see http://sunsite.unc.edu/LDP/HOWTO/Distribution-HOWTO.html **None** of these distributions is ``the official Linux''.

Linux is **not** public domain, nor is it `shareware'. It is `free' software, commonly called **freeware** or **Open Source Software[tm]** (see http://www.opensource.org), and you may give away or sell copies, but you must include the source code or make it available in the same way as any binaries you give or sell. If you distribute any modifications, you are legally bound to distribute the source for those modifications. See the GNU General Public License for details.

Linux is still free as of version 2.0, and will continue to be free. Because of the nature of the GPL to which Linux is subject, it would be illegal for it to be made not free. Note carefully: the `free' part involves access to the source code rather than money; it is perfectly legal to charge money for distributing Linux, so long as you also distribute the source code. This is a generalization; if you want the fine points, read the GPL.

Linux runs on 386/486/Pentium machines with ISA, EISA, PCI and VLB busses. MCA (IBM's proprietary bus) is not well-supported in 2.0.x and earlier versions, but support has been added to the current development tree, 2.1.x. If you are interested, see http://glycerine.itsmm.uni.edu/mca

There is a port to multiple Motorola 680x0 platforms (currently running on some Amigas, Ataris, and VME machines), which now works quite well. It requires a 68020 with an MMU, a 68030, 68040, or a 68060, and also requires an FPU. Networking and X now work. See news:comp.os.linux.m68k

Linux runs well on DEC's Alpha CPU, currently supporting the "Jensen", "NoName", "Cabriolet", "Universal Desktop Box" (better known as the Multia), and many other platforms. For more information, see http://www.azstarnet.com/~axplinux/FAQ.html

Linux runs well on Sun SPARCs; most sun4c, sun4m, and sun4u machines now run Linux, with support for sun4 in development. Red Hat Linux is (as of this writing) the only Linux distribution available for

SPARCs; see http://www.redhat.com/support/docs/rhl-sparc/

Linux is being actively ported to the PowerPC architecture, including PowerMac (Nubus and PCI), Motorola, IBM, and Be machines. See http://www.cs.nmt.edu/~linuxppc/ and http://www.linuxppc.org/

Ports to other machines, including MIPS (see http://linus.linux.sgi.com and http://lena.fnet.fr/) and ARM, are under way and showing various amounts of progress. Don't hold your breath, but if you are interested and able to contribute, you may well find other developers who wish to work with you.

Linux is no longer considered to be in beta testing, as version 1.0 was released on March 14, 1994. There are still bugs in the system, and new bugs will creep up and be fixed as time goes on. Because Linux follows the ``open development model'', all new versions will be released to the public, whether or not they are considered ``production quality''. However, in order to help people tell whether they are getting a stable version or not, the following scheme has been implemented: Versions n.x.y, where x is an even number, are stable versions, and only bug fixes will be applied as y is incremented. So from version 1.2.2 to 1.2.3, there were only bug fixes, and no new features. Versions n.x.y, where x is an odd number, are beta-quality releases for developers only, and may be unstable and may crash, and are having new features added to them all the time. >From time to time, as the currect development kernel stabilizes, it will be frozen as the new ``stable'' kernel, and development will continue on a new development version of the kernel. Note that most releases of the Linux kernel, beta or not, are relatively robust; ``stable'' in this context means ``slow to change'' in addition to ``robust''.

The current stable version is 2.0.35 (this will continue to change as new device drivers get added and bugs fixed), and development has also started on the experimental 2.1.x kernels. The Linux kernel source code contains a file, Documentation/Changes, which explains changes that you should be aware of when upgrading from one kernel version to another. However, the great majority of Linux users simply update their Linux distribution occasionally to get a new kernel version.

Most versions of Linux, beta or not, are quite robust, and you can keep using those if they do what you need and you don't want to be on the bleeding edge. One site had a computer running version 0.97p1 (dating from the summer of 1992) for over 136 days without an error or crash. (It would have been longer if the backhoe operator hadn't mistaken a main power transformer for a dumpster...) Others have posted uptimes in excess of a year. One site still had a computer running Linux 0.99p15s over 600 days at last report.

One thing to be aware of is that Linux is developed using an open and distributed model, instead of a closed and centralized model like much other software. This means that the current development version is always public (with up to a week or two of delay) so that anybody can use it. The result is that whenever a version with new functionality is released, it almost always contains bugs, but it also results in a very rapid development so that the bugs are found and corrected quickly, often in hours, as many people work to fix them.

In contrast, the closed and centralized model means that there is only one person or team working on the project, and they only release software that they think is working well. Often this leads to long intervals between releases, long waiting for bug fixes, and slower development. The latest release of such software to the public is sometimes of higher quality, but the development speed is generally much slower.

For a discussion of these two models, read ``The Cathedral and the Bazaar'' at

http://sagan.earthspace.net/~esr/writings/cathedral-bazaar/ by Eric Raymond.

As of September 1, 1998, the current stable version of Linux is 2.0.35, and the latest development version is 2.1.119.

2. Linux Features

- multitasking: several programs running at the same time.
- multiuser: several users on the same machine at the same time (and **no** two-user licenses!).
- multiplatform: runs on many different CPUs, not just Intel.
- multiprocessor: SMP support is available on the Intel and SPARC platforms (with work currently in progress on other platforms), and Linux is used in several loosely-coupled MP applications, including Beowulf systems (see http://cesdis.gsfc.nasa.gov/linux-web/beowulf/beowulf.html) and the Fujitsu AP1000+ SPARC-based supercomputer.
- multithreading: has native kernel support for multiple independent threads of control within a single process memory space.
- runs in protected mode on the 386.
- has memory protection between processes, so that one program can't bring the whole system down.
- demand loads executables: Linux only reads from disk those parts of a program that are actually used.
- shared copy-on-write pages among executables. This means that multiple process can use the same memory to run in. When one tries to write to that memory, that page (4KB piece of memory) is copied somewhere else. Copy-on-write has two benefits: increasing speed and decreasing memory use.
- virtual memory using paging (not swapping whole processes) to disk: to a separate partition or a file in the filesystem, or both, with the possibility of adding more swapping areas during runtime (yes, they're still called swapping areas). A total of 16 of these 128 MB (2GB in recent kernels) swapping areas can be used at the same time, for a theoretical total of 2 GB of useable swap space. It is simple to increase this if necessary, by changing a few lines of source code.
- a unified memory pool for user programs and disk cache, so that all free memory can be used for caching, and the cache can be reduced when running large programs.
- dynamically linked shared libraries (DLL's), and static libraries too, of course.
- does core dumps for post-mortem analysis, allowing the use of a debugger on a program not only while it is running but also after it has crashed.
- mostly compatible with POSIX, System V, and BSD at the source level.
- through an iBCS2-compliant emulation module, mostly compatible with SCO, SVR3, and SVR4 at the binary level.
- all source code is available, including the whole kernel and all drivers, the development tools and all user programs; also, all of it is freely distributable. Plenty of commercial programs are being provided for Linux without source, but everything that has been free, including the entire base operating system, is still free.
- POSIX job control.

TOC

- pseudoterminals (pty's).
- 387-emulation in the kernel so that programs don't need to do their own math emulation. Every computer running Linux appears to have a math coprocessor. Of course, if your computer already contains an FPU, it will be used instead of the emulation, and you can even compile your own kernel with math emulation removed, for a small memory gain.
- support for many national or customized keyboards, and it is fairly easy to add new ones dynamically.
- multiple virtual consoles: several independent login sessions through the console, you switch by pressing a hot-key combination (not dependent on video hardware). These are dynamically allocated; you can use up to 64.
- Supports several common filesystems, including minix, Xenix, and all the common system V filesystems, and has an advanced filesystem of its own, which offers filesystems of up to 4 TB, and names up to 255 characters long.
- transparent access to MS-DOS partitions (or OS/2 FAT partitions) via a special filesystem: you don't need any special commands to use the MS-DOS partition, it looks just like a normal Unix filesystem (except for funny restrictions on filenames, permissions, and so on). MS-DOS 6 compressed partitions do not work at this time without a patch (dmsdosfs). VFAT (WNT, Windows 95) support and FAT-32 is available in Linux 2.0
- special filesystem called UMSDOS which allows Linux to be installed on a DOS filesystem.
- read-only HPFS-2 support for OS/2 2.1
- HFS (Macintosh) file system support is available separately as a module.
- CD-ROM filesystem which reads all standard formats of CD-ROMs.
- TCP/IP networking, including ftp, telnet, NFS, etc.
- Appletalk server
- Netware client and server
- Lan Manager/Windows Native (SMB) client and server
- Many networking protocols: the base protocols available in the latest development kernels include TCP, IPv4, IPv6, AX.25, X.25, IPX, DDP (Appletalk), Netrom, and others. Stable network protocols included in the stable kernels currently include TCP, IPv4, IPX, DDP, and AX.25.

---

3. Hardware Issues

3.1 Minimal configuration

The following is probably the smallest possible configuration that Linux will work on: 386SX/16, 1 MB RAM, 1.44 MB or 1.2 MB floppy, any supported video card (+ keyboards, monitors, and so on of course). This should allow you to boot and test whether it works at all on the machine, but you won't be able to do anything useful. See http://rsphy1.anu.edu.au/~gpg109/mem.html for minimal Linux configurations

In order to do something, you will want some hard disk space as well, 5 to 10 MB should suffice for a very minimal setup (with only the most important commands and perhaps one or two small applications installed, like, say, a terminal program). This is still very, very limited, and very uncomfortable, as it doesn't leave enough room to do just about anything, unless your applications are quite limited. It's generally not recommended for anything but testing if things work, and of course to be able to brag about small resource requirements.

3.2 Usable configuration

If you are going to run computationally intensive programs, such as gcc, X, and TeX, you will probably want a faster processor than a 386SX/16, but even that should suffice if you are patient.

In practice, you will want at least 4 MB of RAM if you don't use X, and 8 MB if you do. Also, if you want to have several users at a time, or run several large programs (compilations for example) at a time, you may want more than 4 MB of memory. It may still work with a smaller amount of memory, but it will use virtual memory (using the hard drive as **slow** memory) and that will be so slow as to be unusable. If you use many programs at the same time, 16 MB will reduce swapping considerably. If you don't want to swap appreciably under normal load, 32 MB will probably suffice for a single user. If you run lots of memory-intensive applications at once, 64 MB may be necessary to avoid lots of swapping. Of course, if you run memory-hungry applications, you may want more.

The amount of hard disk you need depends on what software you want to install. The normal basic set of Unix utilities, shells, and administrative programs should be comfortable in less than 10 MB, with a bit of room to spare for user files. For a more complete system, get Red Hat, Debian, or another distribution, and assume that you will need 60 to 600 MB, depending on what you choose to install and what distribution you get. Add whatever space you want to reserve for user files to these totals. With today's prices on hard drives, if you are buying a new system, it makes no sense to buy a drive that is too small. Get at least 2 GB, preferably 4 GB or more, and you will not regret it. Linux will happily handle very large hard drive such as the recently popular 11 GB IDE hard drives or 18 GB SCSI hard drives.

Add more memory, more hard disk, a faster processor and other stuff depending on your needs, wishes and budget to go beyond the merely usable. In general, one big difference from DOS is that with Linux,

adding memory makes a large difference, whereas with DOS, extra memory doesn't make that much difference. This of course has something to do with DOS's 640KB limit, which is completely nonexistent under Linux.

TOC

3.3 Supported hardware

CPU:

Anything that runs 386 protected mode programs. All models of 386s 486s, Pentiums, Pentium Pros, Pentium IIs, and clones of these chips should work. (286s and below may someday be supported on a smaller kernel called ELKS (Embeddable Linux Kernel Subset), but don't expect the same capabilities). A version for the 680x0 CPU (for *x* = 2 with external MMU, 3, 4, and 6) which runs on Amigas and Ataris can be found at tsx-11.mit.edu in the 680x0 directory. Many DEC Alphas, SPARCs, and PowerPC machines are supported. Ports are also being done to the ARM, StrongARM, and MIPS architectures. More details are available elsewhere.

Architecture:

PCI, ISA, EISA, and VLB busses. MCA (mostly true blue PS/2s) support is incomplete but improving (see above). Linux puts higher demands on hardware than DOS, Windows, and in fact most operating systems. This means that some marginal hardware that doesn't fail when running less demanding operating system may fail when running Linux. Linux is an excellent memory tester...

RAM:

Up to 1 GB on Intel; more on 64-bit platforms. Some people (including Linus) have noted that adding ram without adding more cache at the same time has slowed down their machine extremely, so if you add memory and find your machine slower, try adding more cache. Some machines can only cache certain amounts of memory regardless of how much RAM is installed (64 MB is the most one popular chipset can cache). Over 64 MB of memory will require a boot-time parameter with kernels 2.0.35 and earlier, as the BIOS was originally designed to be unable to report more than 64MB. Recent 2.1.x kernels and later are able to detect more memory in a system.

Data storage:

Generic AT drives (EIDE, IDE, 16 bit HD controllers with MFM or RLL, or ESDI) are supported, as are SCSI hard disks and CD-ROMs, with a supported SCSI adaptor. Generic XT controllers (8 bit controllers with MFM or RLL) are also supported. Supported SCSI adaptors: Advansys, Adaptec 1542, 1522, 1740, 27xx, and 29xx (with some exceptions) series, Buslogic MultiMaster and Flashpoint, NCR53c8xx-based controllers, DPT controllers, Qlogic ISP and FAS controllers, Seagate ST-01 and ST-02, Future Domain TMC-88x series (or any board based on the TMC950 chip) and TMC1660/1680, Ultrastor 14F, 24F and 34F, Western Digital wd7000, and others. SCSI, QIC-02, and some QIC-80 tapes are also supported. Besides IDE and SCSU cdroms, several proprietary CD-ROM devices are also supported, including Matsushita/Panasonic, Mitsumi, Sony, Soundblaster, Toshiba, ATAPI (EIDE), SCSI, and others. For exact models, check the hardware compatibility HOWTO. **N.B.** These lists are incomplete, and always will be. Distribution vendors maintain more up-to-date lists.

Video:

VGA, EGA, CGA, or Hercules (and compatibles) work in text mode. For graphics and X, there is support for (at least) normal VGA, some super-VGA cards (most of the cards based on Tseng, Paradise, and some Trident chipsets), S3, 8514/A, ATI, Matrox, and Hercules. (Linux uses the XFree86 X server, so that determines what cards are supported. A full list of supported chipsets alone takes over a page. See http://www.XFree86.org/)

Networking:

Ethernet support includes 3COM 503/509/579/589/595/905 (501/505/507 are supported but not recomended), AT&T GIS (neé NCR) WaveLAN, most WD8390-based cards, most WD80x3-based cards, NE1000/2000 and most clones, AC3200, Apricot 82596, AT1700, ATP, DE425/434/435/500, D-Link DE-600/620, DEPCA, DE100/101, DE200/201/202 Turbo, DE210, DE422, Cabletron E2100 (not recommended), Intel EtherExpress (not recommended), EtherExpress Pro, EtherExpress 100, DEC EtherWORKS 3, HP LAN, HP PCLAN/plus, most AMD LANCE-based cards, NI5210, ni6510, SMC Ultra, DEC 21040 (tulip), Zenith Z-Note ethernet, All Zircom cards and all Cabletron cards other than the E2100 are unsupported, due to the manufacturers unwillingness to release programming information freely.

FDDI support currently includes the DEF*xx* cards from DEC.

Point-to-Point networking support include PPP, SLIP, CSLIP, and PLIP. PPP support is available for both standard asyncronous devices like modems, and syncronous connections like ISDN.

Limited Token Ring support is available.

Serial:

Most 16450 and 16550 UART-based boards, including AST Fourport, the Usenet Serial Card II, and others. Intelligent boards supported include Cyclades Cyclom series (supported by the manufacturer), Comtrol Rocketport series (supported by the manufacturer), Stallion (most boards; supported by the manufacturer), and Digi (some boards; supported by the manufacturer). Some ISDN, frame relay, and leased line hardware is supported.

Other hardware:

SoundBlaster, ProAudio Spectrum 16, Gravis Ultrasound, most other sound cards, most (all?) flavours of bus mice (Microsoft, Logitech, PS/2), etc.

---

4. An Incomplete List of Ported Programs and Other Software

Most of the common Unix tools and programs have been ported to Linux, including almost all GNU software and many X clients from various sources. Actually, ported is often too strong a word, since many programs compile out of the box without modifications, or only small modifications, because Linux tracks POSIX quite closely. There are never enough applications for any operating system, but Linux is gaining both end-user applications and server applications. Contact the vendor of your favorite commercial Unix application and ask if they have ported it to Linux.

Here is an incomplete list of software that is known to work under Linux:

Basic Unix commands:

`ls`, `tr`, `sed`, `awk` and so on (you name it, Linux probably has it).

Development tools:

`gcc`, `gdb`, `make`, `bison`, `flex`, `perl`, `rcs`, `cvs`, `prof`.

Languages and Environments:

C, C++, Objective C, Java, Modula-3, Modula-2, Oberon, Ada95, Pascal, Fortran, ML, scheme, Tcl/tk, Perl, Python, Common Lisp, and many others.

Graphical environments:

GNOME and KDE (desktops), X11R6 (XFree86 3.x), X11R5 (XFree86 2.x), MGR.

Editors:

GNU Emacs, XEmacs, MicroEmacs, `jove`, ez, `epoch`, `elvis` (GNU vi), `vim`, `vile`, `joe`, `pico`, `jed`, and others.

Shells:

`bash` (POSIX sh-compatible), `zsh` (includes `ksh` compatiblity mode), `pdksh`, `tcsh`, `csh`, `rc`, `es`, `ash` (mostly sh-compatible shell used as `/bin/sh` by BSD), and many more.

Telecommunication:

PPP, UUCP, SLIP, CSLIP, full TCP/IP communication toolset, `kermit`, `szrz`, `minicom`, `pcomm`, `xcomm`, `term` (runs multiple shells, redirects network activity, and allows remote X, all over one modem line), Seyon (popular X-windows communications program), and several fax and voice-mail (using ZyXEL and other modems) packages are available. Of course, remote serial and network logins are supported.

News and mail:

C-news, `innd`, `trn`, `nn`, `tin`, `smail`, `elm`, `mh`, `exmh`, `pine`, `mutt`, etc.

TOC

Textprocessing:

TeX, `groff`, `doc`, `ez`, LyX, Lout, Linuxdoc-SGML, and others.

Games:

Nethack, several Muds and X games, and lots of others. One of those games is looking through all the games available at tsx-11 and sunsite.

All of these programs (and this isn't even a hundredth of what is available) are freely available. Commercial software is becoming widely available; ask the vendor of your favorite commercial software if they support Linux.

---

5. Who uses Linux?

Linux is freely available, and no one is required to register their copies with any central authority, so it is difficult to know how many people use Linux. Several businesses now survive solely on selling and supporting Linux, and the Linux newsgroups are some of the most heavily read on the internet, so the number is likely in the millions, but firm numbers are hard to come by. The best market research currently indicates between 7.5 and 11 million users.

One brave soul, Harald T. Alvestrand <Harald.T.Alvestrand@uninett.no>, has decided to try to count Linux users the hard way: one by one. If you are willing to be counted as a Linux user, please use the web forms available at http://counter.li.org/ Alternatively, you can send a message to linux-counter@uninett.no with one of the following subjects: `I use Linux at home', `I use Linux at work', or `I use Linux at home and at work'. He will also accept `third-party' registrations; ask him for details.

He posts his counts to news:comp.os.linux.misc each month; they are also available from http://counter.li.org/.

6. Getting Linux

6.1 Anonymous FTP

For freely-redistributable Linux documentation, see the Linux Documentation Project sites at ftp://sunsite.unc.edu/pub/Linux/docs/LDP/ and http://sunsite.unc.edu/LDP/

Stay tuned to the news:comp.os.linux.announce newsgroup for further developments.

At least the following anonymous ftp sites carry Linux.

Textual name	Numeric address	Linux directory
tsx-11.mit.edu	18.172.1.2	/pub/linux
sunsite.unc.edu	152.2.22.81	/pub/Linux
ftp.funet.fi	128.214.248.6	/pub/Linux
net.tamu.edu	128.194.177.1	/pub/linux
ftp.mcc.ac.uk	130.88.203.12	/pub/linux
src.doc.ic.ac.uk	146.169.2.1	/packages/linux
fgb1.fgb.mw.tu-muenchen.de	129.187.200.1	/pub/linux
ftp.informatik.tu-muenchen.de	131.159.0.110	/pub/comp/os/linux
ftp.dfv.rwth-aachen.de	137.226.4.111	/pub/linux
ftp.informatik.rwth-aachen.de	137.226.225.3	/pub/Linux
ftp.Germany.EU.net	192.76.144.75	/pub/os/Linux
ftp.lip6.fr	132.227.77.2	/pub/linux
ftp.uu.net	137.39.1.9	/systems/unix/linux
wuarchive.wustl.edu	128.252.135.4	mirrors/linux
ftp.win.tue.nl	131.155.70.100	/pub/linux
ftp.stack.urc.tue.nl	131.155.2.71	/pub/linux
srawgw.sra.co.jp	133.137.4.3	/pub/os/linux
cair.kaist.ac.kr		/pub/Linux
ftp.denet.dk	129.142.6.74	/pub/OS/linux
NCTUCCCA.edu.tw	140.111.1.10	/Operating-Systems/Linux
nic.switch.ch	130.59.1.40	/mirror/linux
sunsite.cnlab-switch.ch	193.5.24.1	/mirror/linux
cnuce_arch.cnr.it	131.114.1.10	/pub/Linux
ftp.monash.edu.au	130.194.11.8	/pub/linux
ftp.dstc.edu.au	130.102.181.31	/pub/linux
ftp.sydutech.usyd.edu.au	129.78.192.2	/pub/linux

tsx-11.mit.edu and fgb1.fgb.mw.tu-muenchen.de are the official sites for Linux's GCC. Some sites mirror other sites. Please use the site closest (network-wise) to you whenever possible.

At least sunsite.unc.edu and ftp.informatik.tu-muenchen.de offer ftpmail services. Mail ftpmail@sunsite.unc.edu or ftp@informatik.tu-muenchen.de for help.

If you are lost, try looking at http://sunsite.unc.edu/LDP/HOWTO/Distribution-HOWTO.html, where several distributions are enumerated. Red Hat Linux and Debian appear to be the most popular distributions at the moment, at least in the U.S.

6.2 CDROM

Most people now install Linux from CDROM's. The distributions have grown to hundreds of MBs of Linux software, and downloading that over even a 28.8 modem or an ISDN connection takes a **long** time.

There are essentially two ways to purchase a Linux distribution on CDROM: as part of an archive of FTP sites, or directly from the manufacturer. If you purchase an archive, you will often get several different distributions to choose from, but support is generally not included. When you purchase a distribution directly from the vendor, you usually only get one distribution, but you usually get some form of support, usually installation support.

6.3 Other methods of obtaining Linux

There are many BBSs that have Linux files. A list of them is occasionally posted to comp.os.linux.announce. Ask friends and user groups, or order one of the commmercial distributions. A list of these is contained in the Linux distribution HOWTO, available as http://sunsite.unc.edu/LDP/HOWTO/Distribution-HOWTO.html, and posted regularily to the news:comp.os.linux.announce newsgroup.

---

7. Legal Status of Linux

Although Linux is supplied with the complete source code, it is copyrighted software, not public domain. However, it is available for free under the GNU General Public License, sometimes referred to as the ``copyleft''. See the GPL for more information. The programs that run under Linux each have their own copyright, although many of them use the GPL as well. X uses the MIT X copyright, and some utilities are under the BSD copyright. In any case, all of the software on the FTP site is freely distributable (or else it shouldn't be there).

8. News About Linux

A monthly magazine, called *Linux Journal*, was launched over four years ago. It includes articles intended for almost all skill levels, and is intended to be helpful to all Linux users. One-year subscriptions are $22 in the U.S., $27 in Canada and Mexico, and $32 elsewhere, payable in US currency. Subscription inquiries can be sent via email to subs@ssc.com, or faxed to +1-206-782-7191, or phoned to +1-206-782-7733, or mailed to Linux Journal, PO Box 85867, Seattle, WA 98145-1867 USA. SSC has a PGP public key available for encrypting your mail to protect your credit card number; finger info@ssc.com to get the key.

There are several Usenet newsgroups for Linux discussion, and also several mailing lists. See the Linux FAQ for more information about the mailing lists (you should be able to find the FAQ either in the newsgroup or on the FTP sites).

The newsgroup news:comp.os.linux.announce is a moderated newsgroup for announcements about Linux (new programs, bug fixes, etc).

The newsgroup news:comp.os.linux.answers is a moderated newsgroup to which the Linux FAQ, HOWTO documents, and other documentation postings are made.

The newsgroup news:comp.os.linux.admin is an unmoderated newsgroup for discussion of administration of Linux systems.

The newsgroup news:comp.os.linux.development.system is an unmoderated newsgroup specifically for discussion of Linux **kernel** development. The only application development questions that should be discussed here are those that are intimately associated with the kernel. All other development questions are probably generic Unix development questions and should be directed to a comp.unix group instead, unless they are very Linux-specific applications questions, in which case they should be directed at comp.os.linux.development.apps.

The newsgroup news:comp.os.linux.development.apps is an unmoderated newsgroup specifically for discussion of Linux-related applications development. It is not for discussion of where to get applications for Linux, nor a discussion forum for those who would like to see applications for Linux.

The newsgroup news:comp.os.linux.hardware is for Linux-specific hardware questions.

The newsgroup news:comp.os.linux.networking is for Linux-specific networking development and setup questions.

The newsgroup news:comp.os.linux.x is for Linux-specific X Windows questions.

The newsgroup news:comp.os.linux.misc is the replacement for comp.os.linux, and is meant for any discussion that doesn't belong elsewhere.

In general, **do not** crosspost between the Linux newsgroups. The **only** crossposting that is appropriate is

TOC

an occasional posting between one unmoderated group and news:comp.os.linux.announce. The whol point of splitting the old comp.os.linux group into many groups is to reduce traffic in each group. Those that do not follow this rule will be flamed without mercy...

Linux is on the web at the URL http://sunsite.unc.edu/LDP/

---

9. The Future

After Linux 1.0 was released, work was done on several enhancements. Linux 1.2 included disk access speedups, TTY improvements, virtual memory enhancements, multiple platform support, quotas, and more. Linux 2.0, the current stable version, has even more enhancements, including many performance improvements, several new networking protocols, one of the fastest TCP/IP implementations in the world, and far, far more. Even higher performance, more networking protocols, and more device drivers will be available in Linux 2.2.

Even with over 3/4 million lines of code in the kernel, there is plenty of code left to write, and even more documentation. Please join the linux-doc@vger.rutgers.edu mailing list if you would like to contribute to the documentation. Send mail to majordomo@vger.rutgers.edu with a single line containing the word ``help'' in the body (**NOT** the subject) of the message.

10. This document

This document is maintained by Michael K. Johnson <johnsonm@redhat.com>. Please mail me with any comments, no matter how small. I can't do a good job of maintaining this document without your help. A more-or-less current copy of this document can always be found at http://sunsite.unc.edu/LDP/

11. Legalese

Trademarks are owned by their owners. There is no warranty about the information in this document. Use and distribute at your own risk. The content of this document is in the public domain, but please be polite and attribute any quotes.

Linux IPCHAINS-HOWTO

Paul Russell, ipchains@rustcorp.com

v1.0.7, 12 March 1999

This document aims to describe how to obtain, install and configure the enhanced IP firewalling chains software for Linux, and some ideas on how you might use them.

1. Introduction

- 1.1 What?
- 1.2 Why?
- 1.3 How?
- 1.4 Where?

2. Packet Filtering Basics

- 2.1 What?
- 2.2 Why?
- 2.3 How?

3. I'm confused! Routing, masquerading, portforwarding, ipautofw...

- 3.1 Rusty's Three-Line Guide To Masquerading
- 3.2 Gratuitous Promotion: WatchGuard Rules
- 3.3 Common Firewall-like Setups
- 3.4 More Information on Masquerading

TOC

4. IP Firewalling Chains

- 4.1 How Packets Traverse The Filters
- 4.2 Useful Examples

5. Miscellaneous.

- 5.1 How to Organize Your Firewall Rules
- 5.2 What Not To Filter Out
- 5.3 Filtering out Ping of Death
- 5.4 Filtering out Teardrop and Bonk
- 5.5 Filtering out Fragment Bombs
- 5.6 Changing Firewall Rules
- 5.7 How Do I Set Up IP Spoof Protection?
- 5.8 Advanced Projects
- 5.9 Future Enhancements

6. Common Problems

- 6.1 ipchains -L Freezes!
- 6.2 Masquerading/Forwarding Doesn't Work!
- 6.3 -j REDIR doesn't work!
- 6.4 Wildcard Interfaces Don't Work!
- 6.5 TOS Doesn't Work!
- 6.6 ipautofw and ipportfw Don't Work!
- 6.7 xosview is Broken!
- 6.8 Segmentation Fault With `-j REDIRECT'!
- 6.9 I Can't Set Masquerading Timeouts!
- 6.10 I Want to Firewall IPX!

7. A Serious Example.

- 7.1 The Arrangement
- 7.2 Goals
- 7.3 Before Packet Filtering

- 7.4 Packet Filtering for Through Packets
- 7.5 Finally

8. Appendix: Differences between ipchains and ipfwadm.

- 8.1 Quick-Reference table.
- 8.2 Examples of translated ipfwadm commands

9. Appendix: Using the ipfwadm-wrapper script.

10. Appendix: Thanks.

---

1. Introduction

This is the Linux IPCHAINS-HOWTO; see Where? for the master site, which contains the latest copy. You should read the Linux NET-3-HOWTO as well. The IP-Masquerading HOWTO, the PPP-HOWTO, the Ethernet-HOWTO and the Firewall HOWTO might make interesting reading. (Then again, so might the alt.fan.bigfoot FAQ).

If packet filtering is passe to you, read Section Why?, Section How?, and scan through the titles in Section IP Firewalling Chains.

If you are converting from `ipfwadm`, read Section Introduction, Section How?, and Appendices in section Differences between ipchains and ipfwadm and section Using the `ipfwadm-wrapper' script.

1.1 What?

Linux `ipchains` is a rewrite of the Linux IPv4 firewalling code (which was mainly stolen from BSD) and a rewrite of `ipfwadm`, which was a rewrite of BSD's `ipfw`, I believe. It is required to administer the IP packet filters in Linux kernel versions 2.1.102 and above.

1.2 Why?

The older Linux firewalling code doesn't deal with fragments, has 32-bit counters (on Intel at least), doesn't allow specification of protocols other than TCP, UDP or ICMP, can't make large changes atomically, can't specify inverse rules, has some quirks, and can be tough to manage (making it prone to user error).

1.3 How?

Currently the code is in the mainstream kernel from 2.1.102. For the 2.0 kernel series, you will need to download a kernel patch from the web page. If your 2.0 kernel is more recent than the supplied patch, the older patch should be OK; this part of the 2.0 kernels is fairly stable (eg. the 2.0.34 kernel patch works just fine on the 2.0.35 kernel). Since the 2.0 patch is incompatible with the ipportfw and ipautofw patches, I don't recommend applying it unless you really need some functionality that ipchains offers.

1.4 Where?

The official page is The Linux IP Firewall Chains Page

There is a mailing list for bug reports, discussion, development and usage. Join the mailing list by sending a message containing the word ``subscribe'' to ipchains-request at rustcorp.com. To mail to the

list use `ipchains' instead of `ipchains-request'.

2. Packet Filtering Basics

2.1 What?

All traffic through a network is sent in the form of **packets**. For example, downloading this package (say it's 50k long) might cause you to receive 36 or so packets of 1460 bytes each, (to pull numbers at random).

The start of each packet says where it's going, where it came from, the type of the packet, and other administrative details. This start of the packet is called the **header**. The rest of the packet, containing the actual data being transmitted, is usually called the **body**.

Some protocols, such **TCP**, which is used for web traffic, mail, and remote logins, use the concept of a `connection' -- before any packets with actual data are sent, various setup packets (with special headers) are exchanged saying `I want to connect', `OK' and `Thanks'. Then normal packets are exchanged.

A packet filter is a piece of software which looks at the *header* of packets as they pass through, and decides the fate of the entire packet. It might decide to **deny** the packet (ie. discard the packet as if it had never received it), **accept** the packet (ie. let the packet go through), or **reject** the packet (like deny, but tell the source of the packet that it has done so).

Under Linux, packet filtering is built into the kernel, and there are a few trickier things we can do with packets, but the general principle of looking at the headers and deciding the fate of the packet is still there.

2.2 Why?

Control. Security. Watchfulness.

Control:

> when you are using a Linux box to connect your internal network to another network (say, the Internet) you have an opportunity to allow certain types of traffic, and disallow others. For example, the header of a packet contains the destination address of the packet, so you can prevent packets going to a certain part of the outside network. As another example, I use Netscape to access the Dilbert archives. There are advertisements from doubleclick.net on the page, and Netscape wastes my time by cheerfully downloading them. Telling the packet filter not to allow any packets to or from the addresses owned by doubleclick.net solves that problem (there are better ways of doing this though).

Security:

> when your Linux box is the only thing between the chaos of the Internet and your nice, orderly network, it's nice to know you can restrict what comes tromping in your door. For example, you

TOC

might allow anything to go out from your network, but you might be worried about the well-known `Ping of Death' coming in from malicious outsiders. As another example, you might not want outsiders telnetting to your Linux box, even though all your accounts have passwords; maybe you want (like most people) to be an observer on the Internet, and not a server (willing or otherwise) -- simply don't let anyone connect in, by having the packet filter reject incoming packets used to set up connections.

Watchfulness:

sometimes a badly configured machine on the local network will decide to spew packets to the outside world. It's nice to tell the packet filter to let you know if anything abnormal occurs; maybe you can do something about it, or maybe you're just curious by nature.

2.3 How?

A Kernel With Packet Filtering

You need a kernel which has the new IP firewall chains in it. You can tell if the kernel you are running right now has this installed by looking for the file `/proc/net/ip_fwchains'. If it exists, you're in.

If not, you need to make a kernel that has IP firewall chains. First, download the source to the kernel you want. If you have a kernel numbered 2.1.102 or higher, you won't need to patch it (it's in the mainstream kernel now). Otherwise, apply the patch from the web page listed above, and set the configuration as detailed below. If you don't know how to do this, don't panic -- read the Kernel-HOWTO.

The configuration options you will need to set *for the 2.0-series kernel* are:

```
CONFIG_EXPERIMENTAL=y
CONFIG_FIREWALL=y
CONFIG_IP_FIREWALL=y
CONFIG_IP_FIREWALL_CHAINS=y
```

For the *2.1 or 2.2 series kernels*:

```
CONFIG_FIREWALL=y
CONFIG_IP_FIREWALL=y
```

The tool `ipchains` talks to the kernel and tells it what packets to filter. Unless you are a programmer, or overly curious, this is how you will control the packet filtering.

ipchains

The `ipchains` tool inserts and deletes rules from the kernel's packet filtering section. This means that whatever you set up, it will be lost upon reboot; see Making Rules Permanent for how to make sure they are restored the next time Linux is booted.

`ipchains` replaces `ipfwadm`, which was used for the old IP Firewall code. There is a set of useful scripts available from the ipchains ftp site:

ftp://ftp.rustcorp.com/ipchains/ipchains-scripts-1.1.2.tar.gz

This contains a shell script called `ipfwadm-wrapper` which allows you to do packet filtering as it was done before. You probably shouldn't use this script unless you want a quick way of upgrading a system which uses `ipfwadm` (it's slower, and doesn't check arguments, etc). In that case, you don't need this HOWTO much either.

See Appendix Differences between ipchains and ipfwadm and Appendix Using the `ipfwadm-wrapper' script for more details on `ipfwadm` issues.

Making Rules Permanent

Your current firewall setup is stored in the kernel, and thus will be lost on reboot. I recommend using the `ipchains-save' and `ipchains-restore' scripts to make your rules permanent. To do this, set up your rules, then run (as root):

```
# ipchains-save > /etc/ipchains.rules
#
```

Create a script like the following:

```
#! /bin/sh
# Script to control packet filtering.

# If no rules, do nothing.
[ -f /etc/ipchains.rules ] || exit 0

case "$1" in
    start)
        echo -n "Turning on packet filtering:"
        /sbin/ipchains-restore < /etc/ipchains.rules || exit 1
        echo 1 > /proc/sys/net/ipv4/ip_forward
        echo "."
        ;;
    stop)
        echo -n "Turning off packet filtering:"
        echo 0 > /proc/sys/net/ipv4/ip_forward
```

```
            /sbin/ipchains -X
            /sbin/ipchains -F
            /sbin/ipchains -P input ACCEPT
            /sbin/ipchains -P output ACCEPT
            /sbin/ipchains -P forward ACCEPT
            echo "."
            ;;
        *)
            echo "Usage: /etc/init.d/packetfilter {start|stop}"
            exit 1
            ;;
    esac

    exit 0
```

Make sure this is run early in the bootup procedure. In my case (Debian 2.1), I make a symbolic link called `S39packetfilter' in the `/etc/rcS.d' directory (this will be run before S40network).

3. I'm confused! Routing, masquerading, portforwarding, ipautofw...

This HOWTO is about packet filtering. This means deciding whether a packet should be allowed to pass or not. However, Linux being the hacker's playground that it is, you probably want to do more than that.

One problem is that the same tool (``ipchains'') is used to control both masquerading and transparent proxying, although these are notionally separate from packet filtering (the current Linux implementation blurs these together unnaturally, leaving the impression that they are closely related).

Masquerading and proxying are covered by separate HOWTOs, and the auto forwarding and port forwarding features are controlled by separate tools, but since so many people keep asking me about it, I'll include a set of common scenarios and indicate when each one should be applied. The security merits of each setup will not be discussed here.

3.1 Rusty's Three-Line Guide To Masquerading

This assumes that your **external** interface is called `ppp0'. Use ifconfig to find out, and adjust to taste.

```
# ipchains -P forward DENY
# ipchains -A forward -i ppp0 -j MASQ
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

3.2 Gratuitous Promotion: WatchGuard Rules

You can buy off-the-shelf firewalls. An excellent one is WatchGuard's FireBox. It's excellent because I like it, it's secure, it's Linux-based, and because they are funding the maintenance of ipchains as well as the new firewalling code (aimed for 2.3). In short, WatchGuard are paying for me to eat while I work for you. So please consider their stuff.

http://www.watchguard.com

3.3 Common Firewall-like Setups

You run littlecorp.com. You have an internal network, and a single dialup (PPP) connection to the Internet (firewall.littlecorp.com which is 1.2.3.4). You run Ethernet on your local network, and your personal machine is called "myhost".

This section will illustrate the different arrangement which are common. Read carefully, because they are each subtly different.

TOC

Private Network: Traditional Proxies

In this scenario, packets from the private network never traverse the Internet, and vice versa. The IP addresses of the private network should be assigned from the RFC1597 Private Network Allocations (ie. 10.*.*.*, 172.16.*.* or 192.168.*.*).

The only way things ever connect to the Internet is by connecting to the firewall, which is the only machine on both networks which connects onwards. You run a program (on the firewall) called a proxy to do this (there are proxies for FTP, web access, telnet, RealAudio, Usenet News and other services). See the Firewall HOWTO.

Any services you wish the Internet to access must be on the firewall. (But see Limited Internal Services below).

Example: Allowing web access from private network to the Internet.

1. The private network is assigned 192.168.1.* addresses, with myhost being 192.168.1.100, and the firewall's Ethernet interface being assigned 192.168.1.1.
2. A web proxy (eg. "squid") is installed and configured on the firewall, say running on port 8080.
3. Netscape on the private network is configured to use the firewall port 8080 as a proxy.
4. DNS does not need to be configured on the private network.
5. DNS does need to be configured on the firewall.
6. No default route (aka gateway) needs to be configured on the private network.

Netscape on myhost reads http://slashdot.org.

1. Netscape connects to the firewall port 8080, using port 1050 on myhost. It asks for the web page of "http://slashdot.org".
2. The proxy looks up the name "slashdot.org", and gets 207.218.152.131. It then opens a connection to that IP address (using port 1025 on the firewall's external interface), and asks the web server (port 80) for the web page.
3. As it receives the web page from its connection to the web server, it copies the data to the connection from Netscape.
4. Netscape renders the page.

ie. From slashdot.org's point of view, the connection is made from 1.2.3.4 (firewall's PPP interface) port 1025 to 207.218.152.131 (slashdot.org) port 80. From myhost's point of view, the connection is made from 192.168.1.100 (myhost) port 1050, to 192.168.1.1 (firewall's Ethernet interface) port 8080.

Private Network: Transparent Proxies

In this scenario, packets from the private network never traverse the Internet, and vice versa. The IP addresses of the private network should be assigned from the RFC1597 Private Network Allocations (ie. 10.*.*.*, 172.16.*.* or 192.168.*.*).

The only way things ever connect to the Internet is by connecting to the firewall, which is the only machine on both networks, which connects onwards. You run a program (on the firewall) called a

TOC

transparent proxy to do this; the kernel sends outgoing packets to the transparent proxy instead of sending them onwards (ie. it bastardizes routing).

Transparent proxying means that the clients don't need to know there is a proxy involved.

Any services you wish the Internet to access must be on the firewall. (But see Limited Internal Services below).

Example: Allowing web access from private network to the Internet.

1. The private network is assigned 192.168.1.* addresses, with myhost being 192.168.1.100, and the firewall's Ethernet interface being assigned 192.168.1.1.
2. A transparent web proxy (I believe there are patches for squid to allow it to operate in this manner, or try "transproxy") is installed and configured on the firewall, say running on port 8080.
3. The kernel is told to redirect connections to port 80 to the proxy, using ipchains.
4. Netscape on the private network is configured to connect directly.
5. DNS needs to be configured on the private network (ie. you need to run a DNS server as a proxy on the firewall).
6. The default route (aka gateway) needs to be configured on the private network, to send packets to the firewall.

Netscape on myhost reads http://slashdot.org.

1. Netscape looks up the name "slashdot.org", and gets 207.218.152.131. It then opens a connection to that IP address, using local port 1050, and asks the web server (port 80) for the web page.
2. As the packets from myhost (port 1050) to slashdot.org (port 80) pass through the firewall, they are redirected to the waiting transparent proxy on port 8080. The transparent proxy opens a connection (using local port 1025) to 207.218.152.131 port 80 (which is where the original packets were going).
3. As the proxy receives the web page from its connection to the web server, it copies the data to the connection from Netscape.
4. Netscape renders the page.

ie. From slashdot.org's point of view, the connection is made from 1.2.3.4 (firewall's PPP interface) port 1025 to 207.218.152.131 (slashdot.org) port 80. From myhost's point of view, the connection is made from 192.168.1.100 (myhost) port 1050, to 207.218.152.131 (slashdot.org) port 80, but it's actually talking to the transparent proxy.

Private Network: Masquerading

In this scenario, packets from the private network never traverse the Internet without special treatment, and vice versa. The IP addresses of the private network should be assigned from the RFC1597 Private Network Allocations (ie. 10.*.*.*, 172.16.*.* or 192.168.*.*).

Instead of using a proxy, we use a special kernel facility called "masquerading". Masquerading rewrites packets as they pass through the firewall, so that they always seem to come from the firewall itself. It then rewrites the responses so that they look like they are going to the original recipient.

TOC

Masquerading has separate modules to handle "tricky" protocols, such as FTP, RealAudio, Quake, et For really hard-to-handle protocols, the "auto forwarding" facility can handle some of them by automatically setting up port forwarding for related sets of ports: look for ``ipportfw" (2.0 kernels) or ``ipmasqadm" (2.1 kernels).

Any services you wish the Internet to access must be on the firewall. (But see Limited Internal Services below).

Example: Allowing web access from private network to the Internet.

1. The private network is assigned 192.168.1.* addresses, with myhost being 192.168.1.100, and the firewall's Ethernet interface being assigned 192.168.1.1.
2. The firewall is set up to masquerade any packets coming from the private network and going to port 80 on an Internet host.
3. Netscape is configured to connect directly.
4. DNS must be configured correctly on the private network.
5. The firewall should be the default route (aka gateway) for the private network.

Netscape on myhost reads http://slashdot.org.

1. Netscape looks up the name "slashdot.org", and gets 207.218.152.131. It then opens a connection to that IP address, using local port 1050, and asks the web server (port 80) for the web page.
2. As the packets from myhost (port 1050) to slashdot.org (port 80) pass through the firewall, they are rewritten to come from the PPP interface of the firewall, port 65000. The firewall has a valid Internet address (1.2.3.4) so reply packets from www.linuxhq.com get routed back OK.
3. As packets from slashdot.org (port 80) to firewall.littlecorp.com (port 65000) come in, they are rewritten to go to myhost, port 1050. This is the real magic of masquerading: it remembers when it rewrites outgoing packets to it can write them back as replies come in.
4. Netscape renders the page.

ie. From the slashdot.org's point of view, the connection is made from 1.2.3.4 (firewall's PPP interface) port 65000 to 207.218.152.131 (slashdot.org) port 80. From the myhost's point of view, the connection is made from 192.168.1.100 (myhost) port 1050, to 207.218.152.131 (slashdot.org) port 80.

Public Network

In this scenario, your personal network is a part of the Internet: packets can flow without change across both networks. The IP addresses of the internal network must be assigned by applying for a block of IP addresses, so the rest of the network will know how to get packets to you. This implies a permanent connection.

In this role, packet filtering is used to restrict which packets can be forwarded between your network and the rest of the Internet, eg. to restrict the rest of the Internet to only accessing your internal web servers.

Example: Allowing web access from private network to the Internet.

1. Your internal network is assigned according to the IP address block you have registered, (say 1.2.3.*).

2. The firewall is set up to allow all traffic.
3. Netscape is configured to connect directly.
4. DNS must be configured correctly on your network.
5. The firewall should be the default route (aka gateway) for the private network.

Netscape on myhost reads http://slashdot.org.

1. Netscape looks up the name "slashdot.org", and gets 207.218.152.131. It then opens a connection to that IP address, using local port 1050, and asks the web server (port 80) for the web page.
2. Packets pass through your firewall, just as they pass through several other routers between you and slashdot.org.
3. Netscape renders the page.

ie. There is only one connection: from 1.2.3.100 (myhost) port 1050, to 207.218.152.131 (slashdot.org) port 80.

Limited Internal Services

There are a few tricks you can pull to allow the Internet to access your internal services, rather than running the services on the firewall. These will work with either a proxy or masquerading based approach for external connections.

The simplest approach is to run a "redirector", which is a poor-man's proxy which waits for a connection on a given port, and then open a connection a fixed internal host and port, and copies data between the two connections. An example of this is the "redir" program. From the Internet point of view, the connection is made to your firewall. From your internal server's point of view, the connection is made from the internal interface of the firewall to the server.

Another approach (which requires a 2.0 kernel patched for ipportfw, or a 2.1 or later kernel) is to use port forwarding in the kernel. This does the same job as "redir" in a different way: the kernel rewrites packets as they pass through, changing their destination address and ports to point them at an internal host and port. From the Internet's point of view, the connection is made to your firewall. From your internal server's point of view, a direct connection is made from the Internet host to the server.

http://www.ecst.csuchico.edu/~dranch/LINUX/index-LINUX.html#ipmasq

Soon I would expect it to be found under the auspices of the Linux Documentation Project, at http://www.metalab.unc.edu/LDP

The official Masquerading home page is at

http://ipmasq.cjb.net

---

4. IP Firewalling Chains

This section describes all you really need to know to build a packet filter that meets your needs.

4.1 How Packets Traverse The Filters

The kernel starts with three lists of rules; these lists are called **firewall chains** or just **chains**. The three chains are called **input**, **output** and **forward**. When a packet comes in (say, through the Ethernet card) the kernel uses the `input` chain to decide its fate. If it survives that step, then the kernel decides where to send the packet next (this is called **routing**). If it is destined for another machine, it consults the `forward` chain. Finally, just before a packet is to go out, the kernel consults the `output` chain.

A chain is a checklist of **rules**. Each rule says `if the packet header looks like this, then here's what to do with the packet'. If the rule doesn't match the packet, then the next rule in the chain is consulted. Finally, if there are no more rules to consult, then the kernel looks at the chain **policy** to decide what to do. In a security-conscious system, this policy usually tells the kernel to reject or deny the packet.

For ASCII-art fans, this shown the complete path of a packet coming into a machine.

```
        ----------------------------------------------------------------
        |            ACCEPT/                                lo interface |
        v           REDIRECT                     _______                 |
--> C --> S --> ______ --> D --> ~~~~~~~~ -->|forward|----> _______ -->
    h     a    |input |    e    {Routing }   |Chain  |     |output |ACCEPT
    e     n    |Chain |    m    {Decision}   |_______| --->|Chain  |
    c     i    |______|    a     ~~~~~~~~        |     | ->|_______|
    k     t       |        s       |             |     | |     |
    s     y       |        q       |             v     | |     |
    u     |       v        e       v            DENY/  | |     v
    m     |     DENY/      r   Local Process   REJECT  | |   DENY/
    |     v    REJECT      a       |                   | |  REJECT
    |   DENY               d       --------------------- |
    v                      e -----------------------------
   DENY
```

Here is a blow-by-blow description of each stage:

Checksum:

This is a test that the packet hasn't been corrupted in some way. If it has, it is denied.

Sanity:

There is actually one of these sanity checks before each firewall chain, but the input chain's is the most important. Some malformed packets might confuse the rule-checking code, and these are denied here (a message is printed to the syslog if this happens).

input chain:

This is the first firewall chain against which the packet will be tested. If the verdict of the chain is not `DENY` or `REJECT`, the packet continues on.

Demasquerade:

If the packet is a reply to a previously masqueraded packet, it is demasqueraded, and skips straight to the `output` chain. If you don't use IP Masquerading, you can mentally erase this from the diagram.

Routing decision:

The destination field is examined by the routing code, to decide if this packet should go to a local process (see Local process below) or forwarded to a remote machine (see forward chain below).

Local process:

A process running on the machine can receive packets after the Routing Decision step, and can send packets (which go through the Routing Decision step, then traverse the output chain).

lo interface:

If packets from a local process are destined for a local process, they will go through the output chain with interface set to `lo', then return through the input chain with interface also `lo'. The lo interface is usually called the loopback interface.

local:

If the packet was not created by a local process, then the forward chain is checked, otherwise the packet goes to the output chain.

forward chain:

This chain is traversed for any packets which are attempting to pass through this machine to another.

output chain:

This chain is traversed for all packets just before they are sent out.

Using ipchains

First, check that you have the version of ipchains that this document refers to:

```
$ ipchains --version
ipchains 1.3.9, 17-Mar-1999
```

Note that I recommend 1.3.4 (which has no long options, like `--sport'), or 1.3.8 or above; these are very stable.

ipchains has a fairly detailed manual page (`man ipchains`), and if you need more detail on particulars, you can check out the programming interface (`man 4 ipfw`), or the file `net/ipv4/ip_fw.c` in the 2.1.x kernel source, which is (obviously) authoritative.

There is also an excellent quick reference card by Scott Bronson in the source package, in both A4 and US Letter PostScript(TM).

There are several different things you can do with `ipchains`. First the operations to manage whole chains. You start with three built-in chains `input`, `output` and `forward` which you can't delete.

1. Create a new chain (-N).
2. Delete an empty chain (-X).
3. Change the policy for a built-in chain. (-P).
4. List the rules in a chain (-L).
5. Flush the rules out of a chain (-F).
6. Zero the packet and byte counters on all rules in a chain (-Z).

There are several ways to manipulate rules inside a chain:

1. Append a new rule to a chain (-A).
2. Insert a new rule at some position in a chain (-I).
3. Replace a rule at some position in a chain (-R).
4. Delete a rule at some position in a chain (-D).
5. Delete the first rule that matches in a chain (-D).

There are a few operations for masquerading, which are in `ipchains` for want of a good place to put them:

1. List the currently masqueraded connections (-M -L).
2. Set masquerading timeout values (-M -S). (But see I can't set masquerading timeouts!).

The final (and perhaps the most useful) function allows you to check what would happen to a given packet if it were to traverse a given chain.

Operations on a Single Rule

This is the bread-and-butter of ipchains; manipulating rules. Most commonly, you will probably use the append (-A) and delete (-D) commands. The others (-I for insert and -R for replace) are simple extensions of these concepts.

Each rule specifies a set of conditions the packet must meet, and what to do if it meets them (a `target'). For example, you might want to deny all ICMP packets coming from the IP address 127.0.0.1. So in this case our conditions are that the protocol must be ICMP and that the source address must be 127.0.0.1. Our target is `DENY'.

127.0.0.1 is the `loopback' interface, which you will have even if you have no real network connection. You can use the `ping' program to generate such packets (it simply sends an ICMP type 8 (echo request) which all cooperative hosts should obligingly respond to with an ICMP type 0 (echo reply) packet). This makes it useful for testing.

```
# ping -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.2 ms

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
# ipchains -A input -s 127.0.0.1 -p icmp -j DENY
# ping -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 0 packets received, 100% packet loss
#
```

You can see here that the first ping succeeds (the `-c 1' tells ping to only send a single packet).

Then we append (-A) to the `input' chain, a rule specifying that for packets from 127.0.0.1 (`-s 127.0.0.1') with protocol ICMP (`-p ICMP') we should jump to DENY (`-j DENY').

Then we test our rule, using the second ping. There will be a pause before the program gives up waiting for a response that will never come.

We can delete the rule in one of two ways. Firstly, since we know that it is the only rule in the input chain, we can use a numbered delete, as in:

```
# ipchains -D input 1
#
```

To delete rule number 1 in the input chain.

The second way is to mirror the -A command, but replacing the -A with -D. This is useful when you have a complex chain of rules and you don't want to have to count them to figure out that it's rule 37 that you want to get rid of. In this case, we would use:

```
# ipchains -D input -s 127.0.0.1 -p icmp -j DENY
#
```

The syntax of -D must have exactly the same options as the -A (or -I or -R) command. If there are multiple identical rules in the same chain, only the first will be deleted.

Filtering Specifications

We have seen the use of `-p' to specify protocol, and `-s' to specify source address, but there are other options we can use to specify packet characteristics. What follows is an exhaustive compendium.

Specifying Source and Destination IP Addresses

Source (-s) and destination (-d) IP addresses can be specified in four ways. The most common way is to use the full name, such as `localhost' or `www.linuxhq.com'. The second way is to specify the IP address such as `127.0.0.1'.

The third and fourth ways allow specification of a group of IP addresses, such as `199.95.207.0/24' or `199.95.207.0/255.255.255.0'. These both specify any IP address from 192.95.207.0 to 192.95.207.255 inclusive; the digits after the `/' tell which parts of the IP address are significant. `/32' or `/255.255.255.255' is the default (match all of the IP address). To specify any IP address at all `/0' can be used, like so:

```
# ipchains -A input -s 0/0 -j DENY
#
```

This is rarely used, as the effect above is the same as not specifying the `-s' option at all.

Specifying Inversion

Many flags, including the `-s' and `-d' flags can have their arguments preceded by `!' (pronounced `not') to match addresses NOT equal to the ones given. For example. `-s ! localhost' matches any packet not coming from localhost.

Specifying Protocol

The protocol can be specified with the `-p' flag. Protocol can be a number (if you know the numeric protocol values for IP) or a name for the special cases of `TCP', `UDP' or `ICMP'. Case doesn't matter, so `tcp' works as well as `TCP'.

The protocol name can be prefixed by a `!', to invert it, such as `-p ! TCP'.

Specifying UDP and TCP Ports

For the special case where a protocol of TCP or UDP is specified, there can be an extra argument indicating the TCP or UDP port, or an (inclusive) range of ports (but see Handling Fragments below). A range is represented using a `:' character, such as `6000:6010', which covers 11 port numbers, from 6000 to 6010 inclusive. If the lower bound is omitted, it defaults to 0. If the upper bound is omitted, it defaults to 65535. So to specify TCP connections coming from ports under 1024, the syntax would be as `-p TCP -s 0.0.0.0/0 :1023'. Port numbers can be specified by name, eg. `www'.

Note that the port specification can be preceded by a `!', which inverts it. So to specify every TCP packet BUT a WWW packet, you would specify

```
-p TCP -d 0.0.0.0/0 ! www
```

It is important to realize that the specification

```
-p TCP -d ! 192.168.1.1 www
```

is very different from

```
-p TCP -d 192.168.1.1 ! www
```

The first specifies any TCP packet to the WWW port on any machine but 192.168.1.1. The second specifies any TCP connection to any port on 192.168.1.1 but the WWW port.

Finally, this case means not the WWW port and not 192.168.1.1:

```
-p TCP -d ! 192.168.1.1 ! www
```

TOC

Specifying ICMP Type and Code

ICMP also allows an optional argument, but as ICMP doesn't have ports, (ICMP has a **type** and a **code**) they have a different meaning.

You can specify them as ICMP names (use `ipchains -h icmp` to list the names) after the `-s' option, or as a numeric ICMP type and code, where the type follows the `-s' option and the code follows the `-d' option.

The ICMP names are fairly long: you only need use enough letters to make the name distinct from any other.

Here is a small table of some of the most common ICMP packets:

```
Number  Name                      Required by

0       echo-reply                ping
3       destination-unreachable   Any TCP/UDP traffic.
5       redirect                  routing if not running routing daemon
8       echo-request              ping
11      time-exceeded             traceroute
```

Note that the ICMP names cannot be preceeded by `!' at the moment.

DO NOT DO NOT DO NOT block all ICMP type 3 messages! (See ICMP Packets below).

Specifying an Interface

The `-i' option specifies the name of an **interface** to match. An interface is the physical device the packet came in on, or is going out on. You can use the `ifconfig` command to list the interfaces which are `up' (ie. working at the moment).

The interface for incoming packets (ie. packets traversing the `input` chain) is considered to be the interface they came in on. Logically, the interface for outgoing packets (packets traversing the `output` chain) is the interface they will go out on. The interface for packets traversing the `forward` chain is also the interface they will go out on; a fairly arbitrary decision it seems to me.

It is perfectly legal to specify an interface that currently does not exist; the rule will not match anything until the interface comes up. This is extremely useful for dial-up PPP links (usually interface `ppp0`) and the like.

As a special case, an interface name ending with a `+' will match all interfaces (whether they currently exist or not) which begin with that string. For example, to specify a rule which matches all PPP interfaces, the `-i ppp+` option would be used.

The interface name can be preceded by a `!' to match a packet which does NOT match the specified interface(s).

Specifying TCP SYN Packets Only

It is sometimes useful to allow TCP connections in one direction, but not the other. For example, you might want to allow connections to an external WWW server, but not connections from that server.

The naive approach would be to block TCP packets coming from the server. Unfortunately, TCP connections require packets going in both directions to work at all.

The solution is to block only the packets used to request a connection. These packets are called **SYN** packets (ok, technically they're packets with the SYN flag set, and the FIN and ACK flags cleared, but we call them SYN packets). By disallowing only these packets, we can stop attempted connections in their tracks.

The `-y' flag is used for this: it is only valid for rules which specify TCP as their protocol. For example, to specify TCP connection attempts from 192.168.1.1:

```
-p TCP -s 192.168.1.1 -y
```

Once again, this flag can be inverted by preceding it with a `!', which means every packet other than the connection initiation.

Handling Fragments

Sometimes a packet is too large to fit down a wire all at once. When this happens, the packet is divided into **fragments**, and sent as multiple packets. The other end reassembles the fragments to reconstruct the whole packet.

The problem with fragments is that some of the specifications listed above (in particular, source port, destinations port, ICMP type, ICMP code, or TCP SYN flag) require the kernel to peek at the start of the packet, which is only contained in the first fragment.

If your machine is the only connection to an external network, then you can tell the Linux kernel to reassemble all fragments which pass through it, by compiling the kernel with `IP: always defragment` set to `Y'. This sidesteps the issue neatly.

Otherwise, it is important to understand how fragments get treated by the filtering rules. Any filtering rule that asks for information we don't have will *not* match. This means that the first fragment is treated like any other packet. Second and further fragments won't be. Thus a rule `-p TCP -s 192.168.1.1 www` (specifying a source port of `www') will never match a fragment (other than the first fragment). Neither will the opposite rule `-p TCP -s 192.168.1.1 ! www`.

However, you can specify a rule specifically for second and further fragments, using the `-f' flag. Obviously, it is illegal to specify a TCP or UDP port, ICMP type, ICMP code or TCP SYN flag in such a fragment rule.

It is also legal to specify that a rule does *not* apply to second and further fragments, by preceding the `-f' with `!'.

Usually it is regarded as safe to let second and further fragments through, since filtering will effect the first fragment, and thus prevent reassembly on the target host, however, bugs have been known to allow crashing of machines simply by sending fragments. Your call.

Note for network-heads: malformed packets (TCP, UDP and ICMP packets too short for the firewalling code to read the ports or ICMP code and type) are treated as fragments as well. Only TCP fragments starting at position 8 are explicitly dropped by the firewall code (a message should appear in the syslog if this occurs).

As an example, the following rule will drop any fragments going to 192.168.1.1:

```
# ipchains -A output -f -d 192.168.1.1 -j DENY
#
```

Filtering Side Effects

OK, so now we know all the ways we can match a packet using a rule. If a packet matches a rule, the following things happen:

1. The byte counter for that rule is increased by the size of the packet (header and all).
2. The packet counter for that rule is incremented.
3. If the rule requests it, the packet is logged.
4. If the rule requests it, the packet's Type Of Service field is changed.
5. If the rule requests it, the packet is marked (not in 2.0 kernel series).
6. The rule target is examined to decide what to do to the packet next.

For variety, I'll address these in order of importance.

Specifying a Target

A **target** tells the kernel what to do with a packet that matches a rule. ipchains uses `-j' (think `jump-to') for the target specification. The target name must be less than 8 letters, and case matters: "RETURN" and "return" are completely different.

The simplest case is when there is no target specified. This type of rule (often called an `accounting' rule) is useful for simply counting a certain type of packet. Whether this rule matches or not, the kernel simply examines the next rule in the chain. For example, to count the number of packets from 192.168.1.1, we could do this:

```
# ipchains -A input -s 192.168.1.1
#
```

(Using `ipchains -L -v' we can see the byte and packet counters associated with each rule).

TOC

There are six special targets. The first three, `ACCEPT`, `REJECT` and `DENY` are fairly simple. `ACCEPT` allows the packet through. `DENY` drops the packet as if it had never been received. `REJECT` drops the packet, but (if it's not an ICMP packet) generates an ICMP reply to the source to tell it that the destination was unreachable.

The next one, `MASQ` tells the kernel to masquerade the packet. For this to work, your kernel needs to be compiled with IP Masquerading enabled. For details on this, see the Masquerading-HOWTO and the Appendix Differences between ipchains and ipfwadm. This target is only valid for packets traversing the `forward` chain.

The other major special target is `REDIRECT` which tells the kernel to send a packet to a local port instead of wherever it was heading. This can only be specified for rules specifying TCP or UDP as their protocol. Optionally, a port (name or number) can be specified following `-j REDIRECT' which will cause the packet to be redirected to that particular port, even if it was addressed to another port. This target is only valid for packets traversing the `input` chain.

The final special target is `RETURN` which is identical to falling off the end of the chain immediately. (See Setting Policy below).

Any other target indicates a user-defined chain (as described in Operations on an Entire Chain below). The packet will begin traversing the rules in that chain. If that chain doesn't decide the fate of the packet, then once traversal on that chain has finished, traversal resumes on the next rule in the current chain.

Time for more ASCII art. Consider two (silly) chains: `input` (the built-in chain) and `Test` (a user-defined chain).

```
 `input'                          `Test'
----------------------------    ----------------------------
| Rule1: -p ICMP -j REJECT |    | Rule1: -s 192.168.1.1    |
|--------------------------|    |--------------------------|
| Rule2: -p TCP -j Test    |    | Rule2: -d 192.168.1.1    |
|--------------------------|    ----------------------------
| Rule3: -p UDP -j DENY    |
----------------------------
```

Consider a TCP packet coming from 192.168.1.1, going to 1.2.3.4. It enters the `input` chain, and gets tested against Rule1 - no match. Rule2 matches, and its target is `Test`, so the next rule examined is the start of `Test`. Rule1 in `Test` matches, but doesn't specify a target, so the next rule is examined, Rule2. This doesn't match, so we have reached the end of the chain. We return to the `input` chain, where we had just examined Rule2, so we now examine Rule3, which doesn't match either.

So the packet path is:

```
                          v    _________________________
 `input'                  |   /    `Test'                v
------------------------|--/    ------------------------|----
| Rule1                  | /|    | Rule1                  |   |
|------------------------|/-|    |------------------------|---|
| Rule2                  /  |    | Rule2                  |   |
|--------------------------|    ------------------------v----
| Rule3                  /--+_________________________/
------------------------|---
                          v
```

See the section How to Organise Your Firewall Rules for ways to use user-defined chains effectively.

Logging Packets

This is a side effect that matching a rule can have; you can have the matching packet logged using the `-l' flag. You will usually not want this for routine packets, but it is a useful feature if you want to look for exceptional events.

The kernel logs this information looking like:

```
Packet log: input DENY eth0 PROTO=17 192.168.2.1:53 192.168.1.1:1025
  L=34 S=0x00 I=18 F=0x0000 T=254
```

TOC

This log message is designed to be terse, and contain technical information useful only to networking gurus, but it can be useful to the rest of us. It breaks down like so:

1. `input' is the chain which contained the rule which matched the packet, causing the log message.
2. `DENY' is what the rule said to do to the packet. If this is `-' then the rule didn't effect the packet at all (an accounting rule).
3. `eth0' is the interface name. Because this was the input chain, it means that the packet came in `eth0'.
4. `PROTO=17' means that the packet was protocol 17. A list of protocol numbers is given in `/etc/protocols'. The most common are 1 (ICMP), 6 (TCP) and 17 (UDP).
5. `192.168.2.1' means that the packet's source IP address was 192.168.2.1.
6. `:53' means that the source port was port 53. Looking in `/etc/services' shows that this is the `domain' port (ie. this is probably an DNS reply). For UDP and TCP, this number is the source port. For ICMP, it's the ICMP type. For others, it will be 65535.
7. `192.168.1.1' is the destination IP address.
8. `:1025' means that the destination port was 1025. For UDP and TCP, this number is the destination port. For ICMP, it's the ICMP code. For others, it will be 65535.
9. `L=34' means that packet was a total of 34 bytes long.
10. `S=0x00' means the Type of Service field (divide by 4 to get the Type of Service as used by ipchains).
11. `I=18' is the IP ID.
12. `F=0x0000' is the 16-bit fragment offset plus flags. A value starting with `0x4' or `0x5' means that the Don't Fragment bit is set. `0x2' or `0x3' means the `More Fragments' bit is set; expect more fragments after this. The rest of the number is the offset of this fragment, divided by 8.
13. `T=254' is the Time To Live of the packet. One is subtracted from this value for every hop, and it usually starts at 15 or 255.
14. `(#5)' there may be a final number in brackets on more recent kernels (perhaps after 2.2.9). This is the rule number which caused the packet log.

On standard Linux systems, this kernel output is captured by klogd (the kernel logging daemon) which hands it to syslogd (the system logging daemon). The `/etc/syslog.conf' controls the behaviour of syslogd, by specifying a destination for each `facility' (in our case, the facility is "kernel") and `level' (for ipchains, the level used is "info").

For example, my (Debian) /etc/syslog.conf contains two lines which match `kern.info':

```
kern.*                          -/var/log/kern.log
*.=info;*.=notice;*.=warn;\
        auth,authpriv.none;\
        cron,daemon.none;\
        mail,news.none          -/var/log/messages
```

These mean that the messags are duplicated in `/var/log/kern.log' and `/var/log/messages'. For more details, see `man syslog.conf'.

Manipulating the Type Of Service

There are four seldom-used bits in the IP header, called the **Type of Service** (TOS) bits. They effect the way packets are treated; the four bits are "Minimum Delay", "Maximum Throughput", "Maximum Reliability" and "Minimum Cost". Only one of these bits is allowed to be set. Rob van Nieuwkerk, the author of the TOS-mangling code, puts it as follows:

> Especially the "Minimum Delay" is important for me. I switch it on for "interactive" packets in my upstream (Linux) router. I'm behind a 33k6 modem link. Linux prioritizes packets in 3 queues. This way I get acceptable interactive performance while doing bulk downloads at the same time. (It could even be better if there wasn't such a big queue in the serial driver, but latency is kept down 1.5 seconds now).

Note: obviously, you have no control over incoming packets; you can only control the priority of packets leaving your box. To negotiate priorities with the other end, a protocol like RSVP (which I know nothing about, so don't ask me) must be used.

The most common use is to set telnet & ftp control connections to "Minimum Delay" and FTP data to "Maximum Throughput". This would be done as follows:

```
ipchains -A output -p tcp -d 0.0.0.0/0 telnet -t 0x01 0x10
ipchains -A output -p tcp -d 0.0.0.0/0 ftp -t 0x01 0x10
```

```
ipchains -A output -p tcp -s 0.0.0.0/0 ftp-data -t 0x01 0x08
```

The `-t' flag takes two extra parameters, both in hexadecimal. These allow complex twiddling of the TOS bits: the first mask is ANDed with the packet's current TOS, and then the second mask is XORed with it. If this is too confusing, just use the following table:

TOS Name	Value	Typical Uses
Minimum Delay	0x01 0x10	ftp, telnet
Maximum Throughput	0x01 0x08	ftp-data
Maximum Reliability	0x01 0x04	snmp
Minimum Cost	0x01 0x02	nntp

Andi Kleen goes on to point out the following (mildly edited for posterity):

> Maybe it would be useful to add an reference to the txqueuelen parameter of ifconfig to the discussion of TOS bits. The default device queue length is tuned for ethernet cards, on modems it is too long and makes the 3 band scheduler (which queues based on TOS) work suboptimally. It is a good idea to set it to a value between 4-10 on modem or single b channel ISDN links: on bundled devices a longer queue is needed. This is a 2.0 and 2.1 problem, but in 2.1 it is a ifconfig flag (with recent nettools), while in 2.0 it requires source patches in the device drivers to change.

So, to see maximal benifits of TOS manipulation for modem PPP links, do `ifconfig $1 txqueuelen' in your /etc/ppp/ip-up script. The number to use depends on the modem speed and the amount of buffering in the modem; here's Andi setting me straight again:

> The best value for a given configuration needs experiment. If the queues are too short on a router then packets will get dropped. Also of course one gets benefits even without TOS rewriting, just that TOS rewriting helps to give the benefits to non cooperating programs (but all standard linux programs are cooperating).

Marking a Packet

This allows complex and powerful interactions with Alexey Kuznetsov's new Quality of Service implementation, as well as the mark-based forwarding in later 2.1 series kernels. More news as it comes to hand. This option is ignored altogether in the 2.0 kernel series.

Operations on an Entire Chain

A very useful feature of ipchains is the ability to group related rules into chains. You can call the chains whatever you want, as long as the names don't clash with the built-in chains (`input`, `output` and `forward`) or the targets (`MASQ`, `REDIRECT`, `ACCEPT`, `DENY`, `REJECT` or `RETURN`). I suggest avoiding upper-case labels entirely, since I may use these for future extensions. The chain name can be up to 8 characters long.

Creating a New Chain

Let's create a new chain. Because I am such an imaginative fellow, I'll call it `test`.

```
# ipchains -N test
#
```

It's that simple. Now you can put rules in it as detailed above.

Deleting a Chain

Deleting a chain is simple as well.

```
# ipchains -X test
#
```

Why `-X'? Well, all the good letters were taken.

There are a couple of restrictions to deleting chains: they must be empty (see Flushing a Chain below) and they must not be the target of any rule. You can't delete any of the three built-in chains.

TOC

Flushing a Chain

There is a simple way of emptying all rules out of a chain, using the `-F' command.

```
# ipchains -F forward
#
```

If you don't specify a chain, then *all* chains will be flushed.

Listing a Chain

You can list all the rules in a chain by using the `-L' command.

```
# ipchains -L input
Chain input (refcnt = 1): (policy ACCEPT)
target     prot opt    source               destination          ports
ACCEPT     icmp -----  anywhere             anywhere             any
# ipchains -L test
Chain test (refcnt = 0):
target     prot opt    source               destination          ports
DENY       icmp -----  localnet/24          anywhere             any
#
```

The `refcnt' listed for `test` is the number of rules which have `test` as their target. This must be zero (and the chain be empty) before this chain can be deleted.

If the chain name is omitted, all chains are listed, even empty ones.

There are three options which can accompany `-L'. The `-n' (numeric) option is very useful as it prevents `ipchains` from trying to lookup the IP addresses, which (if you are using DNS like most people) will cause large delays if your DNS is not set up properly, or you have filtered out DNS requests. It also causes ports to be printed out as numbers rather than names.

The `-v' options shows you all the details of the rules, such as the the packet and byte counters, the TOS masks, the interface, and the packet mark. Otherwise these values are omitted. For example:

```
# ipchains -v -L input
Chain input (refcnt = 1): (policy ACCEPT)
 pkts bytes target     prot opt   tosa tosx  ifname    mark        source
destination           ports
   10   840 ACCEPT     icmp ----- 0xFF 0x00  lo                    anywhere
anywhere              any
```

Note that the packet and byte counters are printed out using the suffixes `K', `M' or `G' for 1000, 1,000,000 and 1,000,000,000 respectively. Using the `-x' (expand numbers) flag as well prints the full numbers, no matter how large they are.

Resetting (Zeroing) Counters

It is useful to be able to reset the counters. This can be done with the `-Z' (zero counters) option. For example:

```
# ipchains -v -L input
Chain input (refcnt = 1): (policy ACCEPT)
 pkts bytes target     prot opt   tosa tosx  ifname    mark        source
destination           ports
   10   840 ACCEPT     icmp ----- 0xFF 0x00  lo                    anywhere
anywhere              any
# ipchains -Z input
# ipchains -v -L input
Chain input (refcnt = 1): (policy ACCEPT)
```

```
 pkts bytes target     prot opt   tosa tosx  ifname    mark        source
destination           ports
    0     0 ACCEPT     icmp ----- 0xFF 0x00  lo                    anywhere
anywhere              any
#
```

The problem with this approach is that sometimes you need to know the counter values immediately before they are reset. In the above example, some packets could pass through between the `-L' and `-Z' commands. For this reason, you can use the `-L' and `-Z' *together*, to reset the counters while reading them. Unfortunately, if you do this, you can't operate on a single chain: you have to list and zero all the chains at once.

```
# ipchains -L -v -Z
Chain input (policy ACCEPT):
 pkts bytes target     prot opt   tosa tosx  ifname    mark        source
destination           ports
   10   840 ACCEPT     icmp ----- 0xFF 0x00  lo                    anywhere
anywhere              any

Chain forward (refcnt = 1): (policy ACCEPT)
Chain output (refcnt = 1): (policy ACCEPT)
Chain test (refcnt = 0):
    0     0 DENY       icmp ----- 0xFF 0x00  ppp0                  localnet/24
anywhere              any
# ipchains -L -v
Chain input (policy ACCEPT):
 pkts bytes target     prot opt   tosa tosx  ifname    mark        source
destination           ports
   10   840 ACCEPT     icmp ----- 0xFF 0x00  lo                    anywhere
anywhere              any

Chain forward (refcnt = 1): (policy ACCEPT)
Chain output (refcnt = 1): (policy ACCEPT)
Chain test (refcnt = 0):
    0     0 DENY       icmp ----- 0xFF 0x00  ppp0                  localnet/24
anywhere              any
#
```

Setting Policy

We glossed over what happens when a packet hits the end of a built-in chain when we discussed how a packet walks through chains in Specifying a Target above. In this case, the **policy** of the chain determines the fate of the packet. Only built-in chains (`input`, `output` and `forward`) have policies, because if a packet falls off the end of a user-defined chain, traversal resumes at the previous chain.

The policy can be any of the first four special targets: `ACCEPT`, `DENY`, `REJECT` or `MASQ`. `MASQ` is only valid for the `forward' chain.

It is also important to note that a `RETURN` target in a rule in one of the built-in chains is useful to explicitly target the chain policy when a packet matches a rule.

Operations on Masquerading

There are several parameters you can tweak for IP Masquerading. They are bundled with `ipchains` because it's not worth writing a separate tool for them (although this will change).

The IP Masquerading command is `-M', and it can be combined with `-L' to list currently masqueraded connections, or `-S' to set the masquerading parameters.

The `-L' command can be accompanied by `-n' (show numbers instead of hostnames and port names) or `-v' (show deltas in sequence numbers for masqueraded connection, just in case you care).

TOC

The `-S' command should be followed by three timeout values, each in seconds: for TCP sessions, for TCP sessions after a FIN packet, and for UDP packets. If you don't want to change one of these values, simply give a value of `0'.

The default values are listed in `/usr/src/linux/include/net/ip_masq.h', currently 15 minutes, 2 minutes and 5 minutes respectively.

The most common value to change is the first one, for FTP (see FTP Nightmares below).

Note the problems with setting timeouts listed in I can't set masquerading timeouts!.

Checking a Packet

Sometimes you want to see what happens when a certain packet enters your machine, such as for debugging your firewall chains. `ipchains` has the `-C' command to allow this, using the exact same routines that the kernel uses to diagnose real packets.

You specify which chain to test the packet on by following the `-C' argument with its name. Whereas the kernel always starts traversing on the `input`, `output` or `forward` chains, you are allowed to begin traversing on any chain for testing purposes.

The details of the `packet' are specified using the same syntax used to specify firewall rules. In particular, a protocol (`-p'), source address (`-s'), destination address (`-d') and interface (`-i') are compulsory. If the protocol is TCP or UDP, then a single source and a single destination port must be specified, and a ICMP type and code must be specified for the ICMP protocol (unless the `-f' flag is specified to indicate a fragment rule, in which case these options are illegal).

If the protocol is TCP (and the `-f' flag is not specified), the `-y' flag may be specified, to indicate that the test packet should have the SYN bit set.

Here is an example of testing a TCP SYN packet from 192.168.1.1 port 60000 to 192.168.1.2 port www, coming in the eth0 interface, entering the `input' chain. (This is a classic incoming WWW connection initiation):

```
# ipchains -C input -p tcp -y -i eth0 -s 192.168.1.1 60000 -d 192.168.1.2 www
packet accepted
#
```

Multiple Rules at Once and Watching What Happens

Sometimes a single command line can result in multiple rules being effected. This is done in two ways. Firstly, if you specify a hostname which resolves (using DNS) to multiple IP addresses, `ipchains` will act as if you had typed multiple commands with each combination of addresses.

So if the hostname `www.foo.com' resolves to three IP addresses, and the hostname `www.bar.com' resolves to two IP addresses, then the command `ipchains -A input -j reject -s www.bar.com -d www.foo.com' would append six rules to the `input` chain.

The other way to have `ipchains` perform multiple actions is to use the bidirectional flag (`-b'). This flag makes `ipchains` behave as if you had typed the command twice, the second time with the `-s' and `-d' arguments reversed. So, to avoid forwarding either to or from 192.168.1.1, you could do the following:

```
# ipchains -b -A forward -j reject -s 192.168.1.1
#
```

Personally, I don't like the `-b' option much; if you want convenience, see Using ipchains-save below.

The -b option can be used with the insert (`-I'), delete (`-D') (but not the variation which takes a rule number), append (`-A') and check (`-C') commands.

Another useful flag is `-v' (verbose) which prints out exactly what `ipchains` is doing with your commands. This is useful if you are dealing with commands that may effect multiple rules. For example, here we check the behaviour of fragments between 192.168.1.1 and 192.168.1.2.

```
# ipchains -v -b -C input -p tcp -f -s 192.168.1.1 -d 192.168.1.2 -i lo
  tcp opt   ---f- tos 0xFF 0x00  via lo    192.168.1.1  -> 192.168.1.2    * ->   *
packet accepted
  tcp opt   ---f- tos 0xFF 0x00  via lo    192.168.1.2  -> 192.168.1.1    * ->   *
```

```
packet accepted
#
```

TOC

4.2 Useful Examples

I have a dialup PPP connection (`-i ppp0`). I grab news (`-p TCP -s news.virtual.net.au nntp`) and mail (`-p TCP -s mail.virtual.net.au pop-3`) every time I dial up. I use Debian's FTP method to update my machine regularly (`-p TCP -y -s ftp.debian.org.au ftp-data`). I surf the web through my ISP's proxy while this is going on (`-p TCP -d proxy.virtual.net.au 8080`), but hate the ads from doubleclick.net on the Dilbert Archive (`-p TCP -y -d 199.95.207.0/24` and `-p TCP -y -d 199.95.208.0/24`).

I don't mind people trying to ftp to my machine while I'm online (`-p TCP -d $LOCALIP ftp`), but don't want anyone outside pretending to have an IP address of my internal network (`-s 192.168.1.0/24`). This is commonly called IP spoofing, and there is a better way to protect yourself from it in the 2.1.x kernels and above: see How do I set up IP spoof protection?.

This setup is fairly simple, because there are currently no other boxes on my internal network.

I don't want any local process (ie. Netscape, lynx etc.) to connect to doubleclick.net:

```
# ipchains -A output -d 199.95.207.0/24 -j REJECT
# ipchains -A output -d 199.95.208.0/24 -j REJECT
#
```

Now I want to set priorities on various outgoing packets (there isn't much point in doing it on incoming packets). Since I have a fair number of these rules, it makes sense to put them all in a single chain, called `ppp-out`.

```
# ipchains -N ppp-out
# ipchains -A output -i ppp0 -j ppp-out
#
```

Minimum delay for web traffic & telnet.

```
# ipchains -A ppp-out -p TCP -d proxy.virtual.net.au 8080 -t 0x01 0x10
# ipchains -A ppp-out -p TCP -d 0.0.0.0 telnet -t 0x01 0x10
#
```

Low cost for ftp data, nntp, pop-3:

```
# ipchains -A ppp-out -p TCP -d 0.0.0.0/0 ftp-data -t 0x01 0x02
# ipchains -A ppp-out -p TCP -d 0.0.0.0/0 nntp -t 0x01 0x02
# ipchains -A ppp-out -p TCP -d 0.0.0.0/0 pop-3 -t 0x01 0x02
#
```

There are a few restrictions on packets coming in the ppp0 interface: let's create a chain called `ppp-in':

```
# ipchains -N ppp-in
# ipchains -A input -i ppp0 -j ppp-in
#
```

Now, no packets coming in `ppp0` should be claiming a source address of 192.168.1.*, so we log and deny them:

```
# ipchains -A ppp-in -s 192.168.1.0/24 -l -j DENY
#
```

I allow UDP packets in for DNS (I run a caching nameserver which forwards all requests to 203.29.16.1, so I expect DNS replies from them only), incoming ftp, and return ftp-data only (which should only be going to a port above 1023, and not the X11 ports around 6000).

```
# ipchains -A ppp-in -p UDP -s 203.29.16.1 -d $LOCALIP dns -j ACCEPT
```

```
# ipchains -A ppp-in -p TCP -s 0.0.0.0/0 ftp-data -d $LOCALIP 1024:5999 -j ACCEPT
# ipchains -A ppp-in -p TCP -s 0.0.0.0/0 ftp-data -d $LOCALIP 6010: -j ACCEPT
# ipchains -A ppp-in -p TCP -d $LOCALIP ftp -j ACCEPT
#
```

Finally, local-to-local packets are OK:

```
# ipchains -A input -i lo -j ACCEPT
#
```

Now, my default policy on the `input` chain is `DENY`, so everything else gets dropped:

```
# ipchains -P input DENY
#
```

NOTE: I wouldn't set up my chains in this order, as packets might get through while I'm setting up. Safest is usually to set the policy to DENY first, then insert the rules. Of course, if your rules require DNS lookups to resolve hostnames, you could be in trouble.

Using ipchains-save

Setting up firewall chains just the way you want them, and then trying to remember the commands you used so you can do them next time is a pain.

So, `ipchains-save` is a script which reads your current chains setup and saves it to a file. For the moment I'll keep you in suspense with regards to what `ipchains-restore` does.

`ipchains-save` can save a single chain, or all chains (if no chain name is specified). The only option currently permitted is `-v' which prints the rules (to stderr) as they are saved. The policy of the chain is also saved for `input`, `output` and `forward` chains.

```
# ipchains-save > my_firewall
Saving `input'.
Saving `output'.
Saving `forward'.
Saving `ppp-in'.
Saving `ppp-out'.
#
```

Using ipchains-restore

`ipchains-restore` restores chains as saved with `ipchains-save`. It can take two options: `-v' which describes each rule as it is added, and `-f' which forces flushing of user-defined chains if they exist, as described below.

If a user-defined chain is found in the input, `ipchains-restore` checks if that chain already exists. If it does, then you will be prompted whether the chains should be flushed (cleared of all rules) or whether restoring this chain should be skipped. If you specified `-f' on the command line, you will not be prompted; the chain will be flushed.

For example:

```
# ipchains-restore < my_firewall
Restoring `input'.
Restoring `output'.
Restoring `forward'.
Restoring `ppp-in'.
Chain `ppp-in' already exists. Skip or flush? [S/f]? s
Skipping `ppp-in'.
Restoring `ppp-out'.
Chain `ppp-out' already exists. Skip or flush? [S/f]? f
Flushing `ppp-out'.
#
```

5. Miscellaneous.

This section contains all the information and FAQs that I couldn't fit inside the structure above.

5.1 How to Organize Your Firewall Rules

This question requires some thought. You can try to organize them to optimize speed (minimize the number of rule-checks for the most common packets) or to increase manageability.

If you have an intermittent link, say a PPP link, you might want to set the first rule in the input chain to be set to `-i ppp0 -j DENY' at boot time, then have something like this in your `ip-up` script:

```
# Re-create the `ppp-in' chain.
ipchains-restore -f < ppp-in.firewall

# Replace DENY rule with jump to ppp-handling chain.
ipchains -R input 1 -i ppp0 -j ppp-in
```

Your `ip-down` script would look like:

```
ipchains -R input 1 -i ppp0 -j DENY
```

5.2 What Not To Filter Out

There are some things you should be aware of before you start filtering out everything you don't want.

ICMP packets

ICMP packets are used (among other things) to indicate failure for other protocols (such as TCP and UDP). `destination-unreachable' packets in particular. Blocking these packets means that you will never get `Host unreachable' or `No route to host' errors; any connections will just wait for a reply that never comes. This is irritating, but rarely fatal.

A worse problem is the role of ICMP packets in MTU discovery. All good TCP implementations (Linux included) use MTU discovery to try to figure out what the largest packet that can get to a destination without being fragmented (fragmentation slows performance, especially when occasional fragments are lost). MTU discovery works by sending packets with the "Don't Fragment" bit set, and then sending smaller packets if it gets an ICMP packet indicating "Fragmentation needed but DF set" (`fragmentation-needed'). This is a type of `destination-unreachable' packet, and if it is never received, the local host will not reduce MTU, and performance will be abysmal or non-existent.

Note that it is common to block all ICMP redirect messages (type 5); these can be used to manipulate routing (although good IP stacks have safeguards), and so are often seen as slightly risky.

TOC

TCP Connections to DNS (nameservers)

If you're trying to block outgoing TCP connections, remember that DNS doesn't always use UDP; if the reply from the server exceeds 512 bytes, the client uses a TCP connection (still going to port number 53) to get the data.

This can be a trap because DNS will `mostly work' if you disallow such TCP transfers; you may experience strange long delays and other occasional DNS problems if you do.

If your DNS queries are always directed at the same external source (either directly by using the `nameserver` line in `/etc/resolv.conf` or by using a caching nameserver in forward mode), then you need only allow TCP connections to port `domain` on that nameserver from the local `domain` port (if using a caching nameserver) or from a high port (> 1023) if using `/etc/resolv.conf`.

FTP Nightmares

The classic packet filtering problem is FTP. FTP has two **modes**; the traditional one is called **active mode** and the more recent one is called **passive mode**. Web browsers usually default to passive mode, but command-line FTP programs usually default to active mode.

In active mode, when the remote end wants to send a file (or even the results of an `ls` or `dir` command) it tries to open a TCP connection to the local machine. This means you can't filter out these TCP connections without breaking active FTP.

If you have the option of using passive mode, then fine; passive mode makes data connections from client to server, even for incoming data. Otherwise, it is recommended that you only allow TCP connections to ports above 1024 and not between 6000 and 6010 (6000 is used for X-Windows).

5.3 Filtering out Ping of Death

Linux boxes are now immune to the famous **Ping of Death**, which involves sending an illegally-large ICMP packet which overflows buffers in the TCP stack on the receiver and causes havoc.

If you are protecting boxes which might be vulnerable, you could simply block ICMP fragments. Normal ICMP packets aren't large enough to require fragmentation, so you won't break anything except big pings. I have heard (unconfirmed) reports that some systems required only the last fragment of an oversize ICMP packet to corrupt them, so blocking only the first fragment is not recommended.

While the exploit programs I have seen all use ICMP, there is no reasons that TCP or UDP fragments (or an unknown protocol) could not be used for this attack, so blocking ICMP fragments is only a temporary solution.

5.4 Filtering out Teardrop and Bonk

Teardrop and Bonk are two attacks (mainly against Microsoft Windows NT machines) which rely on overlapping fragments. Having your Linux router do defragmentation, or disallowing all fragments to your vulnerable machines are the other options.

TOC

5.5 Filtering out Fragment Bombs

Some less-reliable TCP stacks are said to have problems dealing with large numbers of fragments of packets when they don't receive all the fragments. Linux does not have this problem. You can filter out fragments (which might break legitimate uses) or compile your kernel with `IP: always defragment' set to `Y' (only if your Linux box is the only possible route for these packets).

5.6 Changing Firewall Rules

There are some timing issues involved in altering firewall rules. If you are not careful, you can let packets through while you are half-way through your changes. A simplistic approach is to do the following:

```
# ipchains -I input 1 -j DENY
# ipchains -I output 1 -j DENY
# ipchains -I forward 1 -j DENY

... make changes ...

# ipchains -D input 1
# ipchains -D output 1
# ipchains -D forward 1
#
```

This drops all packets for the duration of the changes.

If you changes are restricted to a single chain, you might want to create a new chain with the new rules, and then replace (`-R') the rule that pointed to the old chain with one that points to the new chain: then you can delete the old chain. This replacement will occur atomically.

5.7 How Do I Set Up IP Spoof Protection?

IP spoofing is a technique where a host sends out packets which claim to be from another host. Since packet filtering makes decisions based on this source address, IP spoofing is uses to fool packet filters. It is also used to hide the identity of attackers using SYN attacks, Teardrop, Ping of Death and the like (don't worry if you don't know what they are).

The best way to protect from IP spoofing is called Source Address Verification, and it is done by the routing code, and not firewalling at all. Look for a file called `/proc/sys/net/ipv4/conf/all/rp_filter`. If this exists, then turning on Source Address Verification at every boot is the right solution for you. To do that, insert the following lines somewhere in your init scripts, before any network interfaces are initialized:

```
# This is the best method: turn on Source Address Verification and get
# spoof protection on all current and future interfaces.
if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]; then
  echo -n "Setting up IP spoofing protection..."
  for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
      echo 1 > $f
  done
  echo "done."
```

```
else
  echo PROBLEMS SETTING UP IP SPOOFING PROTECTION.  BE WORRIED.
  echo "CONTROL-D will exit from this shell and continue system startup."
  echo
  # Start a single user shell on the console
  /sbin/sulogin $CONSOLE
fi
```

If you cannot do this, you can manually insert rules to protect every interface. This requires knowledge of each interface. The 2.1 kernels automatically reject packets claiming to come from the 127.* addresses (reserved for the local loopback interface, `lo`).

For example, say we have three interfaces, `eth0`, `eth1` and `ppp0`. We can use `ifconfig` to tell us the address and netmask of the interfaces. Say `eth0` was attached to a network 192.168.1.0 with netmask 255.255.255.0, `eth1` was attached to a network 10.0.0.0 with netmask 255.0.0.0, and `ppp0` connected to the Internet (where any address except the reserved private IP addresses are allowed), we would insert the following rules:

```
# ipchains -A input -i eth0 -s ! 192.168.1.0/255.255.255.0 -j DENY
# ipchains -A input -i ! eth0 -s 192.168.1.0/255.255.255.0 -j DENY
# ipchains -A input -i eth1 -s ! 10.0.0.0/255.0.0.0 -j DENY
# ipchains -A input -i ! eth1 -s 10.0.0.0/255.0.0.0 -j DENY
#
```

This approach is not as good as the Source Address Verification approach, because if your network changes, you have to change your firewalling rules to keep up.

If you are running a 2.0 series kernel, you might want to protect the loopback interface as well, using a rule like this:

```
# ipchains -A input -i ! lo -s 127.0.0.0/255.0.0.0 -j DENY
#
```

5.8 Advanced Projects

There is a userspace library I have written which is included with the source distribution called `libfw'. It uses the ability of IP Chains 1.3 and above to copy a packet to userspace (using the IP_FIREWALL_NETLINK config option).

The mark value can be used to specify the Quality of Service parameters for packets, or to specify how packets should be port-forwarded. I've never used either, but if you want to write about it, please contact me.

Things such as **stateful inspection** (I prefer the term dynamic firewalling) can be implemented in userspace using this library. Other nifty ideas include controlling packets on a per-user basis by doing a lookup in a userspace daemon. This should be pretty easy.

SPF: Stateful Packet Filtering

ftp://ftp.interlinx.bc.ca/pub/spf is the site of Brian Murrell's SPF project, which does connection tracking in userspace. It adds significant security for low-bandwidth sites.

There's little documentation at present, but here's a post to the mailing list in which Brian answered some

questions:

```
> I believe it does exactly what I want: Installing a temporary
> "backward"-rule to let packets in as a response to an
> outgoing request.

Yup, that is exactly what it does.  The more protocols it
understands, the more "backward" rules it gets right.  Right
now it has support for (from memory, please excuse any errors
or omissions) FTP (both active and passive, in and out), some
RealAudio, traceroute, ICMP and basic ICQ (inbound from the ICQ
servers, and direct TCP connections, but alas the secondary
direct TCP connections for things like file transfer, etc. are
not there yet)

> Is it a replacement for ipchains or a supplement?

It is a supplement.  Think of ipchains as the engine to allow
and prevent packets from travelling across a Linux box.  SPF is
the driver, constantly monitoring traffic and telling ipchains
how to change it's policies to reflect the changes in traffic
patterns.
```

Michael Hasenstein's ftp-data hack

Michael Hasenstein of SuSE has written a kernel patch which adds ftp connection tracking to ipchains. It can currently be found at http://www.csn.tu-chemnitz.de/~mha/patch.ftp-data-2.gz

5.9 Future Enhancements

Firewalling and NAT are being redesigned for 2.3. Plans and discussions are available on the netdev archive, and ipchains-dev list. These enhancements should clear up many outstanding usability issues (really, firewalling and masquerading shouldn't be *this hard*), and allow growth for far more flexible firewalling.

---

6. Common Problems

6.1 ipchains -L Freezes!

You're probably blocking DNS lookups; it will eventually time out. Try using the `-n' (numeric) flag to ipchains, which suppresses the lookup of names.

6.2 Masquerading/Forwarding Doesn't Work!

Make sure that packet forwarding is enabled (in recent kernels it is disabled by default, meaning that packets never even try to traverse the `forward' chain). You can override this (as root) by typing

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
#
```

If this works for you, you can put this somewhere in your bootup scripts so it is enabled every time; you'll want to set up your firewalling before this command runs though, otherwise there's an opportunity for packets to slip through.

6.3 -j REDIR doesn't work!

You must allow forwarding packets (see above) for redirect to work; otherwise the routing code drops the packet. So if you are just using redirect, and don't have any forwarding at all, you should be aware of that.

Note that REDIR (being in the input chain) doesn't effect connections from a local process.

6.4 Wildcard Interfaces Don't Work!

There was a bug in versions 2.1.102 and 2.1.103 of the kernel (and some old patches I produced) which made ipchains commands which specified a wildcard interface (such as `-i ppp+`) fail.

This is fixed in recent kernels, and in the 2.0.34 patch on the web site. You can also fix it by hand in the kernel source by changing line 63 or so in include/linux/ip_fw.h:

```
#define IP_FW_F_MASK    0x002F  /* All possible flag bits mask   */
```

This should read ``0x003F''. Fix this and recompile the kernel.

TOC

6.5 TOS Doesn't Work!

This was my mistake: setting the Type of Service field did not actually set the Type of Service in kernel versions 2.1.102 through 2.1.111. This problem was fixed in 2.1.112.

6.6 ipautofw and ipportfw Don't Work!

For 2.0.x, this is true; I haven't time to create and maintain a jumbo patch for ipchains and ipautofw/ipportfw.

For 2.1.x, download Juan Ciarlante's ipmasqadm from

```
<url url="http://juanjox.linuxhq.com/"
        name="http://juanjox.linuxhq.com/">
```

and use it exactly as you would have used `ipautofw` or `ipportfw`, except instead of `ipportfw` you type `ipmasqadm portfw`, and instead of `ipautofw` you type `ipmasqadm autofw`.

6.7 xosview is Broken!

Upgrade to version 1.6.0 or above, which doesn't require any firewall rules at all for 2.1.x kernels. This seems to have broken again in the 1.6.1 release; please bug the author (it's not my fault!).

6.8 Segmentation Fault With `-j REDIRECT'!

This was a bug in ipchains version 1.3.3. Please upgrade.

6.9 I Can't Set Masquerading Timeouts!

True (for 2.1.x kernels) up to 2.1.123. In 2.1.124, trying to set the masquerading timeouts causes a kernel lockup (change `return` to `ret =` on line 1328 of net/ipv4/ip_fw.c). In 2.1.125, it works fine.

6.10 I Want to Firewall IPX!

So do a number of others, it seems. My code only covers IP, unfortunately. On the good side, all the hooks are there to firewall IPX! You just need to write the code; I will happily help where possible.

---

7. A Serious Example.

This example was extracted from Michael Neuling and my March 1999 LinuxWorld Tutorial; this is not the only way to solve the given problem, but it is probably the simplest. I hope you will find it informative.

7.1 The Arrangement

- Masqueraded internal network (various operating systems), which we call "GOOD".
- Exposed servers in a separate network (called "DMZ" for Demilitarized Zone).
- PPP Connection to the Internet (called "BAD").

```
External Network (BAD)
        |
        |
   ppp0|
 ---------------
 | 192.84.219.1|             Server Network (DMZ)
 |             |eth0
 |             |----------------------------------------------
 |             |192.84.219.250 |             |              |
 |             |               |             |              |
 |192.168.1.250|               |             |              |
 ---------------          --------       -------        -------
        | eth1            | SMTP |       | DNS |        | WWW |
        |                 --------       -------        -------
        |              192.84.219.128  192.84.219.129  192.84.218.130
        |
Internal Network (GOOD)
```

7.2 Goals

Packet Filter box:

PING any network

This is really useful to tell if a machine is down.

TRACEROUTE any network

Once again, useful for diagnosis.

Access DNS

To make ping and DNS more useful.

Within the DMZ:

Mail server

- SMTP to external

- Accept SMTP from internal and external
- Accept POP-3 from internal

Name Server

- Send DNS to external
- Accept DNS from internal, external and packet filter box

Web server

- Accept HTTP from internal and external
- Rsync access from internal

Internal:

Allow WWW, ftp, traceroute, ssh to external

These are fairly standard things to allow: some places start by allowing the internal machines to do just about everything, but here we're being restrictive.

Allow SMTP to Mail server

Obviously, we want them to be able to send mail out.

Allow POP-3 to Mail server

This is how they read their mail.

Allow DNS to Name server

They need to be able to look up external names for WWW, ftp, traceroute and ssh.

Allow rsync to Web server

This is how they synchronize the external web server with the internal one.

Allow WWW to Web server

Obviously, they should be able to connect to our external web server.

Allow ping to packet filter box

This is a courteous thing to allow: it means that they can test if the firewall box is down (so we don't get blamed if an external site is broken).

7.3 Before Packet Filtering

- Anti-spoofing

  Since we don't have any asymmetric routing, we can simply turn on anti-spoofing for all interfaces.

```
# for f in /proc/sys/net/ipv4/conf/*/rp_filter; do echo 1 > $f; done
#
```

- Set filtering rules to DENY all:

  We still allow local loopback traffic, but deny anything else.

```
# ipchains -A input -i ! lo -j DENY
# ipchains -A output -i ! lo -j DENY
```

```
# ipchains -A forward -j DENY
#
```

- Set Up Interfaces

  This is usually done in the boot scripts. Make sure the above steps are done before the interfaces are configured, to prevent packet leakage before the rules are set up.

- Insert per-protocol masquerading modules.

  We need to insert the masquerading module for FTP, so that active and passive FTP `just work' from the internal network.

```
# insmod ip_masq_ftp
#
```

7.4 Packet Filtering for Through Packets

With masquerading, it's best to filter in the forward chain.

Split forward chain into various user chains depending on source/dest interfaces; this breaks the problem down into managable chunks.

```
ipchains -N good-dmz
ipchains -N bad-dmz
ipchains -N good-bad
ipchains -N dmz-good
ipchains -N dmz-bad
ipchains -N bad-good
```

ACCEPTing standard error ICMPs is a common thing to do, so we create a chain for it.

```
ipchains -N icmp-acc
```

Set Up Jumps From forward Chain

Unfortunately, we only know (in the forward chain) the outgoing interface. Thus, to figure out what interface the packet came in on, we use the source address (the anti-spoofing prevents address faking).

Note that we log anything which doesn't match any of these (obviously, this should never happen).

```
ipchains -A forward -s 192.168.1.0/24 -i eth0 -j good-dmz
ipchains -A forward -s 192.168.1.0/24 -i ppp0 -j good-bad
ipchains -A forward -s 192.84.219.0/24 -i ppp0 -j dmz-bad
ipchains -A forward -s 192.84.219.0/24 -i eth1 -j dmz-good
ipchains -A forward -i eth0 -j bad-dmz
ipchains -A forward -i eth1 -j bad-good
ipchains -A forward -j DENY -l
```

Define the icmp-acc Chain

TOC

Packets which are one of the error ICMPs get ACCEPTed, otherwise, control will pass back to the calling chain.

```
ipchains -A icmp-acc -p icmp --icmp-type destination-unreachable -j ACCEPT
ipchains -A icmp-acc -p icmp --icmp-type source-quench -j ACCEPT
ipchains -A icmp-acc -p icmp --icmp-type time-exceeded -j ACCEPT
ipchains -A icmp-acc -p icmp --icmp-type parameter-problem -j ACCEPT
```

Good (Internal) to DMZ (Servers)

Internal restrictions:

- Allow WWW, ftp, traceroute, ssh to external
- **Allow SMTP to Mail server**
- **Allow POP-3 to Mail server**
- **Allow DNS to Name server**
- **Allow rsync to Web server**
- **Allow WWW to Web server**
- Allow ping to packet filter box

Could do masquerading from internal network into DMZ, but here we don't. Since noone in the internal network should be trying to do evil things, we log any packets that get denied.

```
ipchains -A good-dmz -p tcp -d 192.84.219.128 smtp -j ACCEPT
ipchains -A good-dmz -p tcp -d 192.84.219.128 pop-3 -j ACCEPT
ipchains -A good-dmz -p udp -d 192.84.219.129 domain -j ACCEPT
ipchains -A good-dmz -p tcp -d 192.84.219.129 domain -j ACCEPT
ipchains -A good-dmz -p tcp -d 192.84.218.130 www -j ACCEPT
ipchains -A good-dmz -p tcp -d 192.84.218.130 rsync -j ACCEPT
ipchains -A good-dmz -p icmp -j icmp-acc
ipchains -A good-dmz -j DENY -l
```

Bad (external) to DMZ (servers).

- DMZ restrictions:
  - Mail server
    - **SMTP to external**
    - **Accept SMTP from** internal and **external**
    - Accept POP-3 from internal
  - Name server
    - **Send DNS to external**
    - **Accept DNS from** internal, **external** and packet filter box
  - Web server
    - **Accept HTTP from** internal and **external**
    - Rsync access from internal
- Things we allow from external network to DMZ.

TOC

- Don't log violations, as they may happen.

```
ipchains -A bad-dmz -p tcp -d 192.84.219.128 smtp -j ACCEPT
ipchains -A bad-dmz -p udp -d 192.84.219.129 domain -j ACCEPT
ipchains -A bad-dmz -p tcp -d 192.84.219.129 domain -j ACCEPT
ipchains -A bad-dmz -p tcp -d 192.84.218.130 www -j ACCEPT
ipchains -A bad-dmz -p icmp -j icmp-acc
ipchains -A bad-dmz -j DENY
```

Good (internal) to Bad (external).

- Internal restrictions:
  - **Allow WWW, ftp, traceroute, ssh to external**
  - Allow SMTP to Mail server
  - Allow POP-3 to Mail server
  - Allow DNS to Name server
  - Allow rsync to Web server
  - Allow WWW to Web server
  - Allow ping to packet filter box
- Many people allow everything from the internal to external networks, then add restrictions. We're being fascist.
  - Log violations.
  - Passive FTP handled by masq. module.

```
ipchains -A good-bad -p tcp --dport www -j MASQ
ipchains -A good-bad -p tcp --dport ssh -j MASQ
ipchains -A good-bad -p udp --dport 33434:33500 -j MASQ
ipchains -A good-bad -p tcp --dport ftp --j MASQ
ipchains -A good-bad -p icmp --icmp-type ping -j MASQ
ipchains -A good-bad -j REJECT -l
```

DMZ to Good (internal).

- Internal restrictions:
  - Allow WWW, ftp, traceroute, ssh to external
  - **Allow SMTP to Mail server**
  - **Allow POP-3 to Mail server**
  - **Allow DNS to Name server**
  - **Allow rsync to Web server**
  - **Allow WWW to Web server**
  - Allow ping to packet filter box
- If we were masquerading from the internal network to the DMZ, simply refuse any packets coming the other way. As it is, only allow packets which might be part of an established connection.

```
ipchains -A dmz-good -p tcp ! -y -s 192.84.219.128 smtp -j ACCEPT
```

TOC

```
ipchains -A dmz-good -p udp -s 192.84.219.129 domain -j ACCEPT
ipchains -A dmz-good -p tcp ! -y -s 192.84.219.129 domain -j ACCEPT
ipchains -A dmz-good -p tcp ! -y -s 192.84.218.130 www -j ACCEPT
ipchains -A dmz-good -p tcp ! -y -s 192.84.218.130 rsync -j ACCEPT
ipchains -A dmz-good -p icmp -j icmp-acc
ipchains -A dmz-bad -j DENY -l
```

DMZ to bad (external).

- DMZ restrictions:
  - Mail server
    - **SMTP to external**
    - **Accept SMTP from** internal and **external**
    - Accept POP-3 from internal
  - Name server
    - **Send DNS to external**
    - **Accept DNS from** internal, **external** and packet filter box
  - Web server
    - **Accept HTTP from** internal and **external**
    - Rsync access from internal
-

```
ipchains -A dmz-bad -p tcp -s 192.84.219.128 smtp -j ACCEPT
ipchains -A dmz-bad -p udp -s 192.84.219.129 domain -j ACCEPT
ipchains -A dmz-bad -p tcp -s 192.84.219.129 domain -j ACCEPT
ipchains -A dmz-bad -p tcp ! -y -s 192.84.218.130 www -j ACCEPT
ipchains -A dmz-bad -p icmp -j icmp-acc
ipchains -A dmz-bad -j DENY -l
```

Bad (external) to Good (internal).

- We don't allow anything (non-masqueraded) from the external network to the internal network

```
ipchains -A bad-good -j REJECT
```

Packet Filtering for the Linux Box Itself

- If we want to use packet filtering on packets coming into the box itself, we need to do filtering in the input chain. We create one chain for each destination interface:

```
ipchains -N bad-if
ipchains -N dmz-if
ipchains -N good-if
```

- Create jumps to them:

```
ipchains -A input -d 192.84.219.1 -j bad-if
ipchains -A input -d 192.84.219.250 -j dmz-if
ipchains -A input -d 192.168.1.250 -j good-if
```

TOC

Bad (external) interface.

- Packet Filter box:
  - **PING any network**
  - **TRACEROUTE any network**
  - Access DNS
- External interface also receives replies to masqueraded packets, and ICMP errors for them and PING replies.

```
ipchains -A bad-if -i ! ppp0 -j DENY -l
ipchains -A bad-if -p TCP --dport 61000:65096 -j ACCEPT
ipchains -A bad-if -p UDP --dport 61000:65096 -j ACCEPT
ipchains -A bad-if -p ICMP --icmp-type pong -j ACCEPT
ipchains -A bad-if -j icmp-acc
ipchains -A bad-if -j DENY
```

DMZ interface.

- Packet Filter box restrictions:
  - **PING any network**
  - **TRACEROUTE any network**
  - **Access DNS**
- DMZ interface receives DNS replies, ping replies and ICMP errors.

```
ipchains -A dmz-if -i ! eth0 -j DENY
ipchains -A dmz-if -p TCP ! -y -s 192.84.219.129 53 -j ACCEPT
ipchains -A dmz-if -p UDP -s 192.84.219.129 53 -j ACCEPT
ipchains -A dmz-if -p ICMP --icmp-type pong -j ACCEPT
ipchains -A dmz-if -j icmp-acc
ipchains -A dmz-if -j DENY -l
```

Good (internal) interface.

- Packet Filter box restrictions:
  - **PING any network**
  - **TRACEROUTE any network**
  - **Access DNS**
- Internal restrictions:
  - Allow WWW, ftp, traceroute, ssh to external
  - Allow SMTP to Mail server
  - Allow POP-3 to Mail server
  - Allow DNS to Name server
  - Allow rsync to Web server
  - Allow WWW to Web server
  - **Allow ping to packet filter box**

TOC

- Internal interface receives pings, ping replies and ICMP errors.

```
ipchains -A good-if -i ! eth1 -j DENY
ipchains -A good-if -p ICMP --icmp-type ping -j ACCEPT
ipchains -A good-if -p ICMP --icmp-type pong -j ACCEPT
ipchains -A good-if -j icmp-acc
ipchains -A good-if -j DENY -l
```

7.5 Finally

- Delete blocking rules:

```
ipchains -D input 1
ipchains -D forward 1
ipchains -D output 1
```

8. Appendix: Differences between ipchains and ipfwadm.

Some of these changes are a result of kernel changes, and some a result of `ipchains` being different from `ipfwadm`.

1. Many arguments have been remapped: capitals now indicates a command, and lower case now indicates an option.
2. Arbitrary chains are supported, so even built-in chains have full names instead of flags (eg. `input' instead of `-I').
3. The `-k' option has vanished: use `! -y'.
4. The `-b' option actually inserts/appends/deletes two rules, rather than a single `bidirectional' rule.
5. The `-b' option can be passed to `-C' to do two checks (one in each direction).
6. The `-x' option to `-l' has been replaced by `-v'.
7. Multiple source and destination ports are not supported anymore. Hopefully being able to negate the port range will somewhat make up for that.
8. Interfaces can only be specified by name (not address). The old semantics got silently changed in the 2.1 kernel series anyway.
9. Fragments are examined, not automatically allowed through.
10. Explicit accounting chains have been done away with.
11. Arbitrary protocols over IP can be tested for.
12. The old behavior of SYN and ACK matching (which was previously ignored for non-TCP packets) has changed; the SYN option is not valid for non-TCP-specific rules.
13. Counters are now 64-bit on 32-bit machines, not 32-bit.
14. Inverse options are now supported.
15. ICMP codes are now supported.
16. Wildcard interfaces are now supported.
17. TOS manipulations are now sanity-checked: the old kernel code would silently stop you from (illegally) manipulating the `Must Be Zero' TOS bit; ipchains now returns an error if you try, as well as for other illegal cases.

8.1 Quick-Reference table.

[Mainly, command arguments are UPPER CASE, and option arguments are lower case]

One thing to note, masquerading is specified by `-j MASQ'; it is completely different from `-j ACCEPT', and not treated as merely a side-effect, unlike `ipfwadm` does.

TOC

ipfwadm	ipchains	Notes
-A [both]	-N acct & -I 1 input -j acct & -I 1 output -j acct & acct	Create an `acct' chain and have output and input packets traverse it.
-A in	input	A rule with no target
-A out	output	A rule with no target
-F	forward	Use this as [chain].
-I	input	Use this as [chain].
-O	output	Use this as [chain].
-M -l	-M -L	
-M -s	-M -S	
-a policy	-A [chain] -j POLICY	(but see -r and -m).
-d policy	-D [chain] -j POLICY	(but see -r and -m).
-i policy	-I 1 [chain] -j POLICY	(but see -r and -m).
-l	-L	
-z	-Z	
-f	-F	
-p	-P	
-c	-C	
-P	-p	
-S	-s	Only takes one port or range, not multiples.
-D	-d	Only takes one port or range, not multiples.

-V	<none>	Use -i [name].
-W	-i	
-b	-b	Now actually makes 2 rules.
-e	-v	
-k	! -y	Doesn't work unless -p tcp also specified.
-m	-j MASQ	
-n	-n	
-o	-l	
-r [redirpt]	-j REDIRECT [redirpt]	
-t	-t	
-v	-v	
-x	-x	
-y	-y	Doesn't work unless -p tcp also specified.

8.2 Examples of translated ipfwadm commands

Old command: ipfwadm -F -p deny

New command: ipchains -P forward DENY

Old command: ipfwadm -F -a m -S 192.168.0.0/24 -D 0.0.0.0/0

New command: ipchains -A forward -j MASQ -s 192.168.0.0/24 -d 0.0.0.0/0

Old command: ipfwadm -I -a accept -V 10.1.2.1 -S 10.0.0.0/8 -D 0.0.0.0/0

New command: ipchains -A input -j ACCEPT -i eth0 -s 10.0.0.0/8 -d 0.0.0.0/0

(Note that there is no equivalent for specifying interfaces by address: use the interface name. On this machine, 10.1.2.1 corresponds to eth0).

---

9. Appendix: Using the ipfwadm-wrapper script.

The `ipfwadm-wrapper` shell script should be a plug-in replacement of `ipfwadm` for backwards compatibility with ipfwadm 2.3a.

The only feature it can't really handle is the `-V' option. When this is used, a warning is given. If the `-W' option is also used, the `-V' option is ignored. Otherwise, the script tries to find the interface name associated with that address, using `ifconfig`. If that fails (such as for an interface which is down) then it will exit with an error message.

This warning can be suppressed by either changing the `-V' to a `-W', or directing the standard output of the script to /dev/null.

If you should find any mistakes in this script, or any changes between the real ipfwadm and this script, *please* report a bug to me: send an EMail to ipchains@rustcorp.com with "BUG-REPORT" in the subject. Please list your old version of `ipfwadm` (`ipfwadm -h`), your version of `ipchains` (`ipchains --version`), the version of the ipfwadm wrapper script (`ipfwadm-wrapper --version`). Also send the output of `ipchains-save`. Thanks in advance.

Mix `ipchains` with this `ipfwadm-wrapper` script at your own peril.

10. Appendix: Thanks.

Many thanks have to go to Michael Neuling, who wrote the first releasable cut of the IP chains code while working for me. Public apologies for nixing his result-caching idea, which Alan Cox later proposed and I have finally begun implementing, having seen the error of my ways.

Thanks to Alan Cox for his 24-hour EMail tech support, and encouragement.

Thanks to all the authors of the ipfw and ipfwadm code, especially Jos Vos. Standing on the shoulders of giants and all that... This applies to Linus Torvalds and all the kernel and userspace hackers as well.

Thanks to the diligent beta testers and bughunters, especially Jordan Mendelson, Shaw Carruthers, Kevin Moule, Dr. Liviu Daia, Helmut Adams, Franck Sicard, Kevin Littlejohn, Matt Kemner, John D. Hardin, Alexey Kuznetsov, Leos Bitto, Jim Kunzman, Gerard Gerritsen, Serge Sivkov, Andrew Burgess, Steve Schmidtke, Richard Offer, Bernhard Weisshuhn, Larry Auton, Ambrose Li, Pavel Krauz, Steve Chadsey, Francesco Potorti` and Alain Knaff.

Linux IPX-HOWTO

Kevin Thorpe, kevin@pricetrak.com

v2.3, 06 May 1998

This document aims to describe how to obtain, install and configure various tools available for the the Linux operating system that use the Linux kernel IPX protocol support.

1. Introduction.

- 1.1 Changes from the previous release.
- 1.2 Introduction.

2. Disclaimer.

3. Related Documentation.

- 3.1 New versions of this document.
- 3.2 Feedback.
- 3.3 Mailing list support.

4. Some of the terms used in this document.

5. A brief discussion of IPX network topology

6. The IPX related files in the /proc filesystem.

7. Greg Pages IPX tools.

- 7.1 The IPX tools in more detail.

TOC

8. Configuring your Linux machine as an IPX router.

- 8.1 Do I need to configure an internal network ?

9. Configuring your Linux machine as an NCP client.

- 9.1 Obtaining ***ncpfs***.
- 9.2 Building ***ncpfs*** for kernel 1.2.13.
- 9.3 Building ***ncpfs*** for kernels 1.3.71++/2.0.*.
- 9.4 Configuring and using ***ncpfs***.

10. Configuring your Linux machine as an NCP server.

- 10.1 The ***mars_nwe*** package.
- 10.2 The ***lwared*** package.

11. Configuring your Linux machine as a Novell Print Client.

12. Configuring your Linux machine as a Novell Print Server.

- 12.1 Prerequisites
- 12.2 Configuration

13. An overview of the *ncpfs* user and adminstration commands

- 13.1 User commands.
- 13.2 Administration tools.

TOC

14. Configuring PPP for IPX support.

- 14.1 Configuring an IPX/PPP server.
- 14.2 Configuring an IPX/PPP client.

15. IPX tunnel over IP

- 15.1 Obtaining *ipxtunnel*
- 15.2 Building *ipxtunnel*
- 15.3 Configuring *ipxtunnel*
- 15.4 Testing and using *ipxtunnel*

16. Commercial IPX support for Linux.

- 16.1 Caldera'a Network Desktop

17. Some Frequently Asked Questions

18. Copyright Message.

19. Miscellaneous and Acknowledgements.

---

1. Introduction.

This is the Linux IPX-HOWTO. You should read the Linux NET-3-HOWTO in conjunction with this document.

1.1 Changes from the previous release.

```
Change of author:
        Many thanks to Terry Dawson for passing on this document and
        congratulations on becoming a father :-).

Additions:
        Addition of a brief explanation of IPX. This is in response to
        many baffled queries on the discussion lists.

Corrections/Updates:
        New version of ncpfs which now supports NDS logins. This is early
        beta test and may be prohibited in your country due to the use of
        patented technology.

        Addition of support for trustee rights in mars_nwe. This is still
        in beta test.
```

1.2 Introduction.

The Linux Kernel has a completely new network implementation as compared to other Unix like operating systems. The ability to take a fresh approach to developing the kernel networking software has led to the Linux kernel having support for a range of non tcp/ip protocols being built. The IPX protocol is one of those that have been included.

The Linux kernel supports the IPX protocol only. It does not yet support protocols such as IPX/RIP, SAP or NCP, these are supported by other software such as that documented elsewhere in this document.

The IPX support was originally developed by Alan Cox `<alan@lxorguk.ukuu.org.uk>` and has been significantly enhanced by Greg Page `<greg@caldera.com>`.

2. Disclaimer.

I do not and cannot know everything there is to know about the Linux network software. Please accept and be warned that this document probably does contain errors. Please read any README files that are included with any of the various pieces of software described in this document for more detailed and accurate information. I will attempt to keep this document as error-free and up-to-date as possible. Versions of software are current as at time of writing.

In no way do I or the authors of the software in this document offer protection against your own actions. If you configure this software, even as described in this document and it causes problems on your network then you alone must carry the responsibility. I include this warning because IPX network design and configuration is not always a simple matter and sometimes undesirable interaction with other routers and fileservers can result if you do not design or configure your network carefully. I also include this warning because I was asked to by someone unfortunate enough to have discovered this lesson the hard way.

3. Related Documentation.

This document presumes you understand how to build a Linux kernel with the appropriate networking options selected and that you understand how to use the basic network tools such as *ifconfig* and *route*. If you do not, then you should read the NET-3-HOWTO in conjunction with this document as it describes these.

Other Linux HOWTO documents that might be useful are:

The Ethernet-HOWTO, which describes the details of configuring an Ethernet device for Linux.

The PPP-HOWTO as IPX support is available for version 2.2.0d and later of the Linux PPP implementation.

3.1 New versions of this document.

If your copy of this document is more than two months old then I strongly recommend you obtain a newer version. The networking support for Linux is changing very rapidly with new enhancements and features, so this document also changes fairly frequently. The latest released version of this document can always be retrieved by anonymous ftp from:

`ftp:`/sunsite.unc.edu/pub/Linux/docs/HOWTO/IPX-HOWTO>/ or:
`ftp:`/sunsite.unc.edu/pub/Linux/docs/HOWTO/other-formats/IPX-HOWTO{-html.tar,ps,dvi}.gz>/ via the World Wide Web from the Linux Documentation Project Web Server, at page: IPX-HOWTO or directly from me, `<kevin@pricetrak.com>`. It may also be posted to the newsgroups: `comp.os.linux.networking`, `comp.os.linux.answers` and `news.answers` from time to time.

3.2 Feedback.

Please send any comments, updates, or suggestions to me, `<kevin@pricetrak.com>`. The sooner I get feedback, the sooner I can update and correct this document. If you find any problems with it, please mail me directly as I can miss info posted to the newsgroups.

3.3 Mailing list support.

There is a mailing list established for discussion of the various Linux IPX software packages described in this document. You can subscribe to it by sending a mail message to `` `listserv@sh.cvut.cz' `` with `` `add linware' `` in the body of the message. To post to the list your send your mail to `` `linware@sh.cvut.cz' ``. I regularly watch this list.

The mailing list is archived at www.kin.vslib.cz.

4. Some of the terms used in this document.

You will often see the terms `client` and `server` used in this document. They are normally fairly specific terms but in this document I have generalized their definitions a little so that they mean the following:

client

The machine or program that initiates an action or a connection for the purpose of gaining use of some service or data.

server

The machine or program that accepts incoming connections from multiple remote machines and provides a service or data to those.

These definitions are not very reliable either, but they provide a means of distinguishing the ends of peer to peer systems such as *SLIP* or *PPP* which truly do not actually have clients and servers.

Other terms you will see are:

Bindery

The *bindery* is a specialised database storing network configuration information on a Novell fileserver. Netware clients may query the *bindery* to obtain information on available services, routing and user information.

Frame Type

is a term used to describe that actual protocol used to carry the IPX (and IP) datagrams across your ethernet style network segments. There are four common ones. They are:

Ethernet_II

This is a refined version of the original DIX ethernet standard. Novell has been allocated a formal protocol id and this means that both IPX and IP can coexist happily in an Ethernet_II environment quite happily. This is commonly used in Novell environments and is a good choice.

802.3

This is an I.E.E.E. protocol defining a Carrier Sense Multiple Access with Collision Detection (CSMA/CD) mechanism. It was based on the original DIX Ethernet standard, with an important modification, the type (protocol id) field was converted into a length field instead. It is for this reason that IPX really shouldn't be run here. IEEE 802.3 was designed to carry IEEE 802.2 frames **only** but there are implementations that use it to carry IPX frames directly and remarkably it does work. Avoid it unless you are trying to interwork with a network already configured to use it.

802.2

This is an I.E.E.E. protocol that defines a set of Logical Link Control procedures. It pro a simplistic way of allowing different protocols to coexist, but is quite limited in this respect. Novell uses an unofficial Service Address Point (like a protocol id) but since everyone else uses it as well, that hasn't yet presented too much of a problem.

SNAP

SNAP is the Sub Network Access Protocol. This protocol is designed ride on top of 802.3 and 802.2. It expands the multiprotocol capability of 802.2 and provides some measure of compatability with existing Ethernet and Ethernet_II frame types.

IPX

Internet Packet eXchange is a protocol used by the Novell corporation to provide internetworking support for their NetWare(tm) product. IPX is similar in functionality to the IP protocol used by the tcp/ip community.

IPX network address

This is a number which uniquely identifies a particular IPX network. The usual notation for this address is in hexadecimal. An example might look like: `0x23a91002`.

IPX Internal network

This is a virtual IPX network. It is virtual because it does not correspond to a physical network. This is used to provide a means of uniquely identifying and addressing a particular IPX host. This is generally only useful to IPX hosts that exist on more than one physical IPX network such as fileservers. The address is coded in the same form as for a physical IPX network.

RIP

Routing Information Protocol is a protocol used to automatically propagate network routes in an IPX network. It is functionally similar to the RIP used within the tcp/ip community.

NCP

NetWare Core Protocol is a networked filesystem protocol designed by the Novell Corporation for their NetWare(tm) product. NCP is functionally similar to the NFS used in the tcp/ip community.

SAP

Service Advertisement Protocol is a protocol designed by the Novell Corporation that is used to advertise network services in a NetWare(tm) environment.

Hardware address

This is a number that uniquely identifies a host in a physical network at the media access layer. Examples of this are *Ethernet Addresses*. An Ethernet address is generally coded as six hexadecimal values separated by colon characters eg. `00:60:8C:C3:3C:0F`

route

The *route* is the path that your packets take through the network to reach their destination.

---

5. A brief discussion of IPX network topology

This is a much simplified explanation for people new to IPX. Large networks will probably break lots of the rules explained here. In complex IPX networks the administrator should always be consulted.

IPX networking revolves around a scheme of numbered *networks* unlike IP which places more emphasis on the *interface* addresses. A network is a collection of equipment connected to the same LAN segment and *using the same frame type*. Different frame types on the same LAN segment are treated as seperate networks.

Each network must be allocated a number which is unique across the entire internetwork. This is usually performed by a NetWare(tm) server, but can easily be performed by Linux. IPX clients are given this number by the server when starting, they only require to know the correct frame type.

Routing between networks is usually performed by putting two network cards in a server. This server then runs the RIP protocol which holds a routing table for the internetwork. Periodic broadcasts of this routing table are exchanged between servers. Within a short time each server 'discovers' the topology of the internetwork.

If you only wish to use the services of an existing NetWare server, you can use `ipx_configure` (section 7.1) to automatically define the IPX interfaces by using broadcast queries to look for a server. If this fails, or you wish to provide IPX services, you will need to define the interfaces manually using `ipx_interface` or `mars_nwe`.

6. The IPX related files in the `/proc` filesystem.

There are a number of files related to the Linux IPX support that are located within the `/proc` filesystem. They are:

/proc/net/ipx_interface

This file contains information about the IPX interfaces configured on your machine. These may have been configured manually by command or automatically detected and configured.

/proc/net/ipx_route

This file contains a list of the routes that exist in the IPX routing table. These routes may have been added manually by command or automatically by an IPX routing daemon.

/proc/net/ipx

This file is a list of the IPX sockets that are currently open for use on the machine.

7. Greg Pages IPX tools.

Greg Page <greg@caldera.com of Caldera Incorporated has written a suite of IPX configuration tools and enhanced the Linux IPX kernel support.

The kernel enhancements allow linux to be configured as a fully featured IPX bridge or router. The enhanced IPX support has already been fed back into the mainstream kernel distribution so you will probably already have it.

The network configuration tools provide you with the capability to configure your network devices to support IPX and allow you to configure IPX routing and other facilities under Linux. The Linux IPX network tools are available from: sunsite.unc.edu.

7.1 The IPX tools in more detail.

ipx_interface

This command is used to manually add, delete or check ipx capability to an existing network device. Normally the network device would be an Ethernet device such at `eth0`. At least one IPX interface must be designated the *primary* interface and the *-p* flag to this command does this. For example to enable Ethernet device `eth0` for IPX capability as the primary IPX interface using the IEEE 802.2 frame type and IPX network address `39ab0222` you would use:

```
# ipx_interface add -p eth0 802.2 0x39ab0222
```

If the frame type differs from NetWare(tm) servers on this network, they will studiously ignore you. If the frame type is correct but the network number differs, they will still ignore you but complain frequently on the NetWare server console. The latter is guaranteed to gain you flames from your NetWare administrator and may disrupt existing NetWare clients.

If you get an error while running this program and you happen to not have already configured tcp/ip, then you will find that you need to manually start the `eth0` interface using the command:

```
# ifconfig eth0 up
```

ipx_configure

This command enables or disables the automatic setting of the interface configuration and primary interface settings.

`--auto_interface`

allows you to select whether new network devices should be automatically configured as IPX devices or not.

TOC

`--auto_primary`

allows you to select whether the IPX software should automatically select a primary interface or not. Problems have been noted using this with Windows 95 clients on the network.

A typical example would be to enable both automatic interface configuration and automatic primary interface setting with the following command:

```
# ipx_configure --auto_interface=on --auto_primary=on
```

ipx_internal_net

This command allows you to configure or deconfigure an internal network address. An internal network address is optional, but when it is configured it will always be the primary interface. To configure an IPX network address of `ab000000` on IPX node `1` you would use:

```
# ipx_internal_net add 0xab000000 1
```

ipx_route

The command allows you to manually modify the IPX routing table. For example to add a route to IPX network `39ab0222` via a router with node number `00608CC33C0F` on IPX network `39ab0108`:

```
# ipx_route add 0x39ab0222 0x39ab0108 0x00608CC33C0F
```

---

8. Configuring your Linux machine as an IPX router.

If you have a number of IPX segments that you wish to internetwork you need the services of a router. In the Novell environment there are two pieces of information which are necessary to be propagated around the network. They are the network routing information propagated using Novell RIP, and the service advertisement information propagated using Novell SAP. Any router must support both of these protocols to be useful in most situations.

Linux has support for both of these protocols and can be fairly easily made to function as a fully Novell compliant router.

The Linux kernel IPX support actually manages the IPX packet forwarding across interfaces, but it does this according to the rules coded into the IPX routing table. Linux needs a program to implement the Novell RIP and SAP to ensure that the IPX routing table is built correctly and updated periodically to reflect changes in the network status.

Volker Lendecke `<lendecke@namu01.gwdg.de>` has developed a routing daemon *ipxripd* that will do this for you. The *mars_nwe* package mentioned later includes an alternative routing daemon.

You can find *ipxripd* at:

sunsite.unc.edu

or at Volkers home site at:

ftp.gwdg.de

Configuring your Linux machine to act as a router is very straightforward. The steps you must take are:

1. Build your kernel with IPX, Ethernet and `/proc` support.
2. Obtain, compile and install the *ipxd* daemon program.
3. Boot the new kernel and ensure that each of the Ethernet cards has been properly detected and there are no hardware conflicts.
4. Enable the IPX protocol on each of the interfaces using the *ipx_interface* command described above.
5. Start the *ipxd* daemon program.

Consider the following simple network:

```
 IPX Addr: 0x01000000  802.2
|---------------------------|
              |
              \_____________________________
```

```
                                          \          Linux Router
 IPX Addr: 0x02000000  802.2               \
|---------------------------|               \   eth0/-----------\
             |                              \--====|            |
             \_________________________           | IPX route  |
                                          \     eth1|  Table     |
 IPX Addr: 0x03000000  etherII             \----====|    ^       |
|---------------------------|                      |    |       |
             |                                eth2|  IPXd      |
             \_______________________________/====|            |
                                                  |  SAPd      |
 IPX Addr: 0x04000000  etherII                eth3|            |
|---------------------------|                /====|            |
             |                              |    \____________/
             \_______________________________/
```

The configuration for the above network would look like:

```
# ipx_interface add eth0 802.2 0x0100000000
# ipx_interface add eth1 802.2 0x0200000000
# ipx_interface add eth2 etherii 0x0300000000
# ipx_interface add eth3 etherii 0x0400000000
# ipxd
```

You should then wait a moment or two and check your `/proc/net/ipx_route` file and you should see it populated with the IPX routes relevant to your configuration and any learned from any other routers in the network.

8.1 Do I need to configure an internal network ?

Novell has a feature called an internal network, which it uses to simplify routing in situations where a host has more than one network device connected. This is useful in the case of a fileserver connected to multiple networks as it means that only one route needs to be advertised to reach the server regardless of which network you are attempting from.

In the case of a configuration where you are not running a fileserver and your machine acting only as an IPX router the question is not as simple to answer. It has been reported that configuring for IPX/PPP works `better' if you also configure an internal network.

In any case it is easy to do, but may require a rebuild of your kernel. When you are working through the kernel `make config` you must answer `y` when asked `Full internal IPX network` as illustrated:

```
    ...
    ...
  Full internal IPX network (CONFIG_IPX_INTERN) [N/y/?] y
```

. . .
. . .

To configure the internal network interface, use the *ipx_internal_net* command described earlier in the IPX tools section. The main precaution to take is to ensure that they IPX network address you assign is unique on your network and that no other machine or network is using it.

---

9. Configuring your Linux machine as an NCP client.

If you are a user of a mixed technology network that comprises both IP and IPX protocols it is likely that at some time or another you have wanted to have your Linux machine access data stored on a Novell fileserver on your network. Novell have long offered an NFS server package for their fileservers that would allow this, but if you are a small site or have only a small number of people interested in doing this it is difficult to justify the cost of the commercial package.

Volker Lendecke `<lendecke@namu01.gwdg.de>` has written a Linux filesystem kernel module that supports a subset of the Novell NCP that will allow you to mount Novell volumes into your Linux filesystem without requiring any additional products for your fileserver. Volker has called the package *ncpfs* and derived the necessary information mainly from the book "Netzwerkprogrammierung in C" by Manfred Hill and Ralf Zessin (further details of the book are contained within the README file in the *ncpfs* package).

The software causes Linux to emulate a normal Novell workstation for file services. It also includes a small print utility that allows you to print to Novell print queues (This is documented in the Print Client section later). The *ncpfs* package will work with Novell fileservers of version 3.x and later, it will not work the Novell 2.x. The *ncpfs* client will also work with close Novell compatible products, but unfortunately some products that claim to be compatible aren't compatible enough. To use *ncpfs* with Novell 4.x fileservers, it is preferred to use the Novell server in *bindery* emulation mode. The NDS support is a very recent early beta addition to *ncpfs* and additionally its use may be prohibited in your country due to the inclusion of patented technology.

9.1 Obtaining *ncpfs*.

The latest *ncpfs* package was designed to be built against the version `1.2.13` kernel or kernels later than `1.3.71` (this includes 2.x.x). If you not using a kernel in either of these categories then you will have to upgrade your kernel. The Kernel-HOWTO describes how to do this in detail.

You can obtain the *ncpfs* package by anonymous ftp from Volker's home site at: ftp.gwdg.de or sunsite.unc.edu or mirror sites. The current version at the time of writing was:

`ncpfs-2.0.11.tgz` or `ncpfs-2.2.0.tgz` which adds the NDS support.

9.2 Building *ncpfs* for kernel 1.2.13.

Build a kernel with Ethernet and IPX support

The first thing you need to do is ensure that your kernel has been built with IPX support enabled. In the `1.2.13` version kernel you need only ensure that you have answered `Y` to the question: '`The IPX protocol`' as illustrated:

```
 ...
 ...
Assume subnets are local (CONFIG_INET_SNARL) [y]
Disable NAGLE algorithm (normally enabled) (CONFIG_TCP_NAGLE_OFF) [n]
```

```
The IPX protocol (CONFIG_IPX) [n] y
*
* SCSI support
 ...
 ...
```

You will also need to ensure that you include an appropriate driver for your Ethernet card. If you do not know how to do this then you should read the Ethernet-HOWTO.

You can then proceed to build your kernel. Make sure you remember to run *lilo* to install it when you have finished.

Untar the *ncpfs* software

```
# cd /usr/src
# tar xvfz ncpfs-2.0.11.tgz
# cd ncpfs
```

Check the Makefile

If you intend to use *kerneld* to autoload the *ncpfs* kernel module then you must uncomment the line in the `Makefile` that refers to: `KERNELD`. If you are unsure what this means then you should read the Kernel-HOWTO to familiarise yourself with kernel module configuration.

Make the *ncpfs* software

The software should compile cleanly with no other configuration necessary:

```
# make
```

Copy the IPX tools somewhere useful if you don't already have them.

After the *make* has completed you should find all of the tools you need in the `ncpfs/bin` directory. You can use:

```
# make install
```

to install the tools in Volkers choice of directories. If you are running on an ELF based system then you will need to rerun `` `ldconfig -v' `` to ensure that the shared library is able to be found.

Copy the *ncpfs.o* module somewhere useful if necessary.

If you are compiling for a `1.2.*` kernel then you will find a file called `ncpfs.o` in the `ncpfs/bin` directory after the *make* has completed. This is the *ncpfs* kernel module. You should copy this somewhere useful. On my *debian* system I have copied it to the `/lib/modules/1.2.13/fs` directory and added `ncpfs` to the `/etc/modules` file so that it will be automatically started at boot time. If you are using some other distribution you should find where it keeps its modules and copy it there, or just copy it to your `/etc` directory. To load the modules manually you need to use the command:

```
# insmod ncpfs.o
```

9.3 Building *ncpfs* for kernels 1.3.71++/2.0.*.

For the latest version of *ncpfs* you must use kernel `1.3.71` or newer, this includes the `2.0.*` kernels.

If you intend using a kernel that is version `1.3.71` or newer then the *ncpfs* kernel code has been included in the standard kernel distribution. You need only answer `Y` to:

```
Networking options  --->
    ...
    ...
    <*> The IPX protocol
    ...
Filesystems  --->
    ...
    ...
    <*> NCP filesystem support (to mount NetWare volumes)
    ...
```

You will still need to follow the instructions for building for kernels `1.2.*` so that you can build the tools but there will not be a module file for you to install.

9.4 Configuring and using *ncpfs*.

Configure the IPX network software

There are two ways of configuring the IPX network software. You can manually configure all of your IPX network information or you can choose to let the software determine for itself some reasonable settings using the command:

```
# ipx_configure --auto_interface=on --auto_primary=on
```

This should be reasonable in most circumstances, but if it doesn't work for you then read the 'IPX tools' section above to configure your software manually. Problems have been noted using this on networks containing Windows '95 clients.

Test the configuration

After your IPX network is configured you should be able to use the *slist* command to see a list of all of the Novell fileserver on your network:

```
# slist
```

If the slist command displays a message like: `ncp_connect: Invalid argument` then your kernel probably does not support IPX. Check that you have actually booted off the appropriate kernel. When you boot you should see messages about '`IPX`' and '`ncpfs`' in the system startup messages. If the *slist* command does not list all of your fileservers then you may need to use the manual network configuration method.

Mount a Novell(tm) server or volume.

If your IPX network software is working ok you should now be able to mount a Novell fileserver or

TOC

volume into your Linux filesystem. The *ncpmount* command is used for this purpose and requires tha specify at least the following information:

1. The fileserver name
2. (optionally) The fileserver directory to mount
3. The fileserver login id. If it has a password you will also need that.
4. The mount point ie. where you want the mount to go. This will be an existing directory on your machine.

There is an equivalent *ncpumount* command to unmount a mounted NCP filesystem. The NCP filesystems will be unmounted cleanly if you shutdown your machine normally, so you needn't worry about *ncpumount*ing your filesystems manually before a *halt* or *shutdown*.

An example command to mount fileserver `ACCT_FS01`, with a login id of `guest` with no password, under the `/mnt/Accounts` directory might look like the following:

```
# ncpmount -S ACCT_FS01 /mnt/Accounts -U guest -n
```

Note the use of the `-n` option to indicate that no password is required for the login. The same login specifying a password of `secret` would look like:

```
# ncpmount -S ACCT_FS01 /mnt/Accounts -U guest -P secret
```

If you don't specify either the `-n` or the `-P` options you will be prompted for a password.

Check the mount

If the mount is successful you will find all the volumes accessible to the userid used for login listed as directories under the mount point. You should then also be able to traverse the directory structure to find other files. You may alternatively use the `-V` option to mount a single volume.

NCP does not provide uid or gid ownership of files. All the files will have the permission and ownership assigned to the mount point directory restricted by trustee permissions on the Novell server. Bear this in mind when sharing mounts between Linux users.

Configure mounts to be automatically performed.

If you have some need to permanently have an ncp mount then you will want to configure the commands above into your *rc* files so that they occur automatically at boot time. If your distribution doesn't already provide some way of configuring IPX like debian then I recommend you place them in your `/etc/rc.local` file if you have one. You might use something like:

```
#
# Start the ncp filesystem

/sbin/insmod /lib/modules/1.2.13/fs/ncpfs.o

# configure the IPX network
ipx_configure --auto_interface=on --auto_primary=on

# guest login to the Accounting fileserver
ncpmount -S ACCT_FS01 /mnt/Accounts -U guest -n
```

```
#
```

There is another means of configuring NCP mounts and that is by building a `$HOME/.nwclient` file. This file contains details of temporary or user specific NCP mounts that would be performed regularly. It allows you to store the details of mounts so that you can recreate them without having to specify all of the detail each time.

Its format is quite straightforward:

```
# The first entry is the 'preferred server' entry and is
# used whenever you do not specify a server explicitly.
#
# User TERRY login to DOCS_FS01 fileserver with password 'password'
DOCS_FS01/TERRY password
#
# Guest login to the ACCT_FS01 fileserver with no password.
ACCT_FS01/GUEST -
```

To activate these mounts you could use:

```
$ ncpmount /home/terry/docs
```

to mount: DOCS_FS01 with a login of TERRY under the /home/terry/docs directory. Note that this entry was chosen because no fileserver was specified in the mount command. If the following command were used:

```
$ ncpmount -S ACCT_FS01 /home/terry/docs
```

then a GUEST login to ACCT_FS01 would be mounted there instead.

Note: for this mechanism to work the permissions of the `$HOME/.nwclient` file must be `0600` so you would need to use the command:

```
$ chmod 0600 $HOME/.nwclient
```

If non-root users are to be allowed to use this mechanism then the *ncpmount* command must be Set Userid Root, so you would need to give it permissions:

```
# chmod 4755 ncpmount
```

Try out the *nsend* utility

a utility to send messages to Novell users is also included in the package, it is called *nsend* and is used as follows:

```
# nsend rod hello there
```

would send the message "hello there" to a logged in user "rod" on your "primary" fileserver (the first one appearing in your `.nwclient` file. You can specify another fileserver with the same syntax as for the *ncpmount* command.

10. Configuring your Linux machine as an NCP server.

There are two packages available that allow Linux to provide the functions of a Novell Fileserver. They both allow you to share files on your linux machine with users using Novell NetWare client software. Users can attach and map filesystems to appear as local drives on their machines just as they would to a real Novell fileserver. You may want to try both to see which best serves your intended purpose.

10.1 The *mars_nwe* package.

Martin Stover `<mstover@freeway.de>` developed *mars_nwe* to enable linux to provide both file and print services for NetWare clients.

In case you are wondering about the name: *mars_nwe* is Martin Stovers Netware Emulator.

Capability of *mars_nwe*.

mars_nwe implements a subset of the full Novell NCP for file services, disk based bindery and also print services. It is likely to contain bugs but there are many people using it now and the number of bugs is steadily decreasing as new versions are released.

Obtaining *mars_nwe*.

You can obtain *mars_nwe* from ftp.gwdg.de or from ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/.

The version current at the time of writing was: `mars_nwe-0.99.pl10.tgz`.

Building the *mars_nwe* package.

Build a kernel with Ethernet and IPX Support

In the `1.2.13` version kernel you need only ensure that you have answered `Y` to the question: '`The IPX protocol`' and `N` to the question: `Full internal IPX network' as illustrated:

```
 ...
 ...
The IPX protocol (CONFIG_IPX) [n] y
 ...
 ...
Full internal IPX network (CONFIG_IPX_INTERN) [N/y/?] n
 ...
 ...
```

In newer kernels a similar process is adopted but the actual text of the prompt may have changed slightly.

You will also need to ensure that you include an appropriate driver for your Ethernet card. If you do not know how to do this then you should read the Ethernet-HOWTO.

You can then proceed to build your kernel. Make sure you remember to run *lilo* to install it when you have finished.

Untar the *mars_nwe* package.

```
# cd /usr/src
# tar xvfz mars_nwe-0.99.pl10.tgz
```

Make *mars_nwe*.

To make the package is very simple. The first step is to simply run `make`, this will create a `config.h` file for you. Next you should look at and edit the `config.h` file if necessary. It allows you to configure items such as the installation directories that will be used and the maximum number of sessions and volumes that the server will support. The really important entries to look at are:

```
FILENAME_NW_INI       the location of the initialisation file
PATHNAME_PROGS        where the executable support programs will be found.
PATHNAME_BINDERY      where the 'bindery' files will go.
PATHNAME_PIDFILES     the directory for the 'pid' files to be written.
MAX_CONNECTIONS       the maximum number of simultaneous connections allowed.
MAX_NW_VOLS           the maximum number of volumes mars_nwe will support.
MAX_FILE_HANDLES_CONN the maximum number of open files per connection.
WITH_NAME_SPACE_CALLS if you want to support ncpfs clients.
INTERNAL_RIP_SAP      whether you want mars_nwe to provide rip/sap routing.
SHADOW_PWD            whether you use shadow passwords or not.
```

The defaults will probably be ok but you should check anyway.

When this is done:

```
# make
# make install
```

will build the servers and install them in the appropriate directory. The installation script also installs the configuration file `/etc/nwserv.conf`.

Configure the server.

Configuration is fairly simple. You need to edit the `/etc/nwserv.conf` file. The format of this file may at first look a little cryptic, but it is fairly straightforward. The file contains a number of single line configuration items. Each line is whitespace delimited and begins with a number that indicates the contents of the line. All characters following a '#' character are considered a comment and ignored. Martin supplies an example configuration file in the package, but I'll present what I consider to be a simplified example to offer an alternative for you.

```
# VOLUMES (max. 5)
# Only the SYS volume is compulsory. The directory containing the SYS
# volume must contain the directories: LOGIN, PUBLIC, SYSTEM, MAIL.
# The 'i' option ignores case.
# The 'k' option converts all filenames in NCP requests to lowercase.
# The 'm' option marks the volume as removable (useful for cdroms etc.)
# The 'r' option set the volume to read-only.
# The 'o' option indicates the volume is a single mounted filesystem.
# The 'P' option allows commands to be used as files.
# The 'O' option allows use of the OS/2 namespace
# The 'N' option allows use of the NFS namespace
# The default is upper case.
# Syntax:
#    1 <Volumename> <Volumepath>   <Options>

1   SYS         /home/netware/SYS/                # SYS
1   DATA        /home/netware/DATA/      k        # DATA
1   CDROM       /cdrom                   kmr      # CDROM

# SERVER NAME
```

TOC

```
# If not set then the linux hostname will be converted to upper case
# and used. This is optional, the hostname will be used if this is not
# configured.
# Syntax:
#    2 <Servername>

2   LINUX_FS01

# INTERNAL NETWORK ADDRESS
# The Internal IPX Network Address is a feature that simplifies IPX routing
# for multihomed hosts (hosts that have ports on more than one IPX network).
# Syntax:
#    3 <Internal Network Address> [<Node Number>]
# or:
#    3 auto
#
# If you use 'auto' then your host IP address will be used. NOTE: this may
# be dangerous, please be sure you pick a number unique to your network.
# Addresses are 4byte hexadecimal (the leading 0x is required).

3   0x49a01010  1


# NETWORK DEVICE(S)
# This entry configures your IPX network. If you already have your
# IPX network configured then you do not need this. This is the same as
# using ipx_configure/ipx_interface before you start the server.
# Syntax:
#    4 <IPX Network Number> <device_name> <frametype> [<ticks>]
#                           Frame types: ethernet_ii, 802.2, 802.3, SNAP

4  0x39a01010  eth0  802.3  1

# SAVE IPX ROUTES AFTER SERVER IS DOWNED
# Syntax:
#    5 <flag>
#       0 = don't save routes, 1 = do save routes

5 0

# NETWARE VERSION
# Syntax:
#    6 <version>
#      0 = 2.15, 1 = 3.11

6 1

# PASSWORD HANDLING
# Real Novell DOS clients support a feature which encypts your
# password when changing it. You can select whether you want your
# mars server to support this feature or not.
# Syntax
#    7 <flag>
#    <flag> is:
#        0 to force password encryption. (Clients can't change password)
```

TOC

```
#        1 force password encryption, allow unencrypted password change.
#        7 allow non-encrypted password but no empty passwords.
#        8 allow non-encrypted password including empty passwords.
#        9 completely unencrypted passwords (doesn't work with OS/2)

7 1

# MINIMAL GID UID rights
# permissions used for attachments with no login. These permissions
# will be used for the files in your primary server attachment.
# Syntax:
#    10 <gid>
#    11 <uid>
#    <gid> <uid> are from /etc/passwd, /etc/groups

10  200
11  201

# SUPERVISOR password
# May be removed after the server is started once. The server will
# encrypt this information into the bindery file after it is run.
# You should avoid using the 'root' user and instead use another
# account to administer the mars fileserver.
#
# This entry is read and encrypted into the server bindery files, so
# it only needs to exist the first time you start the server to ensure
# that the password isn't stolen.
#
# Syntax:
#    12 <Supervisor-Login> <Unix username> [<password>]

12  SUPERVISOR  terry  secret

# USER ACCOUNTS
# This associates NetWare logins with unix accounts. Password are
# optional.
# Syntax:
     13 <User Login> <Unix Username> [<password>]

13  MARTIN martin
13  TERRY  terry

# LAZY SYSTEM ADMIN CONFIGURATION
# If you have a large numbers of users and could not be bothered using
# type 13 individual user mappings, you can automatically map mars_nwe
# logins to linux user names. BUT, there is currently no means of making
# use of the linux login password so all users configured this way are
# will use the single password supplied here. My recommendation is not
# to do this unless security is absolutely no concern to you.
# Syntax:
#    15 <flag> <common-password>
#    <flag> is: 0  - don't automatically map users.
#               1  - do automatically map users not configured above.
#               99 - automatically map every user in this way.
```

```
15  0  duzzenmatta

# SANITY CHECKING
# mars_nwe will automatically ensure that certain directories exist if
# you set this flag.
# Syntax:
#    16 <flag>
#    <flag> is 0 for no, don't, or 1 for yes, do.

16  0

# PRINT QUEUES
# This associates NetWare printers with unix printers. The queue
# directories must be created manually before printing is attempted.
# The queue directories are NOT lpd queues.
# Syntax:
#    21 <queue_name> <queue_directory> <unix_print_cmd>

21  EPSON  SYS:/PRINT/EPSON lpr -h
21  LASER  SYS:/PRINT/LASER lpr -Plaser

# DEBUG FLAGS
# These are not normally needed, but may be useful if are you debugging
# a problem.
# Syntax:
#    <debug_item> <debug_flag>
#
#    100 = IPX KERNEL
#    101 = NWSERV
#    102 = NCPSERV
#    103 = NWCONN
#    104 = start NWCLIENT
#    105 = NWBIND
#    106 = NWROUTED
#                  0 = disable debug, 1 = enable debug

100 0
101 0
102 0
103 0
104 0
105 0
106 0

# RUN NWSERV IN BACKGROUND AND USE LOGFILE
# Syntax:
#    200 <flag>
#        0 = run NWSERV in foreground and don't use logfile
#        1 = run NWSERV in background and use logfile

200  1

# LOGFILE NAME
# Syntax:
#    201 <logfile>
```

```
201  /tmp/nw.log

# APPEND LOG OR OVERWRITE
# Syntax:
#    202 <flag>
#        0 = append to existing logfile
#        1 = overwrite existing logfile

202  1

# SERVER DOWN TIME
# This item sets the time after a SERVER DOWN is issued that the
# server really goes down.
# Syntax:
#    210 <time>
#        in seconds. (defaults 10)

210  10

# ROUTING BROADCAST INTERVAL
# The time is seconds between server broadcasts
# Syntax:
#    211 <time>
#        in seconds. (defaults 60)

211  60

# ROUTING LOGGING INTERVAL
# Set how many broadcasts take place before logging of routing
# information occurs.
# Syntax:
#    300  <number>

300  5

# ROUTING LOGFILE
# Set the name of the routing logfile
# Syntax:
#    301 <filename>

301  /tmp/nw.routes

# ROUTING APPEND/OVERWRITE
# Set whether you want to append to an existing log file or
# overwrite it.
# Syntax:
#    302 <flag>
#        <flag> is 0 for append, 1 for create/overwrite

302  1

# WATCHDOG TIMING
# Set the timing for watchdog messages that ensure the network is
# still alive.
```

```
# Syntax:
#    310 <value>
#        <value> =   0 - always send watchdogs
#                  < 0 - (-ve) for disable watchdogs
#                  > 0 - send watchdogs when network traffic
#                        drops below 'n' ticks

310  7

# STATION FILE
# Set the filename for the stations file which determine which
# machines this fileserver will act as the primary fileserver for.
# The syntax of this file is described in the 'examples' directory
# of the source code.
# Syntax:
#    400 <filename>

400  /etc/nwserv.stations

# GET NEAREST FILESERVER HANDLING
# Set how SAP Get Nearest Fileserver Requests are handled.
# Syntax:
#    401 <flag>
#        <flag> is: 0 - disable 'Get Nearest Fileserver' requests.
#                   1 - The 'stations' file lists stations to be excluded.
#                   2 - The 'stations' file lists stations to be included.

401  2
```

Start the server

If you've configured the server to expect external programs to configure your network and/or provide the routing function then you should start those before starting the server. Presuming you have configured the server so that it will configure your interfaces for you and provide the routing services you need only issue the command:

```
# nwserv
```

Test the server

To test the server you should first try to attach and login from a NetWare client on your network. You then set a `CAPTURE` from the client and attempt a print. If both of these are successful then the server is working.

10.2 The *lwared* package.

Ales Dryak `<A.Dryak@sh.cvut.cz>` developed *lwared* to allow Linux to function as an NCP based fileserver.

Ales has called the package *lwared*, an abbreviation for *LinWare Daemon*.

Capability of *lwared*.

The *lwared* server is capable of providing a subset of the full function of the Novell NCP. It incorporates messaging but it does not provide any printing facilities at all. It does not currently work very well with either Windows95 or Windows NT clients. The *lwared* server relies on external programs to build and update the IPX routing and SAP tables. Misbehaving clients can cause the server to crash. Importantly, filename translation facilities have not been included.

The server does work for NETX and VLM NetWare shells.

TOC

Obtaining *lwared*

The *lwared* package can be built for any kernel newer than `1.2.0`, I recommend you use version `1.2.13` as no kernel patches are required if you do. Some of the IPX functionality has changed with the version `1.3.*` kernels and this means that patches are now required to make it work properly. Appropriate patches are included for the new kernels, so if you must use an alpha kernel you should still be able to get *lwared* to work properly for you.

You can obtain the *lwared* package by anonymous ftp from: klokan.sh.cvut.cz

or from:

sunsite.unc.edu or mirror sites. The current version at the time of writing was: `lwared-0.95.tar.gz`

Building *lwared*

Untar the *lwared*package

Something like:

```
# cd /usr/src
# tar xvpfz lwared-0.95.tar.gz
```

Build a kernel with Ethernet and IPX support

If you are using an alpha `1.3.*` kernel then you should try and use kernel version `1.3.17` or newer because the supplied patches were built against it. `1.3.*` kernels older than `1.3.17` will require hand patching to install. (*some information on how to do this is included in the `INSTALL` file in the package.*). To install the patches against a `1.3.17` kernel or newer you should try:

```
# make patch
```

After applying the patches if necessary, the next thing you need to do is ensure that your kernel has been built with IPX support enabled. In the `1.2.13` version kernel you need only ensure that you have answered `Y` to the question: '`The IPX protocol`' as illustrated:

```
 ...
 ...
Assume subnets are local (CONFIG_INET_SNARL) [y]
Disable NAGLE algorithm (normally enabled) (CONFIG_TCP_NAGLE_OFF) [n]
The IPX protocol (CONFIG_IPX) [n] y
*
* SCSI support
 ...
 ...
```

In newer kernels a similar process is adopted by the actual text of the prompt may have changed slightly.

You will also need to ensure that you include an appropriate driver for your Ethernet card. If you do not know how to do this then you should read the Ethernet-HOWTO.

You can then proceed to build your kernel. Make sure you remember to run *lilo* to install it when you have finished.

Compile and install *lwared*.

To compile *lwared* you should first check, edit if necessary, the `server/config.h` file. This file contains various settings that will govern the way your server will behave when it is running. The defaults are reasonable, though you might want to check that the directories specified for the log files and configuration files suit your system.

```
# make depend
```

```
# make
# make install
```

I found that the '`make depend`' complained about not finding the `float.h` file on my system but appeared to work anyway. I also found that when I tried compiling with gcc `2.6.3` I found I had to change the line:

```
#include <net/route.h>
```

to

```
#include <net/if_route.h>
```

in `lib/ipxkern.c` as this file changed name sometime.

The '`make install`' will attempt to install the server and routing daemon programs into your `/usr/sbin` directory, the *lwpasswd* program into your `/usr/bin` directory, the IPX utility programs will be installed into your `/sbin` directory and last but not least the manual pages will go into the `/usr/man` directory structure. If any of these locations are not suitable for your system then you should edit the relevant `Makefile` and change the target directories to suit.

Configuring and using *lwared*

Now the fun bit!

Configuring the IPX network

The first thing you must do is configure your Ethernet interfaces to support the IPX networks your server will support. To do this you will need to know the IPX network addresses for each of your LAN segments, which Ethernet device (`eth0`, `eth1` etc.) is on which segment, what frame type (`802.3`, `EtherII` etc.) each LAN segment uses and what Internal Network address your server should use (this is really needed if your server will service more than one LAN segment). A configuration for a server that is on two dis-similar segments with IPX network addresses `23a91300` and `23a91301` and internal network address `bdefaced` might look like:

```
# ipx_internal_net add BDEFACED 1
# ipx_interface add eth0 802.3 23a91300
# ipx_interface add eth1 etherii 23a91301
```

Start the routing daemons

The kernel software itself actually does the IPX packet forwarding as it does for IP, but the kernel requires additional programs to manage the routing table updates. In the case of IPX two daemons are needed and both are supplied with *lwared*: *ipxripd* manages the IPX routing information and *ipxsapd* manages the SAP information. To start the daemons you need only specify the location of where they should write their log messages:

```
# ipxripd /var/adm/ipxrip
# ipxsapd /var/adm/ipxsap
```

Configure the *lwared* server

There are two files that you must manually configure to allow user login to your *lwared* server. They are:

`/etc/lwpasswd`

This is where LinWare user account information is kept. The *lwpasswd* program is to keep it up to date. In its simplest form the `/etc/lwpasswd` file looks like:

```
ales:
terryd:
guest:
```

Its format is a simple list of login id followed by a ':' character and then the encrypted version of the login

passwd. A couple of important caveats here: No encrypted password means no password, LinWare users m have Linux accounts, that is any user you place in `/etc/lwpasswd` must also appear in `/etc/passwd` `root` is the only account that can change the password of another LinWare user. If you are logged in as `root` you can change the password of a LinWare user as this transcript demonstrates:

```
# lwpasswd rodg
Changing password for RODG
Enter new password:
Re-type new password:
Password changed.
```

`/etc/lwvtab`

This is the LinWare volume tables and it stores information about what directories should be made available to LinWare users (this file is similar in nature to the NFS `/etc/exports` file). A simple example of its format is as follows:

```
SYS             /lwfs/sys
DATA            /lwfs/data
HOME            /home
```

The format is simple: Volume name followed by whitespace followed by Linux directory to export. You must have at **least** an entry for the `SYS` volume for the server to start. If you intend your DOS based users to be able use your LinWare server as their primary server then you must install a standard `SYS` volume directory structure underneath the directory you export as your `SYS` volume. Since these files are proprietary and copyright to the Novell corporation you should have a license for these. If you users will be using a Novell fileserver as their primary server then this will not be necessary.

Start the *lwared* server.

tada!

```
# lwared
```

It is almost an anticlimax isn't it ? Ok so you've got a question, right? What is the fileserver name that is being advertised ? If you started the server as shown then the LinWare server name being advertised will be based on what is returned by the Linux *hostname*. If you'd like it to be something else then you can give the server the name when you start it, for example:

```
# lwared -nlinux00
```

would start the server with the name `linux00`.

Test the *lwared* server.

The very first thing to test is that your LinWare server appears in an *slist* from a DOS client on your network. The *slist* program is stored on the `SYS` volume of a Novell fileserver so you must do this from a machine that is already logged in somewhere. If this is not successful then check that *ipxsapd* and *lwared* are both running. If the *slist* is successful then you should try attaching to the server and mapping a volume:

```
C:> attach linux00/ales
 ...
 ...
C:> map l:=linux00/data:
C:> l:
```

You should then be able to treat the new map just like any other map. The file permissions you will have will be based on those allowed to the *linux* account that parallels your LinWare login.

---

11. Configuring your Linux machine as a Novell Print Client.

The *ncpfs* package includes two small programs that allow you to handle printing from you Linux machine to a printer attached to a Novell print server. The *nprint* command allows you to print to a file to a NetWare print queue. The *pqlist* command allows you the list the available print queues on a NetWare server.

To obtain and install these commands just follow the instructions relating to the NCP client described earlier.

Both commands require that you supply username and password so you might normally consider building some shell scripts to make the task of printing easier.

An example might look like:

```
# pqlist -S ACCT_FS01 -U guest -n
# nprint -S ACCT_FS01 -q LASER -U guest -n filename.txt
```

The login syntax is similar to the *ncpmount* command. The examples above assume that fileserver `ACCT_FS01` has a `guest` account with no password, that a print queue called `LASER` exists and that `guest` is allowed to print to it.

On my Linux boxen I have a short shell script for each Novell printer. This can then be used as a print filter to allow printing using the standard Linux spooler.

12. Configuring your Linux machine as a Novell Print Server.

A program to allow your Linux machine to act as a print server on a Netware network is included in the *ncpfs* package. For instructions on how to obtain and build, it follow the directions in the `Netware client' section above. Alternatively, support is included in the *mars_nwe* package.

12.1 Prerequisites

Configuration is quite straightforward but relies on you already having your printer configuration completed and working under Linux. This is covered in the Printing-HOWTO in some depth.

12.2 Configuration

When you have a working printer configuration, and you have built and installed the *pserver* utility then you need to add commands to start it into your `rc` files.

Exactly what command will use will depend on depend on exactly how you want it to operate, but in its simplest form something like the following will work:

```
# pserver -S ACCT_01 -U LASER -P secret -q LASERJET
```

This example asks the *pserver* utility to login in to the `ACCT_01` fileserver with username `LASER` and password `secret` and to take jobs from the `LASERJET` print queue. When an incoming print job is received it will use the default print command of *lpr* to feed the print job to the Linux print daemon. The print queue must already be defined on the fileserver and the username must have server priveliges for the queue.

You could if you wished use any Linux command to accept and print the print job. The `-c` argument allows you to specify the exact print command. For example:

```
# pserver -S ACCT_01 -U LASER -P secret -q LASERJET -c "lpr -Plaserjet"
```

would do exactly the same as the previous example except it would send the job to the `laserjet` *printcap* configuration instead of the default one.

13. An overview of the *ncpfs* user and adminstration commands

Recent versions of Volker's *ncpfs* package include a range of user and administration commands that you might want to use. The tools are built and installed as part of the *ncpfs* installation process, so if you haven't already, follow the instructions supplied in the Novell Client section above to build and install them.

Detailed information is available in the supplied *man* pages but a brief summary of the commands is as follows;

13.1 User commands.

ncopy

Network Copy - allows efficient file copies to be performed by using a Netware function rather than a copy across the network.

nprint

Network Print - allows you to print a file to a Netware print queue on a Netware server.

nsend

Network Send - allows you to send messages to other users on a Netware server.

nwbols

List Bindery Objects - allows you to list the bindery contents of a Netware server.

nwboprops

List Properties of a Bindery Object - allows you to the properties of a Netware bindery object.

nwbpset

Set Bindery Property - allows you to set the properties of a Netware bindery object.

nwbpvalues

Print Netware Bindery Objects Property Contents - allows you to print the contents of a Netware bindery property.

nwfsinfo

Fileserver Information - prints some summary information about a Netware server.

nwpasswd

Netware Password - allows you to change a Netware users password.

nwrights

Netware Rights - displays the rights associated with a particular file or directory.

nwuserlist

Userlist - lists the users currently logged into a Netware fileserver.

pqlist

Print Queue List - displays the contents of a Netware print queue.

slist

Server List - displays a list of know Netware fileserver.

13.2 Administration tools.

nwbocreate

Create a Bindery Object - allows you to create a Netware bindery object.

nwborm

Remove Bindery Object - allows you to delete a Netware bindery object.

nwbpadd

Add Bindery Property - allows you to set the value of an existing property of a Netware bindery object.

nwbpcreate

Create Bindery Property - allows you to create a new property for an existing Netware bindery object.

nwbprm

Remove Bindery Property - allows you to remove a property from a Netware bindery object.

nwgrant

Grant Trustee Rights - allows you to assign trustee rights to a directory on a Netware fileserver.

nwrevoke

Revoke Trustee Rights - allows you to remove trustee rights from a directory on a Netware fileserver.

---

14. Configuring PPP for IPX support.

New versions of the *pppd* PPP daemon for Linux have support that allows you to carry IPX packets across a PPP serial link. You need at least version `ppp-2.2.0d` of the daemon. See the PPP-HOWTO for details on where to find it. When you compile *pppd* you must ensure you enable the IPX support by adding the following two lines:

```
IPX_CHANGE = 1
USE_MS_DNS = 1
```

to: `/usr/src/linux/pppd-2.2.0f/pppd/Makefile.linux`.

The `IPX_CHANGE` is what configures the IPX support into PPP. The `USE_MS_DNS` define allows Microsoft Windows95 machines to do Name Lookups.

The real trick to getting it to work in knowing how to configure it.

There are many ways of doing this, but I'm only going to describe the two that I've received any information on. I've tried neither yet, so consider this section experimental, and if you get something to work, please let me know.

14.1 Configuring an IPX/PPP server.

The first thing you need to do is configure your Linux machine as an IP/PPP server. Don't panic! This isn't difficult. Again, follow the instructions in the PPP-HOWTO and you should be pretty much ok. When you have this done there are a couple of simple modifications you need to make to get IPX working over the same configuration.

First steps.

One of the first steps you must take is to configure your linux machine as an IPX router as described in the appropriate section earlier in this document. You won't need to use the *ipx_route* command for the `ppp` interface because *pppd* will configure these for you as it does for IP. When you have the *ipxd* daemon running it will automatically detect any new IPX interfaces and propogates routes for them. In this way your dialup hosts will be seen by other machines automatically when they connect.

Design.

When you are running as a server it will normally be your responsibility to assign network address to each of the PPP links when they are established. This is an important point, each PPP link will be an IPX network and will have a unique IPX network address. This means that you must decide how you will allocate addresses and what what they will be. A simple convention is to allocate one IPX network

TOC

address to each serial device that will support IPX/PPP. You could allocate IPX network addresses b on the login id of the connecting user, but I don't see any particularly good reason to do so.

I will assume that this is what you have done, and that there are two serial devices (modems) that we will use. The addresses I've assigned in this contrived example are:

```
device IPX Network Address
------ -------------------
ttyS0  0xABCDEF00
ttyS1  0xABCDEF01
```

Configure *pppd*.

Configure your `/etc/ppp/options.ttyS0` file as follows:

```
ipx-network 0xABCDEF00
ipx-node 2:0
ipxcp-accept-remote
```

and your `/etc/ppp/options.ttyS1` file as:

```
ipx-network 0xABCDEF01
ipx-node 3:0
ipxcp-accept-remote
```

These will ask *pppd* to allocate the appropriate IPX network addresses to the link when the link is established, set the local node number to `2` or `3` and will let the remote node overwrite what the remote node number with what it thinks it is. Note that each of the addresses are hexadecimal numbers and that `0x` is required at the start of the network address, but not required at the start of the node address.

There are other places this information could be configured. If you have only one dialin modem then an entry could go into the `/etc/ppp/options` file. Alternatively this information can be passed on the command line to *pppd*.

Test the server configuration.

To test the configuration you will need to have a client configuration that is known to work. When the caller dials in, logs in and *pppd* starts it will assign the network address, advise the client of the servers node number and negotiate the clients node number. When this has completed, and after *ipxd* has detected the new interface the client should be able to establish IPX connections to remote hosts.

14.2 Configuring an IPX/PPP client.

In a client configuration, whether or not you configure your Linux machine as an IPX router depends on whether you have a local LAN that you wish to act as an IPX router for. If you are a standalone machine connecting to an IPX/PPP dialin server then you won't need to run *ipxd*, but if you have a LAN and wish

all of the machines on the LAN to make use of the IPX/PPP route then you must configure and run i as described. This configuration is much simpler because you do not have multiple serial devices to configure.

Configuring *pppd*

The simplest configuration is one that allows the server to supply all of the IPX network configuration information. This configuration would be compatible with the server configuration described above.

Again you need to add some options to your `/etc/ppp/options` file, they are:

```
ipxcp-accept-network
ipxcp-accept-remote
ipxcp-accept-local
```

These options tell *pppd* to act completely passively and accept all of the configuration details from the server. You could supply default values here for servers that don't supply details by adding `ipx-network` and `ipx-node` entries similar to the server configuration.

Testing the IPX/PPP client.

To test the client you will need a known working server to dial into. After you have dialled in and pppd has run you should see the IPX details configured on your `ppp0` device when you run the *ifconfig* command and you should be able to use *ncpmount*.

I'm not sure whether you will have to manually add IPX routes so that you can reach distant fileserver or not. This seems likely. If anyone running this configuration could tell me I'd be grateful.

---

15. IPX tunnel over IP

Many of you will be in a situation where you have two Novell Local Area Netorks with only an IP connection between them. How do you play multiplayer deathmatch DOOM for DOS via this arrangement you might ask ? Andreas Godzina `<ag@agsc.han.de>` has an answer for you in the form of *ipxtunnel*.

ipxtunnel provides a bridge-like facility for IPX by allowing IPX packets to be encapsulated with tcp/ip datagrams so that they can be carried by a tcp/ip connection. It listens for IPX packets and when it hears one it wraps it within a tcp/ip datagram and routes it to a remote IP address that you specify. For this to work of course the machine that you route the encapsulated IPX must also be running a copy of the same version of *ipxtunnel* as you.

15.1 Obtaining *ipxtunnel*

You can obtain *ipxtunnel* from sunsite.unc.edu or mirror sites.

15.2 Building *ipxtunnel*

ipxtunnel built cleanly for me using the following commands:

```
# cd /usr/src
# tar xvfz .../ipxtunnel.tgz
# cd ipxtunnel
# make
```

15.3 Configuring *ipxtunnel*

Configuration for *ipxtunnel* is easy. Lets say that your friends machine is `gau.somewhere.com` and your machine is called `gim.sw.edu`. *ipxtunnel* uses a configuration file called `/etc/ipxtunnel.conf`. This file allows you to specify the default UDP port to use for the tcp/ip connection, where to send the encapsulated data and which of your local interfaces *ipxtunnel* should listen on and deliver IPX packets to.

A simple configuration file would look like the following:

```
#
# /etc/ipxtunnel.conf for gim.sw.edu
#
# The UDP port to use:                          (default 7666)
```

```
port 7777
#
# The remote machine to send IPX packets to: (no default)
remote gau.somewhere.com
#
# The local interfaces to listen for IPX on: (default eth0)
interface eth0
interface eth1
```

Obviously the other machine would have a similar configuration file specifying this machine as a `remote` host.

15.4 Testing and using *ipxtunnel*

ipxtunnel acts **like** an IPX bridge, so the IPX networks at either end of the link should probably be the same. Andreas has never tested the *ipxtunnel* in an environment that actually supports Novell file servers so if you do try this in a real environment let Andreas know if it works or not.

If the *ipxtunnel* is working you should be able to start your DOOM machines up at each end of the link running IPX mode and they should see each other.

Andreas has only used this code over good high speed lines and he makes no claim as to its performance when your link is low speed. Again, let him know what works for you and what doesn't.

---

16. Commercial IPX support for Linux.

16.1 Caldera'a Network Desktop

Caldera Inc., produce a Linux distribution that features a range of commercially supported enhancements including fully functional Novell NetWare client support. The base distribution is the well respected Red Hat Linux Distribution and Caldera have added their "Network Desktop" products to this. The NetWare support provides a fully featured Novell NetWare client built on technology licensed from Novell Corporation. The client provides full client access to Novell 3.x and 4.x fileservers and includes features such as NetWare Directory Service (NDS) and RSA encryption.

You can obtain much more information and ordering details from the: Caldera Inc Web Server.

If you work within a Netware 4.x and/or NDS environment then the Caldera Netware Client is the only solution available.

If you have a business critical application for Novell support for Linux then the Caldera product should be something you take a close look at.

17. Some Frequently Asked Questions

Where can I find commercially supported IPX software for Linux ?

The Caldera Corporation offers a fully licensed and fully supported Netware 3.x and 4.x client. You can obtain information about it from the Caldera Inc Web Server.

Does the IPX software work with Arcnet/Token Ring/etc. ?

The Linux IPX software does work with ArcNet and Token Ring interfaces. I haven't heard of anyone trying it with AX.25 yet. Configuration is the same as for configuring for ethernet except you will have to substitute appropriate device names in place of 'eth0' and appopriate hardware addresses where necessary.

How do I configure more than one IPX interface ?

If you have more than one interface in your machine you should use the *ipx_interface* command to manually configure each one, you should not use the `plug n play' configuration.

How do I choose IPX addresses ?

IPX networking is similar, but not identical to, IP networking. A major difference is the way that addresses are used. IPX does not use the concept of subnetworking and so the sort of associations that you have between network addresses and networks is different. The rules are fairly simple:

- Every IPX network address must be unique on a wide area network. This includes Internal Network Addresses. Many organisations using IPX over a wide area network will have some sort of addressing standard that you should follow.
- Every Host address on an individual network must be unique. This means that every host on each IPX network must have a uniquely assigned address. In the case of ethernet network this isn't difficult as the cards each have a unique address. In the case of IPX/PPP this means you must ensure that you allocate unique addresses to all hosts on the network, irrespective of which end of the link(s) they are connected. Host address do not need to be unique across a wide area network as the network address is used in combination with the host address to uniquely identify a host.

What are frame types, which should I use ?

There are a variety of frame types in use over which you can run IPX. The most common of these are described in the 'common terms' section of this document (under the ``Frame Type` entry').

If you are installing your machine on an existing network then you must use whatever is already in use to allow you to interwork with the other hosts on the network, but if the installation is a brand new network you can use any of a range of protocols to carry your IPX traffic. My recommendation if you are configuring a brand new network and you need to carry both IPX and IP traffic is to use the `Ethernet_II` frame type.

My Windows95 machines mess up my frame type autodetection ?

Apparently they can, yeah. I could make nasty comments, but instead I'll just suggest that you the manual frame type configuration instead of the automatic one. It is probably the better way anyway.

Why do I get the message `invalid argument' when I configure IPX ?

You are probably not running a kernel that supports IPX, either recompile your kernel so it does, or double check that you have actually used lilo to install and run the new kernel.

Why do I get the message `package not installed' when I configure IPX ?

You are probably not running a kernel that supports IPX, either recompile your kernel so it does, or double check that you have actually used lilo to install and run the new kernel.

Why do I get the message `IPX support not in kernel' from *pppd* ?

You've probably compiled IPX as a module and not ensured that it was loaded before started *pppd*.

How do I NFS export a mounted NCP filesystem ?

To use NFS to export an NCP filesystem you must mount it using the *ncpmount* `-V` option. This option allows you to mount only one volume of a fileserver instead of the usual mounting of all of them. When you do this your NFS daemon will allow you to export that filesystem in the usual way.

Why doesn't slist work when I have an internel network with mars_nwe ?

You must have the get nearest server enabled. That is, entry 401 in /etc/nwserv.conf should be 0 unless you have a reason for not responding to get nearest servers. If you just want slist to work and not respond to every get nearest server request, include your internal network and node number in /etc/nwserv.stations and set entry 401 in /etc/nwserv.conf to 2.

Does ncpfs package work with mars_nwe ?

Martin and Volker's code is slowly beginning to converge. Recent versions of *mars_nwe* have an option to enable it to work with *ncpfs*. You must enable the `WITH_NAME_SPACE_CALLS` in the *mars_nwe* `config.h` file.

Is there any free DOS software to work with mars_nwe ?

A contrived question deserves a contrived answer. I'm glad you asked, Martin has a package that he distributes alongside his *mars_nwe* package that offers free DOS client support for the *mars_nwe* server. You can find it at the same sites as the server, and it will be called `mars_dosutils-0.01.tgz`. It includes C source code for programs such as *slist.exe*, *login.exe*, *map.exe* etc. The source is compilable with Borland(tm) C.

---

18. Copyright Message.

The IPX-HOWTO, a guide to software supporting the IPX protocol for Linux. Copyright (c) 1995 Terry Dawson.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the:

Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

19. Miscellaneous and Acknowledgements.

Terry Dawson <terry@perf.no.itg.telstra.com.au> for the original document

David E. Storey <dave@tamos.gmu.edu> and Volker Lendecke <lendecke@namu01.gwdg.de> both assisted greatly by supplying me with information for this document. Gilbert Callaghan <gilbert@pokey.inviso.com>, David Higgins <dave@infra.com> and Chad Robinson <chadr@brtgate.brttech.com> each contributed information on configuring IPX/PPP. Bennie Venter <bjv@Gil-galad.paradigm-sa.com> contributed some useful information relating to frame types. Christopher Wall <vergil@idir.net contributed some useful suggestions to improve the readability and layout of the document. Axel Boldt <boldt@math.ucsb.edu> contributed some useful suggestions and feedback. Erik D. Olson <eriko@wrq.com> provided some useful feedback and information on configuring PPP for IPX. Brian King <root@brian.library.dal.ca> contributed a question for the FAQ section.

"NetWare" is a registered trademark of the Novell Corporation. "Caldera" is a registered trademark of the Caldera Corporation.

regards Kevin Thorpe.

<kevin@pricetrak.com>

Linux PPP HOWTO

Robert Hart, hartr@interweft.com.au

v3.0, 31 March 1997

This document shows how to connect your Linux PC to a PPP server, how to use PPP to link two LANs together and provides one method of setting up your Linux computer as a PPP server.The document also provides help in debugging non-functional PPP connections.

Copyright

This document is distributed under the terms of the GPL (GNU Public License).

Distribution

This document will be posted to comp.os.linux.answers as new versions of the document are produced. It is also available in HTML format at:-

- Linux Howto Index
- PPP-HOWTO

Other formats (SGML, ASCII, postscript, DVI) are available from Howtos - other formats.

As sunsite.unc.edu carries a very heavy load, please use an appropriate mirror site close to you.

Acknowledgements

A growing number of people have provided me with assistance in preparing this document. Special thanks go to Al Longyear for the guidance on PPP itself (if there are mistakes here, they are mine not his), Greg Hankins (maintainer of the Linux Howto system)and Debi Tackett (of MaximumAccess.com) for many helpful suggestions on style, content order, logic and clarity of explanations.

Finally, to the many people who have contacted me by email offering comments - my thanks. As with all HOWTO authors, the satisfaction of helping is all the payment we receive and it is enough. By writing this HOWTO I am repaying in a small way the debt I - and all other Linux users - owe to the people who write and maintain our OS of choice.

1. Introduction

- 1.1 Clients and Servers
- 1.2 Differences between Linux distributions

- 1.3 Distribution specific PPP configuration tools

2. IP Numbers

3. Aims of this Document

- 3.1 Setting up a PPP Client
- 3.2 Linking two LANs or a LAN to the Internet using PPP
- 3.3 Setting up a PPP server
- 3.4 Using PPP over a direct null modem connection
- 3.5 This document at present does NOT cover...

4. Software versions covered

5. Other Useful/Important Documents

- 5.1 Useful Linux Mailing Lists

6. Overview of what has to be done to get PPP working as a client

- 6.1 Obtaining/Installing the software
- 6.2 Compiling PPP support into the kernel
- 6.3 Obtaining information from your ISP
- 6.4 Configuring your modem and serial port
- 6.5 Setting up Name to Address Resolution (DNS)
- 6.6 PPP and root Privileges
- 6.7 Checking your distribution PPP Files and setting up the PPP Options
- 6.8 If your PPP server uses PAP (Password Authentication Protocol)
- 6.9 Connecting to the PPP server by hand
- 6.10 Automating your PPP Connection
- 6.11 Shutting down the link
- 6.12 If you have problems
- 6.13 After the link comes up

- 6.14 Problems with standard IP services on a Dynamic IP number PPP link

7. Configuring your Linux Kernel

- 7.1 Installing the Linux Kernel source
- 7.2 Knowing your hardware
- 7.3 Kernel compilation - the Linux 1.2.13 kernel
- 7.4 Kernel compilation - the Linux 1.3.x and 2.0.x kernels
- 7.5 Note on PPP-2.2 and `/proc/net/dev`
- 7.6 General kernel config considerations for PPP

8. Getting the Information you need about the PPP server

9. Configuring your modem and serial port

- 9.1 A note about serial ports and speed capabilities
- 9.2 Serial Port Names
- 9.3 Configuring your modem
- 9.4 Note on Serial Flow Control
- 9.5 Testing your modem for dial out

10. Setting up Name to Address Resolution (DNS)

- 10.1 The `/etc/resolv.conf` file
- 10.2 The `/etc/host.conf` file

11. Using PPP and root privileges

12. Setting up the PPP connection files

- 12.1 The supplied options.tpl file
- 12.2 What options should I use? (No PAP/CHAP)

TOC

13. If your PPP server uses PAP (Password Authentication

- 13.1 Using MSCHAP
- 13.2 The PAP/CHAP secrets file
- 13.3 The PAP secrets file
- 13.4 The CHAP secrets file
- 13.5 Handling multiple PAP-authenticated connections

14. Setting up the PPP connection manually

15. Automating your connections - Creating the connection scripts

- 15.1 Connection scripts for User name/Password Authentication
- 15.2 The ppp-on script
- 15.3 Editing the supplied PPP startup scripts
- 15.4 What a Chat script means...
- 15.5 A chat script for PAP/CHAP authenticated connections
- 15.6 The pppd `debug` and `file option_file` options

16. Testing your connection script

17. Shutting down the PPP link

18. Debugging

- 18.1 I have compiled PPP support into the kernel, but...
- 18.2 My modem connects but ppp never starts up
- 18.3 The syslog says "`serial line is not 8 bit clean`..."
- 18.4 Default route not set
- 18.5 Other Problems

TOC

19. Getting Help when totally stuck

20. Common Problems once the link is working

- 20.1 I can't see beyond the PPP server I connect to
- 20.2 I can send email, but not receive it
- 20.3 Why can't people finger, WWW, gopher, talk etc to my machine?

21. Using Internet services with Dynamic IP numbers

- 21.1 Setting up email
- 21.2 Setting Up a local Name server

22. Linking two networks using PPP

- 22.1 Setting up the IP numbers
- 22.2 Setting up the routing
- 22.3 Network security

23. After the link comes up - the `/etc/ppp/ip-up` script

- 23.1 Special routing
- 23.2 Handling email queues
- 23.3 A sample `/etc/ppp/ip-up` script
- 23.4 Handling email

24. Using `/etc/ppp/ip-down`

25. Routing issues on a LAN

- 25.1 Note on Security

26. Setting up a PPP server

- 26.1 Kernel compilation
- 26.2 Overview of the server system
- 26.3 Getting the software together
- 26.4 Setting up standard (shell access) dialup.
- 26.5 Setting up the PPP options files
- 26.6 Setting pppd up to allow users to (successfully) run it
- 26.7 Setting up the global alias for pppd

27. Using PPP across a null modem (direct serial) connection

---

1. Introduction

PPP (the Point to Point Protocol) is a mechanism for creating and running IP (the Internet Protocol) and other network protocols over a serial link - be that a direct serial connection (using a null-modem cable), over a telnet established link or a link made using modems and telephone lines (and of course using digital lines such as ISDN).

Using PPP, you can connect your Linux PC to a PPP server and access the resources of the network to which the server is connected (almost) as if you were directly connected to that network.

You can also set up your Linux PC as a PPP server, so that other computers can dial into your computer and access the resources on your local PC and/or network.

As PPP is a peer-to-peer system, you can also use PPP on two Linux PCs to link together two networks (or a local network to the Internet), creating a Wide Area Network (WAN).

One major difference between PPP and an Ethernet connection is of course speed - a standard Ethernet connection operates at 10 Mbs (Mega - million bits per second) maximum theoretical throughput, whereas an analogue modem operates at speeds up to 56 kbps (kilo - thousand bits per second).

Also, depending on the type of PPP connection, there may be some limitations in usage of some applications and services.

1.1 Clients and Servers

PPP is strictly a **peer to peer** protocol; there is (technically) no difference between the machine that dials in and the machine that is dialed into. However, for clarity's sake, it is useful to think in terms of **servers** and **clients**.

When you dial into a site to establish a PPP connection, you are a **client**. The machine to which you connect is the **server**.

When you are setting up a Linux box to receive and handle dial in PPP connections, you are setting up a PPP **server**.

Any Linux PC can be both a PPP server and client - even simultaneously if you have more than one serial port (and modem if necessary). As stated above, there is no real difference between clients and servers as far as PPP is concerned, once the connection is made.

This document refers to the machine that initiates the call (that dials in) as the **CLIENT**, whilst the machine that answers the telephone, checks the authentication of the dial in request (using user names, passwords and possibly other mechanisms) is referred to as the **SERVER**.

The use of PPP as a client to link one or more machines at a location into the Internet is, probably, the one in which most people are interested - that is using their Linux PC as a client.

TOC

The procedure described in this document will allow you to establish and automate your Internet connection.

This document will also give you guidance in setting up your Linux PC as a PPP **server** and in linking two LANs together (with full routing) using PPP (this is frequently characterised as establishing a WAN - wide area network - link).

1.2 Differences between Linux distributions

There are many different Linux distributions and they all have their own idiosyncrasies and ways of doing things.

In particular, there are two different ways a Linux (and Unix) computer actually starts up, configures its interfaces and so forth.

These are **BSD system initialisation** and **System V system initialisation**. If you dip into some of the Unix news groups, you will find occasional religious wars between proponents of these two systems. If that sort of thing amuses you, have fun burning bandwidth and join in!

Possibly the most widely used distributions are

- Slackware
  which uses BSD style system initialisation
- Red Hat (and its former associate Caldera)
  which use SysV system initialisation (although in a slightly modified form)
- Debian
  which uses SysV system initialisation

BSD style initialisation typically keeps its initialisation files in `/etc/...` and these files are:-

---

```
        /etc/rc
        /etc/rc.local
        /etc/rc.serial
                (and possibly other files)
```

Of recent times, some BSD system initialisation schemes use a `/etc/rc.d...` directory to hold the start up file rather than putting everything into `/etc`.

System V initialisation keeps its initialisation files in directories under `/etc/...` or `/etc/rc.d/...` and a number of subdirectories under there:-

---

```
drwxr-xr-x   2 root     root         1024 Jul  6 15:12 init.d
-rwxr-xr-x   1 root     root         1776 Feb  9 05:01 rc
```

TOC

```
-rwxr-xr-x   1 root     root          820 Jan  2  1996 rc.local
-rwxr-xr-x   1 root     root         2567 Jul  5 20:30 rc.sysinit
drwxr-xr-x   2 root     root         1024 Jul  6 15:12 rc0.d
drwxr-xr-x   2 root     root         1024 Jul  6 15:12 rc1.d
drwxr-xr-x   2 root     root         1024 Jul  6 15:12 rc2.d
drwxr-xr-x   2 root     root         1024 Jul 18 18:07 rc3.d
drwxr-xr-x   2 root     root         1024 May 27  1995 rc4.d
drwxr-xr-x   2 root     root         1024 Jul  6 15:12 rc5.d
drwxr-xr-x   2 root     root         1024 Jul  6 15:12 rc6.d
```

If you are trying to track down where your Ethernet interface and associated network routes are actually configured, you will need to track through these files to actually find where the commands are that do this.

1.3 Distribution specific PPP configuration tools

On some installations (for example Red Hat and Caldera), there is a X Windows configured PPP dial up system. This HOWTO does not cover these distribution specific tools. If you are having problems with them, contact the distributors directly!

For Red Hat 4.x users, there is now a Red Hat PPP-TIP in the Linux resources area and also from Red Hat Software in the support area.

---

2. IP Numbers

Every device that connects to the Internet must have its own, unique IP number. These are assigned centrally by a designated authority for each country.

If you are connecting a local area network (LAN) to the Internet, **YOU MUST** use an IP number from your own assigned network range for all the computers and devices you have on your LAN. You **MUST NOT** pick IP numbers out of the air and use these whilst connecting to another LAN (let alone the Internet). At worst this will simply not work at all and could cause total havoc as your 'stolen' IP number starts interfering with the communications of another computer that is already using the IP number you have picked out of the air.

Please note that the IP numbers used throughout this document (with some exceptions) are from the 'unconnected network numbers' series that are reserved for use by networks that are not (ever) connected to the Internet.

There are IP numbers that are specifically dedicated to LANs that do not connect to the Internet. The IP number sequences are:-

- One A Class Network Address
  10.0.0.0 (netmask 255.0.0.0)
- 16 B Class Network Addresses
  172.16.0.0 - 172.31.0.0 (netmask 255.255.0.0)
- 256 C Class Network Addresses
  192.168.0.0 - 192.168.255.0 (netmask 255.255.255.0)

If you have a LAN for which you have **not** been allocated IP numbers by the responsible authority in your country, you should use one of the network numbers from the above sequences for your machines.

These numbers should **never** be used on the Internet.

However, they can be used for the local Ethernet on a machine that is connecting to the Internet. This is because IP numbers are actually allocated to a network interface, not to a computer. So whilst your Ethernet interface may use 10.0.0.1 (for example), when you hook onto the Internet using PPP, your PPP interface will be given another (and valid) IP number by the server. Your PC will have Internet connectivity, but the other computers on your LAN will not.

However, using Linux and the IP Masquerade (also known as NAT - Network address Translation) capabilities of the Linux and the ipfwadm software, you can connect your LAN to the Internet (with some restriction of services), even if you do not have valid IP numbers for the machines on your Ethernet.

For more information on how to do this see the IP Masquerade mini-HOWTO at Linux IP Masquerade mini HOWTO

TOC

For most users, who are connecting a single machine to an Internet service provider via PPP, obtaini an IP number (or more accurately, a network number) will not be necessary.

If you wish to connect a small LAN to the Internet, many Internet Service Providers (ISPs) can provide you with a dedicated subnet (a specific sequence of IP numbers) from their existing IP address space. Alternatively, use IP Masquerading.

For users, who are connecting a single PC to the Internet via an ISP, most providers use **dynamic** IP number assignment. That is, as part of the connection process, the PPP service you contact will tell your machine what IP number to use for the PPP interface during the current session. This number will not be the same every time you connect to your ISP.

With dynamic IP numbers, you are **not** given the same IP number each time you connect. This has implications for server type applications on your Linux machine such as sendmail, ftpd, httpd and so forth. These services are based on the premise that the computer offering the service is accessible at the same IP number all the time (or at least the same fully qualified domain name - FQDN - and that DNS resolution of the name to IP address is available).

The limitations of service due to dynamic IP number assignment (and ways to work around these, where possible) are discussed later in the document.

---

3. Aims of this Document

3.1 Setting up a PPP Client

This document provides guidance to people who wish to use Linux and PPP to dial into a PPP server and set up an IP connection using PPP. It assumes that PPP has been compiled and installed on your Linux machine (but does briefly cover reconfiguring/recompiling your kernel to include PPP support).

Whilst DIP (the standard way of creating a SLIP connection) can be used to set up a PPP connection, DIP scripts are generally quite complex. For this reason, this document does NOT cover using DIP to set up a PPP connection.

Instead, this document describes the standard Linux PPP software (chat/pppd).

3.2 Linking two LANs or a LAN to the Internet using PPP

This document provides (basic) information on linking two LANs or a LAN to the Internet using PPP.

3.3 Setting up a PPP server

This document provides guidance on how to configure your Linux PC as a PPP server (allowing other people to dial into your Linux PC and establish a PPP connection).

You should note that there are a myriad of ways of setting up Linux as a PPP server. This document gives one method - that used by the author to set up several small PPP servers (each of 16 modems).

This method is known to work well. However, it is not necessarily the best method.

3.4 Using PPP over a direct null modem connection

This document provides a brief overview of using PPP to link two Linux PCs via a null modem cable. It is possible to link other OS's to Linux this way as well. To do so, you will need to consult the documentation for the operating system you are interested in.

3.5 This document at present does NOT cover...

- Compiling the PPP daemon software
  See the documentation that comes with the version of pppd you are using.

TOC

- Connecting and configuring a modem to Linux (in detail)
  See the Serial-HOWTO and for modem specific initialisation, see Modem Setup Information f
  information that may help you to configure your modem.
- Using DIP to make PPP connections
  Use chat instead...
- Using socks or IP Masquerade
  There are perfectly good documents already covering these two packages.
- Using `diald` to set up an automated connection
  See the `diald` documentation for information on this.
- Using EQL to gang together two modems into a single PPP link.
- Distribution specific PPP connection methods (such as the Red Hat 4.x network configuration tool.
  See the distribution for documentation on the methods used.
- The growing number of tools available to automate PPP set up
  See the appropriate documentation.

---

4. Software versions covered

This HOWTO assumes that you are using a Linux 1.2.x kernel with the PPP 2.1.2 software or Linux 1.3.X/2.0.x and PPP 2.2.

At the time of writing, the latest official version of PPP available for Linux is ppp-2.2f. The new version (ppp-2.3) is still in beta.

It is possible to use PPP 2.2.0 with kernel 1.2.13. This requires kernel patches. It is recommended that version 1.2.13 kernel users move up to ppp-2.2 as it includes several bug fixes and enhancements.

Also, you should particularly note that you cannot use the PPP 2.1.2 software with Linux kernel version 2.0.X.

Please note that this document does **NOT** cover problems arising from the use of loadable modules for Linux kernel 2.0.x. Please see the kerneld mini-HOWTO and the kernel/module 2.0.x documentation (in the Linux 2.0.x source tree at `/usr/src/linux/Documentation/...`).

As this document is designed to assist new users, it is highly recommended that you use a version of the Linux kernel and the appropriate PPP version that are known to be stable together.

5. Other Useful/Important Documents

Users are advised to read :-

- the documentation that comes with the PPP package;
- the pppd and chat man pages;
  (use `man chat` and `man pppd` to explore these)
- the Linux Network Administration Guide (NAG);
  see The Network Administrators' Guide
- the Net-2/3 HOWTO;
  see Linux NET-2/3-HOWTO
- Linux kernel documentation installed in `/usr/src/linux/Documentation` when you install the Linux source code;
- The modem setup information page - see Modem Setup Information
- The excellent Unix/Linux books published by O'Reilly and Associates. See (O'Reilly and Associates On-Line Catalogue). If you are new to Unix/Linux, **run** (don't walk) to your nearest computer book shop and invest in a number of these immediately!
- The PPP-FAQ maintained by Al Longyear, available from Linux PPP-FAQ.
  This contains a great deal of useful information in question/answer format that is very useful when working out why PPP is not working (properly).
- The growing number of Linux books from various publishing houses and authors;
  You are actively encouraged to check the currency of these books. Linux development and distributions tend to evolve fairly rapidly, whilst the revision of books move (generally) much more slowly! Buying an excellent book (and there are many) that is now out of date will cause new users considerable confusion and frustration.

The best general starting point for Linux documentation is The Linux Documentation Project Home Page. The HOWTO's tend to be revised reasonably regularly.

Whilst you can use this document to create your PPP link without reading any of these documents, you will have a far better understanding of what is going on if you do so! You will also be able to address problems yourself (or at least ask more intelligent questions on the comp.os.linux... newsgroups or Linux mailing lists).

These documents (as well as various others, including the relevant RFCs) provide additional and more detailed explanation than is possible in this HOWTO.

If you are connecting a LAN to the Internet using PPP, you will need to know a reasonable amount about TCP/IP networking. In addition to the documents above, you will find the O'Reilly books "TCP/IP Network Administration" and "Building Internet Firewalls" of considerable benefit!

5.1 Useful Linux Mailing Lists

TOC

There are many Linux mailing lists that operate as a means of communication between users of many levels of ability. By all means subscribe to those that interest you and contribute your expertise and views.

A word to the wise: some lists are specifically aimed at "high powered" users and/or specific topics. Whilst no-one will complain if you 'lurk' (subscribe but don't post messages), you are likely to earn heated comments (if not outright flames) if you post 'newbie' questions to inappropriate lists.

This is not because guru level users hate new users, but because these lists are there to handle the specific issues at particular levels of difficulty.

By all means join the lists that offer open subscription, but keep your comments relevant to the subject of the list!

A good starting point for Linux mailing lists is Linux Mailing List Directory

---

6. Overview of what has to be done to get PPP working as a client

This document contains a great deal of information - and with each version it grows!

As a consequence, this section aims to provide a concise overview of the actions you will need to take to get your Linux system connected as a client to a PPP server.

6.1 Obtaining/Installing the software

If your Linux distribution does not include the PPP software, you will need to obtain this from the Linux PPP daemon.

This is the latest official version at the time of writing. However, choose the latest version available from this site (ppp-2.3 is in beta at the time of writing and should be released soon).

The PPP package contains instructions on how to compile and install the software **so this HOWTO does not**!

6.2 Compiling PPP support into the kernel

Linux PPP operations come in two parts

- the PPP daemon mentioned above
- kernel support for PPP

Many distributions seem to provide PPP kernel support in their default installation kernels, but others do not.

If at boot your kernel reports messages like

```
PPP Dynamic channel allocation code copyright 1995 Caldera, Inc.
PPP line discipline registered.
```

---

your kernel does have PPP support compiled in.

That said, you will probably want to compile your own kernel whatever your distribution to provide the most efficient use of system resources given your particular hardware configuration. It is worth remembering that the kernel cannot be swapped out of memory and so keeping the kernel as small as possible has advantages on a memory limited machine.

TOC

This document provides minimal kernel re-compilation instructions at section Configuring your Linu Kernel.

For greater detail, see the Kernel-HOWTO at The Linux Kernel HOWTO

6.3 Obtaining information from your ISP

There are an almost infinite number of ways in which a PPP server can be set up. In order to connect to your ISP (or corporate PPP server to access your intranet), you will need to obtain information on how the PPP server operates.

Because you are using Linux, you may have some difficulty with some ISP help desks (and work site based PPP intranet servers) which know only about MS Windows clients.

However, a rapidly growing number of ISPs use Linux to provide their service - and Linux is also penetrating the corporate environment as well, so you may be lucky if you do strike problems.

Section Getting the Information you need about the PPP server tells you what you need to know about the PPP server to which you are going to connect - and how to find out the information you need to know.

6.4 Configuring your modem and serial port

In order to connect to a PPP server and to obtain the best possible data transfer rate, your modem needs to be configured correctly.

Similarly, the serial ports on your modem and computer need to be set up correctly.

Section Configuring your modem and serial port provides information on this.

6.5 Setting up Name to Address Resolution (DNS)

In addition to the files that run PPP and perform the automated log in to the PPP server, there are a number of text configuration files that have to be set up for your computer to be able to resolve names like `www.interweft.com.au` to the IP address that is actually used to contact that computer. These are:-

- `/etc/resolv.conf`
- `/etc/host.conf`

Section Setting up Name to Address Resolution for details on setting this up.

In particular, you do **NOT** need to run a name server on your Linux PC in order to connect to the Internet (although you may wish to). All you need is to know the IP number of at least one name server that you can use (preferably one at your ISPs site).

TOC

6.6 PPP and root Privileges

As establishing a PPP link between you Linux computer and another PPP server requires manipulation of network devices (the PPP interface is a network interface) and the kernel routing table, pppd requires root privileges.

For details on this, see section Using PPP and root privileges.

6.7 Checking your distribution PPP Files and setting up the PPP Options

There are a number of configuration and dialer files that need to be set up to get PPP operational. There are examples as part of the PPP distribution and this section shows what files you should have:-

---

```
/etc/ppp/options
/etc/ppp/scripts/ppp-on
/etc/ppp/scripts/ppp-on-dialer
/etc/ppp/options.tpl
```

You may need to create some additional files depending on exactly what you are aiming to achieve with PPP:-

---

```
/etc/ppp/options.ttyXX
/etc/ppp/ip-up
/etc/ppp/pap-secrets
/etc/ppp/chap-secrets
```

In addition, the PPP daemon can use a large number of command line options and it is important to use the right ones; so this section takes you through the standard PPP options and helps you choose the options you should use.

For details on this, see Setting up the PPP connection files.

6.8 If your PPP server uses PAP (Password Authentication Protocol)

Many ISPs and corporate PPP servers use PAP. If your server does **not** require you to use PAP (if you can log in manually and receive the standard user name/password text based prompts it does not use PAP), you can safely ignore this section.

Instead of logging into such a server using a user name and password when prompted to enter them b the server, a PPP server using PAP does not require a text based login.

The user authentication information instead is exchanged as part of the link control protocol (LCP) which is the first part of establishing a PPP link.

Section If your PPP server uses PAP (Password Authentication Protocol) provides information on the files you need to set up to establish a PPP link using PAP.

6.9 Connecting to the PPP server by hand

Having set up the basic files, it is a good idea to test these by connecting (using minicom or seyon) and starting pppd on your Linux PC by hand.

See Section Setting up the PPP connection manually for full details of setting this up.

6.10 Automating your PPP Connection

Once you are able to log in by hand, you can now move to setting up a set of scripts that will automate the establishment of the connection.

Section Automating your connections - Creating the connection scripts covers setting up the necessary scripts, with considerable attention paid to `chat` and scripting the login process to the PPP server.

This section discusses scripts for user name/password authentication as well as scripts for PAP/CHAP authenticating servers.

6.11 Shutting down the link

Once your link is up and working, you need to be able to deactivate the link.

This is covered in Section Shutting down the PPP link.

6.12 If you have problems

Many people have problems getting PPP to work straight away. The variation in PPP servers and how they require you to set up the connection is enormous. Similarly, there are many options to PPP - and some combinations of these just do not work together, ever.

In addition to the problems of logging in and starting the PPP service, there are problems with the modems and the actual telephone lines as well!

Section Fixing problems provides some basic information about common errors, how to isolate these and fix them.

This is **NOT** intended to provide more than just the basics. Al Longyear maintains the PPP-FAQ wh contains much more information on this topic!

6.13 After the link comes up

Once a PPP link is operational (specifically, once the IP layer is operational), Linux PPP can automatically run (as the root user), a script to perform **any** function you can write a script to accomplish.

Section After the link comes up provides information on the `/etc/ppp/ip-up` script, the parameters it receives from PPP and how to use it to do things like acquire your email from your ISP account, send any queued email waiting transmission on your machine and such.

6.14 Problems with standard IP services on a Dynamic IP number PPP link

As noted in the introduction, dynamic IP numbers affect the ability of your Linux PC to act as a server on the Internet.

Section Problems with standard IP services on a Dynamic IP number PPP link provides information on the (main) services affected and what you can do (if anything) to overcome this.

---

7. Configuring your Linux Kernel

In order to use PPP, your Linux kernel must be compiled to include PPP support. Obtain the Linux source code for your kernel if you do not already have this - it belongs in `/usr/src/linux` on Linux's standard file system.

Check out this directory - many Linux distributions install the source tree (the files and subdirectories) as part of their installation process.

At bootup, your Linux kernel prints out a great deal of information. Amongst this is information about PPP support if the kernel includes this. To view this information, look at your syslog file or use `dmesg | less` to display the information to the screen. If your kernel includes PPP support, you will see lines like

---

```
PPP Dynamic channel allocation code copyright 1995 Caldera, Inc.
PPP line discipline registered.
```

(this is for the Linux 2.0.x kernel series).

Linux kernel sources can be obtained by ftp from `sunsite.unc.edu` or its mirror sites.

7.1 Installing the Linux Kernel source

The following are brief instructions for obtaining and installing the Linux kernel sources. Full information can be obtained from The Linux Kernel HOWTO.

In order to install and compile the Linux kernel, you need to be logged in as root.

1. Change directory to the `/usr/src` directory
   `cd /usr/src`
2. Check in `/usr/src/linux` to see if you already have the sources installed.
3. If you don't have the sources, get them from Linux kernel source directory or your nearest mirror.
   If you are looking for earlier versions of the kernel (such as 1.2.X), these are kept in Old Linux kernel source directory.
4. Choose the appropriate kernel - usually the most recent one available is what you are looking for. Retrieve this and put the source tar file in `/usr/src`.
   Note: a 'tar' file is an archive - possibly compressed (as are the Linux kernel source tar files) containing many files in a number of directories. It is the Linux equivalent of a DOS multi-directory zip file.
5. If you already have the Linux sources installed but are upgrading to a new kernel, you must remove the old sources. Use the command
   `rm -rf /usr/src/linux`
6. Now uncompress and extract the sources using the command
   `tar xzf linux-2.0.XX.tar.gz`
7. Now, `cd /usr/src/linux` and read the README file. This contains an excellent explanation of how

to go about configuring and compiling a new kernel. Read this file (it's a good idea to print it out and l a copy handy whilst you are compiling until you have done this enough times to know your way arou

7.2 Knowing your hardware

You **MUST** know what cards/devices you have inside your PC if you are going to recompile your kernel!!! For some devices (such as sound cards) you will also need to know various settings (such as IRQ's, I/O addresses and such).

7.3 Kernel compilation - the Linux 1.2.13 kernel

To start the configuration process, follow the instructions in the README file to properly install the sources. You start the kernel configuration process with

```
make config
```

In order to use PPP, you must configure the kernel to include PPP support (PPP requires BOTH pppd AND kernel support for PPP).

```
PPP (point-to-point) support (CONFIG_PPP) [n] y
```

Answer the other make config questions according to the hardware in your PC and the features of the Linux operating system you want. Then continue to follow the README to compile and install your new kernel.

The 1.2.13 kernel creates only 4 PPP devices. For multi- port serial cards, you will need to edit the kernel PPP sources to obtain more ports. (See the README.linux file that comes as part of the PPP-2.1.2 distribution for full details of the simple edits you need to make).

Note: the 1.2.13 configuration dialogue does NOT allow you to go backwards - so if you make a mistake in answering one of the questions in the `make config` dialogue, exit by typing CTRL C and start again.

7.4 Kernel compilation - the Linux 1.3.x and 2.0.x kernels

For Linux 1.3.x and 2.0.x, you can use a similar process as for Linux 1.2.13. Again, follow the instructions in the README file to properly install the sources. You start the kernel configuration process with

```
make config
```

However, you also have the choice of

```
make menuconfig
```

This provides a menu based configuration system with online help that allows you to move backwards and forwards in the configuration process.

There is also a highly recommended X windows based configuration interface

```
make xconfig
```

You can compile PPP support directly into your kernel or as a loadable module.

If you only use PPP some of the time that your Linux machine is operating, then compiling PPP support as a loadable module is recommended. Using 'kerneld', your kernel will automatically load the module(s) required to provide PPP support when you start your PPP link process. This saves valuable memory space: no part of the kernel can be swapped out of memory, but loadable modules are automatically removed if they are not in use.

To do this, you need to enable loadable module support:-

```
Enable loadable module support (CONFIG_MODULES) [Y/n/?] y
```

To add PPP kernel support, answer the following question:-

```
PPP (point-to-point) support (CONFIG_PPP) [M/n/y/?]
```

For a PPP loadable module, answer **M**, otherwise for PPP compiled in as part of the kernel, answer **Y**.

Unlike kernel 1.2.13, kernel 2.0.x creates PPP devices on the fly as needed and it is not necessary to hack the sources to increase available PPP device numbers at all.

7.5 Note on PPP-2.2 and `/proc/net/dev`

If you are using PPP-2.2, you will find that a side effect of the 'on the fly' creation of the PPP devices is that no devices show up if you look in the `/proc/net` file system until a device is created by starting up pppd:-

```
[hartr@archenland hartr]$ cat /proc/net/dev
Inter-|   Receive                   |  Transmit
 face |packets errs drop fifo frame|packets errs drop fifo colls carrier
    lo:  92792    0    0    0    0    92792    0    0    0     0    0
  eth0: 621737   13   13    0   23   501621    0    0    0  1309    0
```

Once you have one (or more) ppp services started, you will see entries such as this (from a ppp server):-

```
[root@kepler /root]# cat /proc/net/dev
Inter-|   Receive                   |  Transmit
 face |packets errs drop fifo frame|packets errs drop fifo colls carrier
    lo: 428021    0    0    0    0   428021    0    0    0     0    0
  eth0:4788257  648  648  319  650  1423836    0    0    0  4623    5
  ppp0:   2103    3    3    0    0     2017    0    0    0     0    0
```

```
ppp1:  10008    0    0    0    0     8782    0    0    0     0    0
ppp2:    305    0    0    0    0      297    0    0    0     0    0
ppp3:   6720    7    7    0    0     7498    0    0    0     0    0
ppp4: 118231  725  725    0    0   117791    0    0    0     0    0
ppp5:  38915    5    5    0    0    28309    0    0    0     0    0
```

7.6 General kernel config considerations for PPP

If you are setting up your Linux PC as a PPP server, you must compile in IP forwarding support. This is also necessary if you want to use Linux to link to LANs together or your LAN to the Internet.

If you are linking a LAN to the Internet (or linking together two LANs), you should be concerned about security. Adding support for IP fire walls to the kernel is probably a MUST!

You will also need this if you want to use IP masquerade to connect a LAN that uses any of the above mentioned 'unconnected' IP network numbers.

To enable IP Masquerade and IP fire walling, you **MUST** answer yes to the first question in the `make config` process:-

```
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL)?
```

Whilst this may sound a bit off-putting to new users, many users are actively using the IP Masquerade and IP fire walling features of the Linux 2.0.XX kernel with no problems.

Once you have installed and rebooted your new kernel, you can start configuring and testing your PPP link(s).

8. Getting the Information you need about the PPP server

Before you can establish a PPP connection with a server, you need to obtain the following information (from the sysadmin/user support people of the PPP server):-

- The telephone number(s) to dial for the service
  If you are behind a PABX, you also need the PABX number that gives you an outside dial tone - this is frequently digit zero (0) or nine (9).
- Does the server use DYNAMIC or STATIC IP numbers?
  If the server uses STATIC IP numbers, then you may need to know what IP number to use for your end of the PPP connection. If your ISP is providing you with a subnet of valid IP numbers, you will need to know the IP numbers you can use and the network mask (netmask).
  Most Internet Service Providers use DYNAMIC IP numbers. As mentioned above, this has some implications in terms of the services you can use.
  However, even if you are using STATIC IP numbers, most PPP servers will never (for security reasons) allow the client to specify an IP number as this is a security risk. You **do** still need to know this information!
- What are the IP numbers of the ISPs Domain Name Servers?
  There should be at least two although only one is needed.
  There could be a problem here. The MS Windows 95 PPP setup allows the DNS address to be passed to the client as part of its connection process. So your ISP (or corporate help desk) may well tell you you don't need the IP address of the DNS server(s).
  For Linux, you **DO** need the address of at least one DNS. The linux implementation of PPP does not allow the setting of the DNS IP number dynamically at connection time - and quite possibly will never do so.
  Note: whilst Linux (as a PPP client) cannot accept the DNS address from a server, it can, when acting as a server, pass this information to clients using the `dns-addr` pppd option.
- Does the server require the use of PAP/CHAP?
  If this is the case you need to know the "id" and "secret" you are to use in connecting. (These are probably your user name and password at your ISP).
- Does the server automatically start PPP or do you need to issue any commands to start PPP on the server once you are logged in?
  If you must issue a command to start PPP, what is it?
- Is the server a Microsoft Windows NT system and, if so, is it using the MS PAP/CHAP system?
  Many corporate LANs seem to use MS Windows NT this way for increased security.

Carefully note down this information - you are going to use it!

9. Configuring your modem and serial port

You should make sure that your modem is correctly set up and that you know which serial port it is connected to.

Remember:-

- DOS com1: = Linux /dev/cua0 (and /dev/ttyS0)
- DOS com2: = Linux /dev/cua1 (and /dev/ttyS1)
  et cetera

It is also worth remembering that if you have 4 serial ports, the standard PC set up is to have com1 and com3 share IRQ4 and com2 and com4 share IRQ3.

If you have devices on standard serial ports that share an IRQ with your modem you are going to have problems. You need to make sure that your modem serial port is on its own, unique IRQ. Many modern serial cards (and better quality motherboard serial ports) allow you to move the IRQ of the serial ports around.

If you are running Linux kernel 2, you can check the in-use IRQs using `cat /proc/interrupts`, which will produce output like

```
 0:    6766283   timer
 1:      91545   keyboard
 2:          0   cascade
 4:     156944 + serial
 7:     101764   WD8013
10:     134365 + BusLogic BT-958
13:          1   math error
15:    3671702 + serial
```

This shows a serial port on IRQ4 (a mouse) and a serial port on IRQ15 (the permanent modem based PPP link to the Internet. (There is also a serial port on com2, IRQ3 and com4 is on IRQ14, but as they are not in use, they do not show up).

Be warned - you need to know what you are doing if you are going to play with your IRQs! Not only do you have to open up you computer, pull out cards and play with jumpers, but you need to know what is on which IRQ. In my case, this is a totally SCSI based PC, and so I can disable the on motherboard IDE interfaces that normally use IRQ14 and 15!

You should also remember that if your PC boots other operating systems, moving IRQs around may well mean that OS cannot boot properly - or at all!

TOC

If you do move your serial ports to non-standard IRQs, then you need to tell Linux which IRQ each I is using. This is done using `setserial` and is best done as part of the boot process in `rc.local` `rc.serial` which is called from `rc.local` or as part of the SysV initialisation. For the machine illustrated above, the commands used are

```
/bin/setserial -b /dev/ttyS2 IRQ 11
/bin/setserial -b /dev/ttyS3 IRQ 15
```

However, if you are using serial modules dynamically loaded when required by the `kerneld` process, you cannot set and forget the IRQ etc once at boot time. This is because if the serial module is unloaded, Linux forgets the special settings.

So, if you are loading the serial module on demand, you will need to reconfigure the IRQs etc each time the module is loaded.

9.1 A note about serial ports and speed capabilities

If you are using a high speed (external) modem (14,400 Baud or above), your serial port needs to be capable of handling the throughput that such a modem is capable of producing, particularly when the modems are compressing the data.

This requires your serial port to use a modern UART (Universal Asynchronous Receiver Transmitter) such as a 16550(A). If you are using an old machine (or old serial card), it is quite possible that your serial port has only an 8250 UART, which will cause you considerable problems when used with a high speed modem.

Use the command

```
setserial -a /dev/ttySx
```

to get Linux to report to you the type of UART you have. If you do not have a 16550A type UART, invest in a new serial card (available for under $50). When you purchase a new card, make sure you can move the IRQs around on it!

Note: the first versions of the 16550 UART chip had an error. This was rapidly discovered and a revision of the chip was released - the 16550A UART. A relatively small number of the faulty chips did however get into circulation. It is unlikely that you will encounter one of these but you should look for a response that says 16550A, particularly on serial cards of some vintage.

9.2 Serial Port Names

Historically, Linux used `cuaX` devices for dial out and `ttySx` devices for dial in.

The kernel code that required this was changed in kernel version 2.0.x and you should now use `ttySx` for both dial in and dial out. I understand that the `cuaX` device names may well disappear in future

kernel versions.

9.3 Configuring your modem

You will need to configure your modem correctly for PPP - to do this **READ YOUR MODEM MANUAL**! Most modems come with a **factory default setting** that selects the options required for PPP. The minimum configuration specifies:-

- Hardware flow control (RTS/CTS) (&K3 on many Hayes modems)

Other settings (in standard Hayes commands) you should investigate are:-

- E1 Command/usr/src/linux-2.0.27/include/linux/serial.h Echo ON (required for chat to operate)
- Q0 Report result codes (required for chat to operate)
- S0=0 Auto Answer OFF (unless you want your modem to answer the phone)
- &C1 Carrier Detect ON only after connect
- &S0 Data Set Ready (DSR) always ON
- (depends) Data Terminal Ready

There is a site offering modem setups for a growing variety of modem makes and models at Modem setup information which may assist you in this.

It is also worth while investigating how the modem's serial interface between your computer and modem operates. Most modern modems allow you to run the serial interface at a FIXED speed whilst allowing the telephone line interface to change its speed to the highest speed it and the remote modem can both handle.

This is known as split speed operation. If your modem supports this, lock the modem's serial interface to its highest available speed (usually 115,200 baud but maybe 38,400 baud for 14,400 baud modems).

Use your communications software (e.g. minicom or seyon) to find out about your modem configuration and set it to what is required for PPP. Many modems report their current settings in response to AT&V, but you should consult your modem manual.

If you completely mess up the settings, you can return to sanity (usually) by issuing an AT&F - return to factory settings. (For most modem modems I have encountered, the factory settings include all you need for PPP - but you should check).

Once you have worked out the modem setup string required write it down. You now have a decision: you can store these settings in your modem non-volatile memory so they can be recalled by issuing the appropriate AT command. Alternatively you can pass the correct settings to your modem as part of the PPP dialing process.

If you only use your modem from Linux to call into your ISP or corporate server, the simplest set up will have you save your modem configuration in non-volatile RAM.

If on the other hand, you modem is used by other applications and operating systems, it is safest to pass this information to the modem as each call is made so that the modem is guaranteed to be in the correct

state for the call. (This has the added advantage also of recording the modem setup string in case the modem looses the contents of its NV-RAM, which can indeed happen).

TOC

9.4 Note on Serial Flow Control

When data is traveling on serial communication lines, it can happen that data arrives faster than a computer can handle it (the computer may be busy doing something else - remember, Linux is a multi-user, multi- tasking operating system). In order to ensure that data is not lost (data does not over run in the input buffer and hence get lost), some method of controlling the flow of data is necessary.

There are two ways of doing this on serial lines:-

- Using hardware signals (Clear To Send/Request to Send - CTS/RTS)
- Using software signals (control S and control Q, also known as XON/XOFF).

Whilst the latter may be fine for a terminal (text) link, data on a PPP link uses all 8 bits - and it is quite probable that somewhere in the data there will be data bytes that translate as control S and control Q. So, if a modem is set up to use software flow control, things can rapidly go berserk!

For high speed links using PPP (which uses 8 bits of data) hardware flow control is vital and it is for this reason that you must use hardware flow control.

9.5 Testing your modem for dial out

Now that you have sorted out the serial port and modem settings it is a good idea to make sure that these setting do indeed work by dialing you ISP and seeing if you can connect.

Using you terminal communications package (such as minicom), set up the modem initialisation required for PPP and dial into the PPP server you want to connect to with a PPP session.

(Note: at this stage we are **NOT** trying to make a PPP connection - just establishing that we have the right phone number and also to find out **exactly** what the server sends to us in order to get logged in and start PPP).

During this process, either capture (log to a file) the entire login process or carefully (*very carefully*) write down **exactly** what prompts the server gives to let you know it is time to enter your user name and password (and any other commands needed to establish the PPP connection).

If your server uses PAP, you should not see a login prompt, but should instead see the (text representation) of the link control protocol (which looks like garbage) starting on your screen.

A few words of warning:-

- some servers are quite intelligent: you can log in using text based user name/passwords OR using PAP. So if your ISP or corporate site uses PAP but you do not see the garbage start up immediately, this may not mean you have done something wrong.
- some servers require you to enter some text initially and *then* start a standard PAP sequence.
- Some PPP servers are passive - that is they simply sit there sending nothing until the client that is

dialing in sends them a valid lcp packet. If the ppp server you are connecting to operates in pa mode, you will never see the garbage!

- Some servers do not start PPP until you press ENTER - so it is worth trying this if you correctly log in and do not see the garbage!

It is worth dialing in at least twice - some servers change their prompts (e.g. with the time!) every time you log in. The two critical prompts your Linux box needs to be able to identify every time you dial in are:-

- the prompt that requests you to enter your user name;
- the prompt that requests you to enter your password;

If you have to issue a command to start PPP on the server, you will also need to find out the prompt the server gives you once you are logged in to tell you that you can now enter the command to start ppp.

If your server automatically starts PPP, once you have logged in, you will start to see garbage on your screen - this is the PPP server sending your machine information to start up and configure the PPP connection.

This should look something like this :-

```
~y}#.!}!}!} }8}!}$}%U}"}&} } } } }%}& ...}'}"}(}"} .~~y}
```

(and it just keeps on coming!)

On some systems PPP must be explicitly started on the server. This is usually because the server has been set up to allow PPP logins and shell logins using the same user name/password pair. If this is the case, issue this command once you have logged in. Again, you will see the garbage as the server end of the PPP connection starts up.

If you do not see this immediately after connecting (and logging in and starting the PPP server if required), press **Enter** to see if this starts the PPP server...

At this point, you can hang up your modem (usually, type +++ quickly and then issue the ATHO command once your modem responds with OK).

If you can't get your modem to work, read your modem manual, the man pages for your communications software and the Serial HOWTO! Once you have this sorted out, carry on as above.

---

10. Setting up Name to Address Resolution (DNS)

Whilst we humans like to give names to things, computers really like numbers. On a TCP/IP network (which is what the Internet is), we call machines by a particular name - and every machine lives in a particular "domain". For example, my Linux workstation is called **archenland** and it resides in the **interweft.com.au** domain. Its human readable address is thus archenland.interweft.com.au (which is known as the FQDN - fully qualified domain name).

However, for this machine to be found by other computers on the Internet, it is actually known by its IP number when computers are communicating across the Internet.

Translating (resolving) machine (and domain) names into the numbers actually used on the Internet is the business of machines that offer the Domain Name Service.

What happens is this:-

- your machine needs to know the IP address of a particular computer. The application requiring this information asks the 'resolver' on your Linux PC to provide this information;
- the resolver queries the local host file (`/etc/hosts` and/or the domain name servers it knows about (the exact behaviour of the resolver is determined by `/etc/host.conf`);
- if the answer is found in the host file, this answer is returned;
- if a domain name server is specified, your PC queries this machine;
- if the DNS machine already knows the IP number for the required name, it returns it. If it does not, it queries other name servers across the Internet to find the information. The name server than passes this information back to the requesting resolver - which gives the information to the requesting application.

When you make a PPP connection, you need to tell your Linux machine where it can get host name to IP number (address resolution) information so that **you** can use the machine names but your **computer** can translate these to the IP numbers it needs to do its work.

One way is to enter every host that you want to talk to into the `/etc/hosts` file (which is in reality totally impossible if you are connecting to the Internet); another is to use the machine IP numbers as opposed to the names (an impossible memory task for all but the smallest LANs).

The best way is to set up Linux so that it knows where to go to get this name to number information - automatically. This service is provided by the Domain Name Server (DNS) system. All that is necessary is to enter the IP number(s) for the domain name servers into your /etc/resolv.conf file.

10.1 The `/etc/resolv.conf` file

Your PPP server sysadmin/user support people should provide you with two DNS IP numbers (only one is necessary - but two gives some redundancy in the event of failure).

As previously mentioned, Linux cannot set its name server IP number in the way that MS Windows 95 does. So you must **insist** (politely) that your ISP provide you with this information!

Your `/etc/resolv.conf` should look something like :-

```
domain your.isp.domain.name
nameserver 10.25.0.1
nameserver 10.25.1.2
```

Edit this file (creating it if necessary) to represent the information that your ISP has provided. It should have ownership and permissions as follows :-

```
    -rw-r--r--   1 root     root           73 Feb 19 01:46 /etc/resolv.conf
```

If you have already set up a `/etc/resolv.conf` because you are on a LAN, simply add the IP numbers of the PPP DNS servers to your existing file.

10.2 The `/etc/host.conf` file

You should also check that your `/etc/host.conf` file is correctly set up. This should look like

```
order hosts,bind
multi on
```

This tells the resolver to use information in the host file before it sends queries to the DNS for resolution.

11. Using PPP and root privileges

Because PPP needs to set up networking devices, change the kernel routing table and so forth, it requires root privileges to do this.

If users other than root are to set up PPP connections, the pppd program should be setuid root :-

```
-rwsr-xr-x   1 root     root        95225 Jul 11 00:27 /usr/sbin/pppd
```

If /usr/sbin/pppd is not set up this way, then **as root** issue the command:-

```
chmod u+s /usr/sbin/pppd
```

What this does is make pppd run with root privileges **even** if the binary is run by an ordinary user. This allows a normal user to run pppd with the necessary privileges to set up the network interfaces and the kernel routing table.

Programs that run 'set uid root' are potential security holes and you should be extremely cautious about making programs 'suid root'. A number of programs (including pppd) have been carefully written to minimise the danger of running suid root, so you should be safe with this one (but no guarantees).

Depending on how you want your system to operate - specifically if you want ANY user on your system to be able to initiate a PPP link, you should make your ppp-on/off scripts world read/execute. (This is probably fine if your PC is used ONLY by you).

However, if you do NOT want just anyone to be able to start up a PPP connection (for example, your children have accounts on your Linux PC and you do not want them hooking into the Internet without your supervision), you will need to establish a PPP group (as root, edit /etc/group) and :-

- Make pppd suid root, owned by user root and group PPP, with the 'other' permissions on this file empty. It should then look like

  ```
  -rwsr-x---   1 root     PPP        95225 Jul 11 00:27 /usr/sbin/pppd
  ```
- Make the ppp-on/off scripts owned by user root and group PPP
- Make the ppp-on/off scripts read/executable by group PPP

  ```
  -rwxr-x---   1 root     PPP           587 Mar 14  1995 /usr/sbin/ppp-on
  -rwxr-x---   1 root     PPP           631 Mar 14  1995 /usr/sbin/ppp-off
  ```
- Make the other access rights for ppp-on/off nill.
- add the users who will be firing up PPP to the PPP group in /etc/group

Even if you do this, ordinary users will STILL not be able to shut down the link under software control! Running the `ppp-off` script requires root privileges. However, any user can just turn off the modem (or disconnect the telephone line from an internal modem).

An alternative (and better method) to this set up is to use the `sudo` program. This offers superior security and will allow you to set things up so that any (authorised) user can activate/deactivate the link using the scripts. Using `sudo` will allow an authorised user to activate/deactivate the PPP link cleanly and securely.

12. Setting up the PPP connection files

You now need to be logged in as **root** to create the directories and edit the files needed to set up PPP, even if you want PPP to be accessible to all users.

PPP uses a number of files to connect and set up a PPP connection. These differ in name and location between PPP 2.1.2 and 2.2.

For PPP 2.1.2 the files are:-

```
/usr/sbin/pppd          # the PPP binary
/usr/sbin/ppp-on        # the dialer/connection script
/usr/sbin/ppp-off       # the disconnection script
/etc/ppp/options        # the options pppd uses for all connections
/etc/ppp/options.ttyXX  # the options specific to a connection on this port
```

For PPP 2.2 the files are:-

```
/usr/sbin/pppd                  # the PPP binary
/etc/ppp/scripts/ppp-on         # the dialer/connection script
/etc/ppp/scripts/ppp-on-dialer  # part 1 of the dialer script
/etc/ppp/scripts/ppp-off        # the actual chat script itself
/etc/ppp/options                # the options pppd uses for all connections
/etc/ppp/options.ttyXX          # the options specific to a connection on this port
```

Red Hat Linux users should note that the standard Red Hat 4.X installation places these scripts in `/usr/doc/ppp-2.2.0f-2/scripts`.

In your /etc directory there should be a ppp directory:-

```
drwxrwxr-x   2 root     root         1024 Oct  9 11:01 ppp
```

If it does not exist - create it with these ownerships and permissions.

If the directory already existed, it should contain a template options file called **options.tpl**. This file is included below in case it does not.

Print it out as it contains an explanation of nearly all the PPP options (these are useful to read in conjunction with the pppd man pages). Whilst you can use this file as the basis of your `/etc/ppp/options` file, it is probably better to create your own options file that does not include all the comments in the template - it will be much shorter and easier to read/maintain.

If you have multiple serial lines/modems (typically the case for PPP servers), create a general `/etc/ppp/options` file containing the options that are common for all the serial ports on which you are supporting dial in/out and set up individual option files for each serial line on which you will be establishing a PPP connection with the individual settings required for each port.

These port specific option files are named `options.ttyx1`, `options.ttyx2` and so forth (where x is the appropriate letter for your serial ports).

However, for a single PPP connection, you can happily use the `/etc/ppp/options` file. Alternatively, you can pı the options as arguments in the pppd command itself.

It is easier to maintain a setup that uses `/etc/ppp/options.ttySx` files. If you use PPP to connect to a number of different sites, you can create option files for each site in `/etc/ppp/options.site` and then specify the option file as a parameter to the PPP command as you connect (using the `file option-file` pppd option to pppd on the command line).

12.1 The supplied options.tpl file

Some distributions of PPP seem to have lost the options.tpl file, so here is the complete file. I suggest that you do NOT edit this file to create your `/etc/ppp/options` file(s). Rather, copy this to a new file and then edit that. If you mess up your edits, you can then go back to the original and start again.

```
# /etc/ppp/options -*- sh -*- general options for pppd
# created 13-Jul-1995 jmk
# autodate: 01-Aug-1995
# autotime: 19:45

# Use the executable or shell command specified to set up the serial
# line.  This script would typically use the "chat" program to dial the
# modem and start the remote ppp session.
#connect "echo You need to install a connect command."

# Run the executable or shell command specified after pppd has
# terminated the link.  This script could, for example, issue commands
# to the modem to cause it to hang up if hardware modem control signals
# were not available.
#disconnect "chat -- \d+++\d\c OK ath0 OK"

# async character map -- 32-bit hex; each bit is a character
# that needs to be escaped for pppd to receive it.  0x00000001
# represents '\x01', and 0x80000000 represents '\x1f'.
#asyncmap 0

# Require the peer to authenticate itself before allowing network
# packets to be sent or received.
#auth

# Use hardware flow control (i.e. RTS/CTS) to control the flow of data
# on the serial port.
#crtscts

# Use software flow control (i.e. XON/XOFF) to control the flow of data
# on the serial port.
#xonxoff

# Add a default route to the system routing tables, using the peer as
# the gateway, when IPCP negotiation is successfully completed.  This
# entry is removed when the PPP connection is broken.
#defaultroute
```

```
# Specifies that certain characters should be escaped on transmission
# (regardless of whether the peer requests them to be escaped with its
# async control character map).  The characters to be escaped are
# specified as a list of hex numbers separated by commas.  Note that
# almost any character can be specified for the escape option, unlike
# the asyncmap option which only allows control characters to be
# specified.  The characters which may not be escaped are those with hex
# values 0x20 - 0x3f or 0x5e.
#escape 11,13,ff

# Don't use the modem control lines.
#local

# Specifies that pppd should use a UUCP-style lock on the serial device
# to ensure exclusive access to the device.
#lock

# Use the modem control lines.  On Ultrix, this option implies hardware
# flow control, as for the crtscts option.  (This option is not fully
# implemented.)
#modem

# Set the MRU [Maximum Receive Unit] value to <n> for negotiation.  pppd
# will ask the peer to send packets of no more than <n> bytes. The
# minimum MRU value is 128.  The default MRU value is 1500.  A value of
# 296 is recommended for slow links (40 bytes for TCP/IP header + 256
# bytes of data).
#mru 542

# Set the interface netmask to <n>, a 32 bit netmask in "decimal dot"
# notation (e.g. 255.255.255.0).
#netmask 255.255.255.0

# Disables the default behaviour when no local IP address is specified,
# which is to determine (if possible) the local IP address from the
# hostname. With this option, the peer will have to supply the local IP
# address during IPCP negotiation (unless it specified explicitly on the
# command line or in an options file).
#noipdefault

# Enables the "passive" option in the LCP.  With this option, pppd will
# attempt to initiate a connection; if no reply is received from the
# peer, pppd will then just wait passively for a valid LCP packet from
# the peer (instead of exiting, as it does without this option).
#passive

# With this option, pppd will not transmit LCP packets to initiate a
# connection until a valid LCP packet is received from the peer (as for
# the "passive" option with old versions of pppd).
#silent

# Don't request or allow negotiation of any options for LCP and IPCP
# (use default values).
#-all
```

```
# Disable Address/Control compression negotiation (use default, i.e.
# address/control field disabled).
#-ac

# Disable asyncmap negotiation (use the default asyncmap, i.e. escape
# all control characters).
#-am

# Don't fork to become a background process (otherwise pppd will do so
# if a serial device is specified).
#-detach

# Disable IP address negotiation (with this option, the remote IP
# address must be specified with an option on the command line or in an
# options file).
#-ip

# Disable magic number negotiation.  With this option, pppd cannot
# detect a looped-back line.
#-mn

# Disable MRU [Maximum Receive Unit] negotiation (use default, i.e.
# 1500).
#-mru

# Disable protocol field compression negotiation (use default, i.e.
# protocol field compression disabled).
#-pc

# Require the peer to authenticate itself using PAP.
# This requires TWO WAY authentication - do NOT use this for a standard
# PAP authenticated link to an ISP as this will require the ISP machine
# to authenticate itself to your machine (and it will not be able to).
#+pap

# Don't agree to authenticate using PAP.
#-pap

# Require the peer to authenticate itself using CHAP [Cryptographic
# Handshake Authentication Protocol] authentication.
# This requires TWO WAY authentication - do NOT use this for a standard
# CHAP authenticated link to an ISP as this will require the ISP machine
# to authenticate itself to your machine (and it will not be able to).
#+chap

# Don't agree to authenticate using CHAP.
#-chap

# Disable negotiation of Van Jacobson style IP header compression (use
# default, i.e. no compression).
#-vj

# Increase debugging level (same as -d).  If this option is given, pppd
```

```
# will log the contents of all control packets sent or received in a
# readable form.  The packets are logged through syslog with facility
# daemon and level debug. This information can be directed to a file by
# setting up /etc/syslog.conf appropriately (see syslog.conf(5)).  (If
# pppd is compiled with extra debugging enabled, it will log messages
# using facility local2 instead of daemon).
#debug

# Append the domain name <d> to the local host name for authentication
# purposes.  For example, if gethostname() returns the name porsche,
# but the fully qualified domain name is porsche.Quotron.COM, you would
# use the domain option to set the domain name to Quotron.COM.
#domain <d>

# Enable debugging code in the kernel-level PPP driver.  The argument n
# is a number which is the sum of the following values: 1 to enable
# general debug messages, 2 to request that the contents of received
# packets be printed, and 4 to request that the contents of transmitted
# packets be printed.
#kdebug n

# Set the MTU [Maximum Transmit Unit] value to <n>. Unless the peer
# requests a smaller value via MRU negotiation, pppd will request that
# the kernel networking code send data packets of no more than n bytes
# through the PPP network interface.
#mtu <n>

# Set the name of the local system for authentication purposes to <n>.
# This will probably have to be set to your ISP user name if you are
# using PAP/CHAP.
#name <n>

# Set the user name to use for authenticating this machine with the peer
# using PAP to <u>.
# Do NOT use this if you are using 'name' above!
#user <u>

# Enforce the use of the host name as the name of the local system for
# authentication purposes (overrides the name option).
#usehostname

# Set the assumed name of the remote system for authentication purposes
# to <n>.
#remotename <n>

# Add an entry to this system's ARP [Address Resolution Protocol]
# table with the IP address of the peer and the Ethernet address of this
# system.
#proxyarp

# Use the system password database for authenticating the peer using
# PAP.
#login
```

```
# If this option is given, pppd will send an LCP echo-request frame to
# the peer every n seconds. Under Linux, the echo-request is sent when
# no packets have been received from the peer for n seconds. Normally
# the peer should respond to the echo-request by sending an echo-reply.
# This option can be used with the lcp-echo-failure option to detect
# that the peer is no longer connected.
#lcp-echo-interval <n>

# If this option is given, pppd will presume the peer to be dead if n
# LCP echo-requests are sent without receiving a valid LCP echo-reply.
# If this happens, pppd will terminate the connection.  Use of this
# option requires a non-zero value for the lcp-echo-interval parameter.
# This option can be used to enable pppd to terminate after the physical
# connection has been broken (e.g., the modem has hung up) in
# situations where no hardware modem control lines are available.
#lcp-echo-failure <n>

# Set the LCP restart interval (retransmission timeout) to <n> seconds
# (default 3).
#lcp-restart <n>

# Set the maximum number of LCP terminate-request transmissions to <n>
# (default 3).
#lcp-max-terminate <n>

# Set the maximum number of LCP configure-request transmissions to <n>
# (default 10).
# Some PPP servers are slow to start up. You may need to increase this
# if you keep getting 'serial line looped back' errors and your are SURE
# that you have logged in correctly and PPP should be starting on the server.
#lcp-max-configure <n>

# Set the maximum number of LCP configure-NAKs returned before starting
# to send configure-Rejects instead to <n> (default 10).
#lcp-max-failure <n>

# Set the IPCP restart interval (retransmission timeout) to <n>
# seconds (default 3).
#ipcp-restart <n>

# Set the maximum number of IPCP terminate-request transmissions to <n>
# (default 3).
#ipcp-max-terminate <n>

# Set the maximum number of IPCP configure-request transmissions to <n>
# (default 10).
#ipcp-max-configure <n>

# Set the maximum number of IPCP configure-NAKs returned before starting
# to send configure-Rejects instead to <n> (default 10).
#ipcp-max-failure <n>

# Set the PAP restart interval (retransmission timeout) to <n> seconds
# (default 3).
```

```
#pap-restart <n>

# Set the maximum number of PAP authenticate-request transmissions to
# <n> (default 10).
#pap-max-authreq <n>

# Set the CHAP restart interval (retransmission timeout for
# challenges) to <n> seconds (default 3).
#chap-restart <n>

# Set the maximum number of CHAP challenge transmissions to <n>
# (default 10).
#chap-max-challenge

# If this option is given, pppd will re-challenge the peer every <n>
# seconds.
#chap-interval <n>

# With this option, pppd will accept the peer's idea of our local IP
# address, even if the local IP address was specified in an option.
#ipcp-accept-local

# With this option, pppd will accept the peer's idea of its (remote) IP
# address, even if the remote IP address was specified in an option.
#ipcp-accept-remote
```

12.2 What options should I use? (No PAP/CHAP)

Well, as in all things that depends (sigh). The options specified here should work with most servers.

However, if it does NOT work, READ THE TEMPLATE FILE (`/etc/ppp/options.tpl`) **and** the pppd man pages **and** speak to the sysadmin/user support people who run the server to which you are connecting.

You should also note that the connect scripts presented here also use some command line options to pppd to make things a bit easier to change.

```
# /etc/ppp/options (NO PAP/CHAP)
#
# Prevent pppd from forking into the background
-detach
#
# use the modem control lines
modem
# use uucp style locks to ensure exclusive access to the serial device
lock
# use hardware flow control
crtscts
# create a default route for this connection in the routing table
defaultroute
# do NOT set up any "escaped" control sequences
asyncmap 0
```

```
# use a maximum transmission packet size of 552 bytes
mtu 552
# use a maximum receive packet size of 552 bytes
mru 552
#
#-------END OF SAMPLE /etc/ppp/options (no PAP/CHAP)
```

13. If your PPP server uses PAP (Password AuthenticationProtocol)

If the server to which you are connecting requires PAP or CHAP authentication, you have a little bit more work.

To the above options file, add the following lines

```
#
# force pppd to use your ISP user name as your 'host name' during the
# authentication process
name <your ISP user name>       # you need to edit this line
#
# If you are running a PPP *server* and need to force PAP or CHAP
# uncomment the appropriate one of the following lines. Do NOT use
# these is you are a client connecting to a PPP server (even if it uses PAP
# or CHAP) as this tells the SERVER to authenticate itself to your
# machine (which almost certainly can't do - and the link will fail).
#+chap
#+pap
#
# If you are using ENCRYPTED secrets in the /etc/ppp/pap-secrets
# file, then uncomment the following line.
# Note: this is NOT the same as using MS encrypted passwords as can be
# set up in MS RAS on Windows NT.
#+papcrypt
```

13.1 Using MSCHAP

Microsoft Windows NT RAS can be set up to use a variation on CHAP (Challenge/Handshake Authentication Protocol). In your PPP sources tar ball, you will find a file called README.MSCHAP80 that discusses this.

You can determine if the server is requesting authentication using this protocol by enabling debugging for pppd. If the server is requesting MS CHAP authentication, you will see lines like

```
rcvd [LCP ConfReq id=0x2 <asyncmap 0x0> <auth chap 80> <magic 0x46a3>]
```

The critical information here is **auth chap 80**.

In order to use MS CHAP, you will need to recompile pppd to support this. Please see the instructions in the README.MSCHAP80 file in the PPP source file for instructions on how to compile and use this variation.

You should note that at present this code supports only Linux PPP clients connecting to an MS Windows NT server. It does **NOT** support setting up a Linux PPP server to use MSCHAP80 authentication from clients.

TOC

13.2 The PAP/CHAP secrets file

If you are using pap or chap authentication, then you also need to create the secrets file. These are:

```
/etc/ppp/pap-secrets
/etc/ppp/chap-secrets
```

They must be owned by user root, group root and have file permissions 740 for security.

The first point to note about PAP and CHAP is that they are designed to authenticate **computer systems** not **users**.

"Huh? What's the difference?" I hear you ask.

Well now, once your computer has made its PPP connection to the server, **ANY** user on your system can use that connection - not just you. This is why you can set up a WAN (wide area network) link that joins two LANs (local area networks) using PPP.

PAP can (and for CHAP **DOES**) require **bidirectional** authentication - that is a valid name and secret is required on each computer for the other computer involved. However, this is **NOT** the way most PPP servers offering dial up PPP PAP-authenticated connections operate.

That being said, your ISP will probably have given you a user name and password to allow you to connect to their system and thence the Internet. Your ISP is not interested in your computer's name at all, so you will probably need to use the user name at your ISP as the name for your computer.

This is done using the `name user name` option to pppd. So, if you are to use the user name given you by your ISP, add the line

```
name your_user name_at_your_ISP
```

to your `/etc/ppp/options` file.

Technically, you should really use `user our_user name_at_your_ISP` for PAP, but pppd is sufficiently intelligent to interpret `name` as `user` if it is required to use PAP. The advantage of using the `name` option is that this is also valid for CHAP.

As PAP is for authenticating **computers**, technically you need also to specify a remote computer name. However, as most people only have one ISP, you can use a wild card (*) for the remote host name in the secrets file.

It is also worth noting that many ISPs operate multiple modem banks connected to different terminal servers - each with a different name, but ACCESSED from a single (rotary) dial in number. It can therefore be quite difficult in some circumstances to know ahead of time what the name of the remote computer is, as this depends on which terminal server you connect to!

TOC

13.3 The PAP secrets file

The `/etc/ppp/pap-secrets` file looks like

```
# Secrets for authentication using PAP
# client        server       secret     acceptable_local_IP_addresses
```

The four fields are white space delimited and the last one can be blank (which is what you want for a dynamic and probably static IP allocation from your ISP).

Suppose your ISP gave you a user name of `fred` and a password of `flintstone` you would set the `name fred` option in `/etc/ppp/options[.ttySx]` and set up your `/etc/ppp/pap-secrets` file as follows

```
# Secrets for authentication using PAP
# client        server  secret          acceptable local IP addresses
fred            *       flintstone
```

This says for the local machine name `fred` (which we have told pppd to use even though it is not our local machine name) and for **ANY** server, use the password (secret) of `flintstone`.

Note that we do not need to specify a local IP address, unless we are required to FORCE a particular local, static IP address. Even if you try this, it is unlikely to work as most PPP servers (for security reasons) do not allow the remote system to set the IP number they are to be given.

13.4 The CHAP secrets file

This requires that you have mutual authentication methods - that is you must allow for both your machine to authenticate the remote server **AND** the remote server to authenticate your machine.

So, if your machine is `fred` and the remote is `barney`, your machine would set `name fred remotename barney` and the remote machine would set `name barney remotename fred` in their respective `/etc/ppp/options.ttySx` files.

The `/etc/chap-secrets` file for fred would look like

```
# Secrets for authentication using CHAP
# client        server  secret            acceptable local IP addresses
fred            barney  flintstone
barney          fred    wilma
```

and for barney

```
# Secrets for authentication using CHAP
# client        server  secret            acceptable local IP addresses
barney          fred    flintstone
fred            barney  wilma
```

Note in particular that both machines must have entries for bidirectional authentication. This allows the local machine to authenticate itself to the remote **AND** the remote machine to authenticate itself to the local machine.

13.5 Handling multiple PAP-authenticated connections

Some users have more than one server to which they connect that use PAP. Provided that your user name is different on each machine to which you want to connect, this is not a problem.

However, many users have the same user name on two (or more - even all) systems to which they connect. This then presents a problem in correctly selecting the appropriate line from `/etc/ppp/pap-secrets`.

As you might expect, PPP provides a mechanism for overcoming this. PPP allows you to set an 'assumed name' for the remote (server) end of the connection using the **remotename** option to pppd.

Let us suppose that you connect to two PPP servers using the username fred. You set up your `/etc/ppp/pap-secrets` something like

```
fred    pppserver1      barney
fred    pppserver2      wilma
```

Now, to set connect to pppserver1 you would use `name fred remotename pppserver1` in your ppp-options and for pppserver2 `name fred remotename pppserver2`.

As you can select the ppp options file to use with pppd using the `file filename` option, you can set up a script to connect to each of your PPP servers, correctly picking the options file to use and hence selecting the right `remotename` option.

14. Setting up the PPP connection manually

Now that you have created your `/etc/ppp/options` and `/etc/resolv.conf` files (and, if necessary, the `/etc/ppp/pap|chap-secrets` file), you can test the settings by manually establishing a PPP connection. (Once we have the manual connection working, we will automate the process).

To do this, your communications software must be capable of quitting WITHOUT resetting the modem. Minicom can do this - ALT Q (or in older version of minicom CTRL A Q)

Make sure you are logged in as root.

Fire up you communications software (such as minicom), dial into the PPP server and log in as normal. If you need to issue a command to start up PPP on the server, do so. You will now see the garbage you saw before.

If you are using pap or chap, then merely connecting to the remote system should start ppp on the remote and you will see the garbage without logging in (although this may not happen for some servers - try pressing **Enter** and see if the garbage starts up).

Now quit the communications software *without resetting the modem* (ALT Q or CTL A Q in minicom) and at the Linux prompt (as root) type

```
pppd -d -detach /dev/ttySx 38400 &
```

The -d option turns on debugging - the ppp connection start up conversation will be logged to your system log - which is useful if you are having trouble.

Your modem lights should now flash as the PPP connection is established. It will take a short while for the PPP connection to be made.

At this point you can look at the PPP interface, by issuing the command

```
ifconfig
```

In addition to any Ethernet and loop back devices you have, you should see something like :-

```
ppp0      Link encap:Point-Point Protocol
          inet addr:10.144.153.104  P-t-P:10.144.153.51 Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:552  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
```

Where

- inet addr:10.144.153.10 is the IP number of your end of the link.
- P-t-P:10.144.153.5 is the SERVER's IP number.

(Naturally, ifconfig will not report these IP numbers, but the ones used by your PPP server.)

Note: ifconfig also tells you that the link is UP and RUNNING!

If you get no ppp device listed or something like

```
ppp0      Link encap:Point-Point Protocol
          inet addr:0.0.0.0  P-t-P:0.0.0.0  Mask:0.0.0.0
          POINTOPOINT  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
```

Your PPP connection has not been made...see the later section on debugging!

You should also be able to see a route to the the remote host (and beyond). To do this, issue the command

```
route -n
```

You should se something like:-

```
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
10.144.153.3    *               255.255.255.255 UH    1500   0        1 ppp0
127.0.0.0       *               255.0.0.0       U     3584   0       11 lo
10.0.0.0        *               255.0.0.0       U     1500   0       35 eth0
default         10.144.153.3    *               UG    1500   0        5 ppp0
```

Of particular importance here, notice we have TWO entries pointing to our ppp interface.

The first is a HOST route (indicated by the H flag) and that allows us to see the host to which we are connected to - but no further.

The second is the default route (established by giving pppd the option `defaultroute`. This is the route that tells our Linux PC to send any packets NOT destined for the local Ethernet(s) - to which we have specific network routes - to the PPP server itself. The PPP server then is responsible for routing our packets out onto the Internet and routing the return packets back to us.

If you do not see a routing table with two entries, something is wrong. In particular if your syslog shows a message telling you pppd is not replacing an existing default route, then you have a default route pointing at your Ethernet interface - which **MUST** be replaced by a specific network route: **YOU CAN ONLY HAVE ONE DEFAULT ROUTE!!!**

You will need to explore your system initialisation files to find out where this default route is being set up (it will

use a `route add default...` command). Change this command to something like `route add net`.

Now test the link by 'pinging' the server at its IP number as reported by the ifconfig output, i.e.

```
ping 10.144.153.51
```

You should receive output like

```
PING 10.144.153.51 (10.144.153.51): 56 data bytes
64 bytes from 10.144.153.51: icmp_seq=0 ttl=255 time=328.3 ms
64 bytes from 10.144.153.51: icmp_seq=1 ttl=255 time=190.5 ms
64 bytes from 10.144.153.51: icmp_seq=2 ttl=255 time=187.5 ms
64 bytes from 10.144.153.51: icmp_seq=3 ttl=255 time=170.7 ms
```

This listing will go on for ever - to stop it press CTRL C, at which point you will receive some more information :-

---

```
--- 10.144.153.51 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 170.7/219.2/328.3 ms
```

So far so good.

Now try pinging a host by name (not the name of the PPP server itself) but a host at another site that you KNOW is probably going to be up and running...). For example

---

```
ping sunsite.unc.edu
```

This time there will be a bit of a pause as Linux obtains the IP number for the fully qualified host name you have 'ping'ed from the DNS you specified in `/etc/resolv.conf` - so don't worry (but you will see your modem lights flash). Shortly you will receive output like

---

```
 PING sunsite.unc.edu (152.2.254.81): 56 data bytes
64 bytes from 152.2.254.81: icmp_seq=0 ttl=254 time=190.1 ms
64 bytes from 152.2.254.81: icmp_seq=1 ttl=254 time=180.6 ms
64 bytes from 152.2.254.81: icmp_seq=2 ttl=254 time=169.8 ms
64 bytes from 152.2.254.81: icmp_seq=3 ttl=254 time=170.6 ms
64 bytes from 152.2.254.81: icmp_seq=4 ttl=254 time=170.6 ms
```

Again, stop the output by pressing CTRL C and get the statistics...

```
--- sunsite.unc.edu ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 169.8/176.3/190.1 ms
```

If you don't get any response, try pinging the IP address of the DNS server at your ISP's site. If you get a result from this, then it looks like you have a problem with `/etc/resolv.conf`.

If this doesn't work, you have a routing problem, or your ISP has a problem routing packets back to you. Check your routing table as shown above and if that is OK, contact your ISP. A good test of the ISP is to use another operating system to connect. If you can get beyond your ISP with that, then the problem is at your end.

If everything works, shut down the connection by typing

```
ppp-off
```

After a short pause, the modem should hang itself up.

If that does not work, either turn off your modem or fire up your communications software and interrupt the modem with +++ and then hang up with ATH0 when you receive the modem's OK prompt.

You may also need to clean up the lock file created by pppd

```
rm -f /var/lock/LCK..ttySx
```

---

15. Automating your connections - Creating the connection scripts

Whilst you can continue to log in by hand as shown above, it is much neater to set up some scripts to do this automatically for you.

A set of scripts automates the log in and PPP start up so all you have to do (as root or as a member of the PPP group) is issue a single command to fire up your connection.

15.1 Connection scripts for User name/Password Authentication

If your ISP does NOT require the use of PAP/CHAP, these are the scripts for you!

If the ppp package installed correctly, you should have two example files. For PPP 2.1.2 they are in `/usr/sbin` and for PPP 2.2 they are in `/etc/ppp/scripts`. They are called

for PPP-2.1.2

```
ppp-on
ppp-off
```

and for PPP-2.2

```
ppp-off
ppp-on
ppp-on-dialer
```

Now, if you are using PPP 2.1.2, I strongly urge you to delete the sample files. There are potential problems with these - and don't tell me they work fine - I used them for ages too (and recommended them in the first version of this HOWTO)!

For the benefit of PPP 2.1.2 users, here are BETTER template versions, taken from the PPP 2.2 distribution. I suggest you copy and use these scripts **instead of** the old PPP-2.1.2 scripts.

15.2 The ppp-on script

This is the first of a PAIR of scripts that actually fire up the connection.

```
#!/bin/sh
#
# Script to initiate a PPP connection. This is the first part of the
# pair of scripts. This is not a secure pair of scripts as the codes
```

TOC

```
# are visible with the 'ps' command.  However, it is simple.
#
# These are the parameters. Change as needed.
TELEPHONE=555-1212      # The telephone number for the connection
ACCOUNT=george          # The account name for logon (as in 'George Burns')
PASSWORD=gracie         # The password for this account (and 'Gracie Allen')
LOCAL_IP=0.0.0.0        # Local IP address if known. Dynamic = 0.0.0.0
REMOTE_IP=0.0.0.0       # Remote IP address if desired. Normally 0.0.0.0
NETMASK=255.255.255.0   # The proper netmask if needed
#
# Export them so that they will be available to 'ppp-on-dialer'
export TELEPHONE ACCOUNT PASSWORD
#
# This is the location of the script which dials the phone and logs
# in.  Please use the absolute file name as the $PATH variable is not
# used on the connect option.  (To do so on a 'root' account would be
# a security hole so don't ask.)
#
DIALER_SCRIPT=/etc/ppp/ppp-on-dialer
#
# Initiate the connection
#
#
exec /usr/sbin/pppd debug /dev/ttySx 38400 \
        $LOCAL_IP:$REMOTE_IP \
        connect $DIALER_SCRIPT
```

Here is the ppp-on-dialer script:-

```
#!/bin/sh
#
# This is part 2 of the ppp-on script. It will perform the connection
# protocol for the desired connection.
#
/usr/sbin/chat -v                                                   \
        TIMEOUT         3                               \
        ABORT           '\nBUSY\r'                      \
        ABORT           '\nNO ANSWER\r'                 \
        ABORT           '\nRINGING\r\n\r\nRINGING\r'    \
        ''              \rAT                            \
        'OK-+++\c-OK'   ATH0                            \
        TIMEOUT         30                              \
        OK              ATDT$TELEPHONE                  \
        CONNECT         ''                              \
        ogin:--ogin:    $ACCOUNT                        \
        assword:        $PASSWORD
```

For PPP-2.2, the `ppp-off` script looks like:-

TOC

```
#!/bin/sh
#######################################################################
#
# Determine the device to be terminated.
#
if [ "$1" = "" ]; then
        DEVICE=ppp0
else
        DEVICE=$1
fi

#######################################################################
#
# If the ppp0 pid file is present then the program is running. Stop it.
if [ -r /var/run/$DEVICE.pid ]; then
        kill -INT `cat /var/run/$DEVICE.pid`
#
# If the kill did not work then there is no process running for this
# pid. It may also mean that the lock file will be left. You may wish
# to delete the lock file at the same time.
        if [ ! "$?" = "0" ]; then
                rm -f /var/run/$DEVICE.pid
                echo "ERROR: Removed stale pid file"
                exit 1
        fi
#
# Success. Let pppd clean up its own junk.
        echo "PPP link to $DEVICE terminated."
        exit 0
fi
#
# The ppp process is not running for ppp0
echo "ERROR: PPP link is not active on $DEVICE"
exit 1
```

15.3 Editing the supplied PPP startup scripts

As the new scripts come in two parts, we will edit them in turn.

The ppp-on script

You will need to edit the script to reflect YOUR user name at your ISP, YOUR password at your ISP, the telephone number of your ISP.

Each of the lines like `TELEPHONE=` actually set up shell variables that contain the information to the right c '=' (excluding the comments of course). So edit each of these lines so it is correct for your ISP and connecti

Also, as you are setting the IP number (if you need to) in the `/etc/ppp/options` file, DELETE the line that says

---

```
$LOCAL_IP:$REMOTE_IP \
```

Also, make sure that the shell variable DIALER_SCRIPT points at the full path and name of the dialer script that you are actually going to use. So, if you have moved this or renamed the script, make sure you edit this line correctly in the `ppp-on` script!

The ppp-on-dialer script

This is the second of the scripts that actually brings up our ppp link.

Note: a chat script is normally all on one line. the backslashes are used to allow line continuations across several physical lines (for human readability) and do not form part of the script itself.

However, it is very useful to look at it in detail so that we understand what it is actually (supposed) to be doing!

15.4 What a Chat script means...

A chat script is a sequence of `"`expect string`" "`send string`"` pairs. In particular, note that we **ALWAYS** expect **something** before we send something.

If we are to send something **WITHOUT** receiving anything first, we must use an empty expect string (indicated by `""`) and similarly for expecting something without sending anything! Also, if a string consists of several words, (e.g. NO CARRIER), you must quote the string so that it is seen as a single entity by chat.

The chat line in our template is:-

```
exec /usr/sbin/chat -v
```

Invoke chat, the -v tells chat to copy ALL its I/O into the system log (usually /var/log/messages). Once you are happy that the chat script is working reliably, edit this line to remove the -v to save unnecessary clutter in your syslog.

```
TIMEOUT         3
```

This sets the timeout for the receipt of expected input to three seconds. You may need to increase this to say 5 or 10 seconds if you are using a really slow modem!

```
ABORT           '\nBUSY\r'
```

If the string BUSY is received, abort the operation.

```
ABORT             '\nNO ANSWER\r'
```

If the string NO ANSWER is received, abort the operation

```
ABORT             '\nRINGING\r\n\r\nRINGING\r'
```

If the (repeated) string RINGING is received, abort the operation. This is because someone is ringing your phone line!

```
"                 \rAT
```

Expect nothing from the modem and send the string AT.

```
OK-+++\c-OK   ATH0
```

This one is a bit more complicated as it uses some of chat's error recovery capabilities.

What is says is...Expect OK, if it is NOT received (because the modem is not in command mode) then send +++ (the standard Hayes-compatible modem string that returns the modem to command mode) and expect OK. Then send ATH0 (the modem hang up string). This allows your script to cope with the situation of your modem being stuck on-line!

```
TIMEOUT           30
```

Set the timeout to 30 seconds for the remainder of the script. If you experience trouble with the chat script aborting due to timeouts, increase this to 45 seconds or more.

```
OK                ATDT$TELEPHONE
```

Expect OK (the modem's response to the ATH0 command) and dial the number we want to call.

```
CONNECT           ''
```

Expect CONNECT (which our modem sends when the remote modem answers) and send nothing in reply.

```
ogin:--ogin:    $ACCOUNT
```

Again, we have some error recovery built in here. Expect the login prompt (...ogin:) but if we don't receive it by the timeout, send a return and then look for the login prompt again. When the prompt is received, send the username (stored in the shell variable $ACCOUNT).

```
assword:        $PASSWORD
```

Expect the password prompt and send our password (again, stored in a shell variable).

This chat script has reasonable error recovery capability. Chat has considerably more features than demonstrated here. For more information consult the chat manual page (`man 8 chat`).

Starting PPP at the server end

Whilst the ppp-on-dialer script is fine for servers that automatically start pppd at the server end once you have logged in, some servers require that you explicitly start PPP on the server.

If you need to issue a command to start up PPP on the server, you DO need to edit the ppp-on-dialer script.

At the END of the script (after the password line) add an additional **expect send** pair - this one would look for your login prompt (beware of characters that have a special meaning in the Bourne shell - such as $ and or (open and close square brackets).

Once chat has found the shell prompt, chat must issue the ppp start up command required for your ISPs PPP server.

In my case, my PPP server uses the standard Linux Bash prompt

---

```
[hartr@kepler hartr]$
```

---

and requires that I type

```
ppp
```

---

to start up PPP on the server.

It is a good idea to allow for a bit of error recovery here, so in my case I use

---

```
        hartr--hartr    ppp
```

This says, if we don't receive the prompt within the timeout, send a carriage return and looks for the prompt again.

Once the prompt is received, then send the string `ppp`.

Note: don't forget to add a \ to the end of the previous line so chat still thinks the entire chat script is on one line!

Unfortunately, some servers produce a very variable set of prompts! You may need to log in several times using minicom to understand what is going on and pick the stable "expect" strings.

15.5 A chat script for PAP/CHAP authenticated connections

If your ISP is using PAP/CHAP, then your chat script is much simpler. All your chat script needs to do is dial the telephone, wait for a connect and then let pppd handle the logging in!

```
#!/bin/sh
#
# This is part 2 of the ppp-on script. It will perform the connection
# protocol for the desired connection.
#
exec /usr/sbin/chat -v                                  \
        TIMEOUT         3                               \
        ABORT           '\nBUSY\r'                      \
        ABORT           '\nNO ANSWER\r'                 \
        ABORT           '\nRINGING\r\n\r\nRINGING\r'    \
        ''              \rAT                            \
        'OK-+++\c-OK'   ATH0                            \
        TIMEOUT         30                              \
        OK              ATDT$TELEPHONE                  \
        CONNECT         ''                              \
```

15.6 The pppd debug and `file option_file` options

As we have already seen, you can turn on debug information logging with the -d option to pppd. The 'debug' option is equivalent to this.

As we are establishing a new connection with a new script, leave in the debug option for now. (Warning: if your disk space is tight, logging pppd exchanges can rapidly extend your syslog file and run you into trouble - but to do this you must fail to connect and keep on trying for quite a few minutes).

Once you are happy that all is working properly, then you can remove this option.

If you have called your ppp options file anything other than `/etc/ppp/options` or `/etc/ppp/options.ttySx`, specify the file name with the `file` option to pppd - e.g.

```
exec /usr/sbin/pppd debug file options.myserver /dev/ttyS0 38400 \
```

16. Testing your connection script

Open a new root Xterm (if you are in X) or open a new virtual console and log in as root.

In this new session, issue the command

```
tail -f /var/log/messages
```

(or whatever your system log file is).

In the first window (or virtual console) issue the command

```
ppp-on &
```

(or whatever name you have called your edited version of /usr/sbin/ppp- on). If you do not put the script into the background by specifying & at the end of the command, you will not get your terminal prompt back until ppp exits (when the link terminates).

Now switch back to the window that is tracking your system log.

You will see something like the following (provided you specified -v to chat and -d to pppd)....this is the chat script and responses being logged to the system log file followed by the start up information for pppd :-

---

```
Oct 21 16:09:58 hwin chat[19868]: abort on (NO CARRIER)
Oct 21 16:09:59 hwin chat[19868]: abort on (BUSY)
Oct 21 16:09:59 hwin chat[19868]: send (ATZ^M)
Oct 21 16:09:59 hwin chat[19868]: expect (OK)
Oct 21 16:10:00 hwin chat[19868]: ATZ^M^M
Oct 21 16:10:00 hwin chat[19868]: OK -- got it
Oct 21 16:10:00 hwin chat[19868]: send (ATDT722298^M)
Oct 21 16:10:00 hwin chat[19868]: expect (CONNECT)
Oct 21 16:10:00 hwin chat[19868]: ^M
Oct 21 16:10:22 hwin chat[19868]: ATDT722298^M^M
Oct 21 16:10:22 hwin chat[19868]: CONNECT -- got it
Oct 21 16:10:22 hwin chat[19868]: send (^M)
Oct 21 16:10:22 hwin chat[19868]: expect (ogin:)
Oct 21 16:10:23 hwin chat[19868]: kepler login: -- got it
Oct 21 16:10:23 hwin chat[19868]: send (hartr^M)
Oct 21 16:10:23 hwin chat[19868]: expect (ssword:)
Oct 21 16:10:23 hwin chat[19868]:  hartr^M
Oct 21 16:10:23 hwin chat[19868]: Password: -- got it
```

TOC

```
Oct 21 16:10:23 hwin chat[19868]: send (??????^M)
Oct 21 16:10:23 hwin chat[19868]: expect (hartr)
Oct 21 16:10:24 hwin chat[19868]: [hartr -- got it
Oct 21 16:10:24 hwin chat[19868]: send (ppp^M)
Oct 21 16:10:27 hwin pppd[19872]: pppd 2.1.2 started by root, uid 0
Oct 21 16:10:27 hwin pppd[19873]: Using interface ppp0
Oct 21 16:10:27 hwin pppd[19873]: Connect: ppp0 <--> /dev/cua1
Oct 21 16:10:27 hwin pppd[19873]: fsm_sdata(LCP): Sent code 1, id 1.
Oct 21 16:10:27 hwin pppd[19873]: LCP: sending Configure-Request, id 1
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfreq(LCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd MRU
Oct 21 16:10:27 hwin pppd[19873]: (1500)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd ASYNCMAP
Oct 21 16:10:27 hwin pppd[19873]: (0)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd MAGICNUMBER
Oct 21 16:10:27 hwin pppd[19873]: (a098b898)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd PCOMPRESSION
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd ACCOMPRESSION
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: returning CONFACK.
Oct 21 16:10:27 hwin pppd[19873]: fsm_sdata(LCP): Sent code 2, id 1.
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfack(LCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 1, id 1.
Oct 21 16:10:27 hwin pppd[19873]: IPCP: sending Configure-Request, id 1
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfreq(IPCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: ipcp: received ADDR
Oct 21 16:10:27 hwin pppd[19873]: (10.144.153.51)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: ipcp: received COMPRESSTYPE
Oct 21 16:10:27 hwin pppd[19873]: (45)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: ipcp: returning Configure-ACK
Oct 21 16:10:28 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 2, id 1.
Oct 21 16:10:30 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 1, id 1.
Oct 21 16:10:30 hwin pppd[19873]: IPCP: sending Configure-Request, id 1
Oct 21 16:10:30 hwin pppd[19873]: fsm_rconfreq(IPCP): Rcvd id 255.
Oct 21 16:10:31 hwin pppd[19873]: ipcp: received ADDR
Oct 21 16:10:31 hwin pppd[19873]: (10.144.153.51)
Oct 21 16:10:31 hwin pppd[19873]:  (ACK)
Oct 21 16:10:31 hwin pppd[19873]: ipcp: received COMPRESSTYPE
Oct 21 16:10:31 hwin pppd[19873]: (45)
Oct 21 16:10:31 hwin pppd[19873]:  (ACK)
```

```
Oct 21 16:10:31 hwin pppd[19873]: ipcp: returning Configure-ACK
Oct 21 16:10:31 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 2, id 2
Oct 21 16:10:31 hwin pppd[19873]: fsm_rconfack(IPCP): Rcvd id 1.
Oct 21 16:10:31 hwin pppd[19873]: ipcp: up
Oct 21 16:10:31 hwin pppd[19873]: local  IP address 10.144.153.104
Oct 21 16:10:31 hwin pppd[19873]: remote IP address 10.144.153.51
```

(Note - I am using STATIC IP numbers - hence my machine sent that to the PPP server - you won't see this if you are using DYNAMIC IP numbers.) Also, this server requires a specific command to start ppp at its end.

This looks OK - so test it out as before with pings to IP numbers and host names.

Fire up you web browser or whatever and go surfing - you are connected!

---

17. Shutting down the PPP link

When you have finished with the PPP link, use the standard ppp-off command to shut it down (remember - you need to be root or a member of the PPP group!).

In your system log you will see something like:-

```
Oct 21 16:10:45 hwin pppd[19873]: Interrupt received: terminating link
Oct 21 16:10:45 hwin pppd[19873]: ipcp: down
Oct 21 16:10:45 hwin pppd[19873]: default route ioctl(SIOCDELRT): Bad address
Oct 21 16:10:45 hwin pppd[19873]: fsm_sdata(LCP): Sent code 5, id 2.
Oct 21 16:10:46 hwin pppd[19873]: fsm_rtermack(LCP).
Oct 21 16:10:46 hwin pppd[19873]: Connection terminated.
Oct 21 16:10:46 hwin pppd[19873]: Exit.
```

Don't worry about the `SIOCDELRT` - this is just pppd noting that it is terminating and is nothing to worry about.

18. Debugging

There are any number of reasons that your connection does not work - chat has failed to complete correctly, you have a dirty line, etc. So check your syslog for indications.

18.1 I have compiled PPP support into the kernel, but...

A very common problem is that people compile PPP support into the kernel and yet when they try to run pppd, the kernel complains that it does not support ppp! There are a variety of reasons this can occur.

Are you booting the right kernel?Whilst you have recompiled your kernel to support ppp, youare not booting the new kernel. This can happen if you do not update`/etc/lilo.conf` and rerun lilo.

A good check on the kernel can be obtained by issuing the command `uname -a`, which should produce a line like

```
Linux archenland 2.0.28 #2 Thu Feb 13 12:31:37 EST 1997 i586
```

This gives the kernel version and the date on which this kernel was compiled - which should give you a pretty good idea of what is going on.

Did you compile ppp kernel support as a module?

If you compiled your kernel ppp support as a module, but did not make and install the modules, then you can get this error. Check the kernel-HOWTO and the README file in `/usr/src/linux`!

Another module connected possibility is that you are expecting required modules to be automatically loaded, but are not running the `kerneld` daemon (which auto-loads and unloads modules on the fly). Check the kerneld mini-HOWTO for information on setting up kerneld.

Are you using the correct version of PPP for your kernel?

You **must** use ppp-2.2 with kernel version 2.0.x. You can use ppp-2.2 with kernel version 1.2.x (if you patch the kernel) otherwise you must use ppp-2.1.2.

TOC

Are you running pppd as root?

If you are not running pppd as the root user (and pppd is not suid to root), you can receive this message.

18.2 My modem connects but ppp never starts up

There are innumerable variations on this (take a look in comp.os.linux...).

A **VERY** common mistake is that you have mistyped something in your scripts. The only thing to do here is to make sure you are logging the chat conversation between you Linux PC and the server into your syslog (/var/log/messages) and then go through this *line by line* to make. You may need to dial into the ppp server manually to check things out again.

You need to check the log against the actual prompts very carefully - and bear in mind that we humans have a tendency to read what we THINK we have typed - not what is actually there!

18.3 The syslog says "`serial line is not 8 bit clean...`"

There are variations on this too - such as `serial line looped back` etc., and the cause can be one (or a sequence) of a number of things.

To understand what is going on here, it is necessary to grasp a bit of what is going on behind the scenes in pppd itself.

When pppd starts up, it sends LCP (link control protocol) packets to the remote machine. If it receives a valid response it then goes on to the next stage (using IPCP - IP control protocol packets) and only when this negotiation completes is the actual IP layer started so that you can use the PPP link.

If there is no ppp server operating at the remote end when your PC sends lcp packets, these get reflected by the login process at the far end. As these packets use 8 bits, reflecting them strips the 8th bit (remember, ASCII is a 7 bit code). PPP sees this and complains accordingly.

There are several reasons this reflection can occur.

You are not correctly logging into the server

When your chat script completes, pppd starts on your PC. However, if you have not completed the log in process to the server (including sending any command required to start PPP on the server), PPP will not start.

So, the lcp packets are reflected and you receive this error.

You need to carefully check and correct (if necessary) your chat script (see above).

You are not starting PPP on the server

Some PPP servers require you to enter a command and/or a RETURN after completing the log in process before the remote end starts ppp.

Check your chat script (see above).

If you log in manually and find you need to send a RETURN after this to start PPP, simply add a blank expect/send pair to the end of your chat script (an empty send string actually sends a RETURN).

The remote PPP process is slow to start

This one is a bit tricksy!

By default, your Linux pppd is compiled to send a maximum of 10 lcp configuration requests. If the server is a bit slow to start up, all 10 such requests can be sent before the remote PPP is ready to receive them.

On your machine, pppd sees all 10 requests reflected back (with the 8th bit stripped) and exits.

There are two ways round this:-

Add `lcp-max-configure 30` to your ppp options. This increases the maximum number of lcp configure packets pppd sends before giving up. For really slow server, you may need even more than this.

Alternatively, you can get a bit tricksy in return. You may have noticed that when you logged in by hand to the PPP server and PPP started there, the **first** character of the ppp garbage that appears was always the tilde character (~).

Using this knowledge we can add a new `expect/send` pair to the end of the chat script which expects a tilde and sends nothing. This would look like:-

---

```
\~       ''
```

Note: as the tilde character has a special meaning in the shell, it must be escaped (and hence the leading backslash).

18.4 Default route not set

If pppd refuses to set up a default route, it is because (quite correctly) it refuses remove/replace an existing default route.

The usual reason that this error occurs is that some distributions set up a default route via your Ethernet card as opposed to setting up a specific network route.

TOC

See the Linux NAG and the Net2/3 HOWTOs for information on correctly setting up your Ethernet c and associated routes.

An alternative to this is that your LAN uses a gateway/router already and your routing table has been set up to point the default route at this.

Fixing up this last situation can require a fair bit of IP networking knowledge and is beyond the scope of this HOWTO. It is suggested you obtain some expert advice (via the news groups of from someone locally you can ask).

18.5 Other Problems

There are many reasons apart from these that ppp fails to connect and/or operate properly.

Look in the PPP FAQ (which is really a series of questions and answers). This is a very comprehensive document and the answers ARE there! From my own (sad) experience, if the answer to your problems is not there, the problem is NOT ppp's fault! In my case I was using an ELF kernel that I had not upgraded to the appropriate kernel modules. I only wasted about 2 days (and most of one night) cursing what had been a perfect PPP server before the light dawned!

---

19. Getting Help when totally stuck

If you can't get your PPP link to work, go back through this document and check everything - in conjunction with the output created by "chat-v..." and "pppd -d" in you system log.

Also consult the PPP documentation and FAQ plus the other documents mention herein!

If you are still stuck, try the comp.os.linux.misc and comp.os.linux.networking newsgroups are reasonably regularly scanned by people that can help you with PPP as is comp.protocols.ppp

You can try sending me personal email, but I do have a day job (and a life) and I do not guarantee to respond quickly (if at all) as this depends on my current work load and the state of my private life!

In particular - **DO NOT POST REAMS OF DEBUGGING OUTPUT TO THE NEWS GROUPS NOR SEND IT TO ME BY EMAIL** - the former wastes huge amounts of network bandwidth and the latter will be consigned to /dev/null (unless I have specifically requested it).

20. Common Problems once the link is working

One problem you will find is that many service providers will only support the connection software package that they distribute to new accounts. This is (typically) for Microsoft Windows :-(- and many service provider help desks seem to know nothing about Unix (or Linux). So, be prepared for limited assistance from them!

You could of course do the individual a favour and educate then about Linux (any ISP help desk person should be reasonably 'with it' in Internet terms and that means they should have a home Linux box - of course it does)!

20.1 I can't see beyond the PPP server I connect to

OK - your PPP connection is up and running and you can ping the PPP server by IP number (the second or "remote" IP number shown by `ifconfig ppp0`), but you can't reach anything beyond this.

First of all, try pinging the IP numbers you have specified in /etc/resolv.conf as name servers. If this works, you **can** see beyond your PPP server (unless this has the same IP number as the "remote" IP number of your connection). So now try pinging the full Internet name of your service provider - eg

```
ping my.provider.net.au
```

If this does NOT work, you have a problem with the name resolution. This is probably because of a typo in your /etc/resolv.conf file. Check this carefully against the information you acquired by ringing your service provider. If all looks OK, ring your service provider and check that you wrote down the IP numbers correctly.

If it STILL doesn't work (and your service provider confirms that his name servers are up and running), you have a problem somewhere else - and I suggest you check carefully through your Linux installation (looking particularly for file permissions).

If you STILL can't ping your service provider's IP name servers by IP number, either they are down (give them a voice call and check) or there is a routing problem at your service provider's end. Again, ring them and check this out.

One possibility is that the "remote end" is a Linux PPP server where the IP forwarding option has not been specified in the kernel!

A good general test is to try connecting to your service provider using the software that most supply for (gulp) Microsoft Windows. If everything works from another operating system to exactly the same account, then the problem is with your Linux system and NOT your service provider.

TOC

20.2 I can send email, but not receive it

If you are using dynamic IP numbers, this is perfectly normal. See "Setting up Services" below.

20.3 Why can't people finger, WWW, gopher, talk etc to my machine?

Again, if you are using dynamic IP numbers, this is perfectly normal. See "Setting up Services" below.

---

21. Using Internet services with Dynamic IP numbers

If you are using dynamic IP numbers (and many service providers will only give you a dynamic IP number unless you pay significantly more for your connection), then you have to recognise the limitations this imposes.

First of all, outbound service requests will work just fine. That is, you can send email using sendmail (provided you have correctly set up sendmail), ftp files from remote sites, finger users on other machines, browse the web etc.

In particular, you can answer email that you have brought down to your machine whilst you are off line. Mail will simply sit in your mail queue until you dial back into your ISP.

However, your machine is NOT connected to the Internet 24 hours a day, nor does it have the same IP number every time it is connected. So it is impossible for you to receive email directed to your machine, and very difficult to set up a web or ftp server that your friends can access! As far as the Internet is concerned your machine does not exist as a unique, permanently contactable machine as it does not have a unique IP number (remember - other machines will be using the IP number when they are allocated it on dial in).

If you set up a WWW (or any other server), it is totally unknown by any user on the Internet UNLESS they know that your machine is connected AND its actual (current) IP number. There are a number of ways they can get this info, ranging from you ringing them, sending them email to tell them or cunning use of ".plan" files on a shell account at your service provider (assuming that your provider allows shell and finger access).

Now, for most users, this is not a problem - all that most people want is to send and receive email (using your account on your service provider) and make outbound connections to WWW, ftp and other servers on the Internet. If you MUST have inbound connections to your server, you should really get a static IP number. Alternatively you can explore the methods hinted at above...

21.1 Setting up email

Even for dynamic IP numbers, you can certainly configure sendmail on your machine to send out any email that you compose locally. Configuration of sendmail can be obscure and difficult - so this document does not attempt to tell you how to do this. However, you should probably configure sendmail so that your Internet service provider is designated as your "smart relay" host (the `sendmail.cf` **DS** option). (For more sendmail configuration info, see the sendmail documents - and look at the m4 configurations that come with sendmail. There is almost certain to be one there that will meet your needs).

There are also excellent books on Sendmail (notably the 'bible' from O'Reilly and Associates), but these

are almost certainly overkill for most users!

Once you have sendmail configured, you will probably want to have sendmail dispatch any messages that have been sitting in the outbound mail queue as soon as the PPP connection comes up. To do this, add the command

```
sendmail -q &
```

to your /etc/ppp/ip-up script (see below).

Inbound email is a problem for dynamic IP numbers. The way to handle this is to:-

- configure your mail user agent so that all mail is sent out with a "reply to" header giving your email address at your Internet Service provider.
  If you can, you should also set your FROM address to be your email address at your ISP as well.
- use the popclient, fetchmail programs to retrieve your email from your service provider.
  Alternatively, if your ISP is using IMAP, use an IMAP enabled mail user agent (such as pine).

You can automate this process at dial up time by putting the necessary commands in the `/etc/ppp/ip-up` script (see below).

21.2 Setting Up a local Name server

Whilst you can quite happily use the domain name servers located at your ISP, you can also set up a local caching only (secondary) name server that is brought up by the ip-up script. The advantage of running a local (caching only) name server is that it will save you time (and bandwidth) if you frequently contact the same sites during a long on-line session.

DNS configuration for a caching only nameserver (that uses a "forwarders' line in the named.boot file pointing at your ISPs DNS) is relatively simple. The O'Reilly book (DNS and Bind) explains all you want to know about this.

There is also a DNS-HOWTO available.

If you are running a small LAN that can access the Internet through you Linux PC (using IP Masquerade for example), it is probably a good idea to run a local name server (with a forwarders directive) whilst the link is up as this will minimise the bandwidth and delays associated with name resolution.

One point of Nettiquette: ask permission from your ISP before you start using a secondary, caching only name server in your ISP's domain. Properly configured, your DNS will not cause any problems to your ISP at all, but if you get things wrong, it can cause problems.

---

22. Linking two networks using PPP

There is basically no difference between linking a single Linux PC to a PPP server and linking two LANs using PPP on a machine on each LAN. Remember, PPP is a **peer to peer** protocol.

However, you **DEFINITELY** need to understand about how routing is established. Read the NET-2 howto and the Linux Network Administrator Guide (NAG). You will also find " TCP/IP Network Administration" (published by O'Reilly and Assoc - ISBN 0-937175-82-X) to be of invaluable assistance.

If you are going to be sub networking an IP network number on either side of the link, you will also find the Linux (draft) sub networking mini-howto) to be of use. This is available at Linux Sub networking mini-HOWTO.

In order to link two LANs, you **must** be using different IP network numbers (or subnets of the same network number) and you will need to use static IP numbers - or use IP masquerade. If you want to use IP masquerade, see the IP masquerade mini-howto for instructions on setting that up.

22.1 Setting up the IP numbers

Arrange with the network administrator of the other LAN the IP numbers that will be used for each end of the PPP interface. If you are using static IP numbers, this will also probably require you to dial into a specific telephone number.

Now edit the appropriate `/etc/ppp/options[.ttyXX]` file - it's a good idea to have a specific modem and port at your end for this connection. This may well require you to change your `/etc/ppp/options` file - and create appropriate options.ttyXX files for any other connections!

Specify the IP numbers for your end of the PPP link in the appropriate options file exactly as shown above for static IP numbers.

22.2 Setting up the routing

You must arrange that packets on your local LAN are routed across the interface that the PPP link establishes. This is a two stage process.

First of all, you need to establish a route from the machine running the PPP link to the network(s) at the far end of the link. If the link is to the Internet, this can be handled by a default route established by pppd itself at your end of the connection using the 'defaultroute' option to pppd.

If however, the link is only linking two LANs, then a specific network route must be added for each network that is accessible across the link. This is done using a 'route' command for each network in the /etc/ppp/ip-up script (see After the link comes up...) for instructions on doing this.

The second thing you need to do is to tell the other computers on your LAN that your Linux computer is actually the 'gateway' for the network(s) at the far end of the ppp link.

Of course, the network administrator at the other end of the link has to do all this too! However, as s/he will be routing packets to your specific networks, a **specific network route** will be required, not a default route (unless the LANs at the far and of the link are linking into you to access the Internet across your connection).

22.3 Network security

If you are linking you LAN to the Internet using PPP - or even just to a "foreign" LAN, you need to think about security issues. I strongly urge you to think about setting up a firewall!

You should also speak to the LAN administrator at your site **BEFORE** you start linking to foreign LANs or the Internet this way. Failure to do so could earn you anything from no reaction to really serious trouble!

---

23. After the link comes up - the `/etc/ppp/ip-up` script

Once the PPP link is established, pppd looks for `/etc/ppp/ip-up`. If this script exists and is executable, the PPP daemon executes the script. This allows you to automate any special routing commands that may be necessary and any other actions that you want to occur every time the PPP link is activated.

This is just a shell script and can do anything that a shell script can do (i.e. virtually anything you want).

For example, you can get sendmail to dispatch any waiting outbound messages in the mail queue.

Similarly, you can insert the commands into ip-up to collect (using pop) any email waiting for you at your ISP.

There are restrictions on `/etc/ppp/ip-up`:-

- It runs in a deliberately restricted environment to enhance security. This means you must give a full path to binaries etc.
- Technically, `/etc/ppp/ip-up` is a *program* not a script. This means it can be directly executed - and hence it requires the standard file magic (`#!/bin/bash`) at the start of the first line and must be readable and executable by root.

23.1 Special routing

If you are linking two LANs, you will need to set up specific routes to the 'foreign' LANs. This is easily done using the `/etc/ppp/ip-up` script. The only difficulty arises if your machine handles multiple PPP links.

This is because the `/etc/ppp/ip-up` is executed for EVERY ppp connection that comes up, so you need to carefully execute the correct routing commands for the particular link that comes up - and not when any other link comes up!

23.2 Handling email queues

When the link between two LANs comes up, you may well want to make sure that email that is queued at either end is *flushed* - sent out to its destination. This is done by adding the appropriate `sendmail` invocation.

Using the bash 'case' statement on an appropriate parameter that pppd passes into the script accomplishes this. For example, this is the `/etc/ppp/ip-up` script I use to handle our WAN links and the link to my home Ethernet (also handled on the same ppp server).

23.3 A sample `/etc/ppp/ip-up` script

The example below provides a variety of example uses.

```
#!/bin/bash
#
# Script which handles the routing issues as necessary for pppd
# Only the link to Newman requires this handling.
#
# When the ppp link comes up, this script is called with the following
# parameters
```

TOC

```
#       $1      the interface name used by pppd (e.g. ppp3)
#       $2      the tty device name
#       $3      the tty device speed
#       $4      the local IP address for the interface
#       $5      the remote IP address
#       $6      the parameter specified by the 'ipparam' option to pppd
#
case "$5" in
# Handle the routing to the Newman Campus server
        202.12.126.1)
                /sbin/route add -net 202.12.126.0 gw 202.12.126.1
# and flush the mail queue to get their email there asap!
                /usr/sbin/sendmail -q &
                ;;
        139.130.177.2)
# Our Internet link
# When the link comes up, start the time server and synchronise to the world
# provided it is not already running
                if [ ! -f /var/lock/subsys/xntpd ]; then
                        /etc/rc.d/init.d/xntpd.init start &
                fi
# Start the news server (if not already running)
                if [ ! -f /var/lock/subsys/news ]; then
                        /etc/rc.d/init.d/news start &
                fi
                ;;
        203.18.8.104)
# Get the email down to my home machine as soon as the link comes up
# No routing is required as my home Ethernet is handled by IP
# masquerade and proxyarp routing.
                /usr/sbin/sendmail -q &
                ;;
        *)
esac
exit 0
```

As a result of bringing up the ppp link to our Newman campus and this script, we end up with the following routing table entries (this machine also is our general dial up PPP server AND handles our Internet link). I have interspersed comments in the output to help explain what each entry is) :-

---

```
[root@kepler /root]# route -n
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
# the HOST route to our remote internet gateway
139.130.177.2   *               255.255.255.255 UH    1500   0      134 ppp4
# the HOST route to our Newman campus server
202.12.126.1    *               255.255.255.255 UH    1500   0       82 ppp5
# the HOST route to my home ethernet
203.18.8.104    *               255.255.255.255 UH    1500   0       74 ppp3
# two of our general dial up PPP lines
```

```
203.18.8.64     *               255.255.255.255 UH    552    0        0 ppp2
203.18.8.62     *               255.255.255.255 UH    552    0        1 ppp1
# the specific network route to the Newman campus LAN
202.12.126.0    202.12.126.1    255.255.255.0   UG    1500   0        0 ppp5
# the route to our local Ethernet (super-netting two adjacent C classes)
203.18.8.0      *               255.255.254.0   U     1500   0     1683 eth0
# the route to the loop back device
127.0.0.0       *               255.0.0.0       U     3584   0      483 lo
# the default route to the Internet
default         139.130.177.2   *               UG    1500   0     3633 ppp4
```

23.4 Handling email

The previous section shows how to handle the outgoing mail - simply by flushing the mail queue once the link is up.

If you are running a WAN link, you can arrange with the network administrator of the remote LAN to do exactly the same thing. For example, at the Newman Campus end of our WAN link, the `/etc/ppp/ip-up` script looks like :-

```
#!/bin/bash
#
# Script which handles the routing issues as necessary for pppd
# Only the link to Hedland requires this handling.
#
# When the ppp link comes up, this script is called with the following
# parameters
#       $1      the interface name used by pppd (e.g. ppp3)
#       $2      the tty device name
#       $3      the tty device speed
#       $4      the local IP address for the interface
#       $5      the remote IP address
#       $6      the parameter specified by the 'ipparam' option to pppd
#
case "$5" in
        203.18.8.4)
                /usr/sbin/sendmail -q
                ;;
        *)
esac
exit 0
```

If however you have only a dynamic IP PPP link to your ISP, you need to get your email from the account on your ISPs machine. This is usually done using the POP (Post Office Protocol). This process can be handled using the 'popclient' program - and the ip-up script can automate this process for you too!

Simply create a `/etc/ppp/ip-up` script that contains the appropriate invocation of popclient. For my laptop that runs Red Hat Linux (which I take on any travels), this is

```
popclient -3 -c -u hartr -p <password> kepler.hedland.edu.au |formail -s procmail
```

You could use slurp or whatever to do the same for news, and so forth. Remember, the ip-up script is just a standard script and so can be used to automate ANY function that needs to be accomplished every time the appropriate PPP link comes up.

24. Using `/etc/ppp/ip-down`

You can create a script that will be executed once the link has been terminated. This is stored in `/etc/ppp/ip-down`. It can be used to undo anything special that you did in the corresponding /etc/ppp/ip-up script.

25. Routing issues on a LAN

If you are connected to a LAN but still want to use PPP on your personal Linux machine , you need to address some issues of the routes packets need to take from your machine to reach your LAN (through your Ethernet interface) and also to the remote PPP server and beyond.

This section does NOT attempt to teach you about routing - it deals only with a simple, special case of (static) routing!

I strongly urge you to read the Linux Network Administrator Guide (NAG) if you are NOT familiar with routing. Also the O'Reilly book "TCP/IP Network Administration" covers this topic in a very understandable form.

The basic rule of static routing is that the DEFAULT route should be the one that points to the MOST number of network addresses. For other networks, enter specific routes to the routing table.

The ONLY situation I am going to cover here is where your Linux box is on a LAN that is not connected to the Internet - and you want to dial out to the Internet for personal use whilst still connected to the LAN.

First of all, make sure that your Ethernet route is set up to the specific network addresses available across your LAN - NOT set to the default route!

Check this by issuing a route command, you should see something like the following:-

```
[root@hwin /root]# route -n
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
loopback        *               255.255.255.0   U     1936   0       50 lo
10.0.0.0        *               255.255.255.0   U     1436   0      565 eth0
```

If your Ethernet interface (eth0) is pointing at the default route, (the first column will show "default" in the eth0 line) you need to change your Ethernet initialisation scripts to make it point at the specific network numbers rather than the default route (consult the Net2 HOWTO and NAG).

This will allow pppd to set up your default route as shown below:-

```
[root@hwin /root]# route -n
Kernel routing table

Destination     Gateway         Genmask         Flags MSS    Window Use Iface
10.144.153.51   *               255.255.255.255 UH    488    0        0 ppp0
127.0.0.0       *               255.255.255.0   U     1936   0       50 lo
10.1.0.0        *               255.255.255.0   U     1436   0      569 eth0
default         10.144.153.51   *               UG    488    0        3 ppp0
```

As you can see, we have a host route to the PPP server (10.144.153.51) via ppp0 and also a default network route that uses the PPP server as its gateway.

If your set up needs to be more complex than this - read the routing documents already mentioned and consult an expert at your site!

If your LAN already has routers on it, you will already have gateways established to the wider networks available at your site. You should STILL point your default route at the PPP interface - and make the other routes specific to the networks they serve.

25.1 Note on Security

When you set up a Linux box on an existing LAN to link into the Internet, you are potentially opening your entire LAN to the Internet - and the hackers that reside there. Before you do this, I strongly urge you to consult your network administrator and site security policy. If your PPP connection to the Internet is used to successfully attack your site, you will at the very least earn the intense anger of your fellow users, network and system administrators. You may also find yourself in very much more serious trouble!

Before you connect a LAN to the Internet, you should consider the security implications of even a DYNAMIC connection - hence the earlier reference to the O'Reilly "Building Internet Firewalls"!

---

26. Setting up a PPP server

As already mentioned, there are many ways to do this. What I present here is the way I do it (using a Cyclades multi-port serial card) and a rotary dial in set of telephone lines.

If you don't like the method I present here, please feel free to go your own way. I would however, be pleased to include additional methods in future versions of the HOWTO. So, please send me your comments and methods!

Please note, this section only concerns setting up Linux as a PPP server. I do not (ever) intend to include information on setting up special terminal servers and such.

Also, I have yet to experiment with shadow passwords (but will be doing so sometime). Information currently presented does NOT therefore include any bells and whistles that are required by the shadow suite.

26.1 Kernel compilation

All the earlier comments regarding kernel compilation and kernel versions versus pppd versions apply. This section assumes that you have read the earlier sections of this document!

For a PPP server, you **MUST** include IP forwarding in your kernel. You may also wish to include other capabilities (such as IP fire walls, accounting etc etc).

If you are using a multi-port serial card, then you must obviously include the necessary drivers in your kernel too!

26.2 Overview of the server system

We offer dial up PPP (and SLIP) accounts and shell accounts using the same user name/password pair. This has the advantages (for us) that a user requires only one account and can use it for all types of connectivity.

As we are an educational organisation, we do not charge our staff and students for access, and so do not have to worry about accounting and charging issues.

We operate a firewall between our site and the Internet, and this restricts some user access as the dial up lines are inside our (Internet) firewall (for fairly obvious reasons, details of our other internal fire walls are not presented here and are irrelevant in any case).

The process a user goes through to establish a PPP link to our site (once they have a valid account of course) is :-

- Dial into our rotary dialer (this is a single phone number that connects to a bank of modems - the

TOC

first free modem is then used).

- Log in using a valid user name and password pair.
- At the shell prompt, issue the command `ppp` to start PPP on the server.
- Start PPP on their PC (be it running Windows, DOS, Linux MAC OS or whatever - that is their problem).

The server uses individual `/etc/ppp/options.ttyXX` files for each dial in port that set the remote IP number for dynamic IP allocation. The server users proxyarp routing for the remote clients (set via the appropriate option to pppd). This obviates the need for routed or gated.

When the user hangs up at their end, pppd detects this and tells the modem to hang up, bringing down the PPP link at the same time.

26.3 Getting the software together

You will need the following software:-

- Linux, properly compiled to include the necessary options.
- The appropriate version of pppd for your kernel.
- A 'getty' program that intelligently handles modem communications.
  We use getty_ps2.0.7h, but mgetty is highly thought of. I understand that mgetty can detect a call that is using pap/chap (pap is the standard for Windows95) and invoke pppd automatically, but I have yet to explore this.
- An operational domain name server (DNS) that is accessible to your dial up users.
  You should really be running your own DNS if possible...

26.4 Setting up standard (shell access) dialup.

Before you can set up your PPP server, your Linux box must be capable of handling standard dial up access.

This howto does NOT cover setting this up. Please see the documentation of the getty of your choice and serial HOWTO for information on this.

26.5 Setting up the PPP options files

You will need to set up the overall `/etc/ppp/options` with the common options for all dial up ports. The options we use are:-

---

```
asyncmap 0
netmask 255.255.254.0
proxyarp
```

```
lock
crtscts
modem
```

Note - we do NOT use any (obvious) routing - and in particular there is no defaultroute option. The reason for this is that all you (as a PPP server) are required to do is to route packets **from** the ppp client out across your LAN/Internet and route packets **to** the client from your LAN and beyond.

All that is necessary for this is a host route to the client machine and the use of the 'proxyarp' option to pppd.

The 'proxyarp' option sets up (surprise) a proxy ARP entry in the PPP server's ARP table that basically says 'send all packets destined for the PPP client to me'. This is the easiest way to set up routing to a single PPP client - but you cannot use this if you are routing between two LANs - you must add proper network routes which can't use proxy ARP.

You will almost certainly wish to provide dynamic IP number allocation to your dial up users. You can accomplish this by allocating an IP number to each dial up port. Now, create a `/etc/ppp/options.ttyXX` for each dial up port.

In this, simply put the local (server) IP number and the IP number that is to be used for that port. For example

```
kepler:slip01
```

In particular, note that you can use valid host names in this file (I find that I only remember the IP numbers of critical machines and devices on my networks - names are more meaningful)!

26.6 Setting pppd up to allow users to (successfully) run it

As starting a ppp link implies configuring a kernel device (a network interface) and manipulating the kernel routing tables, special privileges are required - in fact full root privileges.

Fortunately, pppd has been designed to be 'safe' to run set uid to root. So you will need to

```
chmod u+s /usr/sbin/pppd
```

When you list the file, it should then appear as

```
-rwsr-xr-x   1 root     root        74224 Apr 28 07:17 /usr/sbin/pppd
```

TOC

If you do not do this, users will be unable to set up their ppp link.

26.7 Setting up the global alias for pppd

In order to simplify things for our dial up PPP users, we create a global alias (in /etc/bashrc) so that one simple command will start ppp on the server once they are logged in.

This looks like

```
alias ppp="exec /usr/sbin/pppd -detach"
```

What this does is

- exec : this means replace the running program (in this case the shell) with the program that is run.
- pppd -detach : start up pppd and do NOT fork into the background. This ensures that when pppd exits there is no process hanging around.

When a user logs in like this, they will appear in the output of 'w' as

```
  6:24pm  up 3 days,  7:00,  4 users,  load average: 0.05, 0.03, 0.00
User     tty       login@  idle   JCPU   PCPU  what
hartr    ttyC0     3:05am  9:14                -
```

And that is it...I told you this was a simple, basic PPP server system!

27. Using PPP across a null modem (direct serial) connection

This is very simple - there is no modem in the way so things are much simpler.

First of all, choose one of the machines as a 'server', setting up a getty on the serial port so you can test that you do have connectivity using minicom to access the serial port on the 'client'.

Once you have this functioning, you can remove the getty UNLESS you want to make sure that the connection is validated using user name/password pairs as for a dial up connection. As you have 'physical control' of both machines, I will presume that you do NOT want to do this.

Now, on the server, remove the getty and make sure that you have the serial ports on both machines configured correctly using 'setserial'.

All you need to do now is to start pppd on both systems. I will assume that the connection uses /dev/ttyS34 on both machines. So, on both machines execute the command:-

```
pppd -detach crtscts lock <local IP>:<remote IP> /dev/ttyS3 38400 &
```

This will bring up the link - but as yet you have no routing specified. You can test the link by pinging to and fro to each machine. If this works, bring down the link by killing one of the pppd processes.

The routing you need will of course depend on exactly what you are trying to do. Generally, one of the machines will be connected to an Ethernet (and beyond) and so the routing required is exactly the same as for a PPP server and client.

So on the Ethernet equipped machine, the pppd command would be

```
pppd -detach crtscts lock proxyarp <local IP>:<remote IP> /dev/ttyS3 38400 &
```

and on the other machine

```
pppd -detach crtscts lock defaultroute <local IP>:<remote IP> /dev/ttyS3 38400 &
```

If you are linking two networks (using a serial link!) or have more complex routing requirements, you can use /etc/ppp/ip-up in exactly the same way as mentioned earlier in this document.

Robert Hart
Port Hedland, Western Australia
Melbourne, Victoria, Australia August/October 1996 January/March 1997

The Linux Printing HOWTO

Grant Taylor <gtaylor+pht@picante.com>

v3.29, 9 March 1999

This is the Linux Printing HOWTO, a collection of information on how to generate, preview, print and fax anything under Linux (and other Unices in general).

1. Introduction

- 1.1 History
- 1.2 Copyright

2. How to print

3. Kernel printer devices

- 3.1 The lp device (kernels <=2.1.32)
- 3.2 The parport device (kernels >= 2.1.33)
- 3.3 Serial devices

4. Supported Printers

- 4.1 Postscript
- 4.2 Non-Postscript
- 4.3 What printers work?

5. Which spooling software?

TOC

6. How it works, basic

7. How to set things up, basic

- 7.1 Traditional lpd configuration
- 7.2 File Permissions

8. Getting Printing Software

9. Vendor Solutions

- 9.1 Red Hat
- 9.2 Debian
- 9.3 Other Distributions

10. Ghostscript.

- 10.1 Invoking Ghostscript
- 10.2 Ghostscript output tuning

11. How to print to a printer over the network

- 11.1 To a Unix/lpd host
- 11.2 To a Win95, WinNT, LanManager, or Samba printer
- 11.3 To a NetWare Printer
- 11.4 To an EtherTalk (Apple) printer
- 11.5 To an HP or other ethernet printer
- 11.6 Running an *if* for remote printers
- 11.7 From Windows.
- 11.8 From an Apple.
- 11.9 From Netware.

TOC

12. Windows-only printers

- 12.1 The Ghostscript Windows redirector
- 12.2 The pbm2ppa program
- 12.3 The pbm2l7k program

13. How to print to a fax machine.

- 13.1 Using a faxmodem
- 13.2 Using the Remote Printing Service

14. How to generate something worth printing.

- 14.1 Markup languages
- 14.2 WYSIWYG Word Processors

15. On-screen previewing of printable things.

- 15.1 PostScript
- 15.2 TeX dvi
- 15.3 Adobe PDF

16. Serial printers under lpd

- 16.1 Setting up in printcap
- 16.2 Older serial printers that drop characters

17. Credits

---

1. Introduction

The Printing HOWTO should contain everything you need to know to help you set up printing services on your Linux box(en). As life would have it, it's a bit more complicated than in the point-and-click world of Microsoft and Apple, but it's also a bit more flexible and certainly easier to administer for large LANs.

This document is structured so that most people will only need to read the first half or so. Most of the more obscure and situation-dependant information in here is in the last half, and can be easily located in the Table of Contents, whereas most of the information through section 9 or 10 is probably needed by most people.

Since version 3.x is a complete rewrite, much information from previous editions has been lost. This is by design, as the previous HOWTOs were so large as to be 60 typeset pages, and had the narrative flow of a dead turtle. If you do not find the answer here, you are encouraged to a) scan the previous version at the Printing HOWTO Home Page and b) drop me a note saying what ought to be here but isn't.

The Printing HOWTO Home Page is a good place to find the latest version; it is also, of course, distributed from Metalab (`metalab.unc.edu`) and your friendly local LDP mirror.

1.1 History

This is the third generation, which is to say the third complete rewrite, of the Printing HOWTO. The history of the PHT may be chronicled thusly:

1. I wrote the printing-howto in 1992 in response to too many printing questions in comp.os.linux, and posted it. This predated the HOWTO project by a few months and was the first FAQlet called a `howto'. This edition was in plain ascii.
2. After joining the HOWTO project, the Printing-HOWTO was merged with an Lpd FAQ by Brian McCauley `<B.A.McCauley@bham.ac.uk>`; we continued to co-author the PHT for two years or so. At some point we incorporated the work of Karl Auer `<Karl.Auer@anu.edu.au>`. This generation of the PHT was in TeXinfo, and available in PS, HTML, Ascii, and Info.
3. After letting the PHT rot and decay for over a year, and an unsuccessful attempt at getting someone else to maintain it, this rewrite happened. This generation of the PHT is written in SGML using the LinuxDoc DTD and the SGML-Tools-1 package. Beginning with version 3.27, it incorporates a summary of a companion printer support database; before 3.27 there was never a printer compatability list in this HOWTO (!).

1.2 Copyright

This document is Copyright (c) 1992-1999 by Grant Taylor. Feel free to copy and redistribute this document according to the terms of the GNU General Public License, revision 2 or later.

2. How to print

If you've already got lpd setup to print to your printer, or your system administrator already did so, or your vendor did so for you, then all you need to do is learn how to use the lpr command. The Printing Usage HOWTO covers this, and a few other queue manipulation commands you should probably know.

If, however, you have a new system or new printer, then you'll have to set up printing services one way or another before you can print. Read on!

3. Kernel printer devices

There are two completely different device drivers for the parallel port; which one you are using depends on your kernel version. The driver changed in Linux 2.1.33.

A few details are the same for both styles of driver. Most notably, many people have found that Linux will not detect their parallel port unless they disable "Plug and Play" in their PC BIOS. (This is no surprise; the track record for PnP of non-PCI devices with Windows and elsewhere has been something of a disaster).

3.1 The lp device (kernels <=2.1.32)

The Linux kernel (<=2.1.32), assuming you have compiled in or loaded the lp device (the output of `cat /proc/devices` should include the device lp if it is loaded), provides one or more of */dev/lp0*, */dev/lp1*, and */dev/lp2*. These are NOT assigned dynamically, rather, each corresponds to a specific hardware I/O address. This means that your first printer may be *lp0* or *lp1* depending on your hardware. Just try both.

A few users have reported that their bidirectional lp ports aren't detected if they use an older unidirectional printer cable. Check that you've got a decent cable.

One cannot run the plip and lp drivers at the same time on any given port (under 2.0, anyway). You can, however, have one or the other driver loaded at any given time either manually, or by kerneld with version 2.x (and later 1.3.x) kernels. By carefully setting the interrupts and such, you can supposedly run plip on one port and lp on the other. One person did so by editing the drivers; I eagerly await a success report of someone doing so with only a clever command line.

There is a little utility called `tunelp` floating about with which you, as root, can tune the Linux 2.0 lp device's interrupt usage, polling rate, and other options.

When the lp driver is built into the kernel, the kernel will accept an `lp=` option to set interrupts and io addresses:

```
When the lp driver is built in to the kernel, you may use the
LILO/LOADLIN command line to set the port addresses and interrupts
that the driver will use.

Syntax:      lp=port0[,irq0[,port1[,irq1[,port2[,irq2]]]]]

For example:   lp=0x378,0   or   lp=0x278,5,0x378,7 **

Note that if this feature is used, you must specify *all* the ports
you want considered, there are no defaults.  You can disable a
built-in driver with lp=0.
```

When loaded as a module, it is possible to specify io addresses and interrupt lines on the insmod com line (or in */etc/conf.modules* so as to affect kerneld) using the usual module argument syntax. The parameters are `io=port0,port1,port2` and `irq=irq0,irq1,irq2`. Read ye the man page for *insmod* for more information on this.

**For those of you who (like me) can never find the standard port numbers when you need them, they are as in the second example above. The other port (*lp0*) is at 0x3bc. I've no idea what interrupt it usually uses.

The source code for the Linux 2.0 parallel port driver is in /usr/src/linux/drivers/char/lp.c.

3.2 The parport device (kernels >= 2.1.33)

Beginning with kernel 2.1.33 (and available as a patch for kernel 2.0.30), the lp device is merely a client of the new parport device. The addition of the parport device corrects a number of the problems that plague the old lp device driver - it can share the port with other drivers, it dynamically assigns available parallel ports to device numbers rather than enforcing a fixed correspondence between I/O addresses and port numbers, and so forth.

The advent of the parport device has enabled a whole flock of new parallel-port drivers for things like Zip drives, Backpack CD-ROMs and disks, and so forth. Some of these are also available in versions for 2.0 kernels; look around on the web.

The main difference that you will notice, so far as printing goes, is that parport-based kernels dynamically assign lp devices to parallel ports. So what was lp1 under Linux 2.0 may well be lp0 under Linux 2.2. Be sure to check this if you upgrade from an lp-driver kernel to a parport-driver kernel.

I'll cover the parport driver more completely when I find myself using one, but in the meantime you can read the file Documentation/parport.txt in your kernel sources, or look at the parport web site.

3.3 Serial devices

Serial devices are usually called something like */dev/ttyS1* under Linux. The utility `stty` will allow you to interactively view or set the settings for a serial port; `setserial` will allow you to control a few extended attributes and configure IRQs and I/O addresses for non-standard ports. Further discussion of serial ports under Linux may be found in the Serial-HOWTO.

When using a slow serial printer with flow control, you may find that some of your print jobs get truncated. This may be due to the serial port, whose default behavior is to purge any untransmitted characters from its buffer 30 seconds after the port device is closed. The buffer can hold up to 4096 characters, and if your printer uses flow control and is slow enough that it can't accept all the data from the buffer within 30 seconds after printing software has closed the serial port, the tail end of the buffer's contents will be lost. If the command `cat file > /dev/ttyS2` produces complete printouts for short files but truncated ones for longer files, you may have this condition.

The 30 second interval can be adjusted through the "closing_wait" commandline option of setserial (version 2.12 and later). A machine's serial ports are usually initialized by a call to setserial in the rc.serial

boot file. The call for the printing serial port can be modified to set the closing_wait at the same time sets that port's other parameters.

4. Supported Printers

The Linux kernel mostly supports any printer that you can plug into a serial or parallel port, but there are things to look out for, and printers that you won't be able to use, even though they can (electrically speaking) communicate with Linux. Primary among these incompatible printers are those referred to as "Windows" or "GDI" printers. They are called this because part or all of the printer control language and the design details of the printing mechanism are not documented. Typically the vendor will provide a Windows driver and happily sell only to Windows users; this is why they are called Winprinters. In some cases the vendor also provides drivers for NT, OS/2, or other operating systems.

Many of these printers *do not work* with Linux. A few of them do, and some of them only work a little bit (usually because someone has reverse engineered the details needed to write a driver). See the printer support list below for details on specific printers.

A few printers are in-between. Some of NEC's models, for example, implement a simple form of the standard printer language PCL that allows PCL-speaking software to print at up to 300dpi, but only NEC knows how to get the full 600dpi out of these printers.

Note that if you already have one of these Winprinters, there are roundabout ways to get Linux to print to one, but they're rather awkward and I've never tried it myself. See Section 12 of this document for more discussion of Windows-only printers.

4.1 Postscript

As for what printers *do* work with Linux, the best choice is to buy a printer with native PostScript support. Nearly all Unix software that produces printable output produces it in PostScript, so obviously it'd be nice to get a printer that supports PostScript directly. Unfortunately, PostScript support is scarce outside the laser printer domain, and is sometimes a costly add-on.

Unix software, and the publishing industry in general, have standardized upon Postscript as the printer control language of choice. This happened for several reasons:

Timing

Postscript arrived as part of the Apple Laserwriter, a perfect companion to the Macintosh, the system largely responsible for the desktop publishing revolution of the 80s.

It's device-independant

Postscript programs can be run to generate output on a pixel screen, a vector screen, a fax machine, or almost any sort of printer mechanism, without the original program needing to be changed. Postscript output will look the same on any Postscript device, at least within the limits of the device's capabilities. Before the creation of PDF, people exchanged complex documents online as Postscript files. The only reason this standard didn't "stick" was because Windows machines didn't usually include a Postscript previewer, so Adobe specified hyperlinks and compression for

Postscript, called the result PDF, distributed previewers for it, and invented a market for their "distiller" tools (the functionality of which is also provided by ghostscript's ps2pdf and pdf2ps programs).

It's a real programming language

Postscript is a complete programming language; you can write software to do most anything in it. This is mostly useful for defining subroutines at the start of your program to reproduce complex things over and over throughout your document, like a logo or a big "DRAFT" in the background.

It's open

Postscript is fully specified in a publically available series of books (which you can find at any good bookstore). Although Adobe invented it and provides the dominant commercial implementation, other vendors like Aladdin produce independantly coded implementations as well.

4.2 Non-Postscript

Failing the (larger) budget necessary to buy a Postscript printer, you can use any printer supported by Ghostscript, the free Postscript interpreter used in lieu of actual printer Postscript support. The Ghostscript Home Page has a list of supported printers and information on the status of new and experimental drivers. Note that this page lists supported printers in the latest version of Ghostscript, while most Linux distributions can only ship a somewhat outdated version of Ghostscript due to the license. Fortunately, there is usually a prepackaged up to date Ghostscript made available in each distribution's contrib area. Please help improve the Ghostscript printer support page by reporting your successes and failures as it asks.

Adobe now has a new printer language called "PrintGear". I think it's a greatly simplified binary format language with some Postscript heritage but no Postscript compatibility. And I haven't heard of Ghostscript supporting it. But some PrintGear printers seem to support another language like PCL, and these printers will work with Linux (iff the PCL is implemented in the printer and not in a Windows driver).

4.3 What printers work?

If you want to buy a printer, you can look in several places to see if it will work. The cooperatively maintained Printing HOWTO printer database aims to be a comprehensive listing of the state of Linux printer support. A summary of it is below; be sure to check online for more details and information on what driver to use.

Ghostscript's printer compatibility page has a list of some working printers, as well as links to other pages. And Dejanews contains hundreds of "it works" and "it doesn't work" testimonials. Try all three, and when you're done, check that your printer is present and correct in the database, so that it will be listed properly in this document in the future.

Printer compatibility list

This section is a summary of the online version. The online version includes basic specifications, notes, links to driver information, user-maintained documentation, manufacturer web pages, and so forth. The online version of this list is also interactive; people can and do add printers all the time, so be sure to check it as well. Finally, if your printer isn't listed, add it!

Printers here are categorized into three types:

Perfectly

Perfect printers work perfectly - you can print to the full ability of the printer, including color, full resolution, etc. In a few cases printers with undocumented "resolution enhancement" modes that don't work are listed as perfect; generally the difference in print quality is small enough that it isn't worth worrying about.

Mostly

You can print fine, but there may be minor limitations or one sort or another in either printing or other features.

Partially

You can print, but maybe not in color, or only at a poor resolution. See the online listing's notes column for information on the limitation.

Paperweight

You can't print a darned thing; typically this will be due to lack of a driver and/or documentation on how to write one.

In all cases, since this information is provided by dozens of people, none of it is guaranteed to be correct. It should, however, be easy to corroborate from the driver web pages and manufacturer web sites.

And without further ado, here is the printer compatability list:

Brother

Perfectly

HL-10V, HL-660, HL-720, HL-760.

Mostly

HL-1050.

Canon

Perfectly

BJ-10e, BJ-20, BJ-200, BJC-210, BJC-240, BJC-250, BJC-610, BJC-620, BJC-70, BJC-800, LBP-8II, LIPS-III.

Mostly

BJ-300, BJC-4000, BJC-4100, BJC-4200, BJC-4300, BJC-4400, BJC-7000, BJC-7004.

Partially

TOC

BJC-4550, MultiPASS C2500, MultiPASS C3500.

Paperweight

BJC-5000, LBP-460, LBP-660.

Epson

Perfectly

ActionLaser 1100, LP 8000, LQ 850, Stylus Color, Stylus Color 400, Stylus Color 500, Stylus Color 600, Stylus Color 640, Stylus Color 850, Stylus Color II, Stylus Color IIs, Stylus Pro XL.

Mostly

Stylus Color 800, Stylus Photo 750.

Partially

Stylus Color 740.

HP

Perfectly

2000Cse, 2500C, DesignJet 650C, DeskJet 1200C, DeskJet 1600C, DeskJet 1600Cm, DeskJet 400, DeskJet 420C, DeskJet 500, DeskJet 550C, DeskJet 600, DeskJet 660Cse, DeskJet 690C, DeskJet 850C, DeskJet 855C, DeskJet 870, DeskJet 870Cxi, DeskJet 890, LaserJet 1100, LaserJet 2100, LaserJet 2100M, LaserJet 4000N, LaserJet 4L, LaserJet 5, LaserJet 5L, LaserJet 5MP, LaserJet 6L, LaserJet 6MP, LaserJet 8000, LaserJet 8100, Laserjet 5000, Mopier 320, PaintJet XL300.

Mostly

HP 660C.

Partially

DeskJet 1000C, DeskJet 670C, DeskJet 710, DeskJet 720C, DeskJet 722C, DeskJet 820C.

Paperweight

LaserJet 3100.

IBM

Perfectly

Jetprinter 3852.

Lexmark

Perfectly

Optra Color 1200, Optra Color 1275, Optra Color 40, Optra Color 45, Optra E, Optra E+, Optra Ep, Optra S 1250.

Partially

1020 Business, 3000, 5700, 7000.

TOC

Paperweight

1000, 1020, 1100, 2030, 2050, 2070, 5000, 7200, Winwriter 100, Winwriter 150c, Winwriter 200.

Minolta

Perfectly

PagePro 6, PagePro 8.

NEC

Perfectly

P2X.

Partially

SuperScript 100C, SuperScript 1260, SuperScript 150C, SuperScript 650C, SuperScript 750C, SuperScript 860, SuperScript 870.

Paperweight

SuperScript 660i, SuperScript 660plus.

Okidata

Perfectly

OL 410e, OL 610e/PS, OL 810e/PS, Okipage 6e, Okipage 6ex, Okipage 8c.

Mostly

Okipage 4w.

Paperweight

Okipage 8w, okijet 2010.

Olivetti

Perfectly

JP350S.

Panasonic

Perfectly

KX-P1123, KX-P4440, KX-P5400, KX-P8420, KX-P8475.

Partially

KX-P6500.

Paperweight

KX-P6100, KX-P8410.

QMS

Perfectly

2425 Turbo EX.

Ricoh

Perfectly

4801, 6000.

Xerox

Perfectly

DocuPrint C55, DocuPrint N17, DocuPrint N32.

---

5. Which spooling software?

Until recently, the choice for Linux users was simple - everyone ran the same old lpd lifted mostly verbatim out of BSD's Net-2 code. Even today, most vendors ship this software. But this is beginning to change. SVR4-like systems including Sun's Solaris come with a completely different print spooling package, centered around lpsched. And there are signs that some Linux vendors will shift to providing LPRng, a far less ancient print spooling implementation that is freely available. LPRng is far easier to administer for large installations (read: more than one printer, any serial printers, or any peculiar non-lpd network printers) and has a less frightfully haphazard codebase than does stock lpd. It can even honestly claim to be secure - there are no SUID binaries, and it supports authentication via PGP or Kerberos.

For the moment, even in light of the new options, lpd is probably fine for most Linux users. While it isn't the snazziest system, it works fine once set up, and it is well understood and extensively documented in third-party Unix books.

If you'd like more information on LPRng, check out the LPRng Web Page. Future versions of this HOWTO will include information on using both LPRng and regular lpd.

6. How it works, basic

In order to get printing working well, you need to understand how the lpd system works.

Lpd stands for Line Printer Daemon, and refers in different contexts to both the daemon and the whole collection of programs which run print spooling. These are:

`lpd`

The spooling daemon. One of these runs to control everything on a machine, AND one is run per printer while the printer is printing.

`lpr`

The user spooling command. Lpr contacts lpd and injects a new print job into the spool.

`lpq`

Lists the jobs in a print queue.

`lpc`

The Lpd system control command. With lpc you can stop, start, reorder, etc, the print queues.

`lprm`

`lprm` will remove a job from the print spool.

So how does it fit together? Well, when the system boots, lpd is run. It scans the file */etc/printcap* to learn which printers it will be managing spools for. Each time someone runs lpr, lpr contacts lpd through the named socket */dev/printer*, and feeds lpd both the file to print and some information about who is printing and how to print it. Lpd then prints the file on the appropriate printer in turn.

The lp system was originally designed when most printers were line printers - that is, people mostly printed plain ascii. As it turns out, only a little extra scripting is needed to make lpd work quite well for today's print jobs, which are often in PostScript, or text, or dvi, or...

7. How to set things up, basic

7.1 Traditional lpd configuration

The minimal setup for lpd rsults in a system that can queue files and print them. It will not pay any attention to wether or not your printer will understand them, and will probably not let you produce attractive output. Nevertheless, it is the first step to understanding, so read on!

Basically, to add a print queue to lpd, you must add an entry in */etc/printcap*, and make the new spool directory under */var/spool/lpd*.

An entry in */etc/printcap* looks like:

```
# LOCAL djet500
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :mx#0:\
        :lp=/dev/lp0:\
        :sh:
```

This defines a spool called *lp*, *dj*, or *deskjet*, spooled in the directory */var/spool/lpd/dj*, with no per-job maximum size limit, which prints to the device */dev/lp0*, and which does not have a banner page (with the name of the person who printed, etc) added to the front of the print job.

Go now and read the man page for *printcap*.

The above looks very simple, but there a catch - unless I send in files a DeskJet 500 can understand, this DeskJet will print strange things. For example, sending an ordinary Unix text file to a deskjet results in literally interpreted newlines, and gets me:

```
This is line one.
                 This is line two.
                                   This is line three.
```

ad nauseam. Printing a PostScript file to this spool would get a beautiful listing of the PostScript commands, printed out with this "staircase effect", but no useful output.

Clearly more is needed, and this is the purpose of filtering. The more observant of you who read the printcap man page might have noticed the spool attributes *if* and *of*. Well, *if*, or the input filter, is just what we need here.

If we write a small shell script called *filter* that adds carriage returns before newlines, the staircasing can be eliminated. So we have to add in an *if* line to our printcap entry above:

TOC

```
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :mx#0:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/dj/filter:\
        :sh:
```

A simple filter script might be:

```
#!perl
# The above line should really have the whole path to perl
# This script must be executable: chmod 755 filter
while(<STDIN>){chop $_; print "$_\r\n";};
# You might also want to end with a form feed: print "\f";
```

If we were to do the above, we'd have a spool to which we could print regular Unix text files and get meaningful results. (Yes, there are four million better ways to write this filter, but few so illustrative. You are encouraged to do this more efficiently.)

The only remaining problem is that printing plain text is really not too hot - surely it would be better to be able to print PostScript and other formatted or graphic types of output. Well, yes, it would, and it's easy to do. The method is simply an extention of the above linefeed-fixing filter. If you write a filter than can accept arbitrary file types as input and produce DeskJet-kosher output for each case, then you've got a clever print spooler indeed!

Such a filter is called a *magic* filter. Don't bother writing one yourself unless you print strange things - there are a good many written for you already on the net. APS Filter is among the best, or your Linux distribution may have a printer setup tool that makes this all really easy.

7.2 File Permissions

By popular demand, I include below a listing of the permissions on interesting files on my system. There are a number of better ways to do this, ideally using only SGID binaries and not making everything SUID root, but this is how my system came out of the box, and it works for me. (Quite frankly, if your vendor can't even ship a working lpd you're in for a rough ride).

```
-r-sr-sr-x   1 root     lp    /usr/bin/lpr*
-r-sr-sr-x   1 root     lp    /usr/bin/lprm*
-rwxr--r--   1 root     root  /usr/sbin/lpd*
-r-xr-sr-x   1 root     lp    /usr/sbin/lpc*
drwxrwxr-x   4 root     lp    /var/spool/lpd/
drwxr-xr-x   2 root     lp    /var/spool/lpd/lp/
```

Lpd must currently be run as root so that it can bind to the low-numbered lp service port. It should probably become UID lp.lp or something after binding, but I don't think it does. Bummer.

---

8. Getting Printing Software

Many prewritten filter packages (and other printer-related software) are available from Metalab. Such utilities as psutils, a2ps, mpage, dvitodvi, flpr, etc can all be found there.

For a while, there were several packages out there all trying to make printer configuration easier. They probably all still exist, but one of the best and most up-to-date is Andreas Klemm's APS Filter package, which has a menu-driven printcap configurator and handles practically any type of input imaginable. If your vendor doesn't ship a nice printer setup tool, APS Filter is the way to go.

9. Vendor Solutions

This section is, by definition, incomplete. Feel free to send in details of your favourite distribution.

9.1 Red Hat

Red Hat has a GUI printer administration tool (in the control panel) which can add remote printers and printers on local devices. It lets you choose a ghostscript-supported printer type and Unix device file to print to, then installs a print queue in */etc/printcap* and writes a short PostScript-and-ascii magic filter based around `gs` and `nenscript`. This solution works fairly well, and is trivial to setup for common cases.

Where Red Hat fails is when you have a printer which isn't supported by their standard Ghostscript (which is GNU rather than Aladdin Ghostscript, and which supports fewer printers). Check in the printer compatibility list above (or online) if you find that you can't print properly with the stock Red Hat software. If your printer isn't supported by Red Hat's tools, you may need to install a contributed verison of Aladdin Ghostscript, and will probably also be better off if you use the apsfilter package, which knows all about the printers supported by late-model Ghostscripts.

9.2 Debian

Debian offers a choice between plain lpd and LPRng; LPRng is probably a better choice. I believe Debian also offers a choice of printer configuration tools; apsfilter version 5 or later is probably your best bet, since that verison adds support for LPRng and Ghostscript's uniprint driver scheme.

9.3 Other Distributions

Please send me info on what other distributions do!

10. Ghostscript.

Ghostscript is an incredibly significant program for Linux printing. Most printing software under Unix generates PostScript, which is typically a $100 option on a printer. Ghostscript, however, is free, and will generate the language of your printer from PostScript. When tied in with your `lpd` input filter, it gives you a virtual PostScript printer and simplifies life immensely.

Ghostscript is available in two forms. The commercial version of Ghostscript, called Aladdin Ghostscript, may be used freely for personal use but may not be distributed by commercial Linux distributions. It is generally a year or so ahead of the free Ghostscript; at the moment, for example, it supports many color inkjets that the older Ghostscripts do not.

The free version of Ghostscript is GNU Ghostscript, and is simply an aged version of Aladdin ghostscript kindly given to GNU. (Kudos to Aladdin for this arrangement; more software vendors should support free software in this way, if they can't handle full-blown GPL distribution of their code).

Whatever you do with `gs`, be very sure to run it with the option for disabling file access (`-dSAFER`). PostScript is a fully functional language, and a bad PostScript program could give you quite a headache.

Speaking of PDF, Adobe's Portable Document Format is actually little more than organized PostScript in a compressed file. Ghostscript can handle PDF input just as it does PostScript. So you can be the first on your block with a PDF-capable printer.

10.1 Invoking Ghostscript

Typically, Ghostscript will be run by whatever magic filter you settle upon (I recommend apsfilter if your vendor didn't supply anything that suits you), but for debugging purposes it's often handy to run it directly.

`gs -help` will give a brief informative listing of options and available drivers (note that this list is the list of drivers compiled in, not the master list of all available drivers).

You might run gs for testing purposes like: `gs` *`options`* `-q -dSAFER -sOutputFile=/dev/lp1 test.ps`.

10.2 Ghostscript output tuning

There are a number of things one can do if gs's output is not satisfactory (actually, you can do anything you darn well please, since you have the source).

TOC

Output location and size

The location, size, and aspect ratio of the image on a page is controlled by the printer-specific driver in ghostscript. If you find that your pages are coming out scrunched too short, or too long, or too big by a factor of two, you might want to look in your driver's source module and adjust whatever parameters jump out at you. Unfortunately, each driver is different, so I can't really tell you what to adjust, but most of them are reasonably well commented.

Gamma, dotsizes, etc.

Most non-laser printers suffer from the fact that their dots are rather large. This results in pictures coming out too dark. If you experience this problem you should use your own transfer function. Simply create the following file in the ghostscript lib-dir and add its name to the gs call just before the actual file. You may need to tweak the actual values to fit your printer. Lower values result in a brighter print. Especially if your driver uses a Floyd-Steinberg algorithm to rasterize colors, lower values (0.2 - 0.15) are probably a good choice.

```
---8<---- gamma.ps ----8<---
%!
%transfer functions for cyan magenta yellow black
{0.3 exp} {0.3 exp} {0.3 exp} {0.3 exp} setcolortransfer
---8<------------------8<---
```

It is also possible to mend printers that have some kind of colour fault by tweaking these values. If you do that kind of thing, I recommend using the file colorcir.ps, that comes with ghostscript (in the examples/ subdir), as a test page.

For many of the newer color inkjet drivers, there are command-line options, or different upp driver files, which implement gamma and other changes to adapt the printer to different paper types. You sould look into this before playing with Postscript to fix things.

---

11. How to print to a printer over the network

One of the features of lpd is that it supports printing over the network to printers physically connected to a different machine. With the careful combination of filter scripts and assorted utilities, you can make lpr print transparently to printers on all sorts of networks.

11.1 To a Unix/lpd host

To allow remote machines to print to your printer, you must list the machines in */etc/hosts.equiv* or */etc/hosts.lpd*. (Note that *hosts.equiv* has a host of other effects; be sure you know what you are doing if you list any machine there). You can allow only certain users on the other machines to print to your printer by usign the *rs* attribute; read the `lpd` man page for information on this.

With `lpd`

To print to another machine, you make an */etc/printcap* entry like this:

```
# REMOTE djet500
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :rm=machine.out.there.com:\
        :rp=printername:\
        :lp=/dev/null:\
        :sh:
```

Note that there is still a spool directory on the local machine managed by `lpd`. If the remote machine is busy or offline, print jobs from the local machine wait in the spool area until they can be sent.

With `rlpr`

You can also use *rlpr* to send a print job directly to a queue on a remote machine without going through the hassle of configuring lpd to handle it. This is mostly useful in situations where you print to a variety of printers only occasionally. From the announcement for *rlpr*:

Rlpr uses TCP/IP to send print jobs to lpd servers anywhere on a network.

Unlike lpr, it *does not* require that the remote printers be explicitly known to the machine you wish to print from, (e.g. through */etc/printcap*) and thus is considerably more flexible and requires less administration.

rlpr can be used anywhere a traditional lpr might be used, and is backwards compatible with traditional BSD lpr.

The main power gained by rlpr is the power to print remotely *from anywhere to anywhere* without regard for how the system you wish to print from was configured. Rlpr can work as a filter just like traditional lpr so that clients executing on a remote machine like netscape, xemacs, etc, etc can print to your local machine

with little effort.

Rlpr is available from Metalab.

11.2 To a Win95, WinNT, LanManager, or Samba printer

There is a Printing to Windows mini-HOWTO out there which has more info than there is here.

It is possible to direct an lpd queue through the `smbclient` program (part of the samba suite) to a TCP/IP based SMB print service. Samba includes a script to do this called `smbprint`. In short, you put a configuration file for the specific printer in question in the spool directory, and install the `smbprint` script as the *if*.

The */etc/printcap* entry goes like this:

```
lp|remote-smbprinter:\
    :lp=/dev/null:sh:\
    :sd=/var/spool/lpd/lp:\
    :if=/usr/local/sbin/smbprint:
```

You should read the documentation inside the `smbprint` script for more information on how to set this up.

You can also use `smbclient` to submit a file directly to an SMB printing service without involving `lpd`. See the man page.

11.3 To a NetWare Printer

The ncpfs suite includes a utility called `nprint` which provides the same functionality as `smbprint` but for NetWare. You can get ncpfs from Metalab. From the LSM entry for version 0.16:

> With ncpfs you can mount volumes of your netware server under Linux. You can also print to netware print queues and spool netware print queues to the Linux printing system. You need kernel 1.2.x or 1.3.54 and above. ncpfs does NOT work with any 1.3.x kernel below 1.3.54.

To make `nprint` work via lpd, you write a little shell script to print stdin on the NetWare printer, and install that as the *if* for an lpd print queue. You'll get something like:

```
sub2|remote-NWprinter:\
        :lp=/dev/null:sh:\
        :sd=/var/spool/lpd/sub2:\
        :if=/var/spool/lpd/nprint-script:
```

The `nprint-script` might look approximately like:

```
#! /bin/sh
# You should try the guest account with no password first!
/usr/local/bin/nprint -S net -U name -P passwd -q printq-name -
```

TOC

11.4 To an EtherTalk (Apple) printer

The netatalk package includes something like `nprint` and `smbclient`. Others have documented the procedure for printing to and from an Apple network far better than I ever will; see the Linux Netatalk-HOWTO.

11.5 To an HP or other ethernet printer

HPs and some other printers come with an ethernet interface which you can print to directly using lpd. You should follow the instructions that came with your printer or its network adaptor, but in general, such printers are "running" lpd, and provide one or more queues which you can print to. An HP, for example, might work with a printcap like:

```
lj-5|remote-hplj:\
        :lp=/dev/null:sh:\
        :sd=/var/spool/lpd/lj-5:\
        :rm=printer.name.com:rp=raw:
```

HP Laserjet printers with Jet Direct interfaces generally support two built in lpd queues - "raw" which accepts PCL (and possibly Postscript) and "text" which accepts straight ascii (and copes automatically with the staircase effect). If you've got a JetDirect Plus3 three-port box, the queues are named "raw1", "text2", and so forth.

Note that the ISS company has identified an assortment of denial of service attacks which hang HP Jetdirect interfaces. Most of these have been addressed beginning in Fall 98.

In a large scale environment, especially a large environment where some printers do not support PostScript, it may be useful to establish a dedicated print server to which all machines print and on which all ghostscript jobs are run.

This also allows your Linux box to act as a spool server for the printer so that your network users can complete their print jobs quickly and get on with things without waiting for the printer to print any other job that someone else has sent. This is suggested too if you have unfixable older HP Jetdirects; it reduces the likelihood of the printers wedging.

To do this, set up a queue on your linux box that points at the ethernet equipped HP LJ (as above). Now set up all the clients on your LAN to point at the Linux queue (eg lj-5 in the example above).

Some HP network printers apparently don't heed the banner page setting sent by clients; you can turn off their internally generated banner page by telnetting to the printer, hitting return twice, typing "banner: 0" followed by "quit". There are other settings you can change this way, as well; type "?" to see a list.

The full range of settings can be controlled with HP's WebJet software. This package runs as a daemon, and accepts http requests on a designated port. It serves up forms and Java applets which can control HP printers on the network. In theory, it can also control Unix print queues, but it does so using the rexec service, which is completely unsecure. I don't advise using that feature.

To older HPs

Some printers (and printer networking "black boxes") support only a cheesy little non-protocol involving plain TCP connections. Notable in this category are early-model JetDirect (including some JetDirectEx) cards. Basically, to print to the printer, you must open a TCP connection to the printer on a specified port (typically 9100, or 9100, 9101 and 9102 for three-port boxes) and stuff your print job into it. LPRng (which you really should be using instead, if you're reading this section) has built-in support for stuffing print jobs into random TCP ports, but with BSD lpd it's not so easy. This can be implemented, among other ways, in Perl:

```
#!/usr/bin/perl
# Thanks to Dan McLaughlin for writing the original version of this
# script (And to Jim W. Jones for sitting next to Dan when writing me
# for help ;)

$fileName = @ARGV[0];

open(IN,"$fileName") || die "Can't open file $fileName";

$dpi300     = "\x1B*t300R";
$dosCr      = "\x1B&k3G";
$ends = "\x0A";

$port =  9100 unless $port;
$them = "bach.sr.hp.com" unless $them;

$AF_INET = 2;
$SOCK_STREAM = 1;
$SIG{'INT'} = 'dokill';
$sockaddr = 'S n a4 x8';

chop($hostname = `hostname`);
($name,$aliases,$proto) = getprotobyname('tcp');
($name,$aliases,$port) = getservbyname($port,'tcp')
    unless $port =~ /^\d+$/;;
($name,$aliases,$type,$len,$thisaddr) =
        gethostbyname($hostname);
($name,$aliases,$type,$len,$thataddr) = gethostbyname($them);
$this = pack($sockaddr, $AF_INET, 0, $thisaddr);
$that = pack($sockaddr, $AF_INET, $port, $thataddr);

if (socket(S, $AF_INET, $SOCK_STREAM, $proto)) {
#    print "socket ok\n";
}
else {
    die $!;
}
```

```
# Give the socket an address.
if (bind(S, $this)) {
#    print "bind ok\n";
}
else {
    die $!;
}

# Call up the server.

if (connect(S,$that)) {
#    print "connect ok\n";
}
else {
    die $!;
}

# Set socket to be command buffered.

select(S); $| = 1; select(STDOUT);

#    print S "@PJL ECHO Hi $hostname! $ends";
#    print S "@PJL OPMSG DISPLAY=\"Job $whoami\" $ends";
#    print S $dpi300;

# Avoid deadlock by forking.

if($child = fork) {
    print S $dosCr;
    print S $TimesNewR;

    while (<IN>) {
        print S;
    }
    sleep 3;
    do dokill();
} else {
    while(<S>) {
        print;
    }
}

sub dokill {
    kill 9,$child if $child;
}
```

11.6 Running an *if* for remote printers

One oddity of lpd is that the *if* is not run for remote printers. If you find that you need to run an *if*, you can do so by setting up a double queue and requeueing the job. As an example, consider this *printcap*:

```
lj-5:\
        :lp=/dev/null:sh:\
        :sd=/var/spool/lpd/lj-5:\
        :if=/usr/lib/lpd/filter-lj-5:
lj-5-remote:lp=/dev/null:sh:rm=printer.name.com:\
        :rp=raw:sd=/var/spool/lpd/lj-5-raw:
```

in light of this `filter-lj-5` script:

```
#!/bin/sh
gs <options> -q -dSAFER -sOutputFile=- - | \
        lpr -Plj-5-remote -U$5
```

The *-U* option to lpr only works if lpr is run as daemon, and it sets the submitter's name for the job in the resubmitted queue correctly. You should probably use a more robust method of getting the username, since in some cases it is not argument 5. See the man page for *printcap*.

11.7 From Windows.

Printing from a Windows (or presumably, OS/2) client to a Linux server is directly supported over SMB through the use of the SAMBA package, which also supports file sharing of your Linux filesystem to Windows clients.

Samba includes fairly complete documentation, and there is a good Samba FAQ which covers it, too. You can either configure a magic filter on the Linux box and print PostScript to it, or run around installing printer-specific drivers on all the Windows machines and having a queue for them with no filters at all. Relying on the Windows drivers may in some cases produce better output, but is a bit more of an administrative hassle if there are many Windows boxen. So try Postscript first.

11.8 From an Apple.

Netatalk supports printing from Apple clients over EtherTalk. See the Netatalk HOWTO Page for more information.

11.9 From Netware.

The ncpfs package includes a daemon named pserver which can be used to privide service to a NetWare print queue. From what I understand, this system requires a Bindery-based NetWare, ie 2.x, 3.x, or 4.x with bindery access enabled.

For more information on ncpfs and it's pserver program, see the ncpfs FTP site.

12. Windows-only printers

As I discussed earlier, some printers are inherently unsupported because they don't speak a normal printer language, instead using the computer's CPU to render a bitmap which is then piped to the printer at a fixed speed. In a few cases, these printers also speak something normal like PCL, but often they do not. In some (really low-end) cases, the printer doesn't even use a normal parallel connection but relies on the vendor's driver to emulate what should be hardware behaviour (most importantly flow control).

In any case, there are a few possible workarounds if you find yourself stuck with such a lemon.

12.1 The Ghostscript Windows redirector

There is now a Windows printer driver available (called mswinpr2) that will run a print job through Ghostscript before finally printing it. (Rather like an if filter in Unix's LPD). There is also a new Ghostscript driver which will print using Windows GDI calls. Taken all together, this should(tm) allow a Windows machine to print PostScript to a Windows-only printer through the vendor's driver.

If you get that working, you can then follow the instructions above for printing to a Windows printer over the network from Linux to let Unix (and other Windows, Mac, etc) hosts print to your lemon printer.

That said, I've never tried this since I don't have a winprinter, so your mileage may vary. All the software involved should be available from the Ghostscript home page (see the links in the Ghostscript section of this document).

12.2 The pbm2ppa program

Some HP printers use "Printing Performance Architecture" (marketingspeak for "we were too cheap to implement PCL"). This is supported in a roundabout way via the pbm2ppa translator written by Tim Norman. Basically, you use ghostscript to render PostScript into a bitmapped image in pbm format and then use pbm2ppa to translate this into a printer-specific ppa format bitmap ready to be dumped to the printer. This program may also come in ghostscript driver format by now.

The ppa software can be had from the ppa home page; pbm2ppa supports some models of the HP 720, 820, and 1000; read the documentation that comes with the package for more details on ppa printer support.

12.3 The pbm2l7k program

TOC

Most of the cheap Lexmark inkjets use a proprietary language and are therefore Winprinters. However, Henryk Paluch has written a program which can print in black and white on a Lexmark 7000. Hopefully he'll be able to figure out color and expand support to other Lexmark inkjets. See here for more info.

---

13. How to print to a fax machine.

13.1 Using a faxmodem

There are a number of fax programs out there that will let you fax and receive documents. One of the most complex is Sam Leffler's *HylaFax*, available from `ftp.sgi.com`. It supports all sorts of things from multiple modems to broadcasting.

SuSE ships a Java HylaFax client which allegedly works on any Java platform (including Windows and Linux). There are also non-Java fax clients for most platforms; Linux can almost certainly handle your network faxing needs.

Also available, and a better choice for most Linux boxen, is `efax`, a simple program which sends faxes. The getty program `mgetty` can receive faxes (and even do voicemail on some modems!).

13.2 Using the Remote Printing Service

There is an experimental service offered that lets you send an email message containing something you'd like printed such that it will appear on a fax machine elsewhere. Nice formats like postscript are supported, so even though global coverage is spotty, this can still be a very useful service. For more information on printing via the remote printing service, see the Remote Printing WWW Site.

14. How to generate something worth printing.

Here we get into a real rat's-nest of software. Basically, Linux can run many types of binaries with varying degrees of success: Linux/x86, Linux/Alpha, Linux/Sparc, Linux/foo, iBCS, Win16/Win32s (with dosemu and, someday, with Wine), Mac/68k (with Executor), and Java. I'll just discuss native Linux and common Unix software.

For Linux itself, choices are mostly limited to those available for Unix in general:

14.1 Markup languages

Most markup languages are more suitable for large or repetitive projects, where you want the computer to control the layout of the text to make things uniform.

`nroff`

This was one of the first Unix markup languages. Man pages are the most common examples of things formatted in *roff macros; many people swear by them, but nroff has, to me at least, a more arcane syntax than needed, and probably makes a poor choice for new works. It is worth knowing, though, that you can typeset a man page directly into postscript with groff. Most man commands will do this for you with `man -t foo | lpr`.

TeX

TeX, and the macro package LaTeX, are one of the most widely used markup languages on Unix. Technical works are frequently written in LaTeX because it greatly simplifies the layout issues and is *still* one of the few text processing systems to support mathematics both completely and well. TeX's output format is `dvi`, and is converted to PostScript or Hewlett Packard's PCL with `dvips` or `dvilj`. If you wish to install TeX or LaTeX, install the whole teTeX group of packages; it contains everything.

SGML

There is at least one free sgml parser available for Unix and Linux; it forms the basis of Linuxdoc-SGML's homegrown document system. It can support other DTD's, as well.

HTML

Someone suggested that for simple projects, it may suffice to write it in HTML and print it out using Netscape. I disagree, but YMMV.

14.2 WYSIWYG Word Processors

There is no longer any shortage of WYSIWYG word processing software. Several complete office suites are available, including one that's free for personal use (StarOffice).

TOC

StarOffice

A German company is distributing StarOffice on the net free for Linux. This full-blown office suite has all the features you'd expect, and you can't beat the price. There's a mini-HOWTO out there which describes how to obtain and install it. It generates PostScript or PCL, so should work with most any printer that works otherwise on Linux. Apparently it's an Office clone and is rather bloated.

WordPerfect

Corel distributes a basic version of Word Perfect 8 free for Linux, and has suggested that they will distribute Corel Draw and Quattro Pro as well, once they are ported. This is probably the best option if you have an ARM machine; Corel makes the ARM-based Netwinder Linux computers and is almost certian to offer ARM Linux versions of everything. You can also buy the full-blown version and support, together or separately. The Linux WordPerfect Fonts and Printers page has information about configuring WordPerfect for use with either Ghostscript or its built-in printer drivers (which are apparently identical the DOS WordPerfect drivers, if your printer's driver isn't included in the WP8 distribution).

Applix

Applix is a cross-platform (ie, various Unices, Windows, and others) office suite sold by the Applix company. Red Hat and SuSE sold it themselves when it was the only game in town; now sales have reverted to Applix.

LyX

LyX is a front-end to LaTeX which looks very promising. See the LyX Homepage for more information. There is a KDE-styled version of LyX, called Klyx; the author of LyX and the instigator of KDE are the same person.

Maxwell

Maxwell is a simple MS RTF-format based word processor which started as a commercial product but is now distributed under the GPL.

The Andrew User Interface System

AUIS includes `ez`, a WYSIWYG-style editor with most basic word processor features, HTML capabilities, and full MIME email and newsgroup support. Unfortunately, AUIS is no longer maintained.

Koffice

The KDE project is working toward a whole office suite. I don't think it's ready for prime time yet. The word processor will apparently be a descendant of LyX.

GNOME

The GNOME project also is working toward various GNU-licensed officey tools. None are available yet, though.

Jeff Phillips `<jeff@I_RATUS.org>` uses Caldera's WordPerfect 7 for Linux (on Slackware, of all things) and says that it works well. It apparently includes built-in printer support, as one would

expect. Caldera should have info on http://www.caldera.com/. You can also buy a newer versi Wordperfect directly from Corel's chosen Unix port company.

Other vendors should feel free to drop me a line with your offerings.

---

15. On-screen previewing of printable things.

Nearly anything you can print can be viewed on the screen, too.

15.1 PostScript

Ghostscript has an X11 driver best used under the management of the PostScript previewer gv. The latest versions of these programs should be able to view PDF files, as well. Note that gv has replaced the older previewer "Ghostview"; the new user interface is mch prettier and featureful that ghostview's plain old Athena gui.

15.2 TeX dvi

TeX DeVice Independant files may be previewed under X11 with xdvi. Modern versions of xdvi call ghostscript to render PostScript specials.

A VT100 driver exists as well. It's called `dgvt`. `Tmview` works with Linux and svgalib, if that's all you can do.

15.3 Adobe PDF

Adobe's Acrobat Reader is available for Linux; just download it from their web site http://www.adobe.com/.

You can also use xpdf, which is freeware and comes with source, and I should think Ghostview supports viewing PDF files with gs under X11 by now.

16. Serial printers under lpd

16.1 Setting up in printcap

Lpd provides five attributes which you can set in */etc/printcap* to control all the settings of the serial port a printer is on. Read the *printcap* man page and note the meanings of *br#*, *fc#*, *xc#*, *fs#* and *xs#*. The last four of these attributes are bitmaps indicating the settings for use the port. The *br#* atrribute is simply the baud rate, ie `br#9600'.

It is very easy to translate from stty settings to printcap flag settings. If you need to, see the man page for stty now.

Use stty to set up the printer port so that you can cat a file to it and have it print correctly. Here's what `*stty -a*' looks like for my printer port:

```
dina:/usr/users/andy/work/lpd/lpd# stty -a < /dev/ttyS2
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr
-igncr -icrnl ixon -ixoff -iuclc -ixany -imaxbel
-opost -olcuc -ocrnl -onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0
bs0 vt0 ff0
-isig -icanon -iexten -echo -echoe -echok -echonl -noflsh -xcase
-tostop -echoprt -echoctl -echoke
```

The only changes between this and the way the port is initialized at bootup are `-clocal`, `-crtscts`, and `ixon`. Your port may well be different depending on how your printer does flow control.

You actually use stty in a somewhat odd way. Since stty operates on the terminal connected to it's standard input, you use it to manipulate a given serial port by using the `<' character as above.

Once you have your stty settings right, so that `*cat file > /dev/ttyS2*' (in my case) sends the file to the printer, look at the file /usr/src/linux/include/asm-i386/termbits.h. This contains a lot of #defines and a few structs (You may wish to cat this file to the printer (you do have that working, right?) and use it as scratch paper). Go to the section that starts out

```
/* c_cflag bit meaning */
#define CBAUD   0000017
```

This section lists the meaning of the *fc#* and *fs#* bits. You will notice that the names there (after the baud rates) match up with one of the lines of stty output. Didn't I say this was going to be easy?

Note which of those settings are preceded with a - in your stty output. Sum up all those numbers (they are octal). This represents the bits you want to clear, so the result is your *fc#* capability. Of course, remember that you will be setting bits directly after you clear, so you can just use `fc#0177777' (I do).

Now do the same for those settings (listed in this section) which do not have a - before them in your stty output. In my example the important ones are CS8 (0000060), HUPCL (0002000), and CREAD (0000200). Also note the flags

for your baud rate (mine is 0000015). Add those all up, and in my example you get 0002275. This goes in your capability (`fs#02275' works fine in my example).

Do the same with set and clear for the next section of the include file, "c_lflag bits". In my case I didn't have to set anything, so I just use `xc#0157777' and `xs#0'.

16.2 Older serial printers that drop characters

Jon Luckey points out that some older serial printers with ten-cent serial interfaces and small buffers *really* mean stop when they say so with flow control. He found that disabling the FIFO in his Linux box's 16550 serial port with `setserial` corrected the problem of dropped characters (you apparently just specify the uart type as an 8250 to do this).

---

17. Credits

The `smbprint` information is from an article by Marcel Roelofs `<marcel@paragon.nl>`.

The `nprint` information for using Netware printers was provided by Michael Smith `<mikes@bioch.ox.ac.uk>`.

The serial printers under lpd section is from Andrew Tefft `<teffta@engr.dnet.ge.com>`.

The blurb about gammas and such for gs was sent in by Andreas `<quasi@hub-fue.franken.de>`.

The two paragraphs about the 30 second closing_wait of the serial driver was contributed by Chris Johnson `<cdj@netcom.com>`.

Robert Hart sent a few excellent paragraphs about setting up a print server to networked HPs which I used verbatim.

And special thanks to the dozens upon dozens of you who've pointed out typos, bad urls, and errors in the document over the years.

The Linux Kernel HOWTO

Brian Ward, bri@cs.uchicago.edu

v1.0, 5 June 1999

This is a detailed guide to kernel configuration, compilation, upgrades, and troubleshooting for ix86-based systems.

1. Introduction

- 1.1 Read this first! (I mean it)
- 1.2 A word on style

2. Important questions and their answers

- 2.1 What does the kernel do, anyway?
- 2.2 Why would I want to upgrade my kernel?
- 2.3 What kind of hardware do the newer kernels support?
- 2.4 What version of gcc and libc do I need?
- 2.5 What's a loadable module?
- 2.6 How much disk space do I need?
- 2.7 How long does it take?

3. How to actually configure the kernel

- 3.1 Getting the source
- 3.2 Unpacking the source
- 3.3 Configuring the kernel
- 3.4 Now what? (The Makefile)

TOC

4. Compiling the kernel

- 4.1 Cleaning and depending
- 4.2 Compile time
- 4.3 Other ``make''ables
- 4.4 Installing the kernel

5. Patching the kernel

- 5.1 Applying a patch
- 5.2 If something goes wrong
- 5.3 Getting rid of the .orig files
- 5.4 Other patches

6. Additional packages

- 6.1 kbd
- 6.2 util-linux
- 6.3 hdparm
- 6.4 gpm

7. Some pitfalls

- 7.1 make clean
- 7.2 Huge or slow kernels
- 7.3 The parallel port doesn't work/my printer doesn't work
- 7.4 Kernel doesn't compile
- 7.5 New version of the kernel doesn't seem to boot
- 7.6 You forgot to run LILO, or system doesn't boot at all
- 7.7 It says `warning: bdflush not running'
- 7.8 I can't get my IDE/ATAPI CD-ROM drive to work
- 7.9 It says weird things about obsolete routing requests
- 7.10 Firewalling not working in 1.2.0
- 7.11 ``Not a compressed kernel Image file''
- 7.12 Problems with console terminal after upgrade to 1.3.x

- 7.13 Can't seem to compile things after kernel upgrade
- 7.14 Increasing limits

TOC

8. Note for upgrade to version 2.0.x, 2.2.x

9. Modules

- 9.1 Installing the module utilities
- 9.2 Modules distributed with the kernel

10. Tips and tricks

- 10.1 Redirecting output of the make or patch commands
- 10.2 Conditional kernel install
- 10.3 Kernel updates

11. Other relevant HOWTOs that might be useful

12. Misc

- 12.1 Author
- 12.2 To do
- 12.3 Contributions
- 12.4 Copyright notice, License, and all that stuff

---

1. Introduction

Should you read this document? Well, see if you've got any of the following symptoms:

- ``Arg! This wizzo-46.5.6 package says it needs kernel release 2.8.193 and I still only have release 1.0.9!"
- There's a device driver in one of the newer kernels that you just gotta have
- You really have no idea at all how to compile a kernel
- ``Is this stuff in the README *really* the whole story?"
- You came, you tried, it didn't work
- You need something to give to people who insist on asking you to install their kernels for them

1.1 Read this first! (I mean it)

Some of the examples in this document assume that you have GNU `tar`, `find`, and `xargs`. These are quite standard; this should not cause problems. It is also assumed that you know your system's filesystem structure; if you don't, it is critical that you keep a written copy of the `mount` command's output during normal system operation (or a listing of `/etc/fstab`, if you can read it). This information is important, and does not change unless you repartition your disk, add a new one, reinstall your system, or something similar.

The latest ``production" kernel version at the time of this writing was 2.2.9, meaning that the references and examples correspond to that release. Even though I try to make this document as version-independent as possible, the kernel is constantly under development, so if you get a newer release, it will inevitably have some differences. Again, this should not cause major problems, but it may create some confusion.

There are two versions of the linux kernel source, ``production" and ``development." Production releases are the even-minor-numbered releases; 1.2.x was production, 2.0.x is production, as well as 2.2.x. These kernels are considered to be the most stable, bug-free versions available at the time of release. The development kernels (2.1.x, 2.3.x, etc) are meant as testing kernels, for people willing to test out new and possibly very buggy kernels. You have been warned.

1.2 A word on style

`Text that looks like this` is either something that appears on your screen, a filename, or something that can be directly typed in, such as a command, or options to a command (if you're looking at a plain-text file, it doesn't look any different). Commands and other input are frequently quoted (with ` '), which causes the following classic punctuation problem: if such an item appears at the end of a sentence in quotes, people often type a `.' along with the command, because the American quoting style says to put the period inside of the quotation marks. Even though common sense (and unfortunately, this

assumes that the one with the ``common sense'' is used to the so-called American style of quotation) should tell one to strip off the punctuation first, many people simply do not remember, so I will place outside the quotation marks in such cases. In other words, when indicating that you should type ```make config`'' I would write `make config', not `make config.'

2. Important questions and their answers

2.1 What does the kernel do, anyway?

The Unix kernel acts as a mediator for your programs and your hardware. First, it does (or arranges for) the memory management for all of the running programs (processes), and makes sure that they all get a fair (or unfair, if you please) share of the processor's cycles. In addition, it provides a nice, fairly portable interface for programs to talk to your hardware.

There is certainly more to the kernel's operation than this, but these basic functions are the most important to know.

2.2 Why would I want to upgrade my kernel?

Newer kernels generally offer the ability to talk to more types of hardware (that is, they have more device drivers), they can have better process management, they can run faster than the older versions, they could be more stable than the older versions, and they fix silly bugs in the older versions. Most people upgrade kernels because they want the device drivers and the bug fixes.

2.3 What kind of hardware do the newer kernels support?

See the Hardware-HOWTO. Alternatively, you can look at the `` `config.in' `` file in the linux source, or just find out when you try `` `make config' ``. This shows you all hardware supported by the standard kernel distribution, but not everything that linux supports; many common device drivers (such as the PCMCIA drivers and some tape drivers) are loadable modules maintained and distributed separately.

2.4 What version of gcc and libc do I need?

Linus recommends a version of gcc in the `README` file included with the linux source. If you don't have this version, the documentation in the recommended version of gcc should tell you if you need to upgrade your libc. This is not a difficult procedure, but it is important to follow the instructions.

2.5 What's a loadable module?

These are pieces of kernel code which are not linked (included) directly in the kernel. One compiles them separately, and can insert and remove them into the running kernel at almost any time. Due to its flexibility, this is now the preferred way to code certain kernel features. Many popular device drivers,

TOC

such as the PCMCIA drivers and the QIC-80/40 tape driver, are loadable modules.

2.6 How much disk space do I need?

It depends on your particular system configuration. First, the compressed linux source is nearly 14 megabytes large at version 2.2.9. Many sites keep this even after unpacking. Uncompressed and built with a moderate configuration, it takes up another 67 MB.

2.7 How long does it take?

With newer machines, the compilation takes dramatically less time than older ones; an AMD K6-2/300 with a fast disk can do a 2.2.x kernel in about four minutes. As for old Pentiums, 486s, and 386s, if you plan to compile one, be prepared to wait, possibly hours, days..

If this troubles you, and you happen to have a faster machine around to compile on, you can build on the fast machines (assuming you give it the right parameters, that your ulilities are up-to-date, and so on), and then transfer the kernel image to the slower machine.

---

3. How to actually configure the kernel

3.1 Getting the source

You can obtain the source via anonymous ftp from `ftp.kernel.org` in `/pub/linux/kernel/vx.y`, where `x.y` is the version (eg 2.2), and as mentioned before, the ones that end with an odd number are development releases and may be unstable. It is typically labelled `linux-x.y.z.tar.gz`, where `x.y.z` is the version number. The sites also typically carry ones with a suffix of `.bz2`, which have been compressed with bzip2 (these files will be smaller and take less time to transfer).

It's best to use `ftp.xx.kernel.org` where `xx` is your country code; examples being `ftp.at.kernel.org` for Austria, and `ftp.us.kernel.org` for the United States.

3.2 Unpacking the source

Log in as or `su` to `root', and `cd` to `/usr/src`. If you installed kernel source when you first installed linux (as most do), there will already be a directory called `linux' there, which contains the entire old source tree. If you have the disk space and you want to play it safe, preserve that directory. A good idea is to figure out what version your system runs now and rename the directory accordingly. The command `uname -r' prints the current kernel version. Therefore, if `uname -r' said `1.0.9', you would rename (with `mv') `linux' to `linux-1.0.9'. If you feel mildly reckless, just wipe out the entire directory. In any case, make certain there is no `linux' directory in `/usr/src` before unpacking the full source code.

Now, in `/usr/src`, unpack the source with `tar zxpvf linux-x.y.z.tar.gz' (if you've just got a `.tar` file with no `.gz` at the end, `tar xpvf linux-x.y.z.tar' works.). The contents of the source will fly by. When finished, there will be a new `linux' directory in `/usr/src`. `cd` to `linux` and look over the `README` file. There will be a section with the label `INSTALLING the kernel'. Carry out the instructions when appropriate -- symbolic links that should be in place, removal of stale `.o` files, etc.

If you have a `.bz2` file and the bzip2 program (read about it at `http://www.muraroa.demon.co.uk/`), do this:

```
bz2cat linux-x.y.z.tar.bz2 | tar xvf -
```

3.3 Configuring the kernel

TOC

Note: Some of this is reiteration/clarification of a similar section in Linus' `README` file.

The command `` `make config' `` while in `/usr/src/linux` starts a configure script which asks you many questions. It requires bash, so verify that bash is `/bin/bash`, `/bin/sh`, or `$BASH`.

However, there are some much more pleasant alternatives to `` `make config' `` and you may very well find them easier and more comfortable to use. `` `make menuconfig' `` is probably the most widely-used. Whatever you choose, it's best to get familiar with the interface because you may find yourself back at it sooner than you think. For those ``running X," you can try `` `make xconfig' `` if you have Tk installed (`click-o-rama' - Nat). `` `make menuconfig' `` is for those who have (n)curses and would prefer a text-based menu. These interfaces have a rather clear advantage: If you goof up and make a wrong choice during configuration, it is simple to go back and fix it.

The configuration options will appear in hierarchies with `` `make menuconfig' `` and `` `make xconfig' ``.

You are ready to answer the questions, usually with `` `y' `` (yes) or `` `n' `` (no). Device drivers typically have an `` `m' `` option. This means ``module," meaning that the system will compile it, but not directly into the kernel, but as a loadable module. A more comical way to describe it is as ``maybe." Some of the more obvious and non-critical options are not described here; see the section ``Other configuration options" for short descriptions of a few others. With `` `make menuconfig' ``, the space bar toggles the selection.

In 2.0.x and later, there is a `` `?' `` option, which provides a brief description of the configuration parameter. That information is likely to be the most up-to-date. Here are a listing of some of the important features, which hierarchy they are in, and brief description.

Kernel math emulation (Processor type and features)

If you don't have a math coprocessor (you have a bare 386 or 486SX), you must say `` `y' `` to this. If you do have a coprocessor and you still say `` `y' ``, don't worry too much -- the coprocessor is still used and the emulation ignored. For any halfway modern machine, the answer will be no, but don't worry if you say yes accidentally; if not needed, it is not used.

Enhanced (MFM/RLL) disk and IDE disk/cdrom support (Block Devices)

You probably need to support this; it means that the kernel will support standard PC hard disks, which most people have. This driver does not include SCSI drives; they come later in the configuration.

You will then be asked about the ``old disk-only" and ``new IDE" drivers. You want to choose one of them; the main difference is that the old driver only supports two disks on a single interface, and the new one supports a secondary interface and IDE/ATAPI cdrom drives. The new driver is 4k larger than the old one and is also supposedly ``improved," meaning that aside from containing a different number of bugs, it might improve your disk performance, especially if you have newer (EIDE-type) hardware.

TOC

Networking support (General Setup)

In principle, you would only say `y' if your machine is on a network such as the internet, or you want to use SLIP, PPP, term, etc to dial up for internet access. However, as many packages (such as the X window system) require networking support even if your machine does not live on a real network, you should say `y'. Later on, you will be asked if you want to support TCP/IP networking; again, say `y' here if you are not absolutely sure.

System V IPC (General Setup)

One of the best definitions of IPC (Interprocess Communication) is in the Perl book's glossary. Not surprisingly, some Perl programmers employ it to let processes talk to each other, as well as many other packages (DOOM, most notably), so it is not a good idea to say n unless you know exactly what you are doing.

Processor family (Processor type and features)

(in older kernels: Use -m486 flag for 486-specific optimizations)

Traditionally, this compiled in certain optimizations for a particular processor; the kernels ran fine on other chips, but the kernel was perhaps a bit larger. In newer kernels, however, this is no longer true, so you should enter the processor for which you are compiling the kernel. A ``386" kernel will work on all machines.

SCSI support

If you have SCSI devices, say `y'. You will be prompted for further information, such as support for CD-ROM, disks, and what kind of SCSI adapter you have. See the SCSI-HOWTO for greater detail.

Network device support

If you have a network card, or you would like to use SLIP, PPP, or a parallel port adapter for connecting to the Internet, say `y'. The config script will prompt for which kind of card you have, and which protocol to use.

Filesystems

The configure script then asks if you wish to support the following filesystems:

Standard (minix) - Newer distributions don't create minix filesystems, and many people don't use it, but it may still be a good idea to configure this one. Some ``rescue disk" programs use it, and still more floppies may have a minix filesystem, since the minix filesystem is less painful to use on a floppy.

Second extended - This is the standard Linux filesystem. You almost definitely have one of these, and need to say `y'.

msdos - If you want to use your MS-DOS hard disk partitions, or mount MS-DOS formatted floppy

TOC

disks, say `y'.

There are various other foreign operating system filesystem types available.

/proc - (idea from Bell Labs, I guess). One doesn't make a proc filesystem on a disk; this is a filesystem interface to the kernel and processes. Many process listers (such as `ps') use it. Try `cat /proc/meminfo' or `cat /proc/devices' sometime. Some shells (rc, in particular) use /proc/self/fd (known as /dev/fd on other systems) for I/O. You should almost certainly say `y' to this; many important linux tools depend on it.

NFS - If your machine lives on a network and you want to use filesystems which reside on other systems with NFS, say `y'.

ISO9660 - Found on most CD-ROMs. If you have a CD-ROM drive and you wish to use it under Linux, say `y'.

But I don't know which filesystems I need!

Ok, type `mount'. The output will look something like this:

```
blah# mount
/dev/hda1 on / type ext2 (defaults)
/dev/hda3 on /usr type ext2 (defaults)
none on /proc type proc (defaults)
/dev/fd0 on /mnt type msdos (defaults)
```

Look at each line; the word next to `type' is the filesystem type. In this example, my / and /usr filesystems are second extended, I'm using /proc, and there's a floppy disk mounted using the msdos (bleah) filesystem.

You can try `cat /proc/filesystems' if you have /proc currently enabled; it will list your current kernel's filesystems.

The configuration of rarely-used, non-critical filesystems can cause kernel bloat; see the section on modules for a way to avoid this and the ``Pitfalls'' section on why a bloated kernel is undesirable.

Character devices

Here, you enable the drivers for your printer (parallel printer, that is), busmouse, PS/2 mouse (many notebooks use the PS/2 mouse protocol for their built-in trackballs), some tape drives, and other such ``character'' devices. Say `y' when appropriate.

Note: gpm is a program which allows the use of the mouse outside of the X window system for cut and paste between virtual consoles. It's fairly nice if you have a serial mouse, because it coexists well with X, but you need to do special tricks for others.

Sound

If you feel a great desire to hear `biff` bark, say `y', and you can tell the configuration program all about your sound board. (A note on sound card configuration: when it asks you if you want to install the full version of the driver, you can say `n' and save some kernel memory by picking only the features which you deem necessary.)

If you are serious about sound card support, have a look at both the free drivers at `http://www.linux.org.uk/OSS/` and the commercial Open Sound System at `http://www.opensound.com/`.

Other configuration options

Not all of the configuration options are listed here because they change too often or fairly self-evident (for instance, 3Com 3C509 support to compile the device drive for this particular ethernet card). There exists a fairly comprehensive list of all the options (plus a way to place them into the `Configure` script) in an effort started and maintained by Axel Boldt (`boldt@math.ucsb.edu`) and it's the online help. It's also available as one big file at the `Documentation/Configure.help` in your Linux kernel source tree as of version 2.0.

Kernel hacking

>From Linus' README:

the ``kernel hacking'' configuration details usually result in a bigger or slower kernel (or both), and can even make the kernel less stable by configuring some routines to actively try to break bad code to find kernel problems (kmalloc()). Thus you should probably answer `n' to the questions for a ``production'' kernel.

3.4 Now what? (The Makefile)

After you finish configuration, a message tells you that your kernel has been configured, and to ``check the top-level `Makefile` for additional configuration,'' etc.

So, look at the `Makefile`. You probably will not need to change it, but it never hurts to look. You can also change its options with the `rdev' command once the new kernel is in place. If you're feel lost when you look at the file, then don't worry about it.

---

4. Compiling the kernel

4.1 Cleaning and depending

When the configure script ends, it also tells you to `make dep' and (possibly) `clean'. So, do the `make dep'. This insures that all of the dependencies, such the include files, are in place. It does not take long, unless your computer is fairly slow to begin with. For older versions of the kernel, when finished, you should do a `make clean'. This removes all of the object files and some other things that an old version leaves behind. In any case, *do not* forget this step before attempting to recompile a kernel.

4.2 Compile time

After depending and cleaning, you may now `make bzImage' or `make bzdisk' (this is the part that takes a long time.). `make bzImage' will compile the kernel, and leave a file in arch/i386/boot called `bzImage' (among other things). This is the new compressed kernel. `make bzdisk' does the same thing, but also places the new bzImage on a floppy disk which you hopefully put in drive ``A:''. `bzdisk' is fairly handy for testing new kernels; if it bombs (or just doesn't work right), just remove the floppy and boot with your old kernel. It can also be a handy way to boot if you accidentally remove your kernel (or something equally as dreadful). You can also use it to install new systems when you just dump the contents of one disk onto the other (``all this and more! NOW how much would you pay?'').

All even halfway reasonably recent kernels are compressed, hence the `bz' in front of the names. A compressed kernel automatically decompresses itself when executed.

In older kernels, you don't have the option to build a bzImage; it was simply a zImage. That option is at the moment still available, however, given the code size of newer kernels, it is now more or less mandatory to build a bzImage because the older methods can't handle a kernel that's just too large.

4.3 Other ``make''ables

`make mrproper' will do a more extensive `clean'ing. It is sometimes necessary; you may wish to do it at every patch. `make mrproper' will also delete your configuration file, so you might want to make a backup of it (.config) if you see it as valuable.

`make oldconfig' will attempt to configure the kernel from an old configuration file; it will run through the `make config' process for you. If you haven't ever compiled a kernel before or don't have an old config file, then you probably shouldn't do this, as you will most likely want to change the default configuration.

See the section on modules for a description of `make modules'.

4.4 Installing the kernel

TOC

After you have a new kernel that seems to work the way you want it to, it's time to install it. Most people use LILO (Linux Loader) for this. `make bzlilo' will install the kernel, run LILO on it, and get you all ready to boot, BUT ONLY if lilo is configured in the following way on your system: kernel is /vmlinuz, lilo is in /sbin, and your lilo config (/etc/lilo.conf) agrees with this.

Otherwise, you need to use LILO directly. It's a fairly easy package to install and work with, but it has a tendency to confuse people with the configuration file. Look at the config file (either /etc/lilo/config for older versions or /etc/lilo.conf for new versions), and see what the current setup is. The config file looks like this:

```
image = /vmlinuz
    label = Linux
    root = /dev/hda1
    ...
```

The `image =' is set to the currently installed kernel. Most people use /vmlinuz. `label' is used by lilo to determine which kernel or operating system to boot, and `root' is the / of that particular operating system. Make a backup copy of your old kernel and copy the bzImage which you just made into place (you would say `cp bzImage /vmlinuz' if you use `/vmlinuz'). Then, rerun lilo -- on newer systems, you can just run `lilo', but on older stuff, you might have to do an /etc/lilo/install or even an /etc/lilo/lilo -C /etc/lilo/config.

If you would like to know more about LILO's configuration, or you don't have LILO, get the newest version from your favorite ftp site and follow the instructions.

To boot one of your old kernels off the hard disk (another way to save yourself in case you screw up the new kernel), copy the lines below (and including) `image = xxx' in the LILO config file to the bottom of the file, and change the `image = xxx' to `image = yyy', where `yyy' is the full pathname of the file you saved your backup kernel to. Then, change the `label = zzz' to `label = linux-backup' and rerun lilo. You may need to put a line in the config file saying `delay=x', where x is an amount in tenths of a second, which tells LILO to wait that much time before booting, so that you can interrupt it (with the shift key, for example), and type in the label of the backup boot image (in case unpleasant things happen).

---

5. Patching the kernel

5.1 Applying a patch

Incremental upgrades of the kernel are distributed as patches. For example, if you have version 1.1.45, and you notice that there's a `patch46.gz' out there for it, it means you can upgrade to version 1.1.46 through application of the patch. You might want to make a backup of the source tree first (`make clean' and then `cd /usr/src; tar zcvf old-tree.tar.gz linux' will make a compressed tar archive for you.).

So, continuing with the example above, let's suppose that you have `patch46.gz' in /usr/src. cd to /usr/src and do a `zcat patch46.gz | patch -p0' (or `patch -p0 < patch46' if the patch isn't compressed). You'll see things whizz by (or flutter by, if your system is that slow) telling you that it is trying to apply hunks, and whether it succeeds or not. Usually, this action goes by too quickly for you to read, and you're not too sure whether it worked or not, so you might want to use the -s flag to patch, which tells patch to only report error messages (you don't get as much of the ``hey, my computer is actually doing something for a change!" feeling, but you may prefer this..). To look for parts which might not have gone smoothly, cd to /usr/src/linux and look for files with a .rej extension. Some versions of patch (older versions which may have been compiled with on an inferior filesystem) leave the rejects with a # extension. You can use `find' to look for you;

```
find .  -name '*.rej' -print
```

prints all files who live in the current directory or any subdirectories with a .rej extension to the standard output.

If everything went right, do a `make clean', `config', and `dep' as described in sections 3 and 4.

There are quite a few options to the patch command. As mentioned above, patch -s will suppress all messages except the errors. If you keep your kernel source in some other place than /usr/src/linux, patch -p1 (in that directory) will patch things cleanly. Other patch options are well-documented in the manual page.

5.2 If something goes wrong

(Note: this section refers mostly to quite old kernels)

The most frequent problem that used to arise was when a patch modified a file called `config.in' and it didn't look quite right, because you changed the options to suit your machine. This has been taken care of, but one still might encounter it with an older release. To fix it, look at the config.in.rej file, and see what remains of the original patch. The changes will typically be marked with `+' and `-' at the beginning of the line. Look at the lines surrounding it, and remember if they were set to `y' or `n'. Now,

TOC

edit `config.in`, and change `y' to `n' and `n' to `y' when appropriate. Do a

```
patch -p0 < config.in.rej
```

and if it reports that it succeeded (no fails), then you can continue on with a configuration and compilation. The `config.in.rej` file will remain, but you can get delete it.

If you encounter further problems, you might have installed a patch out of order. If patch says `previously applied patch detected: Assume -R?', you are probably trying to apply a patch which is below your current version number; if you answer `y', it will attempt to degrade your source, and will most likely fail; thus, you will need to get a whole new source tree (which might not have been such a bad idea in the first place).

To back out (unapply) a patch, use `patch -R' on the original patch.

The best thing to do when patches really turn out wrong is to start over again with a clean, out-of-the-box source tree (for example, from one of the `linux-x.y.z.tar.gz` files), and start again.

5.3 Getting rid of the .orig files

After just a few patches, the `.orig` files will start to pile up. For example, one 1.1.51 tree I had was once last cleaned out at 1.1.48. Removing the .orig files saved over a half a meg.

```
find .  -name '*.orig' -exec rm -f {} ';'
```

will take care of it for you. Versions of `patch` which use # for rejects use a tilde instead of `.orig`.

There are better ways to get rid of the `.orig` files, which depend on GNU `xargs`:

```
find .  -name '*.orig' | xargs rm
```

or the ``quite secure but a little more verbose'' method:

```
find . -name '*.orig' -print0 | xargs --null rm --
```

5.4 Other patches

There are other patches (I'll call them ``nonstandard'') than the ones Linus distributes. If you apply these, Linus' patches may not work correctly and you'll have to either back them out, fix the source or the patch, install a new source tree, or a combination of the above. This can become very frustrating, so if you do not want to modify the source (with the possibility of a very bad outcome), back out the nonstandard patches before applying Linus', or just install a new tree. Then, you can see if the nonstandard patches still work. If they don't, you are either stuck with an old kernel, playing with the patch or source to get it to work, or waiting (possibly begging) for a new version of the patch to come out.

How common are the patches not in the standard distribution? You will probably hear of them. I used to use the noblink patch for my virtual consoles because I hate blinking cursors (This patch is (or at least

was) frequently updated for new kernel releases.). With most newer device drivers being developed a loadable modules, though, the frequecy of ``nonstandard'' patches is decreasing significantly.

6. Additional packages

Your linux kernel has many features which are not explained in the kernel source itself; these features are typically utilized through external packages. Some of the most common are listed here.

6.1 kbd

The linux console probably has more features than it deserves. Among these are the ability to switch fonts, remap your keyboard, switch video modes (in newer kernels), etc. The kbd package has programs which allow the user to do all of this, plus many fonts and keyboard maps for almost any keyboard, and is available from the same sites that carry the kernel source.

6.2 util-linux

Rik Faith (`faith@cs.unc.edu`) put together a large collection of linux utilities which are, by odd coincidence, called util-linux. These are now maintained by Andries Brouwer (`util-linux@math.uio.no`). Available via anonymous ftp from sunsite.unc.edu in `/pub/Linux/system/misc`, it contains programs such as `setterm`, `rdev`, and `ctrlaltdel`, which are relevant to the kernel. As Rik says, *do not install without thinking;* you do not need to install everything in the package, and it could very well cause serious problems if you do.

6.3 hdparm

As with many packages, this was once a kernel patch and support programs. The patches made it into the official kernel, and the programs to optimize and play with your hard disk are distributed separately.

6.4 gpm

gpm stands for general purpose mouse. This program allows you to cut and paste text between virtual consoles and do other things with a large variety of mouse types.

7. Some pitfalls

7.1 make clean

If your new kernel does really weird things after a routine kernel upgrade, chances are you forgot to `make clean` before compiling the new kernel. Symptoms can be anything from your system outright crashing, strange I/O problems, to crummy performance. Make sure you do a `make dep`, too.

7.2 Huge or slow kernels

If your kernel is sucking up a lot of memory, is too large, and/or just takes forever to compile even when you've got your new Quadbazillium-III/4400 working on it, you've probably got lots of unneeded stuff (device drivers, filesystems, etc) configured. If you don't use it, don't configure it, because it does take up memory. The most obvious symptom of kernel bloat is extreme swapping in and out of memory to disk; if your disk is making a lot of noise and it's not one of those old Fujitsu Eagles that sound like like a jet landing when turned off, look over your kernel configuration.

You can find out how much memory the kernel is using by taking the total amount of memory in your machine and subtracting from it the amount of ``total mem'' in `/proc/meminfo` or the output of the command `` `free' ``.

7.3 The parallel port doesn't work/my printer doesn't work

Configuration options for PCs are: First, under the category `General Setup', select `Parallel port support' and `PC-style hardware'. Then under `Character devices', select `Parallel printer support'.

Then there are the names. Linux 2.2 names the printer devices differently than previous releases. The upshot of this is that if you had an `lp1` under your old kernel, it's probably an `lp0` under your new one. Use `` `dmesg' `` or look through the logs in `/var/log` to find out.

7.4 Kernel doesn't compile

If it does not compile, then it is likely that a patch failed, or your source is somehow corrupt. Your version of gcc also might not be correct, or could also be corrupt (for example, the include files might be in error). Make sure that the symbolic links which Linus describes in the `README` are set up correctly. In general, if a standard kernel does not compile, something is seriously wrong with the system, and reinstallation of certain tools is probably necessary.

In some cases, gcc can crash due to hardware problems. The error message will be something like ``xxx exited with signal 15'' and it will generally look very mysterious. I probably would not mention this, except that it happened to me once - I had some bad cache memory, and the compiler would occasionally barf at random. Try reinstalling gcc first if you experience problems. You should only get suspicious if your kernel compiles fine with external cache turned off, a reduced amount of RAM, etc.

TOC

It tends to disturb people when it's suggested that their hardware has problems. Well, I'm not making this u
There is an FAQ for it -- it's at `http://www.bitwizard.nl/sig11/`.

7.5 New version of the kernel doesn't seem to boot

You did not run LILO, or it is not configured correctly. One thing that ``got'' me once was a problem in the config file; it said `` `boot = /dev/hda1' `` instead of `` `boot = /dev/hda' `` (This can be really annoying at first, but once you have a working config file, you shouldn't need to change it.).

7.6 You forgot to run LILO, or system doesn't boot at all

Ooops! The best thing you can do here is to boot off of a floppy disk or CDROM and prepare another bootable floppy (such as `` `make zdisk' `` would do). You need to know where your root (`/`) filesystem is and what type it is (e.g. second extended, minix). In the example below, you also need to know what filesystem your `/usr/src/linux` source tree is on, its type, and where it is normally mounted.

In the following example, `/` is `/dev/hda1`, and the filesystem which holds `/usr/src/linux` is `/dev/hda3`, normally mounted at `/usr`. Both are second extended filesystems. The working kernel image in `/usr/src/linux/arch/i386/boot` is called `bzImage`.

The idea is that if there is a functioning `bzImage`, it is possible to use that for the new floppy. Another alternative, which may or may not work better (it depends on the particular method in which you messed up your system) is discussed after the example.

First, boot from a boot/root disk combo or rescue disk, and mount the filesystem which contains the working kernel image:

```
mkdir /mnt
mount -t ext2 /dev/hda3 /mnt
```

If `mkdir` tells you that the directory already exists, just ignore it. Now, `cd` to the place where the working kernel image was. Note that

```
/mnt + /usr/src/linux/arch/i386/boot - /usr = /mnt/src/linux/arch/i386/boot
```

Place a formatted disk in drive ``A:'' (not your boot or root disk!), dump the image to the disk, and configure it for your root filesystem:

```
cd /mnt/src/linux/arch/i386/boot
dd if=bzImage of=/dev/fd0
rdev /dev/fd0 /dev/hda1
```

`cd` to `/` and unmount the normal `/usr` filesystem:

```
cd /
umount /mnt
```

You should now be able to reboot your system as normal from this floppy. Don't forget to run lilo (or whatever it was that you did wrong) after the reboot!

As mentioned above, there is another common alternative. If you happened to have a working kernel image in / (`/vmlinuz` for example), you can use that for a boot disk. Supposing all of the above conditions, and that my kernel image is `/vmlinuz`, just make these alterations to the example above: change `/dev/hda3` to `/dev/hda1` (the `/` filesystem), `/mnt/src/linux` to `/mnt`, and `if=bzImage` to `if=vmlinuz`. The note explaining how to derive `/mnt/src/linux` may be ignored.

Using LILO with big drives (more than 1024 cylinders) can cause problems. See the LILO mini-HOWTO or documentation for help on that.

7.7 It says `warning: bdflush not running'

This can be a severe problem. Starting with a kernel release after 1.0 (around 20 Apr 1994), a program called `update' which periodically flushes out the filesystem buffers, was upgraded/replaced. Get the sources to `bdflush' (you should find it where you got your kernel source), and install it (you probably want to run your system under the old kernel while doing this). It installs itself as `update' and after a reboot, the new kernel should no longer complain.

7.8 I can't get my IDE/ATAPI CD-ROM drive to work

Strangely enough, lots of people cannot get their ATAPI drives working, probably because there are a number of things that can go wrong.

If your CD-ROM drive is the only device on a particular IDE interface, it must be jumpered as ``master'' or ``single.'' Supposedly, this is the most common error.

Creative Labs (for one) has put IDE interfaces on their sound cards now. However, this leads to the interesting problem that while some people only have one interface to being with, many have two IDE interfaces built-in to their motherboards (at IRQ15, usually), so a common practice is to make the soundblaster interface a third IDE port (IRQ11, or so I'm told).

This causes problems with linux in that versions 1.2.x don't support a third IDE interface (there is support in starting somewhere in the 1.3.x series but that's development, remember, and it doesn't auto-probe). To get around this, you have a few choices.

If you have a second IDE port already, chances are that you are not using it or it doesn't already have two devices on it. Take the ATAPI drive off the sound card and put it on the second interface. You can then disable the sound card's interface, which saves an IRQ anyway.

If you don't have a second interface, jumper the sound card's interface (not the sound card's sound part) as IRQ15, the second interface. It should work.

7.9 It says weird things about obsolete routing requests

Get new versions of the `route` program and any other programs which do route manipulation.

`/usr/include/linux/route.h` (which is actually a file in `/usr/src/linux`) has changed.

TOC

7.10 Firewalling not working in 1.2.0

Upgrade to at least version 1.2.1.

7.11 ``Not a compressed kernel Image file''

Don't use the `vmlinux` file created in `/usr/src/linux` as your boot image; `[..]/arch/i386/boot/bzImage` is the right one.

7.12 Problems with console terminal after upgrade to 1.3.x

Change the word `dumb` to `linux` in the console termcap entry in `/etc/termcap`. You may also have to make a terminfo entry.

7.13 Can't seem to compile things after kernel upgrade

The linux kernel source includes a number of include files (the things that end with `.h`) which are referenced by the standard ones in `/usr/include`. They are typically referenced like this (where `xyzzy.h` would be something in `/usr/include/linux`):

```
#include <linux/xyzzy.h>
```

Normally, there is a link called `linux` in `/usr/include` to the `include/linux` directory of your kernel source (`/usr/src/linux/include/linux` in the typical system). If this link is not there, or points to the wrong place, most things will not compile at all. If you decided that the kernel source was taking too much room on the disk and deleted it, this will obviously be a problem. Another way it might go wrong is with file permissions; if your `root` has a umask which doesn't allow other users to see its files by default, and you extracted the kernel source without the `p` (preserve filemodes) option, those users also won't be able to use the C compiler. Although you could use the `chmod` command to fix this, it is probably easier to re-extract the include files. You can do this the same way you did the whole source at the beginning, only with an additional argument:

```
blah# tar zxvpf linux.x.y.z.tar.gz linux/include
```

Note: ```make config`'' will recreate the `/usr/src/linux` link if it isn't there.

7.14 Increasing limits

The following few *example* commands may be helpful to those wondering how to increase certain soft limits imposed by the kernel:

```
echo 4096 > /proc/sys/kernel/file-max
```

```
echo 12288 > /proc/sys/kernel/inode-max
echo 300 400 500 > /proc/sys/vm/freepages
```

8. Note for upgrade to version 2.0.x, 2.2.x

Kernel versions 2.0.x and 2.2.x introduced quite a bit of changes for kernel installation. The file `Documentation/Changes` in the 2.0.x source tree contains information that you should know when upgrading to either of these versions. You will most likely need to upgrade several key packages, such as gcc, libc, and SysVInit, and perhaps alter some system files, so expect this. Don't panic, though.

9. Modules

Loadable kernel modules can save memory and ease configuration. The scope of modules has grown to include filesystems, ethernet card drivers, tape drivers, printer drivers, and more.

9.1 Installing the module utilities

The module utilities are available from wherever you got your kernel source as `modutils-x.y.z.tar.gz`; choose the highest patchlevel `x.y.z` that is equal to or below that of your current kernel. Unpack it with `` `tar zxvf modutils-x.y.z.tar.gz' ``, `cd` to the directory it creates (`modutils-x.y.z`), look over the `README`, and carry out its installation instructions (which is usually something simple, such as `make install`). You should now have the programs `insmod`, `rmmod`, `ksyms`, `lsmod`, `genksyms`, `modprobe`, and `depmod` in `/sbin`. If you wish, test out the utilities with the ``hw'' example driver in `insmod`; look over the `INSTALL` file in that subdirectory for details.

`insmod` inserts a module into the running kernel. Modules usually have a `.o` extension; the example driver mentioned above is called `drv_hello.o`, so to insert this, one would say `` `insmod drv_hello.o' ``. To see the modules that the kernel is currently using, use `lsmod`. The output looks like this:

```
blah# lsmod
Module:         #pages:  Used by:
drv_hello           1
```

`` `drv_hello' `` is the name of the module, it uses one page (4k) of memory, and no other kernel modules depend on it at the moment. To remove this module, use `` `rmmod drv_hello' ``. Note that `rmmod` wants a *module name*, not a filename; you get this from `lsmod`'s listing. The other module utilities' purposes are documented in their manual pages.

9.2 Modules distributed with the kernel

As of version 2.0.30, most of everything is available as a loadable modules. To use them, first make sure that you don't configure them into the regular kernel; that is, don't say `y` to it during `` `make config' ``. Compile a new kernel and reboot with it. Then, `cd` to `/usr/src/linux` again, and do a `` `make modules' ``. This compiles all of the modules which you did not specify in the kernel configuration, and places links to them in `/usr/src/linux/modules`. You can use them straight from that directory or execute `` `make modules_install' ``, which installs them in `/lib/modules/x.y.z`, where `x.y.z` is the kernel release.

This can be especially handy with filesystems. You may not use the minix or msdos filesystems frequently. For example, if I encountered an msdos (shudder) floppy, I would `insmod`

`/usr/src/linux/modules/msdos.o`, and then `rmmod msdos` when finished. This procedu
saves about 50k of RAM in the kernel during normal operation. A small note is in order for the min
filesystem: you should *always* configure it directly into the kernel for use in ``rescue'' disks.

10. Tips and tricks

10.1 Redirecting output of the make or patch commands

If you would like logs of what those `make' or `patch' commands did, you can redirect output to a file. First, find out what shell you're running: `grep root /etc/passwd' and look for something like `/bin/csh'.

If you use sh or bash,

```
(command) 2>&1 | tee (output file)
```

will place a copy of `(command)'s output in the file `(output file)'.

For csh or tcsh, use

```
(command) |& tee (output file)
```

For rc (Note: you probably do not use rc) it's

```
(command) >[2=1] | tee (output file)
```

10.2 Conditional kernel install

Other than using floppy disks, there are several methods of testing out a new kernel without touching the old one. Unlike many other Unix flavors, LILO has the ability to boot a kernel from anywhere on the disk (if you have a large (500 MB or above) disk, please read over the LILO documentation on how this may cause problems). So, if you add something similar to

```
image = /usr/src/linux/arch/i386/boot/bzImage
    label = new_kernel
```

to the end of your LILO configuration file, you can choose to run a newly compiled kernel without touching your old `/vmlinuz` (after running `lilo`, of course). The easiest way to tell LILO to boot a new kernel is to press the shift key at bootup time (when it says `LILO` on the screen, and nothing else), which gives you a prompt. At this point, you can enter `new_kernel' to boot the new kernel.

If you wish to keep several different kernel source trees on your system at the same time (this can take up a *lot* of disk space; be careful), the most common way is to name them `/usr/src/linux-x.y.z`, where `x.y.z` is the kernel version. You can then ``select'' a source tree with a symbolic link; for example, `ln -sf linux-1.2.2 /usr/src/linux' would make the 1.2.2 tree current. Before

creating a symbolic link like this, make certain that the last argument to `ln` is not a real directory (ol symbolic links are fine); the result will not be what you expect.

10.3 Kernel updates

Russell Nelson (`nelson@crynwr.com`) summarizes the changes in new kernel releases. These are short, and you might like to look at them before an upgrade. They are available with anonymous ftp from `ftp.emlist.com` in `pub/kchanges` or through the URL

```
http://www.crynwr.com/kchanges
```

---

11. Other relevant HOWTOs that might be useful

- Sound-HOWTO: sound cards and utilities
- SCSI-HOWTO: all about SCSI controllers and devices
- NET-2-HOWTO: networking
- PPP-HOWTO: PPP networking in particular
- PCMCIA-HOWTO: about the drivers for your notebook
- ELF-HOWTO: ELF: what it is, converting..
- Hardware-HOWTO: overview of supported hardware
- Module mini-HOWTO: more on kernel modules
- Kerneld mini-HOWTO: about kerneld
- BogoMips mini-HOWTO: in case you were wondering

12. Misc

12.1 Author

The author and maintainer of the Linux Kernel-HOWTO is Brian Ward (`bri@cs.uchicago.edu`). Please send me any comments, additions, corrections (Corrections are, in particular, the most important to me.).

You can take a look at my `home page' at one of these URLs:

```
http://www.math.psu.edu/bri/
http://blah.math.tu-graz.ac.at/~bri/
```

Even though I try to be attentive as possible with mail, please remember that I get a *lot* of it every day, so it may take a little time to get back to you. Especially when emailing me with a question, please try extra hard to be clear and detailed in your message. If you're writing about non-working hardware (or something like that), I need to know what your hardware configureation is. If you report an error, don't just say ``I tried this but it gave an error;'' I need to know what the error was. I would also like to know what versions of the kernel, gcc, and libc you're using. If you just tell me you're using this-or-that distribution, it won't tell me much at all. I don't care if you ask simple questions; remember, if you don't ask, you may never get an answer! I'd like to thank everyone who has given me feedback.

If your question does not relate to the kernel, or is in some language that I don't understand, I may not answer.

If you mailed me and did not get an answer within a resonable amount of time (three weeks or more), then chances are that I accidentally deleted your message or something (sorry). Please try again.

I get a lot of mail about thing which are actually hardware problems or issues. That's OK, but please try to keep in mind that I'm not familiar with all of the hardware in the world. I use AMD processors, Adaptec and Sybios SCSI controllers, and IBM SCSI disks.

Version -0.1 was written on October 3, 1994. This document is available in SGML, PostScript, TeX, roff, and plain-text formats.

12.2 To do

The ``Tips and tricks'' section is a little small. I hope to expand on it with suggestions from others.

So is ``Additional packages.''

More debugging/crash recovery info needed.

TOC

12.3 Contributions

A small part of Linus' README (kernel hacking options) is inclusive. (Thanks, Linus!)

`uc@brian.lunetix.de` (Ulrich Callmeier): patch -s and xargs.

`quinlan@yggdrasil.com` (Daniel Quinlan): corrections and additions in many sections.

`nat@nat@nataa.fr.eu.org` (Nat Makarevitch): mrproper, tar -p, many other things

`boldt@math.ucsb.edu` (Axel Boldt): collected descriptions of kernel configuration options on the net; then provided me with the list

`lembark@wrkhors.psyber.com` (Steve Lembark): multiple boot suggestion

`kbriggs@earwax.pd.uwa.edu.au` (Keith Briggs): some corrections and suggestions

`rmcguire@freenet.columbus.oh.us` (Ryan McGuire): makeables additions

`dumas@excalibur.ibp.fr` (Eric Dumas): French translation

`simazaki@ab11.yamanashi.ac.jp` (Yasutada Shimazaki): Japanese translation

`jjamor@lml.ls.fi.upm.es` (Juan Jose Amor Iglesias): Spanish translation

`mva@sbbs.se` (Martin Wahlen): Swedish translation

`jzp1218@stud.u-szeged.hu` (Zoltan Vamosi): Hungarian translation

`bart@mat.uni.torun.pl` (Bartosz Maruszewski): Polish translation

`donahue@tiber.nist.gov` (Michael J Donahue): typos, winner of the ``sliced bread competition''

`rms@gnu.ai.mit.edu` (Richard Stallman): ``free'' documentation concept/distribution notice

`dak@Pool.Informatik.RWTH-Aachen.DE` (David Kastrup): NFS thing

`esr@snark.thyrsus.com` (Eric Raymond): various tidbits

The people who have sent me mail with questions and problems have also been quite helpful.

12.4 Copyright notice, License, and all that stuff

Copyright © Brian Ward, 1994-1999.

Permission is granted to make and distribute copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the derived work is distributed under the terms of a permission notice identical to this one. Translations fall under the catagory of ``modified versions.''

TOC

Warranty: None.

Recommendations: Commercial redistribution is allowed and encouraged; however, it is strongly recommended that the redistributor contact the author before the redistribution, in the interest of keeping things up-to-date (you could send me a copy of the thing you're making while you're at it). Translators are also advised to contact the author before translating. The printed version looks nicer. Recycle.

Linux NET-3-HOWTO, Linux Networking.

Terry Dawson (main author), VK2KTJ; Alessandro Rubini (maintainer), alessandro.rubini@linux.it

v1.4, August 1998

The Linux Operating System boasts kernel based networking support written almost entirely from scratch. The performance of the tcp/ip implementation in recent kernels makes it a worthy alternative to even the best of its peers. This document aims to describe how to install and configure the Linux networking software and associated tools.

1. Changes from version 1.3 (April 1998)

2. Introduction.

- 2.1 Feedback

3. How to use this HOWTO document (NET-3-HOWTO howto?).

- 3.1 Conventions used in this document

4. General Information about Linux Networking.

- 4.1 A brief history of Linux Networking Kernel Development.
- 4.2 Where to get other information about Linux Networking.
- 4.3 Where to get some non-linux-specific network information.

5. Generic Network Configuration Information.

- 5.1 What do I need to start ?
- 5.2 Where should I put the configuration commands ?
- 5.3 Creating your network interfaces.
- 5.4 Configuring a network interface.
- 5.5 Configuring your Name Resolver.
- 5.6 Configuring your loopback interface.
- 5.7 Routing.
- 5.8 Configuring your network servers and services.
- 5.9 Other miscellaneous network related configuration files.
- 5.10 Network Security and access control.

6. IP- and Ethernet-Related Information

- 6.1 Ethernet
- 6.2 EQL - multiple line traffic equaliser
- 6.3 IP Accounting (for Linux-2.0)
- 6.4 IP Accounting (for Linux-2.2)
- 6.5 IP Aliasing
- 6.6 IP Firewall (for Linux-2.0)
- 6.7 IP Firewall (for Linux-2.2)
- 6.8 IPIP Encapsulation
- 6.9 IP Masquerade (for Linux-2.0)
- 6.10 IP Transparent Proxy
- 6.11 IPv6
- 6.12 Mobile IP
- 6.13 Multicast
- 6.14 NAT - Network Address Translation
- 6.15 Traffic Shaper - Changing allowed bandwidth
- 6.16 Routing in Linux-2.2

TOC

7. Using common PC hardware

- 7.1 ISDN
- 7.2 PLIP for Linux-2.0
- 7.3 PLIP for Linux-2.2
- 7.4 PPP
- 7.5 SLIP client
- 7.6 SLIP server.

8. Other Network Technologies

- 8.1 ARCNet
- 8.2 Appletalk (`AF_APPLETALK`)
- 8.3 ATM
- 8.4 AX25 (`AF_AX25`)
- 8.5 DECNet
- 8.6 FDDI
- 8.7 Frame Relay
- 8.8 IPX (`AF_IPX`)
- 8.9 NetRom (`AF_NETROM`)
- 8.10 Rose protocol (`AF_ROSE`)
- 8.11 SAMBA - `NetBEUI', `NetBios' support.
- 8.12 STRIP support (Starmode Radio IP)
- 8.13 Token Ring
- 8.14 X.25
- 8.15 WaveLan Card

9. Cables and Cabling

- 9.1 Serial NULL Modem cable
- 9.2 Parallel port cable (PLIP cable)
- 9.3 10base2 (thin coax) Ethernet Cabling
- 9.4 Twisted Pair Ethernet Cable

10. Glossary of Terms used in this document.

11. Linux for an ISP ?

12. Acknowledgements

13. Copyright.

---

1. Changes from version 1.3 (April 1998)

```
Additions:
        Traffic Shaper.
        Plip for new kernels

Corrections/Updates:
        Maintainer address for netkit.
        Descriptions of domain names revised.
        General reordering of sections.
        2.0 and 2.2 differences are marked, although some 2.2 info is missing.
        Fixed many references to external documents.
ToDo:
        Describe new routing algorithm
        Add IPv6 kernel compile options
        Describe /proc/sys/net/* entries.
        WanRouter device
        Describe the new firewalling commands for 2.2
```

2. Introduction.

The original NET-FAQ was written by Matt Welsh and Terry Dawson to answer frequently asked questions about networking for Linux at a time before the Linux Documentation Project had formally started. It covered the very early development versions of the Linux Networking Kernel. The NET-2-HOWTO superceded the NET-FAQ and was one of the original LDP HOWTO documents, it covered what was called version 2 and later version 3 of the Linux kernel Networking software. This document in turn supercedes it and relates only to version 3 of the Linux Networking Kernel.

Previous versions of this document became quite large because of the enormous amount of material that fell within its scope. To help reduce this problem a number of HOWTO's dealing with specific networking topics have been produced. This document will provide pointers to them where relevant and cover those areas not yet covered by other documents.

In April 1998 Terry left as NET-3 maintainer, due to his high load. Alessandro Rubini is the new maintainer and would like to keep the document as good as before, although he's new in this kind of stuff.

2.1 Feedback

I always appreciate feedback and especially value contributions. Please direct any feedback or contributions to me by email (rubini@linux.it.

3. How to use this HOWTO document (NET-3-HOWTO howto?).

This document is organized top-down. The first sections include informative material and can be skipped if you are not interested; what follows is a generic discussion of networking issues, and you must ensure you understand this before proceeding to more specific parts. The rest, ``technology specific'' information is grouped in three main sections: Ethernet and IP-related information, technologies pertaining to widespread PC hardware and seldom-used technologies.

The suggested path through the document is thus the following:

Read the generic sections

These sections apply to every, or nearly every, technology described later and so are very important for you to understand. On the other hand, I expect many of the readers to be already confident with this material.

Consider your network

You should know how your network is, or will be, designed and exactly what hardware and technology types you will be implementing.

Read the ``Ethernet and IP'' section if you are directly connected a LAN or the Internet

This section describes basic Ethernet configuration and the various features that Linux offers for IP networks, like firewalling, advanced routing and so on.

Read the next section if you are interested in low-cost local networks or dial-up connections

The section describes PLIP, PPP, SLIP and ISDN, the widespread technologies used on personal workstations.

Read the technology specific sections related to your requirements

If your needs differ from IP and/or common hardware, the final section covers details specific to non-IP protocols and peculiar communication hardware.

Do the configuration work

You should actually try to configure your network and take careful note of any problems you have.

Look for further help if needed

If you experience problems that this document does not help you to resolve then read the section related to where to get help or where to report bugs.

Have fun!

Networking is fun, enjoy it.

3.1 Conventions used in this document

TOC

No special convention is used here, but you must be warned about the way commands are shown. Following the classic Unix documentation, any command you should type to your shell is prefixed by a prompt. This howto shows "`user%`" as the prompt for commands that do not require superuser privileges, and "`root#`" as the prompt for commands that need to run as root. I chose to use "`root#`" instead of a plain "#" to prevent confusion with snapshots from shell scripts, where the hash mark is used to define comment lines.

When ``Kernel Compile Options'' are shown, they are represented in the format used by *menuconfig*. They should be understandable even if you (like me) are not used to *menuconfig*. If you are in doubt about the options' nesting, running the program once can't but help.

Note that any link to other HOWTO's is local to help you browsing your local copy of the LDP documents, in case you are using the html version of this document. If you don't have a complete set of documents, every HOWTO can be retrieved from `sunsite.unc.edu` (directory `/pub/Linux/HOWTO`) and its countless mirrors.

---

4. General Information about Linux Networking.

4.1 A brief history of Linux Networking Kernel Development.

Developing a brand new kernel implementation of the tcp/ip protocol stack that would perform as well as existing implementations was not an easy task. The decision not to port one of the existing implementations was made at a time when there was some uncertainty as to whether the existing implementations may become encumbered by restrictive copyrights because of the court case put by U.S.L. and when there was a lot of fresh enthusiasm for doing it differently and perhaps even better than had already been done.

The original volunteer to lead development of the kernel network code was Ross Biro `<biro@yggdrasil.com>`. Ross produced a simple and incomplete but mostly usable implementation set of routines that were complemented by an ethernet driver for the WD-8003 network interface card. This was enough to get many people testing and experimenting with the software and some people even managed to connect machines in this configuration to live internet connections. The pressure within the Linux community driving development for networking support was building and eventually the cost of a combination of some unfair pressure applied to Ross and his own personal commitments outweighed the benefit he was deriving and he stepped down as lead developer. Ross's efforts in getting the project started and accepting the responsibility for actually producing something useful in such controversial circumstances were what catalyzed all future work and were therefore an essential component of the success of the current product.

Orest Zborowski `<obz@Kodak.COM>` produced the original BSD socket programming interface for the Linux kernel. This was a big step forward as it allowed many of the existing network applications to be ported to linux without serious modification.

Somewhere about this time Laurence Culhane `<loz@holmes.demon.co.uk>` developed the first drivers for Linux to support the SLIP protocol. These enabled many people who did not have access to Ethernet networking to experiment with the new networking software. Again, some people took this driver and pressed it into service to connect them to the Internet. This gave many more people a taste of the possibilities that could be realized if Linux had full networking support and grew the number of users actively using and experimenting with the networking software that existed.

One of the people that had also been actively working on the task of building networking support was Fred van Kempen `<waltje@uwalt.nl.mugnet.org>`. After a period of some uncertainty following Ross's resignation from the lead developer position Fred offered his time and effort and accepted the role essentially unopposed. Fred had some ambitious plans for the direction that he wanted to take the Linux networking software and he set about progressing in those directions. Fred produced a series of networking code called the `NET-2' kernel code (the `NET' code being Ross's) which many people were able to use pretty much usefully. Fred formally put a number of innovations on the

development agenda, such as the dynamic device interface, Amateur Radio AX.25 protocol support a more modularly designed networking implementation. Fred's NET-2 code was used by a fairly large number of enthusiasts, the number increasing all the time as word spread that the software was working. The networking software at this time was still a large number of patches to the standard release of kernel code and was not included in the normal release. The NET-FAQ and subsequent NET-2-HOWTO's described the then fairly complex procedure to get it all working. Fred's focus was on developing innovations to the standard network implementations and this was taking time. The community of users was growing impatient for something that worked reliably and satisfied the 80% of users and, as with Ross, the pressure on Fred as lead developer rose.

Alan Cox `<iialan@www.uk.linux.org>` proposed a solution to the problem designed to resolve the situation. He proposed that he would take Fred's NET-2 code and debug it, making it reliable and stable so that it would satisfy the impatient user base while relieving that pressure from Fred allowing him to continue his work. Alan set about doing this, with some good success and his first version of Linux networking code was called `Net-2D(ebugged)'. The code worked reliably in many typical configurations and the user base was happy. Alan clearly had ideas and skills of his own to contribute to the project and many discussions relating to the direction the NET-2 code was heading ensued. There developed two distinct schools within the Linux networking community, one that had the philosophy of `make it work first, then make it better' and the other of `make it better first'. Linus ultimately arbitrated and offered his support to Alan's development efforts and included Alan's code in the standard kernel source distribution. This placed Fred in a difficult position. Any continued development would lack the large user base actively using and testing the code and this would mean progress would be slow and difficult. Fred continued to work for a short time and eventually stood down and Alan came to be the new leader of the Linux networking kernel development effort.

Donald Becker `<becker@cesdis.gsfc.nasa.gov>` soon revealed his talents in the low level aspects of networking and produced a huge range of ethernet drivers, nearly all of those included in the current kernels were developed by Donald. There have been other people that have made significant contributions, but Donald's work is prolific and so warrants special mention.

Alan continued refining the NET-2-Debugged code for some time while working on progressing some of the matters that remained unaddressed on the `TODO' list. By the time the Linux `1.3.*` kernel source had grown its teeth the kernel networking code had migrated to the NET-3 release on which current versions are based. Alan worked on many different aspects of the networking code and with the assistance of a range of other talented people from the Linux networking community grew the code in all sorts of directions. Alan produced dynamic network devices and the first standard AX.25 and IPX implementations. Alan has continued tinkering with the code, slowly restructuring and enhancing it to the state it is in today.

PPP support was added by Michael Callahan `<callahan@maths.ox.ac.uk>` and Al Longyear `<longyear@netcom.com>` this too was critical to increasing the number of people actively using linux for networking.

Jonathon Naylor `<jsn@cs.nott.ac.uk>` has contributed by significantly enhancing Alan's AX.25 code, adding NetRom and Rose protocol support. The AX.25/NetRom/Rose support itself is quite significant, because no other operating system can boast standard native support for these protocols beside Linux.

There have of course been hundreds of other people who have made significant contribution to the development of the Linux networking software. Some of these you will encounter later in the techno specific sections, other people have contributed modules, drivers, bug-fixes, suggestions, test reports and moral support. In all cases each can claim to have played a part and offered what they could. The Linux kernel networking code is an excellent example of the results that can be obtained from the Linux style of anarchic development, if it hasn't yet surprised you, it is bound to soon enough, the development hasn't stopped.

4.2 Where to get other information about Linux Networking.

There are a number of places where you can find good information about Linux networking.

Alan Cox, the current maintainer of the Linux kernel networking code maintains a world wide web page that contains highlights of current and new developments in linux Networking at: www.uk.linux.org.

Another good place is a book written by Olaf Kirch entitled the `Network Administrators Guide`. It is a work of the Linux Documentation Project and you can read it interactively at Network Administrators Guide HTML version or you can obtain it in various formats by ftp from the sunsite.unc.edu LDP ftp archive. Olaf's book is quite comprehensive and provides a good high level overview of network configuration under linux.

There is a newsgroup in the Linux news hierarchy dedicated to networking and related matters, it is: comp.os.linux.networking

There is a mailing list to which you can subscribe where you may ask questions relating to Linux networking. To subscribe you should send a mail message:

```
To: majordomo@vger.rutgers.edu
Subject: anything at all
Message:

subscribe linux-net
```

On the various IRC networks there are often `#linux` channels on which people will be able to answer questions on linux networking.

Please remember when reporting any problem to include as much relevant detail about the problem as you can. Specifically you should specify the versions of software that you are using, especially the kernel version, the version of tools such as *pppd* or *dip* and the exact nature of the problem you are experiencing. This means taking note of the exact syntax of any error messages you receive and of any commands that you are issuing.

4.3 Where to get some non-linux-specific network information.

TOC

If you are after some basic tutorial information on tcp/ip networking generally, then I recommend you take a look at the following documents:

tcp/ip introduction

this document comes as both a text version and a postscript version.

tcp/ip administration

this document comes as both a text version and a postscript version.

If you are after some more detailed information on tcp/ip networking then I highly recommend:

Internetworking with TCP/IP, Volume 1: principles, protocols and architecture, by Douglas E. Comer, ISBN 0-13-227836-7, Prentice Hall publications, Third Edition, 1995.

If you are wanting to learn about how to write network applications in a Unix compatible environment then I also highly recommend:

Unix Network Programming, by W. Richard Stevens, ISBN 0-13-949876-1, Prentice Hall publications, 1990.

A second edition of this book is appearing on the bookshelves; the new book is made up of three volumes: check Prenice-Hall's web site to probe further.

You might also try the comp.protocols.tcp-ip newsgroup.

An important source of specific technical information relating to the Internet and the tcp/ip suite of protocols are RFC's. RFC is an acronym for `Request For Comment' and is the standard means of submitting and documenting Internet protocol standards. There are many RFC repositories. Many of these sites are ftp sites and other provide World Wide Web access with an associated search engine that allows you to search the RFC database for particular keywords.

One possible source for RFC's is at Nexor RFC database.

---

5. Generic Network Configuration Information.

The following subsections you will pretty much need to know and understand before you actually try to configure your network. They are fundamental principles that apply regardless of the exact nature of the network you wish to deploy.

5.1 What do I need to start ?

Before you start building or configuring your network you will need some things. The most important of these are:

Current Kernel source.

Because the kernel you are running now might not yet have support for the network types or cards that you wish to use you will probably need the kernel source so that you can recompile the kernel with the appropriate options.

You can always obtain the latest kernel source from ftp.kernel.org. Please remember that ftp.kernel.org is seriously overloaded: the preferred way to get current sources is by downloading patches instead of whole tar files; moreover, you should first try to reach mirrors of the main ftp site, like ftp.funet.fi; also remember that every Linux site usually carries updated kernel sources).

Normally the kernel source will be untarred into the `/usr/src/linux` directory. For information on how to apply patches and build the kernel you should read the Kernel-HOWTO. For information on how to configure kernel modules you should read the ``Modules mini-HOWTO''. Also, the `README` file found in the kernel sources and the `Documentation` directory are very informative for the brave reader.

Unless specifically stated otherwise, I recommend you stick with the standard kernel release (the one with the even number as the second digit in the version number). Development release kernels (the ones with the odd second digit) may have structural or other changes that may cause problems working with the other software on your system. If you are uncertain that you could resolve those sorts of problems in addition to the potential for there being other software errors, then don't use them.

On the other hand, some of the features described here have been introduced during the development of 2.1 kernels, so you must take your choice: you can stick to 2.0 while wait for 2.2 and an updated distribution with every new tool, or you can get 2.1 and look around for the various support programs needed to exploit the new features. As I write this paragraph, in August 1998, 2.1.115 is current and 2.2 is expected to appear pretty soon.

Current Network tools.

The network tools are the programs that you use to configure linux network devices. These tools allow you to assign addresses to devices and configure routes for example.

Most modern linux distributions are supplied with the network tools, so if you have installed from a distribution and haven't yet installed the network tools then you should do so.

If you haven't installed from a distribution then you will need to source and compile the tools yourself. This isn't difficult.

The network tools are now maintained by Bernd Eckenfels and are available at: ftp.inka.de and are mirrored at: ftp.uk.linux.org.

Be sure to choose the version that is most appropriate for the kernel you wish to use and follow the instructions in the package to install.

To install and configure the version current at the time of the writing you need do the following:

```
user% tar xvfz net-tools-1.33.tar.gz
user% cd net-tools-1.33
user% make config
user% make
root# make install
```

Additionally, if you intend configuring a firewall or using the IP masquerade feature you will require the *ipfwadm* command. I latest version of it may be obtained from: ftp.xos.nl. Again there are a number of versions available. Be sure to pick the version most closely matches your kernel. Note that the firewalling features of Linux changed during 2.1 development. This only applies to version 2.0 of the kernel.

To install and configure the version current at the time of the writing you need do the following:

```
user% tar xvfz ipfwadm-2.3.0.tar.gz
user% cd ipfwadm-2.3.0
user% make
root# make install
```

Note that if you run version 2.2 (or late 2.1) of the kernel, *ipfwadm* is not the right tool to configure firewalling. This version of the NET-3-HOWTO currently doesn't deal with the new firewalling setup.

Network Application Programs.

The network application programs are programs such as *telnet* and *ftp* and their respective server programs. David Holland has been managing a distribution of the most common of these, which is now maintained by `netbug@ftp.uk.linux.org`. You may obtain the distribution from: ftp.uk.linux.org.

In March 1997 the package has been split to several smaller packages, but in May 1997 the most basic programs has been merged into a package called `netkit-base-0.10`. You might need to get the base package and/or additional packages.

To install and configure the version current at the time of the writing you need do the following:

```
user% tar xvfz netkit-base-0.10
user% cd netkit-base-0.10
user% more README
user% vi MCONFIG
user% make
root# make install
```

Addresses.

Internet Protocol Addresses are composed of four bytes. The convention is to write addresses in what is called `dotted decimal notation'. In this form each byte is converted to a decimal number (0-255) dropping any leading zero's unless the number is zero and written with each byte separated by a `.' character. By convention each interface of a host or router has an IP address. It is legal for the same IP address to be used on each interface of a single machine in some circumstances but usually each interface will have its own address.

Internet Protocol Networks are contiguous sequences of IP addresses. All addresses within a network have a number of digits within the address in common. The portion of the address that is common amongst all addresses within the network is called the `network portion' of the address. The remaining digits are called the `host portion'. The number of bits that are shared by all addresses within a network is called the netmask and it is role of the netmask to determine which addresses belong to the network it is applied to and which don't. For example, consider the following:

```
-----------------  ---------------
Host Address       192.168.110.23
Network Mask       255.255.255.0
Network Portion    192.168.110.
Host portion                  .23
-----------------  ---------------
Network Address    192.168.110.0
Broadcast Address  192.168.110.255
-----------------  ---------------
```

TOC

Any address that is 'bitwise anded' with its netmask will reveal the address of the network it belongs to. The network address is therefore always the lowest numbered address within the range of addresses on the network and always has the host portion of the address coded all zeroes.

The broadcast address is a special address that every host on the network listens to in addition to its own unique address. This address is the one that datagrams are sent to if every host on the network is meant to receive it. Certain types of data like routing information and warning messages are transmitted to the broadcast address so that every host on the network can receive it simultaneously. There are two commonly used standards for what the broadcast address should be. The most widely accepted one is to use the highest possible address on the network as the broadcast address. In the example above this would be `192.168.110.255`. For some reason other sites have adopted the convention of using the network address as the broadcast address. In practice it doesn't matter very much which you use but you must make sure that every host on the network is configured with the same broadcast address.

For administrative reasons some time early in the development of the IP protocol some arbitrary groups of addresses were formed into networks and these networks were grouped into what are called classes. These classes provide a number of standard size networks that could be allocated. The ranges allocated are:

Network Class	Netmask	Network Addresses
A	255.0.0.0	0.0.0.0 - 127.255.255.255
B	255.255.0.0	128.0.0.0 - 191.255.255.255
C	255.255.255.0	192.0.0.0 - 223.255.255.255
Multicast	240.0.0.0	224.0.0.0 - 239.255.255.255

What addresses you should use depends on exactly what it is that you are doing. You may have to use a combination of the following activities to get all the addresses you need:

Installing a linux machine on an existing IP network

If you wish to install a linux machine onto an existing IP network then you should contact whoever administers the network and ask them for the following information:

- Host IP Address
- IP network address
- IP broadcast address
- IP netmask
- Router address
- Domain Name Server Address

You should then configure your linux network device with those details. You can not make them up and expect your configuration to work.

Building a brand new network that will never connect to the Internet

If you are building a private network and you never intend that network to be connected to the Internet then you can choose whatever addresses you like. However, for safety and consistency reasons there have been some IP network addresses that have been reserved specifically for this purpose. These are specified in RFC1597 and are as follows:

RESERVED PRIVATE NETWORK ALLOCATIONS

Network Class	Netmask	Network Addresses
A	255.0.0.0	10.0.0.0 - 10.255.255.255
B	255.255.0.0	172.16.0.0 - 172.31.255.255
C	255.255.255.0	192.168.0.0 - 192.168.255.255

```
-------------------------------------------------------------
```

TOC

You should first decide how large you want your network to be and then choose as many of the addresses as you require.

5.2 Where should I put the configuration commands ?

There are a few different approaches to Linux system boot procedures. After the kernel boots, it always executes a program called `*init*'. The *init* program then reads its configuration file called `/etc/inittab` and commences the boot process. There are a few different flavours of *init* around, although everyone is now converging to the System V (Five) flavour, developed by Miguel van Smoorenburg.

Despite the fact that the *init* program is always the same, the setup of system boot is organized in a different way by each distribution.

Usually the `/etc/inittab` file contains an entry looking something like:

```
si::sysinit:/etc/init.d/boot
```

This line specifies the name of the shell script file that actually manages the boot sequence. This file is somewhat equivalent to the `AUTOEXEC.BAT` file in MS-DOS.

There are usually other scripts that are called by the boot script and often the network is configured within one of many of these.

The following table may be used as a guide for your system:

```
-------------------------------------------------------------------------
Distrib. | Interface Config/Routing          | Server Initialization
-------------------------------------------------------------------------
Debian   | /etc/init.d/network               | /etc/rc2.d/*
-------------------------------------------------------------------------
Slackware| /etc/rc.d/rc.inet1                | /etc/rc.d/rc.inet2
-------------------------------------------------------------------------
RedHat   | /etc/rc.d/init.d/network          | /etc/rc.d/rc3.d/*
-------------------------------------------------------------------------
```

Note that Debian and Red Hat use a whole directory to host scripts that fire up system services (and usually information does not lie within these files, for example Red Hat systems store all of system configuration in files under `/etc/sysconfig`, whence it is retrieved by boot scripts). If you want to grasp the details of the boot process, my suggestion is to check */etc/inittab* and the documentation that accompanies *init*. Linux Journal is also going to publish an article about system initialization, and this document will point to it as soon as it is available on the web.

Most modern distributions include a program that will allow you to configure many of the common sorts of network interfaces. If you have one of these then you should see if it will do what you want before attempting a manual configuration.

```
-----------------------------------------
Distrib   | Network configuration program
-----------------------------------------
RedHat    | /usr/bin/netcfg
Slackware | /sbin/netconfig
-----------------------------------------
```

5.3 Creating your network interfaces.

In many Unix operating systems the network devices have appearances in the */dev* directory. This is not so in Linux. In Linux the network devices are created dynamically in software and do not require device files to be present.

In the majority of cases the network device is automatically created by the device driver while it is initializing and has located your

TOC

hardware. For example, the ethernet device driver creates `eth[0..n]` interfaces sequentially as it locates your ethernet hardware. The first ethernet card found becomes `eth0`, the second `eth1` etc.

In some cases though, notably *slip* and *ppp*, the network devices are created through the action of some user program. The same sequential device numbering applies, but the devices are not created automatically at boot time. The reason for this is that unlike ethernet devices, the number of active *slip* or *ppp* devices may vary during the uptime of the machine. These cases will be covered in more detail in later sections.

5.4 Configuring a network interface.

When you have all of the programs you need and your address and network information you can configure your network interfaces. When we talk about configuring a network interface we are talking about the process of assigning appropriate addresses to a network device and to setting appropriate values for other configurable parameters of a network device. The program most commonly used to do this is the *ifconfig* (interface configure) command.

Typically you would use a command similar to the following:

```
root# ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up
```

In this case I'm configuring an ethernet interface `eth0' with the IP address `192.168.0.1' and a network mask of `255.255.255.0'. The `*up*' that trails the command tells the interface that it should become active, but can usually be omitted, as it is the default. To shutdown an interface, you can just call ``ifconfig eth0 down''.

The kernel assumes certain defaults when configuring interfaces. For example, you may specify the network address and broadcast address for an interface, but if you don't, as in my example above, then the kernel will make reasonable guesses as to what they should be based on the netmask you supply and if you don't supply a netmask then on the network class of the IP address configured. In my example the kernel would assume that it is a class-C network being configured on the interface and configure a network address of `192.168.0.0' and a broadcast address of `192.168.0.255' for the interface.

There are many other options to the *ifconfig* command. The most important of these are:

up

this option activates an interface (and is the default).

down

this option deactivates an interface.

[-]arp

this option enables or disables use of the address resolution protocol on this interface

[-]allmulti

this option enables or disables the reception of all hardware multicast packets. Hardware multicast enables groups of hosts to receive packets addressed to special destinations. This may be of importance if you are using applications like desktop videoconferencing but is normally not used.

mtu N

this parameter allows you to set the *MTU* of this device.

netmask <addr>

this parameter allows you to set the network mask of the network this device belongs to.

irq <addr>

this parameter only works on certain types of hardware and allows you to set the IRQ of the hardware of this device.

[-]broadcast [addr]

this parameter allows you to enable and set the accepting of datagrams destined to the broadcast address, or to disable reception of these datagrams.

[-]pointopoint [addr]

this parameter allows you to set the address of the machine at the remote end of a point to point link such as for *slip* or *p*

hw <type> <addr>

this parameter allows you to set the hardware address of certain types of network devices. This is not often useful for ethernet, but is useful for other network types such as AX.25.

You may use the *ifconfig* command on any network interface. Some user programs such as *pppd* and *dip* automatically configure the network devices as they create them, so manual use of *ifconfig* is unnecessary.

5.5 Configuring your Name Resolver.

The `*Name Resolver*' is a part of the linux standard library. Its prime function is to provide a service to convert human-friendly hostnames like ``ftp.funet.fi'' into machine friendly IP addresses such as `128.214.248.6`.

What's in a name ?

You will probably be familiar with the appearance of Internet host names, but may not understand how they are constructed, or deconstructed. Internet domain names are hierarchical in nature, that is, they have a tree-like structure. A `*domain*' is a family, or group of names. A `*domain*' may be broken down into `*subdomain*'. A `*toplevel domain*' is a domain that is not a subdomain. The Top Level Domains are specified in RFC-920. Some examples of the most common top level domains are:

COM

Commercial Organizations

EDU

Educational Organizations

GOV

Government Organizations

MIL

Military Organizations

ORG

Other organizations

NET

Internet-Related Organizations

Country Designator

these are two letters codes that represent a particular country.

For historical reasons most domains belonging to one of the non-country based top level domains were used by organizations within the United States, although the United States also has its own country code `.us'. This is not true any more for `.com` and `.org` domains, which are commonly used by non-us companies.

Each of these top level domains has subdomains. The top level domains based on country name are often next broken down into subdomains based on the `com`, `edu`, `gov`, `mil` and `org` domains. So for example you end up with: `com.au` and `gov.au` for commercial and government organizations in Australia; note that this is not a general rule, as actual policies depend on the naming authority for each domain.

The next level of division usually represents the name of the organization. Further subdomains vary in nature, often the next level of subdomain is based on the departmental structure of the organization but it may be based on any criterion considered reasonable and meaningful by the network administrators for the organization.

The very left-most portion of the name is always the unique name assigned to the host machine and is called the `*hostname*', the portion of the name to the right of the hostname is called the `*domainname*' and the complete name is called the `*Fully Qualified Domain Name*'.

To use Terry's host as an example, the fully qualified domain name is ``perf.no.itg.telstra.com.au''. This means that the

host name is `perf' and the domain name is `no.itg.telstra.com.au'. The domain name is based on a top level domain based on his country, Australia and as his email address belongs to a commercial organization, `.com' is there as the next level domain. The name of the company is (was) `telstra' and their internal naming structure is based on organizational structure, in this case the machine belongs to the Information Technology Group, Network Operations section.

Usually, the names are fairly shorter; for example, my ISP is called ``systemy.it" and my non-profit organization is called ``linux.it", without any com and org subdomain, so that my own host is just called ``morgana.systemy.it" and rubini@linux.it is a valid email address. Note that the owner of a domain has the rights to register hostnames as well as subdomains; for example, the LUG I belong to uses the domain pluto.linux.it, because the owners of linux.it agreed to open a subdomain for the LUG.

What information you will need.

You will need to know what domain your hosts name will belong to. The name resolver software provides this name translation service by making requests to a `*Domain Name Server*', so you will need to know the IP address of a local nameserver that you can use.

There are three files you need to edit, I'll cover each of these in turn.

/etc/resolv.conf

The /etc/resolv.conf is the main configuration file for the name resolver code. Its format is quite simple. It is a text file with one keyword per line. There are three keywords typically used, they are:

domain

this keyword specifies the local domain name.

search

this keyword specifies a list of alternate domain names to search for a hostname

nameserver

this keyword, which may be used many times, specifies an IP address of a domain name server to query when resolving names

An example /etc/resolv.conf might look something like:

```
domain maths.wu.edu.au
search maths.wu.edu.au wu.edu.au
nameserver 192.168.10.1
nameserver 192.168.12.1
```

This example specifies that the default domain name to append to unqualified names (ie hostnames supplied without a domain) is maths.wu.edu.au and that if the host is not found in that domain to also try the wu.edu.au domain directly. Two nameservers entry are supplied, each of which may be called upon by the name resolver code to resolve the name.

/etc/host.conf

The /etc/host.conf file is where you configure some items that govern the behaviour of the name resolver code. The format of this file is described in detail in the `resolv+' man page. In nearly all circumstances the following example will work for you:

```
order hosts,bind
multi on
```

This configuration tells the name resolver to check the /etc/hosts file before attempting to query a nameserver and to return all valid addresses for a host found in the /etc/hosts file instead of just the first.

/etc/hosts

The `/etc/hosts` file is where you put the name and IP address of local hosts. If you place a host in this file then you do not need to query the domain name server to get its IP Address. The disadvantage of doing this is that you must keep this file up to date yourself if the IP address for that host changes. In a well managed system the only hostnames that usually appear in this file are an entry for the loopback interface and the local hosts name.

```
# /etc/hosts
127.0.0.1       localhost loopback
192.168.0.1     this.host.name
```

You may specify more than one host name per line as demonstrated by the first entry, which is a standard entry for the loopback interface.

Running a name server

If you want to run a local nameserver, you can do it easily. Please refer to the DNS-HOWTO and to any documents included in your version of *BIND* (Berkeley Internet Name Domain).

5.6 Configuring your loopback interface.

The `loopback' interface is a special type of interface that allows you to make connections to yourself. There are various reasons why you might want to do this, for example, you may wish to test some network software without interfering with anybody else on your network. By convention the IP address `127.0.0.1' has been assigned specifically for loopback. So no matter what machine you go to, if you open a telnet connection to `127.0.0.1` you will always reach the local host.

Configuring the loopback interface is simple and you should ensure you do (but note that this task is usually performed by the standard initialization scripts).

```
root# ifconfig lo 127.0.0.1
root# route add -host 127.0.0.1 lo
```

We'll talk more about the *route* command in the next section.

5.7 Routing.

Routing is a big topic. It is easily possible to write large volumes of text about it. Most of you will have fairly simple routing requirements, some of you will not. I will cover some basic fundamentals of routing only. If you are interested in more detailed information then I suggest you refer to the references provided at the start of the document.

Let's start with a definition. What is IP routing ? Here is one that I'm using:

> IP Routing is the process by which a host with multiple network connections decides where to deliver IP datagrams it has received.

It might be useful to illustrate this with an example. Imagine a typical office router, it might have a PPP link off the Internet, a number of ethernet segments feeding the workstations and another PPP link off to another office. When the router receives a datagram on any of its network connections, routing is the mechanism that it uses to determine which interface it should send the datagram to next. Simple hosts also need to route, all Internet hosts have two network devices, one is the loopback interface described above and the other is the one it uses to talk to the rest of the network, perhaps an ethernet, perhaps a PPP or SLIP serial interface.

Ok, so how does routing work ? Each host keeps a special list of routing rules, called a routing table. This table contains rows which typically contain at least three fields, the first is a destination address, the second is the name of the interface to which the datagram is to be routed and the third is optionally the IP address of another machine which will carry the datagram on its next step through the network. In linux you can see this table by using the following command:

TOC

```
user% cat /proc/net/route
```

or by using either of the following commands:

```
user% /sbin/route -n
user% netstat -r
```

The routing process is fairly simple: an incoming datagram is received, the destination address (who it is for) is examined and compared with each entry in the table. The entry that best matches that address is selected and the datagram is forwarded to the specified interface. If the gateway field is filled then the datagram is forwarded to that host via the specified interface, otherwise the destination address is assumed to be on the network supported by the interface.

To manipulate this table a special command is used. This command takes command line arguments and converts them into kernel system calls that request the kernel to add, delete or modify entries in the routing table. The command is called `*route*'.

A simple example. Imagine you have an ethernet network. You've been told it is a class-C network with an address of `192.168.1.0`. You've been supplied with an IP address of `192.168.1.10` for your use and have been told that `192.168.1.1` is a router connected to the Internet.

The first step is to configure the interface as described earlier. You would use a command like:

```
root# ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
```

You now need to add an entry into the routing table to tell the kernel that datagrams for all hosts with addresses that match `192.168.1.*` should be sent to the ethernet device. You would use a command similar to:

```
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
```

Note the use of the `-net' argument to tell the route program that this entry is a network route. Your other choice here is a `-host' route which is a route that is specific to one IP address.

This route will enable you to establish IP connections with all of the hosts on your ethernet segment. But what about all of the IP hosts that aren't on your ethernet segment ?

It would be a very difficult job to have to add routes to every possible destination network, so there is a special trick that is used to simplify this task. The trick is called the `default' route. The `default` route matches every possible destination, but poorly, so that if any other entry exists that matches the required address it will be used instead of the `default` route. The idea of the `default` route is simply to enable you to say "and everything else should go here". In the example I've contrived you would use an entry like:

```
root# route add default gw 192.168.1.1 eth0
```

The `gw' argument tells the route command that the next argument is the IP address, or name, of a gateway or router machine which all datagrams matching this entry should be directed to for further routing.

So, your complete configuration would look like:

```
root# ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
root# route add default gw 192.168.1.1 eth0
```

If you take a close look at your network `rc' files you will find that at least one of them looks very similar to this. This is a very common configuration.

TOC

Let's now look at a slightly more complicated routing configuration. Let's imagine we are configuring the router we looked at e the one supporting the PPP link to the Internet and the lan segments feeding the workstations in the office. Lets imagine the ro has three ethernet segments and one PPP link. Our routing configuration would look something like:

```
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
root# route add -net 192.168.2.0 netmask 255.255.255.0 eth1
root# route add -net 192.168.3.0 netmask 255.255.255.0 eth2
root# route add default ppp0
```

Each of the workstations would use the simpler form presented above, only the router needs to specify each of the network routes separately because for the workstations the `default` route mechanism will capture all of them letting the router worry about splitting them up appropriately. You may be wondering why the default route presented doesn't specify a `gw'. The reason for this is simple, serial link protocols such as PPP and slip only ever have two hosts on their network, one at each end. To specify the host at the other end of the link as the gateway is pointless and redundant as there is no other choice, so you do not need to specify a gateway for these types of network connections. Other network types such as ethernet, arcnet or token ring do require the gateway to be specified as these networks support large numbers of hosts on them.

So what does the *routed* program do ?

The routing configuration described above is best suited to simple network arrangements where there are only ever single possible paths to destinations. When you have a more complex network arrangement things get a little more complicated. Fortunately for most of you this won't be an issue.

The big problem with `manual routing' or `static routing' as described, is that if a machine or link fails in your network then the only way you can direct your datagrams another way, if another way exists, is by manually intervening and executing the appropriate commands. Naturally this is clumsy, slow, impractical and hazard prone. Various techniques have been developed to automatically adjust routing tables in the event of network failures where there are alternate routes, all of these techniques are loosely grouped by the term `dynamic routing protocols'.

You may have heard of some of the more common dynamic routing protocols. The most common are probably RIP (Routing Information Protocol) and OSPF (Open Shortest Path First Protocol). The Routing Information Protocol is very common on small networks such as small-medium sized corporate networks or building networks. OSPF is more modern and more capable at handling large network configurations and better suited to environments where there is a large number of possible paths through the network. Common implementations of these protocols are: `*routed*' - RIP and `*gated*' - RIP, OSPF and others. The `*routed*' program is normally supplied with your Linux distribution or is included in the `NetKit' package detailed above.

An example of where and how you might use a dynamic routing protocol might look something like the following:

```
     192.168.1.0 /                          192.168.2.0 /
        255.255.255.0                          255.255.255.0
      -                                      -
      |                                      |
      |   /-----\                 /-----\    |
      |   |     |ppp0   //    ppp0|     |    |
eth0  |---|  A  |------//---------|  B  |---| eth0
      |   |     |     //          |     |    |
      |   \-----/                 \-----/    |
      |      \ ppp1             ppp1 /       |
      -       \                   /        -
               \                 /
                \               /
                 \             /
                  \           /
                   \         /
                    \       /
                     \     /
                   ppp0\   /ppp1
```

```
                    /-----\
                    |     |
                    |  C  |
                    |     |
                    \-----/
                       |eth0
                       |
                 |---------|
                 192.168.3.0 /
                    255.255.255.0
```

We have three routers A, B and C. Each supports one ethernet segment with a Class C IP network (netmask 255.255.255.0). Each router also has a PPP link to each of the other routers. The network forms a triangle.

It should be clear that the routing table at router A could look like:

```
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
root# route add -net 192.168.2.0 netmask 255.255.255.0 ppp0
root# route add -net 192.168.3.0 netmask 255.255.255.0 ppp1
```

This would work just fine until the link between router A and B should fail. If that link failed then with the routing entry shown above hosts on the ethernet segment of A could not reach hosts on the ethernet segment on B because their datagram would be directed to router A's ppp0 link which is broken. They could still continue to talk to hosts on the ethernet segment of C and hosts on the C's ethernet segment could still talk to hosts on B's ethernet segment because the link between B and C is still intact.

But wait, if A can talk to C and C can still talk to B, why shouldn't A route its datagrams for B via C and let C send them to B ? This is exactly the sort of problem that dynamic routing protocols like RIP were designed to solve. If each of the routers A, B and C were running a routing daemon then their routing tables would be automatically adjusted to reflect the new state of the network should any one of the links in the network fail. To configure such a network is simple, at each router you need only do two things. In this case for Router A:

```
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
root# /usr/sbin/routed
```

The `*routed*' routing daemon automatically finds all active network ports when it starts and sends and listens for messages on each of the network devices to allow it to determine and update the routing table on the host.

This has been a very brief explanation of dynamic routing and where you would use it. If you want more information then you should refer to the suggested references listed at the top of the document.

The important points relating to dynamic routing are:

1. You only need to run a dynamic routing protocol daemon when your Linux machine has the possibility of selecting multiple possible routes to a destination.
2. The dynamic routing daemon will automatically modify your routing table to adjust to changes in your network.
3. RIP is suited to small to medium sized networks.

5.8 Configuring your network servers and services.

Network servers and services are those programs that allow a remote user to make user of your Linux machine. Server programs listen on network ports. Network ports are a means of addressing a particular service on any particular host and are how a server knows the difference between an incoming telnet connection and an incoming ftp connection. The remote user establishes a network connection to your machine and the server program, the network daemon program, listening on that port accepts the connection and executes. There are two ways that network daemons may operate. Both are commonly employed in practice. The two ways are:

standalone

the network daemon program listens on the designated network port and when an incoming connection is made it manages the network connection itself to provide the service.

slave to the *inetd* server

the *inetd* server is a special network daemon program that specializes in managing incoming network connections. It has a configuration file which tells it what program needs to be run when an incoming connection is received. Any service port may be configured for either of the tcp or udp protcols. The ports are described in another file that we will talk about soon.

There are two important files that we need to configure. They are the `/etc/services` file which assigns names to port numbers and the `/etc/inetd.conf` file which is the configuration file for the *inetd* network daemon.

`/etc/services`

The `/etc/services` file is a simple database that associates a human friendly name to a machine friendly service port. Its format is quite simple. The file is a text file with each line representing and entry in the database. Each entry is comprised of three fields separated by any number of whitespace (tab or space) characters. The fields are:

```
name      port/protocol        aliases     # comment
```

name

a single word name that represents the service being described.

port/protocol

this field is split into two subfields.

port

a number that specifies the port number the named service will be available on. Most of the common services have assigned service numbers. These are described in `RFC-1340`.

protocol

this subfield may be set to either `tcp` or `udp`.

It is important to note that an entry of `18/tcp` is very different from an entry of `18/udp` and that there is no technical reason why the same service needs to exist on both. Normally common sense prevails and it is only if a particular service is available via both `tcp` and `udp` that you will see an entry for both.

aliases

other names that may be used to refer to this service entry.

Any text appearing in a line after a `#' character is ignored and treated as a comment.

An example `/etc/services` file.

All modern linux distributions provide a good `/etc/services` file. Just in case you happen to be building a machine from the ground up, here is a copy of the `/etc/services` file supplied with an old Debian distribution:

```
# /etc/services:
# $Id: services,v 1.3 1996/05/06 21:42:37 tobias Exp $
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1340, ``Assigned Numbers'' (July 1992).  Not all ports
# are included, only the more common ones.

tcpmux          1/tcp                           # TCP port service multiplexer
echo            7/tcp
echo            7/udp
```

TOC

```
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
qotd            17/tcp          quote
msp             18/tcp                          # message send protocol
msp             18/udp                          # message send protocol
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
telnet          23/tcp
# 24 - private
smtp            25/tcp          mail
# 26 - unassigned
time            37/tcp          timserver
time            37/udp          timserver
rlp             39/udp          resource        # resource location
nameserver      42/tcp          name            # IEN 116
whois           43/tcp          nicname
re-mail-ck      50/tcp                          # Remote Mail Checking Protocol
re-mail-ck      50/udp                          # Remote Mail Checking Protocol
domain          53/tcp          nameserver      # name-domain server
domain          53/udp          nameserver
mtp             57/tcp                          # deprecated
bootps          67/tcp                          # BOOTP server
bootps          67/udp
bootpc          68/tcp                          # BOOTP client
bootpc          68/udp
tftp            69/udp
gopher          70/tcp                          # Internet Gopher
gopher          70/udp
rje             77/tcp          netrjs
finger          79/tcp
www             80/tcp          http            # WorldWideWeb HTTP
www             80/udp                          # HyperText Transfer Protocol
link            87/tcp          ttylink
kerberos        88/tcp          kerberos5 krb5  # Kerberos v5
kerberos        88/udp          kerberos5 krb5  # Kerberos v5
supdup          95/tcp
# 100 - reserved
hostnames       101/tcp         hostname        # usually from sri-nic
iso-tsap        102/tcp         tsap            # part of ISODE.
csnet-ns        105/tcp         cso-ns          # also used by CSO name server
csnet-ns        105/udp         cso-ns
rtelnet         107/tcp                         # Remote Telnet
rtelnet         107/udp
pop-2           109/tcp         postoffice      # POP version 2
pop-2           109/udp
pop-3           110/tcp                         # POP version 3
pop-3           110/udp
sunrpc          111/tcp         portmapper      # RPC 4.0 portmapper TCP
sunrpc          111/udp         portmapper      # RPC 4.0 portmapper UDP
auth            113/tcp         authentication tap ident
```

TOC

```
sftp            115/tcp
uucp-path       117/tcp
nntp            119/tcp         readnews untp   # USENET News Transfer Protocol
ntp             123/tcp
ntp             123/udp                         # Network Time Protocol
netbios-ns      137/tcp                         # NETBIOS Name Service
netbios-ns      137/udp
netbios-dgm     138/tcp                         # NETBIOS Datagram Service
netbios-dgm     138/udp
netbios-ssn     139/tcp                         # NETBIOS session service
netbios-ssn     139/udp
imap2           143/tcp                         # Interim Mail Access Proto v2
imap2           143/udp
snmp            161/udp                         # Simple Net Mgmt Proto
snmp-trap       162/udp         snmptrap        # Traps for SNMP
cmip-man        163/tcp                         # ISO mgmt over IP (CMOT)
cmip-man        163/udp
cmip-agent      164/tcp
cmip-agent      164/udp
xdmcp           177/tcp                         # X Display Mgr. Control Proto
xdmcp           177/udp
nextstep        178/tcp         NeXTStep NextStep       # NeXTStep window
nextstep        178/udp         NeXTStep NextStep       # server
bgp             179/tcp                         # Border Gateway Proto.
bgp             179/udp
prospero        191/tcp                         # Cliff Neuman's Prospero
prospero        191/udp
irc             194/tcp                         # Internet Relay Chat
irc             194/udp
smux            199/tcp                         # SNMP Unix Multiplexer
smux            199/udp
at-rtmp         201/tcp                         # AppleTalk routing
at-rtmp         201/udp
at-nbp          202/tcp                         # AppleTalk name binding
at-nbp          202/udp
at-echo         204/tcp                         # AppleTalk echo
at-echo         204/udp
at-zis          206/tcp                         # AppleTalk zone information
at-zis          206/udp
z3950           210/tcp         wais            # NISO Z39.50 database
z3950           210/udp         wais
ipx             213/tcp                         # IPX
ipx             213/udp
imap3           220/tcp                         # Interactive Mail Access
imap3           220/udp                         # Protocol v3
ulistserv       372/tcp                         # UNIX Listserv
ulistserv       372/udp
#
# UNIX specific services
#
exec            512/tcp
biff            512/udp         comsat
login           513/tcp
who             513/udp         whod
shell           514/tcp         cmd             # no passwords used
syslog          514/udp
printer         515/tcp         spooler         # line printer spooler
talk            517/udp
```

TOC

```
ntalk           518/udp
route           520/udp         router routed   # RIP
timed           525/udp         timeserver
tempo           526/tcp         newdate
courier         530/tcp         rpc
conference      531/tcp         chat
netnews         532/tcp         readnews
netwall         533/udp                         # -for emergency broadcasts
uucp            540/tcp         uucpd           # uucp daemon
remotefs        556/tcp         rfs_server rfs  # Brunhoff remote filesystem
klogin          543/tcp                         # Kerberized `rlogin' (v5)
kshell          544/tcp         krcmd           # Kerberized `rsh' (v5)
kerberos-adm    749/tcp                         # Kerberos `kadmin' (v5)
#
webster         765/tcp                         # Network dictionary
webster         765/udp
#
# From ``Assigned Numbers'':
#
#> The Registered Ports are not controlled by the IANA and on most systems
#> can be used by ordinary user processes or programs executed by ordinary
#> users.
#
#> Ports are used in the TCP [45,106] to name the ends of logical
#> connections which carry long term conversations.  For the purpose of
#> providing services to unknown callers, a service contact port is
#> defined.  This list specifies the port used by the server process as its
#> contact port.  While the IANA can not control uses of these ports it
#> does register or list uses of these ports as a convenience to the
#> community.
#
ingreslock      1524/tcp
ingreslock      1524/udp
prospero-np     1525/tcp                # Prospero non-privileged
prospero-np     1525/udp
rfe             5002/tcp                # Radio Free Ethernet
rfe             5002/udp                # Actually uses UDP only
bbs             7000/tcp                # BBS service
#
#
# Kerberos (Project Athena/MIT) services
# Note that these are for Kerberos v4 and are unofficial.  Sites running
# v4 should uncomment these and comment out the v5 entries above.
#
kerberos4       750/udp         kdc     # Kerberos (server) udp
kerberos4       750/tcp         kdc     # Kerberos (server) tcp
kerberos_master 751/udp                 # Kerberos authentication
kerberos_master 751/tcp                 # Kerberos authentication
passwd_server   752/udp                 # Kerberos passwd server
krb_prop        754/tcp                 # Kerberos slave propagation
krbupdate       760/tcp         kreg    # Kerberos registration
kpasswd         761/tcp         kpwd    # Kerberos "passwd"
kpop            1109/tcp                # Pop with Kerberos
knetd           2053/tcp                # Kerberos de-multiplexor
zephyr-srv      2102/udp                # Zephyr server
zephyr-clt      2103/udp                # Zephyr serv-hm connection
zephyr-hm       2104/udp                # Zephyr hostmanager
eklogin         2105/tcp                # Kerberos encrypted rlogin
```

TOC

```
#
# Unofficial but necessary (for NetBSD) services
#
supfilesrv      871/tcp                 # SUP server
supfiledbg      1127/tcp                # SUP debugging
#
# Datagram Delivery Protocol services
#
rtmp            1/ddp                   # Routing Table Maintenance Protocol
nbp             2/ddp                   # Name Binding Protocol
echo            4/ddp                   # AppleTalk Echo Protocol
zip             6/ddp                   # Zone Information Protocol
#
# Debian GNU/Linux services
rmtcfg          1236/tcp                # Gracilis Packeten remote config server
xtel            1313/tcp                # french minitel
cfinger         2003/tcp                # GNU Finger
postgres        4321/tcp                # POSTGRES
mandelspawn     9359/udp        mandelbrot      # network mandelbrot

# Local services
```

In the real world, the actual file is always growing as new services are being created. If you fear your own copy is incomplete, I'd suggest to copy a new `/etc/services` from a recent distribution.

/etc/inetd.conf

The `/etc/inetd.conf` file is the configuration file for the *inetd* server daemon. Its function is to tell *inetd* what to do when it receives a connection request for a particular service. For each service that you wish to accept connections for you must tell *inetd* what network server daemon to run and how to run it.

Its format is also fairly simple. It is a text file with each line describing a service that you wish to provide. Any text in a line following a `#' is ignored and considered a comment. Each line contains seven fields separated by any number of whitespace (tab or space) characters. The general format is as follows:

```
service  socket_type  proto  flags  user  server_path  server_args
```

service

is the service relevant to this configuration as taken from the `/etc/services` file.

socket_type

this field describes the type of socket that this entry will consider relevant, allowable values are: `stream`, `dgram`, `raw`, `rdm`, or `seqpacket`. This is a little technical in nature, but as a rule of thumb nearly all `tcp` based services use `stream` and nearly all `udp` based services use `dgram`. It is only very special types of server daemons that would use any of the other values.

proto

the protocol to considered valid for this entry. This should match the appropriate entry in the `/etc/services` file and will typically be either `tcp` or `udp`. Sun RPC (Remote Procedure Call) based servers will use `rpc/tcp` or `rpc/udp`.

flags

there are really only two possible settings for this field. This field setting tells *inetd* whether the network server program frees the socket after it has been started and therefore whether *inetd* can start another one on the next connection request, or whether *inetd* should wait and assume that any server daemon already running will handle the new connection request. Again this is a little tricky to work out, but as a rule of thumb all `tcp` servers should have this entry set to `nowait` and most `udp` servers should have this entry set to `wait`. Be warned there are some notable exceptions to this, so let the example guide you if you are not sure.

user

this field describes which user account from `/etc/passwd` will be set as the owner of the network daemon when it is started. This is often useful if you want to safeguard against security risks. You can set the user of an entry to the `nobody` user so that if the network server security is breached the possible damage is minimized. Typically this field is set to `root` though, because many servers require root privileges in order to function correctly.

server_path

this field is pathname to the actual server program to execute for this entry.

server_args

this field comprises the rest of the line and is optional. This field is where you place any command line arguments that you wish to pass to the server daemon program when it is launched.

An example `/etc/inetd.conf`

As for the `/etc/services` file all modern distributions will include a good `/etc/inetd.conf` file for you to work with. Here, for completeness is the `/etc/inetd.conf` file from the Debian distribution.

```
# /etc/inetd.conf:  see inetd(8) for further informations.
#
# Internet server configuration database
#
#
# Modified for Debian by Peter Tobias <tobias@et-inf.fho-emden.de>
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
# Internal services
#
#echo           stream  tcp     nowait  root    internal
#echo           dgram   udp     wait    root    internal
discard         stream  tcp     nowait  root    internal
discard         dgram   udp     wait    root    internal
daytime         stream  tcp     nowait  root    internal
daytime         dgram   udp     wait    root    internal
#chargen        stream  tcp     nowait  root    internal
#chargen        dgram   udp     wait    root    internal
time            stream  tcp     nowait  root    internal
time            dgram   udp     wait    root    internal
#
# These are standard services.
#
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.telnetd
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.ftpd
#fsp    dgram   udp     wait    root    /usr/sbin/tcpd  /usr/sbin/in.fspd
#
# Shell, login, exec and talk are BSD protocols.
#
shell   stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rshd
login   stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rlogind
#exec   stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rexecd
talk    dgram   udp     wait    root    /usr/sbin/tcpd  /usr/sbin/in.talkd
ntalk   dgram   udp     wait    root    /usr/sbin/tcpd  /usr/sbin/in.ntalkd
#
# Mail, news and uucp services.
#
smtp    stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.smtpd
```

TOC

```
#nntp   stream  tcp     nowait  news    /usr/sbin/tcpd  /usr/sbin/in.nntpd
#uucp   stream  tcp     nowait  uucp    /usr/sbin/tcpd  /usr/lib/uucp/uucico
#comsat dgram   udp     wait    root    /usr/sbin/tcpd  /usr/sbin/in.comsat
#
# Pop et al
#
#pop-2  stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.pop2d
#pop-3  stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.pop3d
#
# `cfinger' is for the GNU finger server available for Debian.  (NOTE: The
# current implementation of the `finger' daemon allows it to be run as `root'.)
#
#cfinger stream tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.cfingerd
#finger stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.fingerd
#netstat        stream  tcp     nowait  nobody  /usr/sbin/tcpd  /bin/netstat
#systat stream  tcp     nowait  nobody  /usr/sbin/tcpd  /bin/ps -auwwx
#
# Tftp service is provided primarily for booting.  Most sites
# run this only on machines acting as "boot servers."
#
#tftp   dgram   udp     wait    nobody  /usr/sbin/tcpd  /usr/sbin/in.tftpd
#tftp   dgram   udp     wait    nobody  /usr/sbin/tcpd  /usr/sbin/in.tftpd /boot
#bootps dgram   udp     wait    root    /usr/sbin/bootpd        bootpd -i -t 120
#
# Kerberos authenticated services (these probably need to be corrected)
#
#klogin         stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rlogind
-k
#eklogin        stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rlogind
-k -x
#kshell         stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rshd -k
#
# Services run ONLY on the Kerberos server (these probably need to be corrected)
#
#krbupdate      stream tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/registerd
#kpasswd        stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/kpasswdd
#
# RPC based services
#
#mountd/1       dgram   rpc/udp wait    root    /usr/sbin/tcpd  /usr/sbin/rpc.mountd
#rstatd/1-3     dgram   rpc/udp wait    root    /usr/sbin/tcpd  /usr/sbin/rpc.rstatd
#rusersd/2-3    dgram   rpc/udp wait    root    /usr/sbin/tcpd  /usr/sbin/rpc.rusersd
#walld/1        dgram   rpc/udp wait    root    /usr/sbin/tcpd  /usr/sbin/rpc.rwalld
#
# End of inetd.conf.
ident           stream  tcp     nowait  nobody  /usr/sbin/identd        identd -i
```

5.9 Other miscellaneous network related configuration files.

There are a number of miscellaneous files relating to network configuration under linux that you might be interested in. You may never have to modify these files, but it is worth describing them so you know what they contain and what they are for.

`/etc/protocols`

The `/etc/protocols` file is a database that maps protocol id numbers against protocol names. This is used by programmers to allow them to specify protocols by name in their programs and also by some programs such as *tcpdump* to allow them to display names instead of numbers in their output. The general syntax of the file is:

TOC

```
protocolname  number  aliases
```

The `/etc/protocols` file supplied with the Debian distribution is as follows:

```
# /etc/protocols:
# $Id: protocols,v 1.1 1995/02/24 01:09:41 imurdock Exp $
#
# Internet (IP) protocols
#
#       from: @(#)protocols     5.1 (Berkeley) 4/17/89
#
# Updated for NetBSD based on RFC 1340, Assigned Numbers (July 1992).

ip      0       IP              # internet protocol, pseudo protocol number
icmp    1       ICMP            # internet control message protocol
igmp    2       IGMP            # Internet Group Management
ggp     3       GGP             # gateway-gateway protocol
ipencap 4       IP-ENCAP        # IP encapsulated in IP (officially ``IP'')
st      5       ST              # ST datagram mode
tcp     6       TCP             # transmission control protocol
egp     8       EGP             # exterior gateway protocol
pup     12      PUP             # PARC universal packet protocol
udp     17      UDP             # user datagram protocol
hmp     20      HMP             # host monitoring protocol
xns-idp 22      XNS-IDP         # Xerox NS IDP
rdp     27      RDP             # "reliable datagram" protocol
iso-tp4 29      ISO-TP4         # ISO Transport Protocol class 4
xtp     36      XTP             # Xpress Tranfer Protocol
ddp     37      DDP             # Datagram Delivery Protocol
idpr-cmtp       39      IDPR-CMTP       # IDPR Control Message Transport
rspf    73      RSPF            # Radio Shortest Path First.
vmtp    81      VMTP            # Versatile Message Transport
ospf    89      OSPFIGP         # Open Shortest Path First IGP
ipip    94      IPIP            # Yet Another IP encapsulation
encap   98      ENCAP           # Yet Another IP encapsulation
```

/etc/networks

The `/etc/networks` file has a similar function to that of the `/etc/hosts` file. It provides a simple database of network names against network addresses. Its format differs in that there may be only two fields per line and that the fields are coded as:

```
networkname networkaddress
```

An example might look like:

```
loopnet    127.0.0.0
localnet   192.168.0.0
amprnet    44.0.0.0
```

When you use commands like the *route* command, if a destination is a network and that network has an entry in the `/etc/networks` file then the route command will display that network name instead of its address.

5.10 Network Security and access control.

Let me start this section by warning you that securing your machine and network against malicious attack is a complex art. I do not consider myself an expert in this field at all and while the following mechanisms I describe will help, if you are serious about security then I recommend you do some research of your own into the subject. There are many good references on the Internet relating to the subject, including the Security-HOWTO

An important rule of thumb is: `**Don't run servers you don't intend to use**'. Many distributions come configured with all sorts of services configured and automatically started. To ensure even a minimum level of safety you should go through your `/etc/inetd.conf` file and comment out (*place a `#' at the start of the line*) any entries for services you don't intend to use. Good candidates are services such as: `shell`, `login`, `exec`, `uucp`, `ftp` and informational services such as `finger`, `netstat` and `systat`.

There are all sorts of security and access control mechanisms, I'll describe the most elementary of them.

/etc/ftpusers

The `/etc/ftpusers` file is a simple mechanism that allows you to deny certain users from logging into your machine via ftp. The `/etc/ftpusers` file is read by the ftp daemon program (*ftpd*) when an incoming ftp connection is received. The file is a simple list of those users who are disallowed from logging in. It might looks something like:

```
# /etc/ftpusers - users not allowed to login via ftp
root
uucp
bin
mail
```

/etc/securetty

The `/etc/securetty` file allows you to specify which `tty` devices `root` is allowed to login on. The `/etc/securetty` file is read by the login program (usually */bin/login*). Its format is a list of the tty devices names allowed, on all others `root` login is disallowed:

```
# /etc/securetty - tty's on which root is allowed to login
tty1
tty2
tty3
tty4
```

The *tcpd* hosts access control mechanism.

The *tcpd* program you will have seen listed in the same `/etc/inetd.conf` provides logging and access control mechanisms to services it is configured to protect.

When it is invoked by the *inetd* program it reads two files containing access rules and either allows or denies access to the server it is protecting accordingly.

It will search the rules files until the first match is found. If no match is found then it assumes that access should be allowed to anyone. The files it searches in sequence are: `/etc/hosts.allow`, `/etc/hosts.deny`. I'll describe each of these in turn. For a complete description of this facility you should refer to the appropriate *man* pages (`hosts_access(5)` is a good starting point).

/etc/hosts.allow

The `/etc/hosts.allow` file is a configuration file of the */usr/sbin/tcpd* program. The `hosts.allow` file contains rules describing which hosts are *allowed* access to a service on your machine.

The file format is quite simple:

```
# /etc/hosts.allow
#
# <service list>: <host list> [: command]
```

service list

is a comma delimited list of server names that this rule applies to. Example server names are: `ftpd`, `telnetd` and `fingerd`.

host list

is a comma delimited list of host names. You may also use IP addresses here. You may additionally specify hostnames or addresses using wildcard characters to match groups of hosts. Examples include: `gw.vk2ktj.ampr.org` to match a specific host, `.uts.edu.au` to match any hostname ending in that string, `44.` to match any IP address commencing with those digits. There are some special tokens to simplify configuration, some of these are: `ALL` matches every host, `LOCAL` matches any host whose name does not contain a `.' ie is in the same domain as your machine and `PARANOID` matches any host whose name does not match its address (name spoofing). There is one last token that is also useful. The `EXCEPT` token allows you to provide a list with exceptions. This will be covered in an example later.

command

is an optional parameter. This parameter is the full pathname of a command that would be executed everytime this rule is matched. It could for example run a command that would attempt to identify who is logged onto the connecting host, or to generate a mail message or some other warning to a system administrator that someone is attempting to connect. There are a number of expansions that may be included, some common examples are: `%h` expands to the name of the connecting host or address if it doesn't have a name, `%d` the daemon name being called.

An example:

```
# /etc/hosts.allow
#
# Allow mail to anyone
in.smtpd: ALL
# All telnet and ftp to only hosts within my domain and my host at home.
telnetd, ftpd: LOCAL, myhost.athome.org.au
# Allow finger to anyone but keep a record of who they are.
fingerd: ALL: (finger @%h | mail -s "finger from %h" root)
```

/etc/hosts.deny

The `/etc/hosts.deny` file is a configuration file of the */usr/sbin/tcpd* program. The `hosts.deny` file contains rules describing which hosts are *disallowed* access to a service on your machine.

A simple sample would look something like this:

```
# /etc/hosts.deny
#
# Disallow all hosts with suspect hostnames
ALL: PARANOID
#
# Disallow all hosts.
ALL: ALL
```

The `PARANOID` entry is really redundant because the other entry traps everything in any case. Either of these entry would make a reasonable default depending on your particular requirement.

Having an `ALL: ALL` default in the `/etc/hosts.deny` and then specifically enabling on those services and hosts that you want

in the `/etc/hosts.allow` file is the safest configuration.

/etc/hosts.equiv

The `hosts.equiv` file is used to grant certain hosts and users access rights to accounts on your machine without having to supply a password. This is useful in a secure environment where you control all machines, but is a security hazard otherwise. Your machine is only as secure as the least secure of the trusted hosts. To maximize security, don't use this mechanism and encourage your users not to use the `.rhosts` file as well.

Configure your *ftp* daemon properly.

Many sites will be interested in running an anonymous *ftp* server to allow other people to upload and download files without requiring a specific userid. If you decide to offer this facility make sure you configure the *ftp* daemon properly for anonymous access. Most *man* pages for `ftpd(8)` describe in some length how to go about this. You should always ensure that you follow these instructions. An important tip is to not use a copy of your `/etc/passwd` file in the anonymous account `/etc` directory, make sure you strip out all account details except those that you must have, otherwise you will be vulnerable to brute force password cracking techniques.

Network Firewalling.

Not allowing datagrams to even reach your machine or servers is an excellent means of security. This is covered in depth in the Firewall-HOWTO, and (more concisely) in a later section of this document.

Other suggestions.

Here are some other, potentially religious suggestions for you to consider.

sendmail

despite its popularity the *sendmail* daemon appears with frightening regularity on security warning announcements. Its up to you, but I choose not to run it.

NFS and other Sun RPC services

be wary of these. There are all sorts of possible exploits for these services. It is difficult finding an option to services like NFS, but if you configure them, make sure you are careful with who you allow mount rights to.

---

6. IP- and Ethernet-Related Information

This section covers information specific to Ethernet and IP. These subsections have been grouped together because I think they are the most interesting ones in the formerly-called ``Technology Specific'' Section. Anyone with a LAN should be able to benefit from these goodies.

6.1 Ethernet

Ethernet device names are `eth0', `eth1', `eth2' etc. The first card detected by the kernel is assigned `eth0' and the rest are assigned sequentially in the order they are detected.

By default, the Linux kernel only probes for one Ethernet device, you need to pass command line arguments to the kernel in order to force detection of furter boards.

To learn how to make your ethernet card(s) working under Linux you should refer to the Ethernet-HOWTO.

Once you have your kernel properly built to support your ethernet card then configuration of the card is easy.

Typically you would use something like (which most distributions already do for you, if you configured them to support your ethernet):

```
root# ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up
root# route add -net 192.168.0.0 netmask 255.255.255.0 eth0
```

Most of the ethernet drivers were developed by Donald Becker, `becker@CESDIS.gsfc.nasa.gov`.

6.2 EQL - multiple line traffic equaliser

The EQL device name is `eql'. With the standard kernel source you may have only one EQL device per machine. EQL provides a means of utilizing multiple point to point lines such as PPP, slip or plip as a single logical link to carry tcp/ip. Often it is cheaper to use multiple lower speed lines than to have one high speed line installed.

Kernel Compile Options:

```
Network device support  --->
    [*] Network device support
    <*> EQL (serial line load balancing) support
```

To support this mechanism the machine at the other end of the lines must also support EQL. Linux, Livingstone Portmasters and newer dial-in servers support compatible facilities.

To configure EQL you will need the eql tools which are available from: sunsite.unc.edu.

Configuration is fairly straightforward. You start by configuring the eql interface. The eql interface is just like any other network device. You configure the IP address and mtu using the *ifconfig* utility, so something like:

```
root# ifconfig eql 192.168.10.1 mtu 1006
```

Next you need to manually initiate each of the lines you will use. These may be any combination of point to point network devices. How you initiate the connections will depend on what sort of link they are, refer to the appropriate sections for further information.

Lastly you need to associate the serial link with the EQL device, this is called `enslaving' and is done with the *eql_enslave* command as shown:

```
root# eql_enslave eql sl0 28800
root# eql_enslave eql ppp0 14400
```

The `*estimated speed*' parameter you supply *eql_enslave* doesn't do anything directly. It is used by the EQL driver to determine what share of the datagrams that device should receive, so you can fine tune the balancing of the lines by playing with this value.

To disassociate a line from an EQL device you use the *eql_emancipate* command as shown:

```
root# eql_emancipate eql sl0
```

You add routing as you would for any other point to point link, except your routes should refer to the `eql` device rather than the actual serial devices themselves, typically you would use:

```
root# route add default eql
```

The EQL driver was developed by Simon Janes, `simon@ncm.com`.

6.3 IP Accounting (for Linux-2.0)

The IP accounting features of the Linux kernel allow you to collect and analyze some network usage data. The data collected comprises the number of packets and the number of bytes accumulated since the figures were last reset. You may specify a variety of rules to categorize the figures to suit whatever purpose you may have. This option has been removed in kernel 2.1.102, because the old ipfwadm-based firewalling was replaced by ``ipfwchains''.

Kernel Compile Options:

```
Networking options  --->
    [*] IP: accounting
```

After you have compiled and installed the kernel you need to use the *ipfwadm* command to configure IP accounting. There are many different ways of breaking down the accounting information that you might choose. I've picked a simple example of what might be useful to use, you should read the *ipfwadm* man page for more information.

Scenario: You have a ethernet network that is linked to the internet via a PPP link. On the ethernet you have a machine that offers a number of services and that you are interested in knowing how much traffic is generated by each of ftp and world wide web traffic, as well as total tcp and udp traffic.

You might use a command set that looks like the following, which is shown as a shell script:

```
#!/bin/sh
#
# Flush the accounting rules
```

TOC

```
ipfwadm -A -f
#
# Set shortcuts
localnet=44.136.8.96/29
any=0/0
# Add rules for local ethernet segment
ipfwadm -A in  -a -P tcp -D $localnet ftp-data
ipfwadm -A out -a -P tcp -S $localnet ftp-data
ipfwadm -A in  -a -P tcp -D $localnet www
ipfwadm -A out -a -P tcp -S $localnet www
ipfwadm -A in  -a -P tcp -D $localnet
ipfwadm -A out -a -P tcp -S $localnet
ipfwadm -A in  -a -P udp -D $localnet
ipfwadm -A out -a -P udp -S $localnet
#
# Rules for default
ipfwadm -A in  -a -P tcp -D $any ftp-data
ipfwadm -A out -a -P tcp -S $any ftp-data
ipfwadm -A in  -a -P tcp -D $any www
ipfwadm -A out -a -P tcp -S $any www
ipfwadm -A in  -a -P tcp -D $any
ipfwadm -A out -a -P tcp -S $any
ipfwadm -A in  -a -P udp -D $any
ipfwadm -A out -a -P udp -S $any
#
# List the rules
ipfwadm -A -l -n
#
```

The names ``ftp-data'' and ``www'' refer to lines in `/etc/services`. The last command lists each of the Accounting rules and displays the collected totals.

An important point to note when analyzing IP accounting is that **totals for all rules that match will be incremented** so that to obtain differential figures you need to perform appropriate maths. For example if I wanted to know how much data was not ftp nor www I would substract the individual totals from the rule that matches all ports.

```
root# ipfwadm -A -l -n
IP accounting rules
 pkts bytes dir prot source              destination         ports
    0     0 in  tcp  0.0.0.0/0           44.136.8.96/29      * -> 20
    0     0 out tcp  44.136.8.96/29      0.0.0.0/0           20 -> *
   10  1166 in  tcp  0.0.0.0/0           44.136.8.96/29      * -> 80
   10   572 out tcp  44.136.8.96/29      0.0.0.0/0           80 -> *
  252 10943 in  tcp  0.0.0.0/0           44.136.8.96/29      * -> *
  231 18831 out tcp  44.136.8.96/29      0.0.0.0/0            * -> *
    0     0 in  udp  0.0.0.0/0           44.136.8.96/29      * -> *
    0     0 out udp  44.136.8.96/29      0.0.0.0/0           * -> *
    0     0 in  tcp  0.0.0.0/0           0.0.0.0/0           * -> 20
    0     0 out tcp  0.0.0.0/0           0.0.0.0/0           20 -> *
   10  1166 in  tcp  0.0.0.0/0           0.0.0.0/0           * -> 80
   10   572 out tcp  0.0.0.0/0           0.0.0.0/0           80 -> *
  253 10983 in  tcp  0.0.0.0/0           0.0.0.0/0           * -> *
```

```
  231 18831 out tcp  0.0.0.0/0            0.0.0.0/0            * -> *
    0     0 in  udp  0.0.0.0/0            0.0.0.0/0            * -> *
    0     0 out udp  0.0.0.0/0            0.0.0.0/0            * -> *
```

TOC

6.4 IP Accounting (for Linux-2.2)

The new accounting code is accessed via ``IP Firewall Chains''. See the IP chanins home page for more information. Among other things, you'll now need to use *ipchains* instead of `ipfwadm` to configure your filters. (From `Documentation/Changes` in the latest kernel sources).

6.5 IP Aliasing

There are some applications where being able to configure multiple IP addresses to a single network device is useful. Internet Service Providers often use this facility to provide a `customized' to their World Wide Web and ftp offerings for their customers. You can refer to the ``IP-Alias mini-HOWTO'' for more information than you find here.

Kernel Compile Options:

```
Networking options  --->
    ....
    [*] Network aliasing
    ....
    <*> IP: aliasing support
```

After compiling and installing your kernel with IP_Alias support configuration is very simple. The aliases are added to virtual network devices associated with the actual network device. A simple naming convention applies to these devices being `<devname>:<virtual dev num>`, e.g. `eth0:0`, `ppp0:10` etc. Note that the the ifname:number device can only be configured *after* the main interface has been set up.

For example, assume you have an ethernet network that supports two different IP subnetworks simultaneously and you wish your machine to have direct access to both, you could use something like:

```
root# ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0

root# ifconfig eth0:0 192.168.10.1 netmask 255.255.255.0 up
root# route add -net 192.168.10.0 netmask 255.255.255.0 eth0:0
```

To delete an alias you simply add a `-' to the end of its name and refer to it and is as simple as:

```
root# ifconfig eth0:0- 0
```

All routes associated with that alias will also be deleted automatically.

6.6 IP Firewall (for Linux-2.0)

IP Firewall and Firewalling issues are covered in more depth in the Firewall-HOWTO. IP Firewalling allows you to secure your machine against unauthorized network access by filtering or allowing datagrams from or to IP addresses that

you nominate. There are three different classes of rules, incoming filtering, outgoing filtering and forwarding filtering. Incoming rules are applied to datagrams that are received by a network device. Outgoing rules are applied to datagrams that are to be transmitted by a network device. Forwarding rules are applied to datagrams that are received and are not for this machine, ie datagrams that would be routed.

Kernel Compile Options:

```
Networking options  --->
    [*] Network firewalls
    ....
    [*] IP: forwarding/gatewaying
    ....
    [*] IP: firewalling
    [ ] IP: firewall packet logging
```

Configuration of the IP firewall rules is performed using the *ipfwadm* command. As I mentioned earlier, security is not something I am expert at, so while I will present an example you can use, you should do your own research and develop your own rules if security is important to you.

Probably the most common use of IP firewall is when you are using your linux machine as a router and firewall gateway to protect your local network from unauthorized access from outside your network.

The following configuration is based on a contribution from Arnt Gulbrandsen, `<agulbra@troll.no>`.

The example describes the configuration of the firewall rules on the Linux firewall/router machine illustrated in this diagram:

```
-                                   -
 \                                  | 172.16.37.0
  \                                 |   /255.255.255.0
   \                  ---------     |
    |  172.16.174.30 | Linux |      |
NET ================|  f/w  |------|    ..37.19
    |    PPP         | router|      |  --------
   /                  ---------     |--| Mail |
  /                                 |  | /DNS |
 /                                  |  --------
-                                   -
```

The following commands would normally be placed in an `rc` file so that they were automatically started each time the system boots. For maximum security they would be performed after the network interfaces are configured, but before the interfaces are actually brought up to prevent anyone gaining access while the firewall machine is rebooting.

```
#!/bin/sh

# Flush the 'Forwarding' rules table
# Change the default policy to 'accept'
#
/sbin/ipfwadm -F -f
/sbin/ipfwadm -F -p accept
#
# .. and for 'Incoming'
#
```

TOC

```
/sbin/ipfwadm -I -f
/sbin/ipfwadm -I -p accept

# First off, seal off the PPP interface
# I'd love to use '-a deny' instead of '-a reject -y' but then it
# would be impossible to originate connections on that interface too.
# The -o causes all rejected datagrams to be logged. This trades
# disk space against knowledge of an attack of configuration error.
#
/sbin/ipfwadm -I -a reject -y -o -P tcp -S 0/0 -D 172.16.174.30

# Throw away certain kinds of obviously forged packets right away:
# Nothing should come from multicast/anycast/broadcast addresses
#
/sbin/ipfwadm -F -a deny -o -S 224.0/3 -D 172.16.37.0/24
#
# and nothing coming from the loopback network should ever be
# seen on a wire
#
/sbin/ipfwadm -F -a deny -o -S 127.0/8 -D 172.16.37.0/24

# accept incoming SMTP and DNS connections, but only
# to the Mail/Name Server
#
/sbin/ipfwadm -F -a accept -P tcp -S 0/0 -D 172.16.37.19 25 53
#
# DNS uses UDP as well as TCP, so allow that too
# for questions to our name server
#
/sbin/ipfwadm -F -a accept -P udp -S 0/0 -D 172.16.37.19 53
#
# but not "answers" coming to dangerous ports like NFS and
# Larry McVoy's NFS extension.  If you run squid, add its port here.
#
/sbin/ipfwadm -F -a deny -o -P udp -S 0/0 53 \
        -D 172.16.37.0/24 2049 2050

# answers to other user ports are okay
#
/sbin/ipfwadm -F -a accept -P udp -S 0/0 53 \
        -D 172.16.37.0/24 53 1024:65535

# Reject incoming connections to identd
# We use 'reject' here so that the connecting host is told
# straight away not to bother continuing, otherwise we'd experience
# delays while ident timed out.
#
/sbin/ipfwadm -F -a reject -o -P tcp -S 0/0 -D 172.16.37.0/24 113

# Accept some common service connections from the 192.168.64 and
# 192.168.65 networks, they are friends that we trust.
#
/sbin/ipfwadm -F -a accept -P tcp -S 192.168.64.0/23 \
```

```
            -D 172.16.37.0/24 20:23

# accept and pass through anything originating inside
#
/sbin/ipfwadm -F -a accept -P tcp -S 172.16.37.0/24 -D 0/0

# deny most other incoming TCP connections and log them
# (append 1:1023 if you have problems with ftp not working)
#
/sbin/ipfwadm -F -a deny -o -y -P tcp -S 0/0 -D 172.16.37.0/24

# ... for UDP too
#
/sbin/ipfwadm -F -a deny -o -P udp -S 0/0 -D 172.16.37.0/24
```

Good firewall configurations are a little tricky. This example should be a reasonable starting point for you. The *ipfwadm* manual page offers some assistance in how to use the tool. If you intend to configure a firewall, be sure to ask around and get as much advice from sources you consider reliable and get someone to test/sanity check your configuration from the outside.

6.7 IP Firewall (for Linux-2.2)

The new firewalling code is accessed via ``IP Firewall Chains''. See the IP chanins home page for more information. Among other things, you'll now need to use *ipchains* instead of `ipfwadm` to configure your filters. (From `Documentation/Changes` in the latest kernel sources).

6.8 IPIP Encapsulation

Why would you want to encapsulate IP datagrams within IP datagrams? It must seem an odd thing to do if you've never seen an application of it before. Ok, here are a couple of common places where it is used: Mobile-IP and IP-Multicast. What is perhaps the most widely spread use of it though is also the least well known, Amateur Radio.

Kernel Compile Options:

```
Networking options  --->
    [*] TCP/IP networking
    [*] IP: forwarding/gatewaying
    ....
    <*> IP: tunneling
```

IP tunnel devices are called `tunl0', `tunl1' etc.

"But why ?". Ok, ok. Conventional IP routing rules mandate that an IP network comprises a network address and a network mask. This produces a series of contiguous addresses that may all be routed via a single routing entry. This is very convenient, but it means that you may only use any particular IP address while you are connected to the particular piece of network to which it belongs. In most instances this is ok, but if you are a mobile netizen then you may not be able to stay connected to the one place all the time. IP/IP encapsulation (IP tunneling) allows you to overcome this restriction by allowing datagrams destined for your IP address to be wrapped up and redirected to another IP address. If you know that you're going to be operating from some other IP network for some time you can set up a machine on your home network to accept datagrams to your IP address and redirect them to the address that you will actually be using

temporarily.

A tunneled network configuration.

As always, I believe a diagram will save me lots of confusing text, so here is one:

```
192.168.1/24                              192.168.2/24

     -                                          -
     |      ppp0 =            ppp0 =            |
     |  aaa.bbb.ccc.ddd  fff.ggg.hhh.iii        |
     |                                          |
     |   /-----\                  /-----\       |
     |   |     |       //         |     |       |
     |---|  A  |------//---------|  B  |---|
     |   |     |     //           |     |       |
     |   \-----/                  \-----/       |
     |                                          |
     -                                          -
```

The diagram illustrates another possible reason to use IPIP encapsulation, virtual private networking. This example presupposes that you have two machines each with a simple dial up internet connection. Each host is allocated just a single IP address. Behind each of these machines are some private local area networks configured with reserved IP network addresses. Suppose that you want to allow any host on network A to connect to any host on network B, just as if they were properly connected to the Internet with a network route. IPIP encapsulation will allow you to do this. Note, encapsulation does not solve the problem of how you get the hosts on networks A and B to talk to any other on the Internet, you still need tricks like IP Masquerade for that. Encapsulation is normally performed by machine functioning as routers.

Linux router `A' would be configured with a script like the following:

```
#!/bin/sh
PATH=/sbin:/usr/sbin
mask=255.255.255.0
remotegw=fff.ggg.hhh.iii
#
# Ethernet configuration
ifconfig eth0 192.168.1.1 netmask $mask up
route add -net 192.168.1.0 netmask $mask eth0
#
# ppp0 configuration (start ppp link, set default route)
pppd
route add default ppp0
#
# Tunnel device configuration
ifconfig tunl0 192.168.1.1 up
route add -net 192.168.2.0 netmask $mask gw $remotegw tunl0
```

Linux router `B' would be configured with a similar script:

```
#!/bin/sh
PATH=/sbin:/usr/sbin
```

```
mask=255.255.255.0
remotegw=aaa.bbb.ccc.ddd
#
# Ethernet configuration
ifconfig eth0 192.168.2.1 netmask $mask up
route add -net 192.168.2.0 netmask $mask eth0
#
# ppp0 configuration (start ppp link, set default route)
pppd
route add default ppp0
#
# Tunnel device configuration
ifconfig tunl0 192.168.2.1 up
route add -net 192.168.1.0 netmask $mask gw $remotegw tunl0
```

The command:

```
route add -net 192.168.1.0 netmask $mask gw $remotegw tunl0
```

reads: `Send any datagrams destined for `192.168.1.0/24` inside an IPIP encap datagram with a destination address of `aaa.bbb.ccc.ddd'.

Note that the configurations are reciprocated at either end. The tunnel device uses the `gw' in the route as the *destination* of the IP datagram in which it will place the datagram it has received to route. That machine must know how to decapsulate IPIP datagrams, that is, it must also be configured with a tunnel device.

A tunneled host configuration.

It doesn't have to be a whole network you route. You could for example route just a single IP address. In that instance you might configure the `tunl` device on the `remote' machine with its home IP address and at the A end just use a host route (and Proxy Arp) rather than a network route via the tunnel device. Let's redraw and modify our configuration appropriately. Now we have just host `B' which to want to act and behave as if it is both fully connected to the Internet and also part of the remote network supported by host `A':

```
192.168.1/24

    -
    |      ppp0 =                ppp0 =
    |  aaa.bbb.ccc.ddd      fff.ggg.hhh.iii
    |
    |   /-----\                 /-----\
    |   |     |       //        |     |
    |---|  A  |------//---------|  B  |
    |   |     |     //          |     |
    |   \-----/                 \-----/
    |                      also: 192.168.1.12
    -
```

Linux router `A' would be configured with:

```
#!/bin/sh
PATH=/sbin:/usr/sbin
```

```
mask=255.255.255.0
remotegw=fff.ggg.hhh.iii
#
# Ethernet configuration
ifconfig eth0 192.168.1.1 netmask $mask up
route add -net 192.168.1.0 netmask $mask eth0
#
# ppp0 configuration (start ppp link, set default route)
pppd
route add default ppp0
#
# Tunnel device configuration
ifconfig tunl0 192.168.1.1 up
route add -host 192.168.1.12 gw $remotegw tunl0
#
# Proxy ARP for the remote host
arp -s 192.168.1.12 xx:xx:xx:xx:xx:xx pub
```

Linux host `B' would be configured with:

```
#!/bin/sh
PATH=/sbin:/usr/sbin
mask=255.255.255.0
remotegw=aaa.bbb.ccc.ddd
#
# ppp0 configuration (start ppp link, set default route)
pppd
route add default ppp0
#
# Tunnel device configuration
ifconfig tunl0 192.168.1.12 up
route add -net 192.168.1.0 netmask $mask gw $remotegwtunl0
```

This sort of configuration is more typical of a Mobile-IP application. Where a single host wants to roam around the Internet and maintain a single usable IP address the whole time. You should refer to the Mobile-IP section for more information on how that is handled in practice.

6.9 IP Masquerade (for Linux-2.0)

Many people have a simple dialup account to connect to the Internet. Nearly everybody using this sort of configuration is allocated a single IP address by the Internet Service Provider. This is normally enough to allow only one host full access to the network. IP Masquerade is a clever trick that enables you to have many machines make use of that one IP address, by causing the other hosts to look like, hence the term masquerade, the machine supporting the dialup connection. There is a small caveat and that is that the masquerade function nearly always works only in one direction, that is the masqueraded hosts can make calls out, but they cannot accept or receive network connections from remote hosts. This means that some network services do not work such as *talk* and others such as *ftp* must be configured to operate in passive (PASV) mode to operate. Fortunately the most common network services such as *telnet*, World Wide Web and *irc* do work just fine.

Kernel Compile Options:

```
Code maturity level options  --->
    [*] Prompt for development and/or incomplete code/drivers
Networking options  --->
    [*] Network firewalls
    ....
    [*] TCP/IP networking
    [*] IP: forwarding/gatewaying
    ....
    [*] IP: masquerading (EXPERIMENTAL)
```

Normally you have your linux machine supporting a slip or PPP dialup line just as it would if it were a standalone machine. Additionally it would have another network device configured, perhaps an ethernet, configured with one of the reserved network addresses. The hosts to be masqueraded would be on this second network. Each of these hosts would have the IP address of the ethernet port of the linux machine set as their default gateway or router.

A typical configuration might look something like this:

```
-                                   -
 \                                  | 192.168.1.0
  \                                 |   /255.255.255.0
   \                 ---------      |
    |                | Linux | .1.1 |
NET ================| masq  |------|
    |    PPP/slip    | router|      |  --------
   /                 ---------      |--| host |
  /                                  |  |      |
 /                                   |  --------
-                                   -
```

The most relevant commands for this configuration are:

```
# Network route for ethernet
route add -net 192.168.1.0 netmask 255.255.255.0 eth0
#
# Default route to the rest of the internet.
route add default ppp0
#
# Cause all hosts on the 192.168.1/24 network to be masqueraded.
ipfwadm -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0
```

If you are minimalist and a lazy typist, like me, and your masquerading host has only two interfaces (so that every packet being forwarded must be masqueraded), the following command will suffice:

```
root# /sbin/ipfwadm -F -a accept -m
```

You can get more information on the Linux IP Masquerade feature from the IP Masquerade Resource Page. Also, a *very* detailed document about masquesrading is the ``IP-Masquerade mini-HOWTO'' (which also intructs to configure other OS's to run with a Linux masquerade server).

6.10 IP Transparent Proxy

IP transparent proxy is a feature that enables you to redirect servers or services destined for another machine to those services on this machine. Typically this would be useful where you have a linux machine as a router and also provides a proxy server. You would redirect all connections destined for that service remotely to the local proxy server.

Kernel Compile Options:

```
Code maturity level options  --->
        [*] Prompt for development and/or incomplete code/drivers
Networking options  --->
        [*] Network firewalls
        ....
        [*] TCP/IP networking
        ....
        [*] IP: firewalling
        ....
        [*] IP: transparent proxy support (EXPERIMENTAL)
```

Configuration of the transparent proxy feature is performed using the *ipfwadm* command

An example that might be useful is as follows:

```
root# ipfwadm -I -a accept -D 0/0 telnet -r 2323
```

This example will cause any connection attempts to port `telnet` (23) on any host to be redirected to port 2323 on this host. If you run a service on that port, you could forward telnet connections, log them or do whatever fits your need.

A more interesting example is redirecting all `http` traffic through a local cache. However, the protocol used by proxy servers is different from native http: where a client connects to `www.server.com:80` and asks for `/path/page`, when it connects to the local cache it contacts `proxy.local.domain:8080` and asks for `www.server.com/path/page`.

To filter an `http` request through the local proxy, you need to adapt the protocol by inserting a small server, called `transproxy` (you can find it on the world wide web). You can choose to run `transproxy` on port 8081, and issue this command:

```
root# ipfwadm -I -a accept -D 0/0 80 -r 8081
```

The `transproxy` program, then, will receive all connections meant to reach external servers and will pass them to the local proxy after fixing protocol differences.

6.11 IPv6

Just when you thought you were beginning to understand IP networking the rules get changed! IPv6 is the shorthand notation for version 6 of the Internet Protocol. IPv6 was developed primarily to overcome the concerns in the Internet community that there would soon be a shortage of IP addresses to allocate. IPv6 addresses are 16 bytes long (128 bits). IPv6 incorporates a number of other changes, mostly simplifications, that will make IPv6 networks more managable than IPv4 networks.

Linux already has a working, but not complete, IPv6 implementation in the `2.1.*` series kernels.

TOC

If you wish to experiment with this next generation Internet technology, or have a requirement for it, then you should read the IPv6-FAQ which is available from www.terra.net.

6.12 Mobile IP

The term "IP mobility" describes the ability of a host that is able to move its network connection from one point on the Internet to another without changing its IP address or losing connectivity. Usually when an IP host changes its point of connectivity it must also change its IP address. IP Mobility overcomes this problem by allocating a fixed IP address to the mobile host and using IP encapsulation (tunneling) with automatic routing to ensure that datagrams destined for it are routed to the actual IP address it is currently using.

A project is underway to provide a complete set of IP mobility tools for Linux. The Status of the project and tools may be obtained from the: Linux Mobile IP Home Page.

6.13 Multicast

IP Multicast allows an arbitrary number of IP hosts on disparate IP networks to have IP datagrams simultaneously routed to them. This mechanism is exploited to provide Internet wide "broadcast" material such as audio and video transmissions and other novel applications.

Kernel Compile Options:

```
Networking options  --->
        [*] TCP/IP networking
        ....
        [*] IP: multicasting
```

A suite of tools and some minor network configuration is required. Please check the Multicast-HOWTO for more information on Multicast support in Linux.

6.14 NAT - Network Address Translation

The IP Network Address Translation facility is pretty much the standardized big brother of the Linux IP Masquerade facility. It is specified in some detail in RFC-1631 at your nearest RFC archive. NAT provides features that IP-Masquerade does not that make it eminently more suitable for use in corporate firewall router designs and larger scale installations.

An alpha implementation of NAT for Linux 2.0.29 kernel has been developed by Michael.Hasenstein, `Michael.Hasenstein@informatik.tu-chemnitz.de`. Michaels documentation and implementation are available from: Linux IP Network Address Web Page

Newer Linux 2.1.* kernels also include some NAT functionality in the routing algorithm.

6.15 Traffic Shaper - Changing allowed bandwidth

The traffic shaper is a driver that creates new interface devices, those devices are traffic-limited in a user-defined way, they rely on physical network devices for actual transmission and can be used as outgoing routed for network traffic.

The shaper was introduced in Linux-2.1.15 and was backported to Linux-2.0.36 (it appeared in `2.0.36-pre-patch-2` distributed by Alan Cox, the author of the shaper device and maintainer of Linux-2.0).

The traffic shaper can only be compiled as a module and is configured by the *shapecfg* program with commands like following:

```
shapecfg attach shaper0 eth1
shapecfg speed shaper0 64000
```

The shaper device can only control the bandwidth of outgoing traffic, as packets are transmitted via the shaper only according to the routing tables; therefore, a ``route by source address'' functionality could help in limiting the overall bandwidth of specific hosts using a Linux router.

Linux-2.1 already has support for such routing, if you need it for Linux-2.0 please check the patch by Mike McLagan, at `ftp.invlogic.com`. Refer to `Documentation`networking/shaper.txt for further information about the shaper.

If you want to try out a (tentative) shaping for incoming packets, try out `rshaper-1.01` (or newer), from ftp.systemy.it.

6.16 Routing in Linux-2.2

The latest versions of Linux-2.1 offer a lot of flexibility in routing policy. Unfortunately, you have to wait for the next version of this howto, or go read the kernel sources.

---

7. Using common PC hardware

7.1 ISDN

The Integrated Services Digital Network (ISDN) is a series of standards that specify a general purpose switched digital data network. An ISDN `call' creates a synchronous point to point data service to the destination. ISDN is generally delivered on a high speed link that is broken down into a number of discrete channels. There are two different types of channels, the `B Channels' which will actually carry the user data and a single channel called the `D channel' which is used to send control information to the ISDN exchange to establish calls and other functions. In Australia for example, ISDN may be delivered on a 2Mbps link that is broken into 30 discrete 64kbps B channels with one 64kbps D channel. Any number of channels may be used at a time and in any combination. You could for example establish 30 separate calls to 30 different destinations at 64kbps each, or you could establish 15 calls to 15 different destinations at 128kbps each (two channels used per call), or just a small number of calls and leave the rest idle. A channel may be used for either incoming or outgoing calls. The original intention of ISDN was to allow Telecommunications companies to provide a single data service which could deliver either telephone (via digitised voice) or data services to your home or business without requiring you to make any special configuration changes.

There are a few different ways to connect your computer to an ISDN service. One way is to use a device called a `Terminal Adaptor' which plugs into the Network Terminating Unit that you telecommunications carrier will have installed when you got your ISDN service and presents a number of serial interfaces. One of those interfaces is used to enter commands to establish calls and configuration and the others are actually connected to the network devices that will use the data circuits when they are established. Linux will work in this sort of configuration without modification, you just treat the port on the Terminal Adaptor like you would treat any other serial device. Another way, which is the way the kernel ISDN support is designed for allows you to install an ISDN card into your Linux machine and then has your Linux software handle the protocols and make the calls itself.

Kernel Compile Options:

```
ISDN subsystem  --->
        <*> ISDN support
        [ ] Support synchronous PPP
        [ ] Support audio via ISDN
        < > ICN 2B and 4B support
        < > PCBIT-D support
        < > Teles/NICCY1016PC/Creatix support
```

The Linux implementation of ISDN supports a number of different types of internal ISDN cards. These are those listed in the kernel configuration options:

- ICN 2B and 4B
- Octal PCBIT-D
- Teles ISDN-cards and compatibles

Some of these cards require software to be downloaded to them to make them operational. There is a separate utility to do this with.

TOC

Full details on how to configure the Linux ISDN support is available from the `/usr/src/linux/Documentation/isdn/` directory and an FAQ dedicated to *isdn4linux* is available at www.lrz-muenchen.de. (You can click on the english flag to get an english version).

A note about PPP. The PPP suite of protocols will operate over either asynchronous or synchronous serial lines. The commonly distributed PPP daemon for Linux `*pppd*' supports only asynchronous mode. If you wish to run the PPP protocols over your ISDN service you need a specially modified version. Details of where to find it are available in the documentation referred to above.

7.2 PLIP for Linux-2.0

PLIP device names are `plip0', `plip1 and plip2.

Kernel Compile Options:

```
Network device support  --->
    <*> PLIP (parallel port) support
```

plip (Parallel Line IP), is like SLIP, in that it is used for providing a *point to point* network connection between two machines, except that it is designed to use the parallel printer ports on your machine instead of the serial ports (a cabling diagram in included in the cabling diagram section later in this document). Because it is possible to transfer more than one bit at a time with a parallel port, it is possible to attain higher speeds with the *plip* interface than with a standard serial device. In addition, even the simplest of parallel ports, printer ports, can be used in lieu of you having to purchase comparatively expensive 16550AFN UART's for your serial ports. PLIP uses a lot of CPU compared to a serial link and is most certainly not a good option if you can obtain some cheap ethernet cards, but it will work when nothing else is available and will work quite well. You should expect a data transfer rate of about 20 kilobytes per second when a link is running well.

The PLIP device drivers competes with the parallel device driver for the parallel port hardware. If you wish to use both drivers then you should compile them both as modules to ensure that you are able to select which port you want to use for PLIP and which ports you want for the printer driver. Refer to the ``Mudules mini-HOWTO'' for more information on kernel module configuration.

Please note that some laptops use chipsets that will not work with PLIP because they do not allow some combinations of signals that PLIP relies on, that printers don't use.

The Linux *plip* interface is compatible with the *Crynwyr Packet Driver PLIP* and this will mean that you can connect your Linux machine to a DOS machine running any other sort of tcp/ip software via *plip*.

In the 2.0.* series kernel the plip devices are mapped to i/o port and IRQ as follows:

```
device  i/o     IRQ
------  -----   ---
plip0   0x3bc   5
plip1   0x378   7
plip2   0x278   2
```

If your parallel ports don't match any of the above combinations then you can change the IRQ of a port using the *ifconfig* command using the `irq' parameter (be sure to enable IRQ's on your printer ports in your ROM BIOS if it

supports this option). As an alternative, you can specify ``io=" annd ``irq=" options on the *insmod* command line, if you use modules. For example:

```
root# insmod plip.o io=0x288 irq=5
```

PLIP operation is controlled by two timeouts, whose default values are probably ok in most cases. You will probably need to increase them if you have an especially slow computer, in which case the timers to increase are actually on the **other** computer. A program called *plipconfig* exists that allows you to change these timer settings without recompiling your kernel. It is supplied with many Linux distributions.

To configure a *plip* interface, you will need to invoke the following commands (or **add** them to your initialization scripts):

```
root# /sbin/ifconfig plip1 localplip pointopoint remoteplip
root# /sbin/route add remoteplip plip1
```

Here, the port being used is the one at I/O address 0x378; *localplip* amd *remoteplip* are the names or IP addresses used over the PLIP cable. I personally keep them in my `/etc/hosts` database:

```
# plip entries
192.168.3.1   localplip
192.168.3.2   remoteplip
```

The *pointopoint* parameter has the same meaning as for SLIP, in that it specifies the address of the machine at the other end of the link.

In almost all respects you can treat a *plip* interface as though it were a *SLIP* interface, except that neither *dip* nor *slattach* need be, nor can be, used.

Further information on PLIP may be obtained from the ``PLIP mini-HOWTO".

7.3 PLIP for Linux-2.2

During development of the 2.1 kernel versions, support for the parallel port was changed to a better setup.

Kernel Compile Options:

```
General setup  --->
    [*] Parallel port support
Network device support  --->
    <*> PLIP (parallel port) support
```

The new code for PLIP behaves like the old one (use the same *ifconfig* and *route* commands as in the previous section, but initialization of the device is different due to the advanced parallel port support.

The ``first" PLIP device is always called ``plip0", where first is the first device detected by the system, similarly to what happens for Ethernet devices. The actual parallel port being used is one of the available ports, as shown in `/proc/parport`. For example, if you have only one parallel port, you'll only have a directory called

`/proc/parport/0`.

If your kernel didn't detect the IRQ number used by your port, ``insmod plip'' will fail; in this case just write the right number to `/proc/parport/0/irq` and reinvoke *insmod*.

Complete information about parallel port management is available in the file `Documentation/parport.txt`, part of your kernel sources.

7.4 PPP

PPP devices names are `ppp0', `ppp1, etc. Devices are numbered sequentially with the first device configured receiving `0'.

Kernel Compile Options:

```
Networking options  --->
    <*> PPP (point-to-point) support
```

PPP configuration is covered in detail in the PPP-HOWTO.

Maintaining a permanent connection to the net with *pppd*.

If you are fortunate enough to have a semi permanent connection to the net and would like to have your machine automatically redial your PPP connection if it is lost then here is a simple trick to do so.

Configure PPP such that it can be started by the `root` user by issuing the command:

```
# pppd
```

Be sure that you have the `-detach' option configured in your `/etc/ppp/options` file. Then, insert the following line into your `/etc/inittab` file, down with the *getty* definitions:

```
pd:23:respawn:/usr/sbin/pppd
```

This will cause the *init* program to spawn and monitor the *pppd* program and automatically restart it if it dies.

7.5 SLIP client

SLIP devices are named `sl0', `sl1' etc. with the first device configured being assigned `0' and the rest incrementing sequentially as they are configured.

Kernel Compile Options:

```
Network device support  --->
    [*] Network device support
    <*> SLIP (serial line) support
    [ ]  CSLIP compressed headers
    [ ]  Keepalive and linefill
    [ ]  Six bit SLIP encapsulation
```

TOC

SLIP (Serial Line Internet Protocol) allows you to use tcp/ip over a serial line, be that a phone line with a dial u modem, or a leased line of some sort. Of course to use SLIP you need access to a *SLIP-server* in your area. M universities and businesses provide SLIP access all over the world.

Slip uses the serial ports on your machine to carry IP datagrams. To do this it must take control of the serial device. Slip device names are named *sl0*, *sl1* etc. How do these correspond to your serial devices ? The networking code uses what is called an *ioctl* (i/o control) call to change the serial devices into SLIP devices. There are two programs supplied that can do this, they are called *dip* and *slattach*

dip

dip (Dialup IP) is a smart program that is able to set the speed of the serial device, command your modem to dial the remote end of the link, automatically log you into the remote server, search for messages sent to you by the server and extract information for them such as your IP address and perform the *ioctl* necessary to switch your serial port into SLIP mode. *dip* has a powerful scripting ability and it is this that you can exploit to automate your logon procedure.

You can find it at: sunsite.unc.edu.

To install it, try the following:

```
user% tar xvzf dip337o-uri.tgz
user% cd dip-3.3.7o
user% vi Makefile
root# make install
```

The `Makefile` assumes the existence of a group called *uucp*, but you might like to change this to either *dip* or *SLIP* depending on your configuration.

slattach

slattach as contrasted with *dip* is a very simple program, that is very easy to use, but does not have the sophistication of *dip*. It does not have the scripting ability, all it does is configure your serial device as a SLIP device. It assumes you have all the information you need and the serial line is established before you invoke it. *slattach* is ideal to use where you have a permanent connection to your server, such as a physical cable, or a leased line.

When do I use which ?

You would use *dip* when your link to the machine that is your SLIP server is a dialup modem, or some other temporary link. You would use *slattach* when you have a leased line, perhaps a cable, between your machine and the server and there is no special action needed to get the link working. See section `Permanent Slip connection' for more information.

Configuring SLIP is much like configuring an Ethernet interface (read section `Configuring an ethernet device' above). However there are a few key differences.

First of all, SLIP links are unlike ethernet networks in that there is only ever two hosts on the network, one at each end of the link. Unlike an ethernet that is available for use as soon are you are cabled, with SLIP, depending on the type of link you have, you may have to initialize your network connection in some special way.

TOC

If you are using *dip* then this would not normally be done at boot time, but at some time later, when you were to use the link. It is possible to automate this procedure. If you are using *slattach* then you will probably want add a section to your *rc.inet1* file. This will be described soon.

There are two major types of SLIP servers: Dynamic IP address servers and static IP address servers. Almost every SLIP server will prompt you to login using a username and password when dialing in. *dip* can handle logging you in automatically.

Static SLIP server with a dialup line and DIP.

A static SLIP server is one in which you have been supplied an IP address that is exclusively yours. Each time you connect to the server, you will configure your SLIP port with that address. The static SLIP server will answer your modem call, possibly prompt you for a username and password, and then route any datagrams destined for your address to you via that connection. If you have a static server, then you may want to put entries for your hostname and IP address (since you know what it will be) into your `/etc/hosts`. You should also configure some other files such as: `rc.inet2`, `host.conf`, `resolv.conf`, `/etc/HOSTNAME` and `rc.local`. Remember that when configuring `rc.inet1`, you don't need to add any special commands for your SLIP connection since it is *dip* that does all of the hard work for you in configuring your interface. You will need to give *dip* the appropriate information and it will configure the interface for you after commanding the modem to establish the call and logging you into your SLIP server.

If this is how your SLIP server works then you can move to section `Using Dip' to learn how to configure *dip* appropriately.

Dynamic SLIP server with a dialup line and DIP.

A *dynamic* SLIP server is one which allocates you an IP address randomly, from a pool of addresses, each time you logon. This means that there is no guarantee that you will have any particular address each time, and that address may well be used by someone else after you have logged off. The network administrator who configured the SLIP server will have assigned a pool of address for the SLIP server to use, when the server receives a new incoming call, it finds the first unused address, guides the caller through the login process and then prints a welcome message that contains the IP address it has allocated and will proceed to use that IP address for the duration of that call.

Configuring for this type of server is similar to configuring for a static server, except that you must add a step where you obtain the IP address that the server has allocated for you and configure your SLIP device with that.

Again, *dip* does the hard work and new versions are smart enough to not only log you in, but to also be able to automatically read the IP address printed in the welcome message and store it so that you can have it configure your SLIP device with it.

If this is how your SLIP server works then you can move to section `Using Dip' to learn how to configure *dip* appropriately.

Using DIP.

As explained earlier, *dip* is a powerful program that can simplify and automate the process of dialing into the SLIP server, logging you in, starting the connection and configuring your SLIP devices with the appropriate *ifconfig* and *route* commands.

Essentially to use *dip* you'll write a `dip script', which is basically a list of commands that *dip* understands that tell *dip* how to perform each of the actions you want it to perform. See `sample.dip` that comes supplied with *dip* to get an idea of how it works. *dip* is quite a powerful program, with many options. Instead of going into all of them

here you should look at the *man* page, README and sample files that will have come with your version of *dip*.

You may notice that the `sample.dip` script assumes that you're using a static SLIP server, so you know what your IP address is beforehand. For dynamic SLIP servers, the newer versions of *dip* include a command you can use to automatically read and configure your SLIP device with the IP address that the dynamic server allocates for you. The following sample is a modified version of the `sample.dip` that came supplied with *dip337j-uri.tgz* and is probably a good starting point for you. You might like to save it as `/etc/dipscript` and edit it to suit your configuration:

```
#
# sample.dip    Dialup IP connection support program.
#
#               This file (should show) shows how to use the DIP
#       This file should work for Annex type dynamic servers, if you
#       use a static address server then use the sample.dip file that
#       comes as part of the dip337-uri.tgz package.
#
#
# Version:      @(#)sample.dip  1.40    07/20/93
#
# Author:       Fred N. van Kempen, <waltje@uWalt.NL.Mugnet.ORG>
#

main:
# Next, set up the other side's name and address.
# My dialin machine is called 'xs4all.hacktic.nl' (== 193.78.33.42)
get $remote xs4all.hacktic.nl
# Set netmask on sl0 to 255.255.255.0
netmask 255.255.255.0
# Set the desired serial port and speed.
port cua02
speed 38400

# Reset the modem and terminal line.
# This seems to cause trouble for some people!
reset

# Note! "Standard" pre-defined "errlevel" values:
#  0 - OK
#  1 - CONNECT
#  2 - ERROR
#
# You can change those grep'ping for "addchat()" in *.c...

# Prepare for dialing.
send ATQ0V1E1X4\r
wait OK 2
if $errlvl != 0 goto modem_trouble
dial 555-1234567
if $errlvl != 1 goto modem_trouble
```

TOC

```
# We are connected.  Login to the system.
login:
sleep 2
wait ogin: 20
if $errlvl != 0 goto login_trouble
send MYLOGIN\n
wait ord: 20
if $errlvl != 0 goto password_error
send MYPASSWD\n
loggedin:

# We are now logged in.
wait SOMEPROMPT 30
if $errlvl != 0 goto prompt_error

# Command the server into SLIP mode
send SLIP\n
wait SLIP 30
if $errlvl != 0 goto prompt_error

# Get and Set your IP address from the server.
#   Here we assume that after commanding the SLIP server into SLIP
#   mode that it prints your IP address
get $locip remote 30
if $errlvl != 0 goto prompt_error

# Set up the SLIP operating parameters.
get $mtu 296
# Ensure "route add -net default xs4all.hacktic.nl" will be done
default

# Say hello and fire up!
done:
print CONNECTED $locip ---> $rmtip
mode CSLIP
goto exit

prompt_error:
print TIME-OUT waiting for sliplogin to fire up...
goto error

login_trouble:
print Trouble waiting for the Login: prompt...
goto error

password:error:
print Trouble waiting for the Password: prompt...
goto error
```

```
modem_trouble:
print Trouble occurred with the modem...
error:
print CONNECT FAILED to $remote
quit

exit:
exit
```

The above example assumes you are calling a *dynamic* SLIP server, if you are calling a *static* SLIP server, then the `sample.dip` file that comes with *dip337j-uri.tgz* should work for you.

When *dip* is given the *get $local* command it searches the incoming text from the remote end for a string that looks like an IP address, ie strings numbers separated by `.' characters. This modification was put in place specifically for *dynamic* SLIP servers, so that the process of reading the IP address granted by the server could be automated.

The example above will automatically create a default route via your SLIP link, if this is not what you want, you might have an ethernet connection that should be your default route, then remove the *default* command from the script. After this script has finished running, if you do an *ifconfig* command, you will see that you have a device *sl0*. This is your SLIP device. Should you need to, you can modify its configuration manually, after the *dip* command has finished, using the *ifconfig* and *route* commands. Please note that *dip* allows you to select a number of different protocols to use with the `mode` command, the most common example is *cSLIP* for SLIP with compression. Please note that both ends of the link must agree, so you should ensure that whatever you select agrees with what your server is set to. The above example is fairly robust and should cope with most errors. Please refer to the *dip* man page for more information. Naturally you could, for example, code the script to do such things as redial the server if it doesn't get a connection within a prescribed period of time, or even try a series of servers if you have access to more than one.

Permanent SLIP connection using a leased line and slattach.

If you have a cable between two machines, or are fortunate enough to have a leased line, or some other permanent serial connection between your machine and another, then you don't need to go to all the trouble of using *dip* to set up your serial link. *slattach* is a very simple to use utility that will allow you just enough functionality to configure your connection. Since your connection will be a permanent one, you will want to add some commands to your `rc.inet1` file. In essence all you need to do for a permanent connection is ensure that you configure the serial device to the correct speed and switch the serial device into SLIP mode. *slattach* allows you to do this with one command. **Add** the following to your `rc.inet1` file:

```
#
# Attach a leased line static SLIP connection
#
#  configure /dev/cua0 for 19.2kbps and cslip
/sbin/slattach -p cslip -s 19200 /dev/cua0 &
/sbin/ifconfig sl0 IPA.IPA.IPA.IPA pointopoint IPR.IPR.IPR.IPR up
#
# End static SLIP.
```

Where:

IPA.IPA.IPA.IPA

represents your IP address.

IPR.IPR.IPR.IPR

represents the IP address of the remote end.

slattach allocates the first unallocated SLIP device to the serial device specified. *slattach* starts with *sl0*. Therefore the first *slattach* command attaches SLIP device *sl0* to the serial device specified and *sl1* the next time, etc.

slattach allows you to configure a number of different protocols with the `-p` argument. In your case you will use either *SLIP* or *cSLIP* depending on whether you want to use compression or not. Note: both ends must agree on whether you want compression or not.

7.6 SLIP server.

If you have a machine that is perhaps network connected, that you'd like other people be able to dial into and provide network services, then you will need to configure your machine as a server. If you want to use SLIP as the serial line protocol, then currently you have three options as to how to configure your Linux machine as a SLIP server. My preference would be to use the first presented, *sliplogin*, as it seems the easiest to configure and understand, but I will present a summary of each, so you can make your own decision.

Slip Server using *sliplogin*.

sliplogin is a program that you can use in place of the normal login shell for SLIP users that converts the terminal line into a SLIP line. It allows you to configure your Linux machine as either a *static address server*, users get the same address everytime they call in, or a *dynamic address server*, where users get an address allocated for them which will not necessarily be the same as the last time they called.

The caller will login as per the standard login process, entering their username and password, but instead of being presented with a shell after their login, *sliplogin* is executed which searches its configuration file (`/etc/slip.hosts`) for an entry with a login name that matches that of the caller. If it locates one, it configures the line as an 8bit clean line, and uses an *ioctl* call to convert the line discipline to SLIP. When this process is complete, the last stage of configuration takes place, where *sliplogin* invokes a shell script which configures the SLIP interface with the relevant ip address, netmask and sets appropriate routing in place. This script is usually called `/etc/slip.login`, but in a similar manner to *getty*, if you have certain callers that require special initialization, then you can create configuration scripts called `/etc/slip.login.loginname` that will be run instead of the default specifically for them.

There are either three or four files that you need to configure to get *sliplogin* working for you. I will detail how and where to get the software and how each is configured in detail. The files are:

- `/etc/passwd`, for the dialin user accounts.
- `/etc/slip.hosts`, to contain the information unique to each dial-in user.
- `/etc/slip.login`, which manages the configuration of the routing that needs to be performed for the user.
- `/etc/slip.tty`, which is required only if you are configuring your server for *dynamic address allocation* and contains a table of addresses to allocate
- `/etc/slip.logout`, which contains commands to clean up after the user has hung up or logged out.

TOC

Where to get *sliplogin*

You may already have the *sliplogin* package installed as part of your distribution, if not then *sliplogin* can be obtained from: sunsite.unc.edu. The tar file contains both source, precompiled binaries and a *man* page.

To ensure that only authorized users will be able to run *sliplogin* program, you should add an entry to your `/etc/group` file similar to the following:

```
 ..
slip::13:radio,fred
 ..
```

When you install the *sliplogin* package, the `Makefile` will change the group ownership of the *sliplogin* program to `slip`, and this will mean that only users who belong to that group will be able to execute it. The example above will allow only users `radio` and `fred` to execute *sliplogin*.

To install the binaries into your `/sbin` directory and the *man* page into section 8, do the following:

```
# cd /usr/src
# gzip -dc .../sliplogin-2.1.1.tar.gz | tar xvf -
# cd sliplogin-2.1.1
# <..edit the Makefile if you don't use shadow passwords..>
# make install
```

If you want to recompile the binaries before installation, add a `make clean` before the `make install`. If you want to install the binaries somewhere else, you will need to edit the `Makefile` *install* rule.

Please read the `README` files that come with the package for more information.

Configuring `/etc/passwd` for Slip hosts.

Normally you would create some special logins for Slip callers in your `/etc/passwd` file. A convention commonly followed is to use the *hostname* of the calling host with a capital `S' prefixing it. So, for example, if the calling host is called `radio` then you could create a `/etc/passwd` entry that looked like:

```
Sradio:FvKurok73:1427:1:radio SLIP login:/tmp:/sbin/sliplogin
```

It doesn't really matter what the account is called, so long as it is meaningful to you.

Note: the caller doesn't need any special home directory, as they will not be presented with a shell from this machine, so `/tmp` is a good choice. Also note that *sliplogin* is used in place of the normal login shell.

Configuring `/etc/slip.hosts`

The `/etc/slip.hosts` file is the file that *sliplogin* searches for entries matching the login name to obtain configuration details for this caller. It is this file where you specify the ip address and netmask that will be assigned to the caller and configured for their use. Sample entries for two hosts, one a static configuration for host `radio` and another, a dynamic configuration for user host `albert` might look like:

```
#
Sradio   44.136.8.99   44.136.8.100  255.255.255.0  normal      -1
Salbert  44.136.8.99   DYNAMIC       255.255.255.0  compressed  60
```

```
#
```

The `/etc/slip.hosts` file entries are:

1. the login name of the caller.
2. ip address of the server machine, ie this machine.
3. ip address that the caller will be assigned. If this field is coded `DYNAMIC` then an ip address will be allocated based on the information contained in your `/etc/slip.tty` file discussed later. **Note:** you must be using at least version 1.3 of sliplogin for this to work.
4. the netmask assigned to the calling machine in dotted decimal notation eg 255.255.255.0 for a Class C network mask.
5. the slip mode setting which allows you to enable/disable compression and slip other features. Allowable values here are "`normal`" or "`compressed`".
6. a timeout parameter which specifies how long the line can remain idle (no datagrams received) before the line is automatically disconnected. A negative value disables this feature.
7. optional arguments.

Note: You can use either hostnames or IP addresses in dotted decimal notation for fields 2 and 3. If you use hostnames then those hosts must be resolvable, that is, your machine must be able to locate an ip address for those hostnames, otherwise the script will fail when it is called. You can test this by trying trying to telnet to the hostname, if you get the `*Trying nnn.nnn.nnn...*' message then your machine has been able to find an ip address for that name. If you get the message `*Unknown host*', then it has not. If not, either use ip addresses in dotted decimal notation, or fix up your name resolver configuration (See section `Name Resolution`).

The most common slip modes are:

normal

to enable normal uncompressed SLIP.

compressed

to enable van Jacobsen header compression (cSLIP)

Naturally these are mutually exclusive, you can use one or the other. For more information on the other options available, refer to the *man* pages.

Configuring the `/etc/slip.login` file.

After *sliplogin* has searched the `/etc/slip.hosts` and found a matching entry, it will attempt to execute the `/etc/slip.login` file to actually configure the SLIP interface with its ip address and netmask. The sample `/etc/slip.login` file supplied with the *sliplogin* package looks like this:

```
#!/bin/sh -
#
#       @(#)slip.login  5.1 (Berkeley) 7/1/90
#
# generic login file for a SLIP line.  sliplogin invokes this with
# the parameters:
#     $1       $2       $3    $4, $5, $6 ...
#   SLIPunit ttyspeed   pid   the arguments from the slip.host entry
#
/sbin/ifconfig $1 $5 pointopoint $6 mtu 1500 -trailers up
```

```
/sbin/route add $6
arp -s $6 <hw_addr> pub
exit 0
#
```

You will note that this script simply uses the *ifconfig* and *route* commands to configure the SLIP device with its ipaddress, remote ip address and netmask and creates a route for the remote address via the SLIP device. Just the same as you would if you were using the *slattach* command.

Note also the use of *Proxy ARP* to ensure that other hosts on the same ethernet as the server machine will know how to reach the dial-in host. The `<hw_addr>` field should be the hardware address of the ethernet card in the machine. If your server machine isn't on an ethernet network then you can leave this line out completely.

Configuring the `/etc/slip.logout` file.

When the call drops out, you want to ensure that the serial device is restored to its normal state so that future callers will be able to login correctly. This is achieved with the use of the `/etc/slip.logout` file. It is quite simple in format and is called with the same argument as the `/etc/slip.login` file.

```
#!/bin/sh -
#
#               slip.logout
#
/sbin/ifconfig $1 down
arp -d $6
exit 0
#
```

All it does is `down' the interface which will delete the manual route previously created. It also uses the *arp* command to delete any proxy arp put in place, again, you don't need the *arp* command in the script if your server machine does not have an ethernet port.

Configuring the `/etc/slip.tty` file.

If you are using dynamic ip address allocation (have any hosts configured with the `DYNAMIC` keyword in the `/etc/slip.hosts` file, then you must configure the `/etc/slip.tty` file to list what addresses are assigned to what port. You only need this file if you wish your server to dynamically allocate addresses to users.

The file is a table that lists the *tty* devices that will support dial-in SLIP connections and the ip address that should be assigned to users who call in on that port. Its format is as follows:

```
# slip.tty    tty -> IP address mappings for dynamic SLIP
# format: /dev/tty?? xxx.xxx.xxx.xxx
#
/dev/ttyS0      192.168.0.100
/dev/ttyS1      192.168.0.101
#
```

What this table says is that callers that dial in on port `/dev/ttyS0` who have their remote address field in the `/etc/slip.hosts` file set to `DYNAMIC` will be assigned an address of `192.168.0.100`.

In this way you need only allocate one address per port for all users who do not require an dedicated address themselves. This helps you keep the number of addresses you need down to a minimum to avoid wastage.

Slip Server using *dip*.

Let me start by saying that some of the information below came from the *dip* man pages, where how to run Linux as a SLIP server is briefly documented. Please also beware that the following has been based on the *dip337o-uri.tgz* package and probably will not apply to other versions of *dip*. *dip* has an input mode of operation, where it automatically locates an entry for the user who invoked it and configures the serial line as a SLIP link according to information it finds in the `/etc/diphosts` file. This input mode of operation is activated by invoking *dip* as *diplogin*. This therefore is how you use *dip* as a SLIP server, by creating special accounts where *diplogin* is used as the login shell. The first thing you will need to do is to make a symbolic link as follows:

```
# ln -sf /usr/sbin/dip /usr/sbin/diplogin
```

You then need to add entries to both your `/etc/passwd` and your `/etc/diphosts` files. The entries you need to make are formatted as follows: To configure Linux as a SLIP server with *dip*, you need to create some special SLIP accounts for users, where *dip* (in input mode) is used as the login shell. A suggested convention is that of having all SLIP accounts begin with a capital `S', eg `Sfredm'. A sample `/etc/passwd` entry for a SLIP user looks like:

```
Sfredm:ij/SMxiTlGVCo:1004:10:Fred:/tmp:/usr/sbin/diplogin
^^         ^^        ^^  ^^   ^^   ^^   ^^
|          |         |   |    |    |    \__ diplogin as login shell
|          |         |   |    |    \_______ Home directory
|          |         |   |    \____________ User Full Name
|          |         |   \_________________ User Group ID
|          |         \______________________ User ID
|          \________________________________ Encrypted User Password
\___________________________________________ Slip User Login Name
```

After the user logs in, the *login* program, if it finds and verifies the user ok, will execute the *diplogin* command. *dip*, when invoked as *diplogin* knows that it should automatically assume that it is being used a login shell. When it is started as *diplogin* the first thing it does is use the *getuid()* function call to get the userid of whoever has invoked it. It then searches the `/etc/diphosts` file for the first entry that matches either the userid or the name of the *tty* device that the call has come in on and configures itself appropriately. By judicious decision as to whether to give a user an entry in the `diphosts` file, or whether to let the user be given the default configuration you can build your server in such a way that you can have a mix of static and dynamically assigned address users. *dip* will automatically add a `Proxy-ARP' entry if invoked in input mode, so you do not need to worry about manually adding such entries.

Configuring `/etc/diphosts`

`/etc/diphosts` is used by *dip* to lookup preset configurations for remote hosts. These remote hosts might be users dialing into your linux machine, or they might be for machines that you dial into with your linux machine. The general format for `/etc/diphosts` is as follows:

```
 ..
Suwalt::145.71.34.1:145.71.34.2:255.255.255.0:SLIP uwalt:CSLIP,1006
ttyS1::145.71.34.3:145.71.34.2:255.255.255.0:Dynamic ttyS1:CSLIP,296
```

```
..
```

The fields are:

1. `login name`: as returned by getpwuid(getuid()) or tty name.
2. `unused`: compat. with passwd
3. `Remote Address`: IP address of the calling host, either numeric or by name
4. `Local Address`: IP address of this machine, again numeric or by name
5. `Netmask`: in dotted decimal notation
6. `Comment field`: put whatever you want here.
7. `protocol`: Slip, CSlip etc.
8. `MTU`: decimal number

An example `/etc/net/diphosts` entry for a remote SLIP user might be:

```
Sfredm::145.71.34.1:145.71.34.2:255.255.255.0:SLIP uwalt:SLIP,296
```

which specifies a SLIP link with remote address of 145.71.34.1 and MTU of 296, or:

```
Sfredm::145.71.34.1:145.71.34.2:255.255.255.0:SLIP uwalt:CSLIP,1006
```

which specifies a cSLIP-capable link with remote address 145.71.34.1 and MTU of 1006.

Therefore, all users who you wish to be allowed a statically allocated dial-up IP access should have an entry in the `/etc/diphosts`. If you want users who call a particular port to have their details dynamically allocated then you must have an entry for the `tty` device and do not configure a user based entry. You should remember to configure at least one entry for each `tty` device that your dialup users use to ensure that a suitable configuration is available for them regardless of which modem they call in on.

When a user logs in they will receive a normal login and password prompt at which they should enter their SLIP-login userid and password. If these verify ok then the user will see no special messages and they should just change into SLIP mode at their end. The user should then be able to connect ok and be configured with the relevant parameters from the `diphosts` file.

SLIP server using the *dSLIP* package.

Matt Dillon `<dillon@apollo.west.oic.com>` has written a package that does not only dial-in but also dial-out SLIP. Matt's package is a combination of small programs and scripts that manage your connections for you. You will need to have *tcsh* installed as at least one of the scripts requires it. Matt supplies a binary copy of the *expect* utility as it too is needed by one of the scripts. You will most likely need some experience with *expect* to get this package working to your liking, but don't let that put you off.

Matt has written a good set of installation instructions in the README file, so I won't bother repeating them.

You can get the *dSLIP* package from its home site at: **apollo.west.oic.com**

```
/pub/linux/dillon_src/dSLIP203.tgz
```

or from: **sunsite.unc.edu**

```
/pub/Linux/system/Network/serial/dSLIP203.tgz
```

Read the `README` file and create the `/etc/passwd` and `/etc/group` entries **before** doing a `make install`.

---

8. Other Network Technologies

The following subsections are specific to particular network technologies. The information contained in these sections does not necessarily apply to any other type of network technology. The topics are sorted alphabetically.

8.1 ARCNet

ARCNet device names are `arc0e', `arc1e', `arc2e' etc. or `arc0s', `arc1s', `arc2s' etc. The first card detected by the kernel is assigned `arc0e' or `arc0s' and the rest are assigned sequentially in the order they are detected. The letter at the end signifies whether you've selected ethernet encapsulation packet format or RFC1051 packet format.

Kernel Compile Options:

```
Network device support  --->
    [*] Network device support
    <*> ARCnet support
    [ ]   Enable arc0e (ARCnet "Ether-Encap" packet format)
    [ ]   Enable arc0s (ARCnet RFC1051 packet format)
```

Once you have your kernel properly built to support your ethernet card then configuration of the card is easy.

Typically you would use something like:

```
root# ifconfig arc0e 192.168.0.1 netmask 255.255.255.0 up
root# route add -net 192.168.0.0 netmask 255.255.255.0 arc0e
```

Please refer to the `/usr/src/linux/Documentation/networking/arcnet.txt` and `/usr/src/linux/Documentation/networking/arcnet-hardware.txt` files for further information.

ARCNet support was developed by Avery Pennarun, `apenwarr@foxnet.net`.

8.2 Appletalk (AF_APPLETALK)

The Appletalk support has no special device names as it uses existing network devices.

Kernel Compile Options:

```
Networking options  --->
    <*> Appletalk DDP
```

Appletalk support allows your Linux machine to interwork with Apple networks. An important use for this is to share resources such as printers and disks between both your Linux and Apple computers. Additional software is required, this is called *netatalk*. Wesley Craig `netatalk@umich.edu` represents a team called the `Research Systems Unix Group' at the University of Michigan and they have produced the *netatalk* package which provides software that implements the Appletalk protocol stack and some useful utilities. The *netatalk* package will either have been supplied with your Linux distribution, or you will have to ftp it from its home site at the University of Michigan

To build and install the package do something like:

```
user% tar xvfz .../netatalk-1.4b2.tar.Z
user% make
root# make install
```

You may want to edit the `Makefile' before calling *make* to actually compile the software. Specifically, you might want to change the DESTDIR variable which defines where the files will be installed later. The default of /usr/local/atalk is fairly safe.

Configuring the Appletalk software.

The first thing you need to do to make it all work is to ensure that the appropriate entries in the `/etc/services` file are present. The entries you need are:

```
rtmp  1/ddp   # Routing Table Maintenance Protocol
nbp   2/ddp   # Name Binding Protocol
echo  4/ddp   # AppleTalk Echo Protocol
zip   6/ddp   # Zone Information Protocol
```

The next step is to create the Appletalk configuration files in the `/usr/local/atalk/etc` directory (or wherever you installed the package).

The first file to create is the `/usr/local/atalk/etc/atalkd.conf` file. Initially this file needs only one line that gives the name of the network device that supports the network that your Apple machines are on:

```
eth0
```

The Appletalk daemon program will add extra details after it is run.

Exporting a Linux filesystems via Appletalk.

You can export filesystems from your linux machine to the network so that Apple machine on the network can share them.

To do this you must configure the `/usr/local/atalk/etc/AppleVolumes.system` file. There is another configuration file called `/usr/local/atalk/etc/AppleVolumes.default` which has exactly the same format and describes which filesystems users connecting with guest privileges will receive.

Full details on how to configure these files and what the various options are can be found in the *afpd* man page.

A simple example might look like:

```
/tmp Scratch
/home/ftp/pub "Public Area"
```

Which would export your `/tmp` filesystem as AppleShare Volume `Scratch' and your ftp public directory as AppleShare Volume `Public Area'. The volume names are not mandatory, the daemon will choose some for you, but it won't hurt to specify them anyway.

Sharing your Linux printer across Appletalk.

You can share your linux printer with your Apple machines quite simply. You need to run the *papd* program which is the Appletalk Printer Access Protocol Daemon. When you run this program it will accept requests from your Apple machines and spool the print job to your local line printer daemon for printing.

You need to edit the `/usr/local/atalk/etc/papd.conf` file to configure the daemon. The syntax of this file is the same as that of your usual `/etc/printcap` file. The name you give to the definition is registered with the Appletalk naming protocol, NBP.

A sample configuration might look like:

```
TricWriter:\
   :pr=lp:op=cg:
```

Which would make a printer named `TricWriter' available to your Appletalk network and all accepted jobs would be printed to the linux printer `lp' (as defined in the `/etc/printcap` file) using *lpd*. The entry `op=cg' says that the linux user `cg' is the operator of the printer.

Starting the appletalk software.

Ok, you should now be ready to test this basic configuration. There is an *rc.atalk* file supplied with the *netatalk* package that should work ok for you, so all you should have to do is:

```
root# /usr/local/atalk/etc/rc.atalk
```

and all should startup and run ok. You should see no error messages and the software will send messages to the console indicating each stage as it starts.

Testing the appletalk software.

To test that the software is functioning properly, go to one of your Apple machines, pull down the Apple menu, select the Chooser, click on AppleShare, and your Linux box should appear.

Caveats of the appletalk software.

- You may need to start the Appletalk support before you configure your IP network. If you have problems starting the Appletalk programs, or if after you start them you have trouble with your IP network, then try starting the Appletalk software before you run your `/etc/rc.d/rc.inet1` file.
- The *afpd* (Apple Filing Protocol Daemon) severely messes up your hard disk. Below the mount points it creates a couple of directories called ``.AppleDesktop" and `Network Trash Folder`. Then, for each directory you access it will create a `.AppleDouble` below it so it can store resource forks, etc. So think twice before exporting `/`, you will have a great time cleaning up afterwards.
- The *afpd* program expects clear text passwords from the Macs. Security could be a problem, so be very careful when you run this daemon on a machine connected to the Internet, you have yourself to blame if somebody nasty does something bad.
- The existing diagnostic tools such as *netstat* and *ifconfig* don't support Appletalk. The raw information is available in the `/proc/net/` directory if you need it.

TOC

More information

For a much more detailed description of how to configure Appletalk for Linux refer to Anders Brownworth *Linux Netatalk-HOWTO* page at thehamptons.com.

8.3 ATM

Werner Almesberger `<werner.almesberger@lrc.di.epfl.ch>` is managing a project to provide Asynchronous Transfer Mode support for Linux. Current information on the status of the project may be obtained from: lrcwww.epfl.ch.

8.4 AX25 (AF_AX25)

AX.25 device names are `sl0', `sl1', etc. in `2.0.*` kernels or `ax0', `ax1', etc. in `2.1.*` kernels.

Kernel Compile Options:

```
Networking options  --->
    [*] Amateur Radio AX.25 Level 2
```

The AX25, Netrom and Rose protocols are covered by the AX25-HOWTO. These protocols are used by Amateur Radio Operators world wide in packet radio experimentation.

Most of the work for implementation of these protocols has been done by Jonathon Naylor, `jsn@cs.nott.ac.uk`.

8.5 DECNet

Support for DECNet is currently being worked on. You should expect it to appear in a late `2.1.*` kernel.

8.6 FDDI

FDDI device names are `fddi0', `fddi1', `fddi2' etc. The first card detected by the kernel is assigned `fddi0' and the rest are assigned sequentially in the order they are detected.

Larry Stefani, `lstefani@ultranet.com`, has developed a driver for the Digital Equipment Corporation FDDI EISA and PCI cards.

Kernel Compile Options:

```
Network device support  --->
    [*] FDDI driver support
    [*] Digital DEFEA and DEFPA adapter support
```

When you have your kernel built to support the FDDI driver and installed, configuration of the FDDI interface is almost identical to that of an ethernet interface. You just specify the appropriate FDDI interface name in the *ifconfig* and *route* commands.

8.7 Frame Relay

TOC

The Frame Relay device names are `dlci00', `dlci01' etc for the DLCI encapsulation devices and `sdla0', `sdla1' etc for the FRAD(s).

Frame Relay is a new networking technology that is designed to suit data communications traffic that is of a `bursty' or intermittent nature. You connect to a Frame Relay network using a Frame Relay Access Device (FRAD). The Linux Frame Relay supports IP over Frame Relay as described in RFC-1490.

Kernel Compile Options:

```
Network device support  --->
    <*> Frame relay DLCI support (EXPERIMENTAL)
    (24)   Max open DLCI
    (8)   Max DLCI per device
    <*>   SDLA (Sangoma S502/S508) support
```

Mike McLagan, `mike.mclagan@linux.org`, developed the Frame Relay support and configuration tools.

Currently the only FRAD supported are the Sangoma Technologies `S502A`, `S502E` and `S508`.

To configure the FRAD and DLCI devices after you have rebuilt your kernel you will need the Frame Relay configuration tools. These are available from ftp.invlogic.com. Compiling and installing the tools is straightforward, but the lack of a top level Makefile makes it a fairly manual process:

```
user% tar xvfz .../frad-0.15.tgz
user% cd frad-0.15
user% for i in common dlci frad; make -C $i clean; make -C $i; done
root# mkdir /etc/frad
root# install -m 644 -o root -g root bin/*.sfm /etc/frad
root# install -m 700 -o root -g root frad/fradcfg /sbin
rppt# install -m 700 -o root -g root dlci/dlcicfg /sbin
```

Note that the previous commands use *sh* syntax, if you use a *csh* flavour instead (like *tcsh*), the *for* loop will look different.

After installing the tools you need to create an `/etc/frad/router.conf` file. You can use this template, which is a modified version of one of the example files:

```
# /etc/frad/router.conf
# This is a template configuration for frame relay.
# All tags are included. The default values are based on the code
# supplied with the DOS drivers for the Sangoma S502A card.
#
# A '#' anywhere in a line constitutes a comment
# Blanks are ignored (you can indent with tabs too)
# Unknown [] entries and unknown keys are ignored
#

[Devices]
Count=1                 # number of devices to configure
```

```
Dev_1=sdla0             # the name of a device
#Dev_2=sdla1            # the name of a device

# Specified here, these are applied to all devices and can be overridden for
# each individual board.
#
Access=CPE
Clock=Internal
KBaud=64
Flags=TX
#
# MTU=1500              # Maximum transmit IFrame length, default is 4096
# T391=10               # T391 value    5 - 30, default is 10
# T392=15               # T392 value    5 - 30, default is 15
# N391=6                # N391 value    1 - 255, default is 6
# N392=3                # N392 value    1 - 10, default is 3
# N393=4                # N393 value    1 - 10, default is 4

# Specified here, these set the defaults for all boards
# CIRfwd=16             # CIR forward   1 - 64
# Bc_fwd=16             # Bc forward    1 - 512
# Be_fwd=0              # Be forward    0 - 511
# CIRbak=16             # CIR backward  1 - 64
# Bc_bak=16             # Bc backward   1 - 512
# Be_bak=0              # Be backward   0 - 511


#
#
# Device specific configuration
#
#

#
# The first device is a Sangoma S502E
#
[sdla0]
Type=Sangoma            # Type of the device to configure, currently only
                        # SANGOMA is recognized
#
# These keys are specific to the 'Sangoma' type
#
# The type of Sangoma board - S502A, S502E, S508
Board=S502E
#
# The name of the test firmware for the Sangoma board
# Testware=/usr/src/frad-0.10/bin/sdla_tst.502
#
# The name of the FR firmware
# Firmware=/usr/src/frad-0.10/bin/frm_rel.502
#
Port=360                # Port for this particular card
Mem=C8                  # Address of memory window, A0-EE, depending on card
```

```
IRQ=5                   # IRQ number, do not supply for S502A
DLCIs=1                 # Number of DLCI's attached to this device
DLCI_1=16               # DLCI #1's number, 16 - 991
# DLCI_2=17
# DLCI_3=18
# DLCI_4=19
# DLCI_5=20
#
# Specified here, these apply to this device only,
# and override defaults from above
#
# Access=CPE            # CPE or NODE, default is CPE
# Flags=TXIgnore,RXIgnore,BufferFrames,DropAborted,Stats,MCI,AutoDLCI
# Clock=Internal        # External or Internal, default is Internal
# Baud=128              # Specified baud rate of attached CSU/DSU
# MTU=2048              # Maximum transmit IFrame length, default is 4096
# T391=10               # T391 value    5 - 30, default is 10
# T392=15               # T392 value    5 - 30, default is 15
# N391=6                # N391 value    1 - 255, default is 6
# N392=3                # N392 value    1 - 10, default is 3
# N393=4                # N393 value    1 - 10, default is 4

#
# The second device is some other card
#
# [sdla1]
# Type=FancyCard        # Type of the device to configure.
# Board=                # Type of Sangoma board
# Key=Value             # values specific to this type of device


#
# DLCI Default configuration parameters
# These may be overridden in the DLCI specific configurations
#
CIRfwd=64               # CIR forward   1 - 64
# Bc_fwd=16             # Bc forward    1 - 512
# Be_fwd=0              # Be forward    0 - 511
# CIRbak=16             # CIR backward  1 - 64
# Bc_bak=16             # Bc backward   1 - 512
# Be_bak=0              # Be backward   0 - 511

#
# DLCI Configuration
# These are all optional. The naming convention is
# [DLCI_D<devicenum>_<DLCI_Num>]
#

[DLCI_D1_16]
# IP=
# Net=
# Mask=
```

```
# Flags defined by Sangoma: TXIgnore,RXIgnore,BufferFrames
# DLCIFlags=TXIgnore,RXIgnore,BufferFrames
# CIRfwd=64
# Bc_fwd=512
# Be_fwd=0
# CIRbak=64
# Bc_bak=512
# Be_bak=0

[DLCI_D2_16]
# IP=
# Net=
# Mask=
# Flags defined by Sangoma: TXIgnore,RXIgnore,BufferFrames
# DLCIFlags=TXIgnore,RXIgnore,BufferFrames
# CIRfwd=16
# Bc_fwd=16
# Be_fwd=0
# CIRbak=16
# Bc_bak=16
# Be_bak=0
```

When you've built your `/etc/frad/router.conf` file the only step remaining is to configure the actual devices themselves. This is only a little trickier than a normal network device configuration, you need to remember to bring up the FRAD device before the DLCI encapsulation devices. These commands are best hosted in a shell script, due to their number:

```
#!/bin/sh
# Configure the frad hardware and the DLCI parameters
/sbin/fradcfg /etc/frad/router.conf || exit 1
/sbin/dlcicfg file /etc/frad/router.conf
#
# Bring up the FRAD device
ifconfig sdla0 up
#
# Configure the DLCI encapsulation interfaces and routing
ifconfig dlci00 192.168.10.1 pointopoint 192.168.10.2 up
route add -net 192.168.10.0 netmask 255.255.255.0 dlci00
#
ifconfig dlci01 192.168.11.1 pointopoint 192.168.11.2 up
route add -net 192.168.11.0 netmask 255.255.255.0 dlci00
#
route add default dev dlci00
#
```

8.8 IPX (AF_IPX)

The IPX protocol is most commonly utilized in Novell NetWare(tm) local area network environments. Linux includes support for this protocol and may be configured to act as a network endpoint, or as a router for IPX.

Kernel Compile Options:

```
Networking options  --->
    [*] The IPX protocol
    [ ] Full internal IPX network
```

The IPX protocol and the NCPFS are covered in greater depth in the IPX-HOWTO.

8.9 NetRom (AF_NETROM)

NetRom device names are `nr0', `nr1', etc.

Kernel Compile Options:

```
Networking options  --->
    [*] Amateur Radio AX.25 Level 2
    [*] Amateur Radio NET/ROM
```

The AX25, Netrom and Rose protocols are covered by the AX25-HOWTO. These protocols are used by Amateur Radio Operators world wide in packet radio experimentation.

Most of the work for implementation of these protocols has been done by Jonathon Naylor, `jsn@cs.nott.ac.uk`.

8.10 Rose protocol (AF_ROSE)

Rose device names are `rs0', `rs1', etc. in `2.1.*` kernels. Rose is available in the `2.1.*` kernels.

Kernel Compile Options:

```
Networking options  --->
    [*] Amateur Radio AX.25 Level 2
    <*> Amateur Radio X.25 PLP (Rose)
```

The AX25, Netrom and Rose protocols are covered by the AX25-HOWTO. These protocols are used by Amateur Radio Operators world wide in packet radio experimentation.

Most of the work for implementation of these protocols has been done by Jonathon Naylor, `jsn@cs.nott.ac.uk`.

8.11 SAMBA - `NetBEUI', `NetBios' support.

SAMBA is an implementation of the Session Management Block protocol. Samba allows Microsoft and other systems to mount and use your disks and printers.

SAMBA and its configuration are covered in detail in the SMB-HOWTO.

8.12 STRIP support (Starmode Radio IP)

STRIP device names are `st0', `st1', etc.

Kernel Compile Options:

```
Network device support  --->
        [*] Network device support
        ....
        [*] Radio network interfaces
        < > STRIP (Metricom starmode radio IP)
```

STRIP is a protocol designed specifically for a range of Metricom radio modems for a research project being conducted by Stanford University called the MosquitoNet Project. There is a lot of interesting reading here, even if you aren't directly interested in the project.

The Metricom radios connect to a serial port, employ spread spectrum technology and are typically capable of about 100kbps. Information on the Metricom radios is available from the: Metricom Web Server.

At present the standard network tools and utilities do not support the STRIP driver, so you will have to download some customized tools from the MosquitoNet web server. Details on what software you need is available at the: MosquitoNet STRIP Page.

A summary of configuration is that you use a modified *slattach* program to set the line discipline of a serial tty device to STRIP and then configure the resulting `st[0-9]' device as you would for ethernet with one important exception, for technical reasons STRIP does not support the ARP protocol, so you must manually configure the ARP entries for each of the hosts on your subnet. This shouldn't prove too onerous.

8.13 Token Ring

Token ring device names are `tr0', `tr1' etc. Token Ring is an IBM standard LAN protocol that avoids collisions by providing a mechanism that allows only one station on the LAN the right to transmit at a time. A `token' is held by one station at a time and the station holding the token is the only station allowed to transmit. When it has transmitted its data it passes the token onto the next station. The token loops amongst all active stations, hence the name `Token Ring'.

Kernel Compile Options:

```
Network device support  --->
        [*] Network device support
        ....
        [*] Token Ring driver support
        < > IBM Tropic chipset based adaptor support
```

Configuration of token ring is identical to that of ethernet with the exception of the network device name to configure.

8.14 X.25

X.25 is a circuit based packet switching protocol defined by the `C.C.I.T.T.` (a standards body recognized by Telecommunications companies in most parts of the world). An implementation of X.25 and LAPB are being worked on and recent `2.1.*` kernels include the work in progress.

Jonathon Naylor `jsn@cs.nott.ac.uk` is leading the development and a mailing list has been established to discuss Linux X.25 related matters. To subscribe send a message to: `majordomo@vger.rutgers.edu` with the text "`subscribe linux-x25`" in the body of the message.

Early versions of the configuration tools may be obtained from Jonathon's ftp site at ftp.cs.nott.ac.uk.

8.15 WaveLan Card

TOC

Wavelan device names are `eth0', `eth1', etc.

Kernel Compile Options:

```
Network device support  --->
        [*] Network device support
        ....
        [*] Radio network interfaces
        ....
        <*> WaveLAN support
```

The WaveLAN card is a spread spectrum wireless lan card. The card looks very like an ethernet card in practice and is configured in much the same way.

You can get information on the Wavelan card from Wavelan.com.

---

9. Cables and Cabling

Those of you handy with a soldering iron may want to build your own cables to interconnect two linux machines. The following cabling diagrams should assist you in this.

9.1 Serial NULL Modem cable

Not all NULL modem cables are alike. Many null modem cables do little more than trick your computer into thinking all the appropriate signals are present and swap transmit and receive data. This is ok but means that you must use software flow control (XON/XOFF) which is less efficient than hardware flow control. The following cable provides the best possible signalling between machines and allows you to use hardware (RTS/CTS) flow control.

```
Pin Name  Pin                                Pin
Tx Data    2  ------------------------------  3
Rx Data    3  ------------------------------  2
RTS        4  ------------------------------  5
CTS        5  ------------------------------  4
Ground     7  ------------------------------  7
DTR        20 -\----------------------------  8
DSR        6  -/
RLSD/DCD   8  ----------------------------/-  20
                                           \-  6
```

9.2 Parallel port cable (PLIP cable)

If you intend to use the PLIP protocol between two machines then this cable will work for you irrespective of what sort of parallel ports you have installed.

```
Pin Name    pin            pin
STROBE      1*
D0->ERROR   2  ----------- 15
D1->SLCT    3  ----------- 13
D2->PAPOUT  4  ----------- 12
D3->ACK     5  ----------- 10
D4->BUSY    6  ----------- 11
D5          7*
D6          8*
D7          9*
ACK->D3     10 ----------- 5
```

```
BUSY->D4     11 ----------- 6
PAPOUT->D2   12 ----------- 4
SLCT->D1     13 ----------- 3
FEED         14*
ERROR->D0    15 ----------- 2
INIT         16*
SLCTIN       17*
GROUND       25 ----------- 25
```

Notes:

- Do not connect the pins marked with an asterisk `*'.
- Extra grounds are 18,19,20,21,22,23 and 24.
- If the cable you are using has a metallic shield, it should be connected to the metallic DB-25 shell at **one end only**.

Warning: A miswired PLIP cable can destroy your controller card. Be very careful and double check every connection to ensure you don't cause yourself any unnecessary work or heartache.

While you may be able to run PLIP cables for long distances, you should avoid it if you can. The specifications for the cable allow for a cable length of about 1 metre or so. Please be very careful when running long plip cables as sources of strong electromagnetic fields such as lightning, power lines and radio transmitters can interfere with and sometimes even damage your controller. If you really want to connect two of your computers over a large distance you really should be looking at obtaining a pair of thin-net ethernet cards and running some coaxial cable.

9.3 10base2 (thin coax) Ethernet Cabling

10base2 is an ethernet cabling standard that specifies the use of 52 ohm coaxial cable with a diameter of about 5 millimeters. There are a couple of important rules to remember when interconnecting machines with 10base2 cabling. The first is that you must use terminators at **both ends** of the cabling. A terminator is a 52 ohm resistor that helps to ensure that the signal is absorbed and not reflected when it reaches the end of the cable. Without a terminator at each end of the cabling you may find that the ethernet is unreliable or doesn't work at all. Normally you'd use `T pieces' to interconnect the machines, so that you end up with something that looks like:

```
|==========T=============T=============T==========T==========|
           |             |             |          |
           |             |             |          |
         -----         -----         -----      -----
         |   |         |   |         |   |      |   |
         -----         -----         -----      -----
```

where the `|' at either end represents a terminator, the `======' represents a length of coaxial cable with BNC plugs at either end and the `T' represents a `T piece' connector. You should keep the length of cable between the `T piece' and the actual ethernet card in the PC as short as possible, ideally the `T piece' will be plugged directly into the ethernet card.

9.4 Twisted Pair Ethernet Cable

TOC

If you have only two twisted pair ethernet cards and you wish to connect them you do not require a hub. You can cable the two cards directly together. A diagram showing how to do this is included in the Ethernet-HOWTO

---

10. Glossary of Terms used in this document.

The following is a list of some of the most important terms used in this document.

ARP

This is an acronym for the *Address Resolution Protocol* and this is how a network machine associates an IP Address with a hardware address.

ATM

This is an acronym for *Asynchronous Transfer Mode*. An ATM network packages data into standard size blocks which it can convey efficiently from point to point. ATM is a circuit switched packet network technology.

client

This is usually the piece of software at the end of a system where the user is. There are exceptions to this, for example, in the X11 window system it is actually the server with the user and the client runs on the remote machine. The client is the program or end of a system that is receiving the service provided by the server. In the case of *peer to peer* systems such as *slip* or *ppp* the client is taken to be the end that initiates the connection and the remote end, being called, is taken to be the server.

datagram

A datagram is a discrete package of data and headers which contain addresses, which is the basic unit of transmission across an IP network. You might also hear this called a `packet'.

DLCI

The DLCI is the Data Link Connection Identifier and is used to identify a unique virtual point to point connection via a Frame Relay network. The DLCI's are normally assigned by the Frame Relay network provider.

Frame Relay

Frame Relay is a network technology ideally suited to carrying traffic that is of bursty or sporadic nature. Network costs are reduced by having many Frame Relay customer sharing the same network capacity and relying on them wanting to make use of the network at slightly different times.

Hardware address

This is a number that uniquely identifies a host in a physical network at the media access layer. Examples of this are *Ethernet Addresses* and *AX.25 Addresses*.

ISDN

This is an acronym for *Integrated Services Digital Network*. ISDN provides a standardized means by which Telecommunications companies may deliver either voice or data information to a

TOC

customers premises. Technically ISDN is a circuit switched data network.

ISP

This is an acronym of Internet Service Provider. These are organizations or companies that provide people with network connectivity to the Internet.

IP address

This is a number that uniquely identifies a TCP/IP host on the network. The address is 4 bytes long and is usually represented in what is called the "dotted decimal notation", where each byte is represented in decimal from with dots `.' between them.

MSS

The Maximum Segment Size (*MSS*) is the largest quantity of data that can be transmitted at one time. If you want to prevent local fragmentation MSS would equal MTU-IP header.

MTU

The Maximum Transmission Unit (*MTU*) is a parameter that determines the largest datagram than can be transmitted by an IP interface without it needing to be broken down into smaller units. The MTU should be larger than the largest datagram you wish to transmit unfragmented. Note, this only prevents fragmentation locally, some other link in the path may have a smaller MTU and the datagram will be fragmented there. Typical values are 1500 bytes for an ethernet interface, or 576 bytes for a SLIP interface.

route

The *route* is the path that your datagrams take through the network to reach their destination.

server

This is usually the piece of software or end of a system remote from the user. The server provides some service to one or many clients. Examples of servers include *ftp*, *Networked File System*, or *Domain Name Server*. In the case of *peer to peer* systems such as *slip* or *ppp* the server is taken to be the end of the link that is called and the end calling is taken to be the client.

window

The *window* is the largest amount of data that the receiving end can accept at a given point in time.

---

11. Linux for an ISP ?

If you are interested in using Linux for ISP purposes the I recommend you take a look at the Linux ISP homepage for a good list of pointers to information you might need and use.

12. Acknowledgements

I'd like to thank the following people for their contributions to this document (in no particular order): Terry Dawson, Axel Boldt, Arnt Gulbrandsen, Gary Allpike, Cees de Groot, Alan Cox, Jonathon Naylor, Claes Ensson, Ron Nessim, John Minack, Jean-Pierre Cocatrix, Erez Strauss.

13. Copyright.

The NET-3-HOWTO, information on how to install and configure networking support for Linux. Copyright (c) 1997 Terry Dawson, 1998 Alessandro Rubini This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the: Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

SMB HOWTO

David Wood, dwood@plugged.net.au

v1.1.1, 25 June 1999

This is the SMB HOWTO. This document describes how to use the Server Message Block (SMB) protocol, also called the Session Message Block, NetBIOS or LanManager protocol, with Linux.

1. Introduction

2. Further Information

3. Installation

4. Running The Daemons

5. General Configuration (/etc/smb.conf)

6. Sharing A Linux Drive With Windows Machines

7. Sharing A Windows Drive With Linux Machines

8. Sharing A Linux Printer With Windows Machines

9. Sharing A Windows Printer With Linux Machines

10. Backing Up Windows Machines to a Linux Host

TOC

11. Copyright

12. Acknowledgements

1. Introduction

This is the SMB HOWTO. This document describes how to use the Server Message Block (SMB) protocol, also called the Session Message Block, NetBIOS or LanManager protocol, with Linux. Although this document is Linux-centric, Samba runs on most Unix-like operating systems.

This document is maintained by David Wood (dwood@plugged.net.au). Additions, modifications or corrections may be mailed there for inclusion in the next release.

Much more Samba documentation is available at the Samba Web site, located at http://www.samba.org/. You also might try the comp.protocols.smb newsgroup.

The SMB protocol is used by Microsoft Windows 3.11, NT and 95/98 to share disks and printers. Using the Samba suite of tools by Andrew Tridgell (Andrew.Tridgell@anu.edu.au), UNIX (including Linux) machines can share disk and printers with Windows hosts. The smbfs tools by Paal-Kr. Engstad (engstad@intermetrics.com) and Volker Lendecke (lendecke@namu01.gwdg.de) enable Unix machines to mount SMB shares from Windows or Samba hosts.

There are four things that one can do with Samba:

1. Share a Linux drive with Windows machines.
2. Share a Windows drive with Linux machines.
3. Share a Linux printer with Windows machines.
4. Share a Windows printer with Linux machines.

All of these are covered in this document.

Disclaimer: The procedures and scripts either work for the author or have been reported to work by the people that provided them. Different configurations may not work with the information given here. If you encounter such a situation, you may e-mail the author with suggestions for improvement in this document, but the author guarantees nothing. What did you expect? The author is, after all, a consultant...

Please note that for Windows 3.x machines to access SMB shares, they must have a TCP/IP stack and the Win32s DLLs. Both of these are available on Microsoft's Web site (http://www.microsoft.com).

2. Further Information

This HOWTO attempts to explain how to configure basic SMB file and print services on a Linux machine. Samba is a very complex and complete package. There would be no point in attempting to duplicate all of the documentation for Samba here.

For further information, please see the following documents:

- The Samba documentation, available as part of the Samba distribution. The distribution is available at: ftp://ftp.samba.org/
- The Linux Printing HOWTO.
- The Print2Win Mini-HOWTO.
- Protocol Standard For A NetBIOS Service On A TCP/UDP Transport.

  RFC 1001

  Concepts and Methods.

  RFC 1002

  Detailed Specifications.

3. Installation

First, in order to use Samba your machines must be on a single ethernet LAN segment using the TCP/IP protocol. Samba will not work using other network protocols. This is generally easy since Linux and Windows 95/98/NT ship with TCP/IP support. However, if you are using Windows 3.X machines TCP/IP support will need to be added.

SMB services cannot be used across routers. If you want to do something like that, you would need to set up an IP tunnel which is beyond the scope of this document.

To get the latest source version of Samba, go to this URL and pick the closest mirror site to you: ftp://ftp.samba.org/.

However, if you have installed the RedHat distribution of Linux, you have the option of installing it as a package. Some other distributions also include the Samba binaries.

The following two daemons are required for the Samba package. They are typically installed in /usr/sbin and run either on boot from the systems startup scripts or from inetd. Example scripts are shown in Running the Daemons.

```
smbd (The SMB daemon)
nmbd (Provides NetBIOS nameserver support to clients)
```

Please note that the name service provided by the nmbd daemon is different from the name service provided by the Domain Name Service (DNS). NetBIOS name service is a 'Windows-style' name service used for SMB. In other words, having DNS name service tells you nothing about the state of the ability for Samba to resolve host names.

Typically, the following Samba binaries are installed in /usr/bin or /usr/local/samba/bin, although the location is optional.

```
smbclient       (An SMB client for UNIX machines)
smbprint        (A script to print to a printer on an SMB host)
smbprint.sysv   (As above, but for SVR4 UNIX machines)
smbstatus       (Lists the cuurent SMB connections for the local host)
smbrun          (A 'glue' script to facilitate runnning applciations
                 on SMB hosts)
```

The binaries for smbfs file system support are discussed later in this document.

Additionally, a script called 'print' is included with this HOWTO, which serves as a useful front end to the smbprint script.

The Samba package is simple to install. Simply retrieve the source from the location mentioned above, and read the file README in the distribution. There is also a file called docs/INSTALL.txt in the distribution that provides a simple step-by-step set of instructions.

Following installation, place the daemons in /usr/sbin and the binaries in /usr/bin. Install the man pages in

/usr/local/man.

When you made the Samba package, you would have specified in the Makefile the location for the configuration file, smb.conf. This is generally in /etc, but you can put it anywhere you like. For these directions, we will presume that you specified the location of the configuration file as /etc/smb.conf, the log file location as log file = /var/log/samba-log.%m and the lock directory as lock directory = /var/lock/samba.

Install the configuration file, smb.conf. Go to the directory where Samba was built. Look in the subdirectory examples/simple and read the file README. Copy the file smb.conf found in that directory to /etc. BE CAREFUL! If you have a Linux distribution that already has Samba installed, you may already have a Samba configuration file in /etc. You should probably start with that one.

If you don't want to have your configuration file in /etc, put it wherever you want to, then put a symlink in /etc:

```
ln -s /path/to/smb.conf /etc/smb.conf
```

4. Running The Daemons

The two SMB daemons are /usr/sbin/smbd and /usr/sbin/nmbd.

You can run the Samba daemons from inetd or as stand-alone processes. Samba will respond slightly faster as a standalone daemon than running from inetd.

In either case, you should check the file /etc/services for lines that look like this:

```
netbios-ns      137/tcp         nbns
netbios-ns      137/udp         nbns
netbios-dgm     138/tcp         nbdgm
netbios-dgm     138/udp         nbdgm
netbios-ssn     139/tcp         nbssn
```

Make sure they are all uncommented. Depending on your distribution, you may even need to add them. Samba will not be able to bind to the appropriate ports unless /etc/services has these entries.

To run the daemons from inetd, place the following lines in the inetd configuration file, /etc/inetd.conf:

```
# SAMBA NetBIOS services (for PC file and print sharing)
netbios-ssn stream tcp nowait root /usr/sbin/smbd smbd
netbios-ns dgram udp wait root /usr/sbin/nmbd nmbd
```

Then restart the inetd daemon by running the command:

```
kill -HUP `cat /var/run/inetd.pid`
```

To run the daemons from the system startup scripts, put the following script in file called /etc/rc.d/init.d/smb (for a RedHat distribution) and symbolically link it to the files specified in the comments:

```
#!/bin/sh

#
# /etc/rc.d/init.d/smb - starts and stops SMB services.
#
# The following files should be synbolic links to this file:
# symlinks: /etc/rc.d/rc1.d/K35smb  (Kills SMB services on shutdown)
#           /etc/rc.d/rc3.d/S91smb  (Starts SMB services in multiuser mode)
#           /etc/rc.d/rc6.d/K35smb  (Kills SMB services on reboot)
#

# Source function library.
. /etc/rc.d/init.d/functions
```

TOC

```
# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# See how we were called.
case "$1" in
  start)
    echo -n "Starting SMB services: "
    daemon smbd -D
    daemon nmbd -D
    echo
    touch /var/lock/subsys/smb
    ;;
  stop)
    echo -n "Shutting down SMB services: "
    killproc smbd
    killproc nmbd
    rm -f /var/lock/subsys/smb
    echo ""
    ;;
  *)
    echo "Usage: smb {start|stop}"
    exit 1
esac
```

If when starting Samba you get an error that says something about the daemon failing to bind to port 139, then you probably have another Samba process already running that hasn't yet shut down. Check a process list (with 'ps auxww | grep mbd') to determine if another Samba service is running.

---

5. General Configuration (/etc/smb.conf)

Samba configuration on a Linux (or other UNIX machine) is controlled by a single file, /etc/smb.conf. This file determines which system resources you want to share with the outside world and what restrictions you wish to place on them.

Since the following sections will address sharing Linux drives and printers with Windows machines, the smb.conf file shown in this section is as simple as you can get, just for introductory purposes.

Don't worry about the details, yet. Later sections will introduce the major concepts.

Each section of the file starts with a section header such as [global], [homes], [printers], etc.

The [global] section defines a few variables that Samba will use to define sharing for all resources.

The [homes] section allows a remote users to access their (and only their) home directory on the local (Linux) machine). That is, users trying to connect to this share from Windows machines, will be connected to their personal home directories. Note that to do this, they must have an account on the Linux box.

The sample smb.conf file below allows remote users to get to their home directories on the local machine and to write to a temporary directory. For a Windows user to see these shares, the Linux box has to be on the local network. Then the user simply connects a network drive from the Windows File Manager or Windows Explorer.

Note that in the following sections, additional entries for this file will be given to allow more resources to be shared.

```
; /etc/smb.conf
;
; Make sure and restart the server after making changes to this file, ex:
; /etc/rc.d/init.d/smb stop
; /etc/rc.d/init.d/smb start

[global]
; Uncomment this if you want a guest account
; guest account = nobody
   log file = /var/log/samba-log.%m
   lock directory = /var/lock/samba
   share modes = yes

[homes]
   comment = Home Directories
   browseable = no
   read only = no
   create mode = 0750

[tmp]
   comment = Temporary file space
   path = /tmp
```

```
read only = no
public = yes
```

Having written a new smb.conf, it is useful to test it to verify its correctness. You can test the correctness of a smb.conf file , using the 'testparm' utility (man page: testparm); if testparm reports no problems, smbd will correctly load the configuration file.

Here's a good trick: If your Samba server has more than one ethernet interface, the smbd may bind to the wrong one. If so, you can force it to bind to the intended one by adding a line that looks like this to the [global] section of /etc/smb.conf:

```
interfaces = 192.168.1.1/24
```

where you replace the IP address above with the one that is assigned to the correct ethernet interface. The "24" is correct for a Class C network, but may have to be recalculated if you have subnetted the network. The number relates to the netmask. Numbers for other classes of networks are given in the IP-Masquerade mini-HOWTO.

There is now a GUI configuration tool for Samba: GtkSamba. See http://www.open-systems.com/gtksamba.html.

6. Sharing A Linux Drive With Windows Machines

As shown in the simple smb.conf above, sharing Linux drives with Windows users is easy. However, like everything else with Samba, you can control things to a large degree. Here are some examples:

To share a directory with the public, create a clone of the [tmp] section above by adding something like this to smb.conf:

```
[public]
   comment = Public Stuff
   path = /home/public
   public = yes
   writable = yes
   printable = no
```

To make the above directory readable by the public, but only writable by people in group staff, modify the entry like this:

```
[public]
   comment = Public Stuff
   path = /home/public
   public = yes
   writable = yes
   printable = no
   write list = @staff
```

It used to be that easy; you would now be able to start Samba and browse the shares from a Windows PC. However, Microsoft has recently made life slightly more difficult for those using Samba. Windows 98, Windows NT (service pack 3 or higher) and later builds of Windows 95 now use encrypted passwords by default. Samba uses unencrypted passwords by default. You can't browse servers when either the client or server is using encrypted passwords, because a connection cannot be made anonymously.

You can tell if you have a password type mismatch between client and server if when you try to connect to a share you see a dialog box which reads something like "You are not authorized to access that account from this machine".

You can either configure your Samba server to use encrypted passwords, or configure the Windows machines to use unencrypted passwords.

To get Windows to work with encrypted SMB passwords:

Windows 95/98 ==========

Using the registry editor (regedit), create the registry setting HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP Add a new DWORD value: Value Name: EnablePlainTextPassword Data: 0x01.

Windows NT ==========

Using the registry editor (regedit), create the registry setting HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Rdr\Parameters Add a new DWORD value: Value Name: EnablePlainTextPassword Data: 0x01

Once these registry changes have been made, reboot the Windows machine and try to map a network drive on the Samba server again. It should work as long as the Samba server is using plain text passwords.

To configure Samba to use encrypted passwords:

In the [global] section of /etc/smb.conf, add the following lines:

```
encrypt passwords = yes
smb passwd file = /etc/smbpasswd
```

You are highly encouraged to read the files ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation before doing this!

If your clients and server are using encrypted passwords, you will not be able to browse the available shares on the server until an initial connection has been made with the appropriate authentication. To get the initial connection, enter the share name manually in the Windows File Manager or Explorer dialog box, in the form '\\<hostname>\<sharename>'. Log onto the server with a username and password that is valid on the server!

If you suspect that your NetBIOS name service is not correctly configured (perhaps because you get 'host not found' errors when trying to connect), try using just the IP address of the server: '\\<host ip address>\<sharename>'.

In order to get filenames to appear correctly, you may also need to set some options in the appropriate share section. These work for Windows 95/98/NT clients, but may need to be modified if you have Windows 3.X clients:

```
; Mangle case = yes seems to give the correct filenames
; for Win95/98/NT.
mangle case = yes

; If samba is case sensitive when looking for files
case sensitive = no
```

```
; Default case of files that are created
default case = lower

; Preserve case for all filenames
preserve case = yes

; Preserve case for dos (8.3) filenames
short preserve case = no
```

For other tricks to play with drive shares, see the Samba documentation or man pages.

```
interfaces = 192.168.1.1/24
```

Note: The bit after the / is a reference to the subnet mask. "24" is the value to use for an unsegmented Class C network. For more information on subnet calculations, you might want to see http://www.ziplink.net/~ralphb/IPSubnet/index.html.

There is a lot more to Samba configuration than this, but this will get you started. If you want to do something more advanced, I refer you to the Samba Web site mentioned above.

---

7. Sharing A Windows Drive With Linux Machines

An SMB client program for UNIX machines is included with the Samba distribution. It provides an ftp-like interface on the command line. You can use this utility to transfer files between a Windows 'server' and a Linux client.

To see which shares are available on a given host, run:

```
/usr/sbin/smbclient -L host
```

where 'host' is the name of the machine that you wish to view. this will return a list of 'service' names - that is, names of drives or printers that it can share with you. Unless the SMB server has no security configured, it will ask you for a password. Get it the password for the 'guest' account or for your personal account on that machine.

For example:

```
smbclient -L zimmerman
```

The output of this command should look something like this:

```
Server time is Sat Aug 10 15:58:27 1996
Timezone is UTC+10.0
Password:
Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]

Server=[ZIMMERMAN] User=[] Workgroup=[WORKGROUP] Domain=[]

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        public          Disk      Public
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        OReilly         Printer   OReilly
        print$          Disk      Printer Drivers


This machine has a browse list:

        Server               Comment
        ---------            -------
        HOPPER               Samba 1.9.15p8
        KERNIGAN             Samba 1.9.15p8
        LOVELACE             Samba 1.9.15p8
```

```
        RITCHIE              Samba 1.9.15p8
        ZIMMERMAN
```

TOC

The browse list shows other SMB servers with resources to share on the network.

To use the client, run:

```
/usr/sbin/smbclient service <password>
```

where 'service' is a machine and share name. For example, if you are trying to reach a directory that has been shared as 'public' on a machine called zimmerman, the service would be called \\zimmerman\public. However, due to shell restrictions, you will need to escape the backslashes, so you end up with something like this:

```
/usr/sbin/smbclient \\\\zimmerman\\public mypasswd
```

where 'mypasswd' is the literal string of your password.

You will get the smbclient prompt:

```
Server time is Sat Aug 10 15:58:44 1996
Timezone is UTC+10.0
Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]
smb: \>
```

Type 'h' to get help using smbclient:

```
smb: \> h
ls             dir            lcd            cd             pwd
get            mget           put            mput           rename
more           mask           del            rm             mkdir
md             rmdir          rd             prompt         recurse
translate      lowercase      print          printmode      queue
cancel         stat           quit           q              exit
newer          archive        tar            blocksize      tarmode
setmode        help           ?              !
smb: \>
```

If you can use ftp, you shouldn't need the man pages for smbclient.

Although you can use smbclient for testing, you will soon tire of it for real work. For that you will probably want to use the smbfs package. Smbfs comes with two simple utilties, smbmount and smbumount. They work just like mount and umount for SMB shares.

The smbfs package is now included in most Linux distributions. One important thing to note: You must have smbfs support compiled into your kernel to use these utilities!

8. Sharing A Linux Printer With Windows Machines

To share a Linux printer with Windows machines, you need to make certain that your printer is set up to work under Linux. If you can print from Linux, setting up an SMB share of the printer is stright forward.

See the Printing HOWTO to set up local printing.

Since the author used a printer connected to a Windows NT machine (a long time ago - before converting our network nearly totally to Linux), this section should not be taken as definitive, but merely a suggestion. Anyone with details to share, please send them to dwood@plugged.net.au so this section can be completed.

Add printing configuration to your smb.conf:

```
[global]
   printing = bsd
   printcap name = /etc/printcap
   load printers = yes
   log file = /var/log/samba-log.%m
   lock directory = /var/lock/samba

[printers]
   comment = All Printers
   security = server
   path = /var/spool/lpd/lp
   browseable = no
   printable = yes
   public = yes
   writable = no
   create mode = 0700

[ljet]
   security = server
   path = /var/spool/lpd/lp
   printer name = lp
   writable = yes
   public = yes
   printable = yes
   print command = lpr -r -h -P %p %s
```

Make certain that the printer path (in this case under [ljet]) matches the spool directory in /etc/printcap!

The lines:

```
printcap name = /etc/printcap
load printers = yes
```

controls whether all the printers in /etc/printcap should be loaded by default. If you do this, there is no reason to set up printers individually. The section [printers] specifies options for the printers that you wish to explicitly difine. If the printing subsystem you are using doesn't work this way (BSD), you need to set up a fake printcap file (or to use the 'print command' technique, see below). For more information on the printcap system see the Printing HOWTO.

A useful technique to test the network connection is to change the print command to:

```
print command = cp %S /tmp/print.%P.%S
```

The resulting file can then be analyzed.

NOTE: There are some problems sharing printers on UNIX boxes with Windows NT machines using Samba. One problem is with NT seeing the shared printer properly. To fix this, see the notes in the Samba distribution in the file docs/WinNT.txt. The other deals with password problems. See the comments in the same file for an annoying gain of understanding and failure to fix the problem.

Oleg L. Machulskiy (machulsk@shade.msu.ru) suggests that a better print command to use in the above example would be:

```
print command = smb2ps %s | lpr -r -h -P %p
```

where 'smb2ps' is a script which transforms the spool file received from Windows into usual a usable Postscript file. It must cut off first 3 lines and last 2 lines, because these lines contain some PJL or PCL codes.

That approach is only needed if your Windows machine is printing PCL and not real Postscript. I have found that Windows 95/98/NT don't have a generic Postscript driver per se, but the "Digital turbo Printserver 20" driver acts as a good general Postscript driver for most setups. I have also heard that the "Apple LaserWriter II NTX" driver works for this purpose.

Jeff Stern (jstern@eclectic.ss.uci.edu) reported the following that may be of help to some of you:

----------%<---------- Problem I was having was that I could print via lpd/lpr to my Linux printer, from the Linux box itself. But I couldn't get it to print from a remote Win95 machine. When I tried

TOC

smbclient \\eclectic\belen -P

and then tried to

print myfile

I got errors about access denied. SO I reset the permissions on my /var/spool/lpd/lp1 directory to 777. I'm sorry, but that's what I had to do. I suppose alternatively I could have message with putting everyone on the Linux box in the 'lp' group, or made the directory be owned by the group 'users', etc.. But for now, this works (with ownership root:lp).

Finally, the other thing administrators should know is that the name of the account on the WIndows machine (which is trying to use the Linux printer via Samba) should have an equivalently-named account on the Linux box. Thus, if there is a user named 'joe' on the Windows machine, 'mywinbox' trying to print to the printer belen on Linux machine eclectic (\\eclectic\belen) then there should be a user named 'joe' on the Linux box. Then 'joe's login password will be the password to access eclectic's belen printer. This password will be asked for on the Windows machine when setting up the printer on the Windows box with Printers|Add Printer.

I kind of thought maybe this wouldn't be the case, since I have set up my printer to be 'public' in the smb.conf. But apparently it still asks for password. (The windows box unfortunately doesn't give you the opportunity to supply a different user name to the remote (Linux) printer. It just uses your local user name which you signed in as when you started up Win95.
----------%<----------

Dr. Michael Langner (langner@fiz-chemie.de points out that write permission problems on the /var/spool/lpd/ tree could be avoided by use something like "path = /tmp" and "print command = lpr -r -P%p %s" instead.

Sometimes, a Postscript parsing error will occur with Postscript printing from Windows machines that causes an extra page to be printed at the end of every print job. The last page will always have "%%[Lastpage]%%" at the top of it. This seems to happen with Windows 95 and 98 only and is because the Postscript is malformed.

One way to handle that is to use a script to remove that bit of bad Postscript from the spooled jobs. Another way is to try to find a better Windows Postscript driver. Probably the best way is to us LPRng instead of Postscript to print to a Samba server.

Erik Ratcliffe (erik@caldera.com) Caldera tells me that using LPRng means that any printer driver can be used from Windows machines. On the Samba server, they used an /etc/printcap entry that looked like this:

```
raw:\
        :rw:sh:
        :lp=/dev/lp1
        :sd=/var/spool/lpd/raw
        :fx=flp
```

LPRng doesn't require :\ at the end of every line. A printer entry will still need to be made in /etc/smb.conf for the physical printer. The print command line needs to use the "raw" entry in /etc/printcap and data must be sent to the printer in binary form. Try a print command line like this:

```
print command = lpr -b -Praw %s
```

You may also need to set the spooling on the Windows95 end to print directly to the printer instead of spooling.

9. Sharing A Windows Printer With Linux Machines

To share a printer on a Windows machine, you must do the following:

1. You must have the proper entries in /etc/printcap and they must correspond to the local directory structure (for the spool directory, etc).
2. You must have the script /usr/bin/smbprint. This comes with the Samba source, but not with all Samba binary distributions. A slightly modifed copy is discussed below.
3. If you want to convert ASCII files to Postscript, you must have nenscript, or its equivalent. nenscript is a Postscript converter and is generally installed in /usr/bin.
4. You may wish to make Samba printing easier by having an easy-to-use front end. A simple perl script to handle ASCII, Postscript or created Postscript is given below.
5. You could also use MagicFilter to do the above. The details on setting up MagicFilter are given below the perl script. MagicFilter has advantages because it knows how to automatically convert a lot of file formats.
6. 1. The /etc/printcap entry below is for an HP 5MP printer on a Windows NT host. The entries are as follows:

```
cm - comment
lp - device name to open for output
sd - the printer's spool directory (on the local machine)
af - the accounting file
mx - the maximum file size (zero is unlimited)
if - name of the input filter (script)
```

For more information, see the Printing HOWTO or the man page for printcap.

```
# /etc/printcap
#
# //zimmerman/oreilly via smbprint
#
lp:\
        :cm=HP 5MP Postscript OReilly on zimmerman:\
        :lp=/dev/lp1:\
        :sd=/var/spool/lpd/lp:\
        :af=/var/spool/lpd/lp/acct:\
        :mx#0:\
        :if=/usr/bin/smbprint:
```

Make certain that the spool and accounting directories exist and are writable. Ensure that the 'if' line holds the proper path to the smbprint script (given below) and make sure that the proper device is pointed to (the /dev special file). Next is the smbprint script itself. It is usually placed in /usr/bin and is attributable to Andrew Tridgell, the person who created Samba as far as I know. It comes with the Samba source distribution, but is absent from some binary distributions, so I have recreated it here. You may wish to look at this carefully. There are some minor alterations that have shown themselves to be useful.

```
#!/bin/sh -x

# This script is an input filter for printcap printing on a unix machine. It
# uses the smbclient program to print the file to the specified smb-based
# server and service.
# For example you could have a printcap entry like this
#
# smb:lp=/dev/null:sd=/usr/spool/smb:sh:if=/usr/local/samba/smbprint
#
# which would create a unix printer called "smb" that will print via this
# script. You will need to create the spool directory /usr/spool/smb with
# appropriate permissions and ownerships for your system.
```

TOC

```
# Set these to the server and service you wish to print to
# In this example I have a WfWg PC called "lapland" that has a printer
# exported called "printer" with no password.

#
# Script further altered by hamiltom@ecnz.co.nz (Michael Hamilton)
# so that the server, service, and password can be read from
# a /usr/var/spool/lpd/PRINTNAME/.config file.
#
# In order for this to work the /etc/printcap entry must include an
# accounting file (af=...):
#
#   cdcolour:\
#       :cm=CD IBM Colorjet on 6th:\
#       :sd=/var/spool/lpd/cdcolour:\
#       :af=/var/spool/lpd/cdcolour/acct:\
#       :if=/usr/local/etc/smbprint:\
#       :mx=0:\
#       :lp=/dev/null:
#
# The /usr/var/spool/lpd/PRINTNAME/.config file should contain:
#   server=PC_SERVER
#   service=PR_SHARENAME
#   password="password"
#
# E.g.
#   server=PAULS_PC
#   service=CJET_371
#   password=""

#
# Debugging log file, change to /dev/null if you like.
#
logfile=/tmp/smb-print.log
# logfile=/dev/null


#
# The last parameter to the filter is the accounting file name.
#
spool_dir=/var/spool/lpd/lp
config_file=$spool_dir/.config

# Should read the following variables set in the config file:
#   server
#   service
#   password
#   user
eval `cat $config_file`

#
# Some debugging help, change the >> to > if you want to same space.
#
echo "server $server, service $service" >> $logfile

(
# NOTE You may wish to add the line `echo translate' if you want automatic
# CR/LF translation when printing.
        echo translate
        echo "print -"
        cat
) | /usr/bin/smbclient "\\\\$server\\$service" $password -U $user -N -P >> $logfile
```

Most Linux distributions come with nenscript for converting ASCII documents to Postscript. The following perl script mak easier be providing a simple interface to Linux printing via smbprint.

```
Usage: print [-a|c|p] <filename>
       -a prints <filename> as ASCII
       -c prints <filename> formatted as source code
       -p prints <filename> as Postscript
        If no switch is given, print attempts to
        guess the file type and print appropriately.
```

Using smbprint to print ASCII files tends to truncate long lines. This script breaks long lines on whitespace (instead of in the middle of a word), if possible. The source code formatting is done with nenscript. It takes an ASCII file and foramts it in 2 columns with a fancy header (date, filename, etc). It also numbers the lines. Using this as an example, other types of formatting can be accomplished. Postscript documents are already properly formatted, so they pass through directly.

```
#!/usr/bin/perl

# Script:   print
# Authors:  Brad Marshall, David Wood
#           Plugged In Communications
# Date:     960808
#
# Script to print to a Postscript printer via Samba.
# Purpose:  Takes files of various types as arguments and
# processes them appropriately for piping to a Samba print script.
#
# Currently supported file types:
#
# ASCII      - ensures that lines longer than $line_length characters wrap on
#              whitespace.
# Postscript - Takes no action.
# Code       - Formats in Postscript (using nenscript) to display
#              properly (landscape, font, etc).
#

# Set the maximum allowable length for each line of ASCII text.
$line_length = 76;

# Set the path and name of the Samba print script
$print_prog = "/usr/bin/smbprint";

# Set the path and name to nenscript (the ASCII-->Postscript converter)
$nenscript = "/usr/bin/nenscript";

unless ( -f $print_prog ) {
        die "Can't find $print_prog!";
}
unless ( -f $nenscript ) {
        die "Can't find $nenscript!";
}

&ParseCmdLine(@ARGV);

# DBG
print "filetype is $filetype\n";

if ($filetype eq "ASCII") {
        &wrap($line_length);
} elsif ($filetype eq "code") {
```

TOC

```
        &codeformat;
} elsif ($filetype eq "ps") {
        &createarray;
} else {
        print "Sorry..no known file type.\n";
        exit 0;
}
# Pipe the array to smbprint
open(PRINTER, "|$print_prog") || die "Can't open $print_prog: $!\n";
foreach $line (@newlines) {
        print PRINTER $line;
}
# Send an extra linefeed in case a file has an incomplete last line.
print PRINTER "\n";
close(PRINTER);
print "Completed\n";
exit 0;

# ---------------------------------------------------- #
#        Everything below here is a subroutine         #
# ---------------------------------------------------- #

sub ParseCmdLine {
        # Parses the command line, finding out what file type the file is

        # Gets $arg and $file to be the arguments (if the exists)
        # and the filename
        if ($#_ < 0) {
                &usage;
        }
        # DBG
#       foreach $element (@_) {
#               print "*$element* \n";
#       }

        $arg = shift(@_);
        if ($arg =~ /\-./) {
                $cmd = $arg;
        # DBG
#       print "\$cmd found.\n";

                $file = shift(@_);
        } else {
                $file = $arg;
        }

        # Defining the file type
        unless ($cmd) {
                # We have no arguments

                if ($file =~ /\.ps$/) {
                        $filetype = "ps";
                } elsif ($file =~
/\.java$|\.c$|\.h$|\.pl$|\.sh$|\.csh$|\.m4$|\.inc$|\.html$|\.htm$/) {
                        $filetype = "code";
                } else {
                        $filetype = "ASCII";
                }

                # Process $file for what type is it and return $filetype
        } else {
                # We have what type it is in $arg
```

TOC

```
                        if ($cmd =~ /^-p$/) {
                                $filetype = "ps";
                        } elsif ($cmd =~ /^-c$/) {
                                $filetype = "code";
                        } elsif ($cmd =~ /^-a$/) {
                                $filetype = "ASCII"
                        }
                }
        }

        sub usage {
                print "
        Usage: print [-a|c|p] <filename>
               -a prints <filename> as ASCII
               -c prints <filename> formatted as source code
               -p prints <filename> as Postscript
                If no switch is given, print attempts to
                guess the file type and print appropriately.\n
        ";
                exit(0);
        }

        sub wrap {
                # Create an array of file lines, where each line is < the
                # number of characters specified, and wrapped only on whitespace

                # Get the number of characters to limit the line to.
                $limit = pop(@_);

                # DBG
                #print "Entering subroutine wrap\n";
                #print "The line length limit is $limit\n";

                # Read in the file, parse and put into an array.
                open(FILE, "<$file") || die "Can't open $file: $!\n";
                while(<FILE>) {
                        $line = $_;

                        # DBG
                        #print "The line is:\n$line\n";

                        # Wrap the line if it is over the limit.
                        while ( length($line) > $limit ) {

                                # DBG
                                #print "Wrapping...";

                                # Get the first $limit +1 characters.
                                $part = substr($line,0,$limit +1);

                                # DBG
                                #print "The partial line is:\n$part\n";

                                # Check to see if the last character is a space.
                                $last_char = substr($part,-1, 1);
                                if ( " " eq $last_char ) {
                                    # If it is, print the rest.

                                    # DBG
                                    #print "The last character was a space\n";

                                    substr($line,0,$limit + 1) = "";
```

TOC

```
                                substr($part,-1,1) = "";
                                push(@newlines,"$part\n");
                            } else {
                                 # If it is not, find the last space in the
                                 # sub-line and print up to there.

                                # DBG
                                #print "The last character was not a space\n";

                                 # Remove the character past $limit
                                 substr($part,-1,1) = "";
                                 # Reverse the line to make it easy to find
                                 # the last space.
                                 $revpart = reverse($part);
                                 $index = index($revpart," ");
                                 if ( $index > 0 ) {
                                   substr($line,0,$limit-$index) = "";
                                   push(@newlines,substr($part,0,$limit-$index)
                                       . "\n");
                                 } else {
                                   # There was no space in the line, so
                                   # print it up to $limit.
                                   substr($line,0,$limit) = "";
                                   push(@newlines,substr($part,0,$limit)
                                       . "\n");
                                 }
                            }
                    }
                    push(@newlines,$line);
            }
            close(FILE);
    }

    sub codeformat {
            # Call subroutine wrap then filter through nenscript
            &wrap($line_length);

            # Pipe the results through nenscript to create a Postscript
            # file that adheres to some decent format for printing
            # source code (landscape, Courier font, line numbers).
            # Print this to a temporary file first.
            $tmpfile = "/tmp/nenscript$$";
            open(FILE, "|$nenscript -2G -i$file -N -p$tmpfile -r") ||
                    die "Can't open nenscript: $!\n";
            foreach $line (@newlines) {
                    print FILE $line;
            }
            close(FILE);

            # Read the temporary file back into an array so it can be
            # passed to the Samba print script.
            @newlines = ("");
            open(FILE, "<$tmpfile") || die "Can't open $file: $!\n";
            while(<FILE>) {
                    push(@newlines,$_);
            }
            close(FILE);
            system("rm $tmpfile");
    }

    sub createarray {
            # Create the array for postscript
```

```
        open(FILE, "<$file") || die "Can't open $file: $!\n";
        while(<FILE>) {
                push(@newlines,$_);
        }
        close(FILE);
}
```

Now the MagicFilter way. Thanks to Alberto Menegazzi (flash.egon@iol.it) for this information. Alberto says: ----------%<---------- 1) Install MagicFilter with the filter for the printers you need in /usr/bin/local but DON'T fill /etc/printcap with the suggestion given by the documentation from MagicFilter. 2) Write the /etc/printcap like this way (it's done for my LaserJet 4L): lp|ljet4l:\ :cm=HP LaserJet 4L:\ :lp=/dev/null:\ # or /dev/lp1 :sd=/var/spool/lpd/ljet4l:\ :af=/var/spool/lpd/ljet4l/acct:\ :sh:mx#0:\ :if=/usr/local/bin/main-filter: You should explain that the lp=/dev/... is opened for locking so "virtual" devices one for every remote printer should be used. Example creating with : touch /dev/ljet4l 3) Write the filter /usr/local/bin/main-filter the same you suggest using the ljet4l-filter instead of cat. Here's mine. #! /bin/sh logfile=/var/log/smb-print.log spool_dir=/var/spool/lpd/ljet4l (echo "print -" /usr/local/bin/ljet4l-filter) | /usr/bin/smbclient "\\\\SHIR\\HPLJ4" -N -P >> $logfile P.S. : here is the quote from the Print2Win mini-Howto about locking and why creating virtual printers ---Starts here Hint from Rick Bressler : Good tip sheet. I use something very similar. One helpful tip, this is not a particularly good idea: :lp=/dev/null:\ lpr does an 'exclusive' open on the file you specify as lp=. It does this in order to prevent multiple processes from trying to print to the dame printer at the same time. The side effect of this is that in your case, eng and colour can't print at the same time, (usually more or less transparent since they probably print quickly and since they queue you probably don't notice) but any other process that tries to write to /dev/null will break! On a single user system, probably not a big problem. I have a system with over 50 printers. It would be a problem there. The solution is to create a dummy printer for each. Eg: touch /dev/eng. I have modified the lp entries in the printcap file above to take into account Rick's suggestion. I did the following: #touch /dev/eng #touch /dev/colour ---Ends here ----------%<----------

---

10. Backing Up Windows Machines to a Linux Host

Adam Neat (adamneat@ipax.com.au) kindly contributed the following script to back up Windows machines to a Linux host, using the smbclient utility. Adam says that it is used to backup Windows 3.x and NT machines to a Linux based DAT SCSI Drive.

Adam is not proud of the coding style used here, but it works. As I like to say, "If it works and its stupid, then it is not stupid".

In this script, the string 'agnea1' is the username on the Linux machine that does the backups.

```
#!/bin/bash

clear
echo Initialising ...
checkdate=`date | awk '{print $1}'`

if [ -f "~agnea1/backup-dir/backup-data" ]; then

        echo "ERROR: No config file for today!"
        echo "FATAL!"
        exit 1
fi

if [ -d "~agnea1/backup-dir/temp" ]; then

        echo "ERROR: No tempoary directory found!"
        echo
        echo "Attempting to create"
        cd ~agnea1
        cd backup-dir
        mkdir temp
        echo "Directory Made - temp"
fi

if [ "$1" = "" ]; then

        echo "ERROR: enter in a machine name (ie: cdwriter)"
        exit 1
fi

if [ "$2" = "" ]; then

        echo "ERROR: enter in a SMB (Lan Manager) Resource (ie: work)"
        exit 1
fi

if [ "$3" = "" ]; then
```

TOC

```
        echo "ERROR: enter in an IP address for $1 (ie:
        130.xxx.xxx.52)" exit 1
fi


##############################################################################
# Main Section
#
##############################################################################

cd ~agnea1/backup-dir/temp
rm -r ~agnea1/backup-dir/temp/*
cd ~agnea1/backup-dir/

case "$checkdate"
in
        Mon)
                echo "Backuping for Monday"
                cat backup-data | /usr/local/samba/bin/smbclient
                \\\\$1\\$2 -I$3 -N echo "Complete"

                        if [ -d "~agnea1/backup-dir/Monday" ]; then
                                echo "Directory Monday Not found ...
                                making" mkdir
                                ~agnea1/backup-dir/Monday
                        fi

                echo "Archiving ..."
                cd ~agnea1/backup-dir/temp
                tar -cf monday.tar *                echo "done ..."
                rm ~agnea1/backup-dir/Monday/monday.tar
                mv monday.tar ~agnea1/backup-dir/Monday
                ;;


        Tue)
                echo "Backuping for Tuesday"
                cat backup-data | /usr/local/samba/bin/smbclient
                \\\\$1\\$2 -I$3 -N echo "Complete"

                        if [ -d "~agnea1/backup-dir/Tuesday" ]; then
                                echo "Directory Tuesday Not found ...
                                making" mkdir
                                ~agnea1/backup-dir/Tuesday
                        fi
                echo "Archiving ..."
                cd ~agnea1/backup-dir/temp
                tar -cf tuesday.tar *
                echo "done ..."
                rm ~agnea1/backup-dir/Tuesday/tuesday.tar
                mv tuesday.tar ~agnea1/backup-dir/Tuesday
                ;;
```

TOC

```
Wed)
        echo "Backuping for Wednesday"
        cat backup-data | /usr/local/samba/bin/smbclient
        \\\\$1\\$2 -I$3 -N echo "Complete"

                if [ -d "~agnea1/backup-dir/Wednesday" ]; then
                        echo "Directory Wednesday Not found
                        ... making" mkdir
                        ~agnea1/backup-dir/Wednesday
                fi
        echo "Archiving ..."
        cd ~agnea1/backup-dir/temp
        tar -cf wednesday.tar *
        echo "done ..."
     rm ~agnea1/backup-dir/Wednesday/wednesday.tar
        mv wednesday.tar ~agnea1/backup-dir/Wednesday
        ;;

Thu)
        echo "Backuping for Thrusday"
        cat backup-data | /usr/local/samba/bin/smbclient
        \\\\$1\\$2 -I$3 -N echo "Complete"

                if [ -d "~agnea1/backup-dir/Thursday" ]; then
                        echo "Directory Thrusday Not found ...
                        making" mkdir
                        ~agnea1/backup-dir/Thursday
                fi
        echo "Archiving ..."
        cd ~agnea1/backup-dir/temp
        tar -cf thursday.tar *
        echo "done ..."
        rm ~agnea1/backup-dir/Thursday/thursday.tar
        mv thursday.tar ~agnea1/backup-dir/Thursday
        ;;


Fri)
        echo "Backuping for Friday"
        cat backup-data | /usr/local/samba/bin/smbclient
        \\\\$1\\$2 -I$3 -N echo "Complete"

                if [ -d "~agnea1/backup-dir/Friday" ]; then
                        echo "Directory Friday Not found ...
                        making" mkdir
                        ~agnea1/backup-dir/Friday
                fi
        echo "Archiving ..."
        cd ~agnea1/backup-dir/temp
        tar -cf friday.tar *
        echo "done ..."
```

```
                rm ~agnea1/backup-dir/Friday/friday.tar
                mv friday.tar ~agnea1/backup-dir/Friday
                ;;

        *)
                echo "FATAL ERROR: Unknown variable passed for day"
                exit 1;;

esac
###########
```

11. Copyright

This HOWTO is copyright 1996-9 by David Wood. It may be reproduced in any form and freely distributed as long as the file stays intact, including this statement.

12. Acknowledgements

Brad Marshall (bmarshall@plugged.net.au) and Jason Parker (jparker@plugged.net.au) contributed time, patience, scripting and research.

Adam Neat (adamneat@ipax.com.au) contributed the bash script used to back up Windows machines to a Linux host.

Matthew Flint () told me about the use of the 'interfaces' option in smb.conf.

Oleg L. Machulskiy (machulsk@shade.msu.ru), Jeff Stern (jstern@eclectic.ss.uci.edu), Dr. Michael Langner (langner@fiz-chemie.de and Erik Ratcliffe (erik@caldera.com) suggested modifications to the section on Sharing A Linux Printer With Windows Machines.

Alberto Menegazzi (flash.egon@iol.it) contributed the MagicFilter setup to enable a Linux machine to share a Windows printer.

Andrea Girotto (icarus@inca.dei.unipd.it) contributed a number of valuable suggestions throughout the document.

Thanks, also, to all of the international translators that have brought this HOWTO to the non-English speaking world: Takeo Nakano (nakano@apm.seikei.ac.jp), Klaus-Dieter Schumacher (Klaus-Dieter.Schumacher@fernuni-hagen.de), Andrea Girotto (icarus@inca.dei.unipd.it) and many others for whom I don't have contact details.

Multicast over TCP/IP HOWTO

Juan-Mariano de Goyeneche <jmseyas@dit.upm.es>

v1.0, 20 March 1998

This HOWTO tries to cover most aspects related to multicast over TCP/IP networks. So, a lot of information within it is not Linux-specific (just in case you don't use GNU/Linux... yet). Multicast is currently an active area of research and, at the time of writing, many of the "standards" are merely drafts. Keep it in mind while reading the lines that follow.

1. Introduction.

- 1.1 What is Multicast.
- 1.2 The problem with Unicast.

2. Multicast Explained.

- 2.1 Multicast addresses.
- 2.2 Levels of conformance.
- 2.3 Sending Multicast Datagrams.
- 2.4 Receiving Multicast Datagrams.

3. Kernel requirements and configuration.

4. The MBone.

5. Multicast applications.

6. Multicast programming.

- 6.1 IP_MULTICAST_LOOP.
- 6.2 IP_MULTICAST_TTL.
- 6.3 IP_MULTICAST_IF.
- 6.4 IP_ADD_MEMBERSHIP.
- 6.5 IP_DROP_MEMBERSHIP.

7. The internals.

- 7.1 IGMP.
- 7.2 Kernel corner.

8. Routing Policies and Forwarding Techniques.

9. Multicast Transport Protocols.

10. References.

- 10.1 RFCs.
- 10.2 Internet Drafts.
- 10.3 Web pages.
- 10.4 Books.

11. Copyright and Disclaimer.

12. Acknowledgements.

1. Introduction.

I'll try to give here the most wide range, up to date and accurate information related to multicasting over TCP/IP networks that I can. Any feedback is very welcome. If you find any mistakes in this document, have any comments about its contents or an update or addition, please send them to me at the address listed at the top of this howto.

1.1 What is Multicast.

Multicast is... a need. Well, at least in some scenarios. If you have information (a *lot* of information, usually) that should be transmitted to various (but usually not *all*) hosts over an internet, then Multicast is the answer. One common situation in which it is used is when distributing real time audio and video to the set of hosts which have joined a distributed conference.

Multicast is much like radio or TV in the sense that only those who have tuned their receivers (by selecting a particular frequency they are interested on) receive the information. That is: you hear the channel you are interested in, but not the others.

1.2 The problem with Unicast.

Unicast is anything that is not broadcast nor multicast. All right, the definition is not very bright... When you send a packet and there is only one sender process -yours- and one recipient process (the *one* you are sending the packet to), then this is unicast. TCP is, by its own nature, unicast oriented. UDP supports a lot more paradigms, but if you are sending UDP packets and there is only one precess supposed to receive them, this is unicast too.

For years unicast transmissions proved to be enough for the Internet. It was not until 1993 when the first implementation of multicast saw the light in the 4.4 BSD release. It seems nobody needed it until then. Which were those new problems that multicast addressed?

Needless to say that the Internet has changed a lot since the "early days". Particularly, the appearance of the Web strongly transformed the situation: people didn't just want connections to remote hosts, mail and FTP. First they wanted to see the pictures people placed in their home pages, but later they also wanted to *see* and *hear* that people.

With today's technology it is possible to afford the "cost" of making a unicast connection with everyone who wants to see your web page. However, if you are to send audio and video, which needs a *huge* amount of bandwidth compared with web applications, you have -you *had*, until multicast came into scene- two options: to establish a separate unicast connection with *each* of the recipients, or to use broadcast. The first solution is not affordable: if we said that a *single* connection sending audio/video consumes a huge bandwidth, imagine having to establish hundreds or, may be, thousands of those connections. Both the sending computer and your network would collapse.

Broadcast seems to be *a* solution, but it's not certainly *the* solution. If you want all the hosts in your LAN to attend the conference, you may use broadcast. Packets will be sent only once and every host will receive them as they are sent to the broadcast address. The problem is that perhaps only a *few* of the hosts and not *all* are interested in those packets. Furthermore: perhaps some hosts are really interested in your conference, but they are outside of your LAN, a few routers away. And you know that broadcast works fine inside a LAN, but problems arise when you want broadcast packets to be routed across different LANs.

The best solution seems to be one in which you send packets to a certain special address (a certain frequency in radio/TV transmissions). Then, all hosts which have decided to join the conference will be aware of packets with that destination address, read them when they traverse the network, and pass them to the IP layer to be demultiplexed. This is similar to broadcasting in that you send only one broadcast packet and all the hosts in the network recognize and read it; it differs, however, in that not all multicast packets are read and processed, but only those that were previously registered in the kernel as being "of interest".

Those special packets are routed at kernel level like any packet because they *are* IP packets. The only difference might reside in the routing algorithm which tells the kernel where to route or not to route them.

---

2. Multicast Explained.

2.1 Multicast addresses.

As you probably know, the range of IP addresses is divided into "classes" based on the high order bits of a 32 bits IP address:

```
Bit -->  0                            31            Address Range:
        +-+-----------------------------+
        |0|       Class A Address       |       0.0.0.0 - 127.255.255.255
        +-+-----------------------------+
        +-+-+---------------------------+
        |1 0|     Class B Address       |     128.0.0.0 - 191.255.255.255
        +-+-+---------------------------+
        +-+-+-+-----------------------+
        |1 1 0|   Class C Address       |     192.0.0.0 - 223.255.255.255
        +-+-+-+-----------------------+
        +-+-+-+-+-----------------------+
        |1 1 1 0|  MULTICAST Address    |     224.0.0.0 - 239.255.255.255
        +-+-+-+-+-----------------------+
        +-+-+-+-+-+---------------------+
        |1 1 1 1 0|     Reserved        |     240.0.0.0 - 247.255.255.255
        +-+-+-+-+-+---------------------+
```

The one which concerns us is the "Class D Address". Every IP datagram whose destination address starts with "1110" is an IP Multicast datagram.

The remaining 28 bits identify the multicast "*group*" the datagram is sent to. Following with the previous analogy, you have to tune your radio to hear a program that is transmitted at some specific frequency, in the same way you have to "tune" your kernel to receive packets sent to an specific multicast group. When you do that, it's said that the host has *joined* that group in the interface you specified. More on this later.

There are some special multicast groups, say "well known multicast groups", you should not use in your particular applications due the special purpose they are destined to:

- 224.0.0.1 is the *all-hosts* group. If you ping that group, all multicast capable hosts on the network should answer, as every multicast capable host *must* join that group at start-up on all it's multicast capable interfaces.
- 224.0.0.2 is the *all-routers* group. All multicast routers must join that group on all it's multicast capable interfaces.
- 224.0.0.4 is the *all DVMRP routers*, 224.0.0.5 the *all OSPF routers*, 224.0.013 the *all PIM routers*, etc.

All this special multicast groups are regularly published in the "Assigned Numbers" RFC.

TOC

In any case, range 224.0.0.0 through 224.0.0.255 is reserved for local purposes (as administrative and maintenance tasks) and datagrams destined to them are never forwarded by multicast routers. Similarly, the range 239.0.0.0 to 239.255.255.255 has been reserved for "administrative scoping" (see section 2.3.1 for information on administrative scoping).

2.2 Levels of conformance.

Hosts can be in three different levels of conformance with the Multicast specification, according to the requirements they meet.

Level 0 is the "no support for IP Multicasting" level. Lots of hosts and routers in the Internet are in this state, as multicast support is not mandatory in IPv4 (it is, however, in IPv6). Not too much explanation is needed here: hosts in this level can neither send nor receive multicast packets. They must ignore the ones sent by other multicast capable hosts.

Level 1 is the "support for sending but not receiving multicast IP datagrams" level. Thus, note that it is not necessary to join a multicast group to be able to send datagrams to it. Very few additions are needed in the IP module to make a "Level 0" host "Level 1-compliant", as shown in section 2.3.

Level 2 is the "full support for IP multicasting" level. Level 2 hosts must be able to both send and receive multicast traffic. They must know the way to join and leave multicast groups and to propagate this information to multicast routers. Thus, they must include an Internet Group Management Protocol (IGMP) implementation in their TCP/IP stack.

2.3 Sending Multicast Datagrams.

By now, it should be obvious that multicast traffic is handled at the transport layer with UDP, as TCP provides point-to-point connections, not feasibles for multicast traffic. (Heavy research is taking place to define and implement new multicast-oriented transport protocols. See section Multicast Transport Protocols for details).

In principle, an application just needs to open a UDP socket and fill with a class D multicast address the destination address where it wants to send data to. However, there are some operations that a sending process must be able to control.

TTL.

The TTL (Time To Live) field in the IP header has a double significance in multicast. As always, it controls the live time of the datagram to avoid it being looped forever due to routing errors. Routers decrement the TTL of every datagram as it traverses from one network to another and when its value reaches 0 the packet is dropped.

The TTL in IPv4 multicasting has also the meaning of "threshold". Its use becomes evident with an example: suppose you set a long, bandwidth consuming, video conference between all the hosts belonging to your department. You want that huge amount of traffic to remain in your LAN. Perhaps your department is big enough to have various LANs. In that case you want those hosts belonging to each of *your* LANs to attend the conference, but in any case you want to collapse the entire Internet with your multicast traffic. There is a need to limit how "long" multicast traffic will expand across routers. That's what the TTL is used for. Routers have a TTL threshold assigned to each of its interfaces, and only datagrams with a TTL greater than the interface's threshold are forwarded. Note that when a datagram traverses a router with a certain threshold assigned, the datagram's TTL is *not* decremented by the value of the threshold. Only a comparison is made. (As before, the

TTL is decremented by 1 each time a datagram passes across a router).

A list of TTL thresholds and their associated scope follows:

```
TTL     Scope
-----------------------------------------------------------------------
   0    Restricted to the same host. Won't be output by any interface.
   1    Restricted to the same subnet. Won't be forwarded by a router.
 <32         Restricted to the same site, organization or department.
 <64 Restricted to the same region.
<128 Restricted to the same continent.
<255 Unrestricted in scope. Global.
```

Nobody knows what "site" or "region" mean exactly. It is up to the administrators to decide what this limits apply to.

The TTL-trick is not always flexible enough for all needs, specially when dealing with overlapping regions or trying to establish geographic, topologic and bandwidth limits simultaneously. To solve this problems, administratively scoped IPv4 multicast regions were established in 1994. (see D. Meyer's "*Administratively Scoped IP Multicast*" Internet draft). It does scoping based on multicast addresses rather than on TTLs. The range 239.0.0.0 to 239.255.255.255 is reserved for this administrative scoping.

Loopback.

When the sending host is Level 2 conformant and is also a member of the group datagrams are being sent to, a copy is looped back by default. This does not mean that the interface card reads its own transmission, recognizes it as belonging to a group the interface belongs to, and reads it from the network. On the contrary, is the IP layer which, by default, recognizes the to-be-sent datagram and copies and queues it on the IP input queue before sending it.

This feature is desirable in some cases, but not in others. So the sending process can turn it on and off at wish.

Interface selection.

Hosts attached to more than one network should provide a way for applications to decide which network interface will be used to output the transmissions. If not specified, the kernel chooses a default one based on system administrator's configuration.

2.4 Receiving Multicast Datagrams.

Joining a Multicast Group.

Broadcast is (in comparison) easier to implement than multicast. It doesn't require processes to give the kernel some rules regarding what to do with broadcast packets. The kernel just knows what to do: read and deliver *all* of them to the proper applications.

With multicast, however, it is necessary to advise the kernel which multicast groups we are interested in. That

TOC

is, we have to ask the kernel to "join" those multicast groups. Depending on the underlying hardware, mult datagrams are filtered by the hardware or by the IP layer (and, in some cases, by both). Only those with a destination group previously registered via a join are accepted.

Essentially, when we join a group we are telling the kernel: "OK. I know that, by default, you ignore multicast datagrams, but remember that I am interested in *this* multicast group. So, do read and deliver (to any process interested in them, not only to me) any datagram that you see in this network interface with this multicast group in its destination field".

Some considerations: first, note that you don't just join a group. You join a group *on* a particular network interface. Of course, it is possible to join the same group on more than one interface. If you don't specify a concrete interface, then the kernel will choose it based on its routing tables when datagrams are to be sent. It is also possible that more than one process joins the same multicast group on the same interface. They will all receive the datagrams sent to that group via that interface.

As said before, any multicast-capable hosts join the *all-hosts* group at start-up , so "pinging" 224.0.0.1 returns all hosts in the network that have multicast enabled.

Finally, consider that for a process to receive multicast datagrams it has to ask the kernel to join the group *and* bind the port those datagrams were being sent to. The UDP layer uses both the destination address and port to demultiplex the packets and decide which socket(s) deliver them to.

Leaving a Multicast Group.

When a process is no longer interested in a multicast group, it informs the kernel that *it* wants to leave that group. It is important to understand that this doesn't mean that the kernel will no longer accept multicast datagrams destined to that multicast group. It will still do so if there are more precesses who issued a "multicast join" petition for that group and are still interested. In that case *the host* remains member of the group, until all the processes decide to leave the group.

Even more: if you leave the group, but remain bound to the port you were receiving the multicast traffic on, and there are more processes that joined the group, you will still receive the multicast transmissions.

The idea is that joining a multicast group only tells the IP and data link layer (which in some cases explicitly tells the hardware) to accept multicast datagrams destined to that group. It is not a per-process membership, but a per-host membership.

Mapping of IP Multicast Addresses to Ethernet/FDDI addresses.

Both Ethernet and FDDI frames have a 48 bit destination address field. In order to avoid a kind of multicast ARP to map multicast IP addresses to ethernet/FDDI ones, the IANA reserved a range of addresses for multicast: every ethernet/FDDI frame with its destination in the range 01-00-5e-00-00-00 to 01-00-5e-ff-ff-ff (hex) contains data for a multicast group. The prefix 01-00-5e identifies the frame as multicast, the next bit is always 0 and so only 23 bits are left to the multicast address. As IP multicast groups are 28 bits long, the mapping can not be one-to-one. Only the 23 least significant bits of the IP multicast group are placed in the frame. The remaining 5 high-order bits are ignored, resulting in 32 different multicast groups being mapped to the same ethernet/FDDI address. This means that the ethernet layer acts as an imperfect filter, and the IP layer will have to decide whether to accept the datagrams the data-link layer passed to it. The IP layer acts as a definitive perfect filter.

Full details on IP Multicasting over FDDI are given in RFC 1390: "*Transmission of IP and ARP over FDDI*

Networks". For more information on mapping IP Multicast addresses to ethernet ones, you may consult `draft-ietf-mboned-intro-multicast-03.txt`: "*Introduction to IP Multicast Routing*".

If you are interested in IP Multicasting over Token-Ring Local Area Networks, see RFC 1469 for details.

3. Kernel requirements and configuration.

Linux is, of course (you doubted it?), full Level-2 Multicast-Compliant. It meets all requirements to send, receive and act as a router (mrouter) for multicast datagrams.

If you want just to send and receive, you must say yes to "*IP: multicasting*" when configuring your kernel. If you also want your Linux box to act as a multicast router (mrouter) you also need to enable multicast routing in the kernel by selecting "*IP: forwarding/gateweying*", "*IP: multicast routing*" and "*IP: tunneling*", the latter because new versions of `mrouted` relay on IP tunneling to send multicast datagrams encapsulated into unicast ones. This is necessary when establishing tunnels between multicast hosts separated by unicast-only networks and routers. (The `mrouted` is a daemon that implements the multicast routing algorithm -the routing policy- and instructs the kernel on how to route multicast datagrams).

Some kernel versions label multicast routing as "*EXPERIMENTAL*", so you should enable "*Prompt for development and/or incomplete code/drivers*" in the "*Code maturity level options*" section.

If, when running the `mrouted`, traffic generated in the same network your Linux box is connected to is correctly forwarded to the other network, but you can't see the other's network traffic on your local network, check whether you are receiving ICMP protocol error messages. Almost sure you forgot to turn on IP tunneling in your Linux router. It's a kind of stupid error when you know it but, believe me, its quite time-consuming when you don't, and there is no apparent reason that explains what is going wrong. A sniffer proves to be quite useful in these situations!

(You can see more on multicast routing on section Routing Policies and Forwarding Techniques; `mrouted` and tunnels are also explained in sections The MBone and Multicast applications).

Once you have compiled and installed your new kernel, you should provide a default route for multicast traffic. The goal is to add a route to the network 224.0.0.0.

The problem most people seem to face in this stage of the configuration is with the value of the mask to supply. If you have read Terry Dawson's excellent NET-3-HOWTO, it should not be difficult to guess the correct value, though. As explained there, the netmask is a 32 bit number filled with all-1s in the network part of your IP address, and with all-0s in the host part. Recall from section 2.1 that a class D multicast address has no netwok/host sections. Instead it has a 28-bit group identifier and a 4-bit class D identifier. Well, this 4 bits are the network part and the remaining 28 the host part. So the netmask needed is 11110000000000000000000000000000 or, easier to read: 240.0.0.0. Then, the full command should be:

```
route add 224.0.0.0 netmask 240.0.0.0 dev eth0
```

Depending on how old your `route` program is, you might need to add the `-net` flag after the `add`.

Here we supposed that `eth0` was multicast-capable and that, when not otherwise specified, we wanted

multicast traffic to be output there. If this is not your case, change the `dev` parameter as appropriate.

The `/proc` filesystem proves here to be useful once again: you can check `/proc/net/igmp` to see the groups your host is currently subscribed to.

---

4. The MBone.

Using a new technology usually carries some advantages and disadvantages. The advantages of multicast are -I think- clear. The main disadvantage is that hundreds of hosts and, specially, routers don't support it yet. As a consequence, people who started working on multicast, bought new equipment, modified their operating systems, and built *multicast islands* in their local places. Then they discovered that it was difficult to communicate with people doing similar things because if only one of the routers between them didn't support multicast there was nothing to do...

The solution was clear: they decided to build a *virtual multicast network* in the top of the Internet. That is: sites with multicast routers between them could communicate directly. But sites joined across unicast routers would send their island's multicast traffic encapsulated in unicast packets to other multicast islands. Routers in the middle would not have problems, as they would be dealing with unicast traffic. Finally, in the receiving site, traffic would be de-encapsulated, and sent to the island in the original multicast way. Two ends converting from multicast to unicast, and then again to multicast define what is called a multicast *tunnel*.

The *MBone* or *Multicast Backbone* is that virtual multicast network based on multicast islands connected by multicast tunnels.

Several activities take place in the MBone daily, but it deserves to be remarked the profusion of tele-conferences with real time audio and video taking place across the whole Internet. As an example, it was recently transmitted (live) the talk Linus Torvalds gave to the Silicon Valley Linux Users Group.

For more information on the MBone, see:

http://www.mediadesign.co.at/newmedia/more/mbone-faq.html

5. Multicast applications.

Most people dealing with multicast, sooner or later decide to connect to the MBone, and then they usually need an `mrouted`. You'll also need it if you don't have a multicast-capable router and you want multicast traffic generated in one of your subnets to be "heard" on another. `mrouted` does circunvect the problem of sending multicast traffic across unicast routers -it encapsulates multicast datagrams into unicast ones (IP into IP)- but this is not the only feature it provides. Most important, it instructs the kernel on how to route (or not-to-route) multicast datagrams based on their source and destination. So, even having a multicast capable router, `mrouted` can be used to tell it *what* to do with the datagrams (note I said *what*, and not *how*; `mrouted` says "forward this to the network connected to that interface", but actual forwarding is performed by the kernel). This distinction between actual-forwarding and the algorithm that decides who and how to forward is very useful as it allows to write forwarding code only once and place it into the kernel. Forwarding algorithms and policies are then implemented in user space daemons, so it is very easy to change from one policy to another without the need of kernel re-compilation.

You can get a version of *mrouted* ported to Linux from:

ftp://www.video.ja.net/mice/mrouted/Linux/. This site is mirrored all across the world. Be sure to read the ftp://www.video.ja.net/mice/README.mirrors file to choose the one nearest you.

Next, we'll focus specially on multicast applications written to connect to the MBone, which have been ported to Linux. The list is picked up from Michael Esler's "Linux Multicast Information" page http://www.cs.virginia.edu/~mke2e/multicast/. I recommend you that page for lots of information and resources on multicast and Linux.

Audio Conferencing

- NeVoT - Network Voice Terminal http://www.fokus.gmd.de/step/nevot
- RAT - UCL Robust-Audio Tool http://www-mice.cs.ucl.ac.uk/mice/rat
- vat - LBL visual audio tool http://www-nrg.ee.lbl.gov/vat/

Video Conferencing

- ivs - Inria video conferencing system http://www.inria.fr/rodeo/ivs.html
- nv - Network video tool ftp://ftp.parc.xerox.com/pub/net-research/
- nv w/ Meteor - Release of nv w/ support for the Matrox Meteor (UVa) ftp://ftp.cs.virginia.edu/pub/gwtts/Linux/nv-meteor.tar.gz
- vic - LBL video conferencing tool http://www-nrg.ee.lbl.gov/vic/
- vic w/ Meteor - Release of vic w/ support for the Matrox Meteor (UVa) ftp://ftp.cs.virginia.edu/pub/gwtts/Linux/vic2.7a38-meteor.tar.gz

Other Utilities

- mmphone Multimedia phone service http://www.eit.com/software/mmphone/phoneform.html
- wb - LBL shared white board http://www-nrg.ee.lbl.gov/wb/
- webcast - Reliable multicast application for linking Mosaic browsers http://www.ncsa.uiuc.edu/SDG/Software/XMosaic/CCI/webcast.html

Session Tools

I placed session tools later because I think they deserve some explanation. When a conference takes places, several multicast groups and ports are assigned to each service you want for your conference (audio, video, shared white-boards, etc...) Announces of the conferences that will take place, along with information on multicast groups, ports and programs that will be used (vic, vat, ...) are periodically multicasted to the MBone. Session tools "hear" this information and present you in an easy way which conferences are taking (or will take) place, so you can decide which interest you. Also, they facilitate the task of joining a session. Instead of launching each program that will be used and telling which multicast group/port to join, you usually just need to click and the session tool launches the proper programs suppling them all information needed to join the conference. Session tools usually let you announce your own conferences on the MBone.

- gwTTS - University of Virginia tele-tutoring system http://www.cs.Virginia.EDU/~gwtts
- isc - Integrated session controller http://www.fokus.gmd.de/step/isc
- mmcc - Multimedia conference control ftp://ftp.isi.edu/confctrl/mmcc
- sd - LBL session directory tool ftp://ftp.ee.lbl.gov/conferencing/sd
- sd-snoop - Tenet Group session directory snoop utility ftp://tenet.berkeley.edu/pub/software
- sdr - UCL's next generation session directory ftp://cs.ucl.ac.uk/mice/sdr

---

6. Multicast programming.

Multicast programming... or writing your own multicast applications.

Several extensions to the programming API are needed in order to support multicast. All of them are handled via two system calls: `setsockopt()` (used to pass information to the kernel) and `getsockopt()` (to retrieve information regarded multicast behavior). This does *not* mean that 2 new system calls were added to support multicast. The pair `setsockopt()`/`getsockopt()` has been there for years. Since 4.2 BSD at least. The addition consists on a new set of options (multicast options) that are passed to these system calls, that the kernel must understand.

The following are the `setsockopt()`/`getsockopt()` function prototypes:

```
int getsockopt(int s, int level, int optname, void* optval, int* optlen);

int setsockopt(int s, int level, int optname, const void* optval, int optlen);
```

The first parameter, `s`, is the socket the system call applies to. For multicasting, it must be a socket of the family `AF_INET` and its type may be either `SOCK_DGRAM` or `SOCK_RAW`. The most common use is with `SOCK_DGRAM` sockets, but if you plan to write a routing daemon or modify some existing one, you will probably need to use `SOCK_RAW` ones.

The second one, `level`, identifies the layer that is to handle the option, message or query, whatever you want to call it. So, `SOL_SOCKET` is for the socket layer, `IPPROTO_IP` for the IP layer, etc... For multicast programming, `level` will always be `IPPROTO_IP`.

`optname` identifies the option we are setting/getting. Its value (either supplied by the program or returned by the kernel) is `optval`. The optnames involved in multicast programming are the following:

	setsockopt()	getsockopt()
IP_MULTICAST_LOOP	yes	yes
IP_MULTICAST_TTL	yes	yes
IP_MULTICAST_IF	yes	yes
IP_ADD_MEMBERSHIP	yes	no
IP_DROP_MEMBERSHIP	yes	no

`optlen` carries the size of the data structure `optval` points to. Note that in `getsockopt()` it is a value-result rather than a value: the kernel writes the value of `optname` in the buffer pointed by `optval` and informs us of that value's size via `optlen`.

Both `setsockopt()` and `getsockopt()` return 0 on success and -1 on error.

6.1 IP_MULTICAST_LOOP.

You have to decide, as the application writer, whether you want the data you send to be looped back to your host or not. If you plan to have more than one process or user "listening", loopback must be enabled. On the other hand, if you are sending the images your video camera is producing, you probably don't want loopback, even if you want to see yourself on the screen. In that latter case, your application will probably receive the images from a device attached to the computer and send them to the socket. As the application already "has" that data, it is improbable it wants to receive it again on the socket. Loopback is by default enabled.

TOC

Regard that `optval` is a pointer. You can't write:

```
setsockopt(socket, IPPROTO_IP, IP_MULTICAST_LOOP, 0, 1);
```

to disable loopback. Instead write:

```
u_char loop;
setsockopt(socket, IPPROTO_IP, IP_MULTICAST_LOOP, &loop, sizeof(loop));
```

and set `loop` to 1 to enable loopback or 0 to disable it.

To know whether a socket is currently looping-back or not use something like:

```
u_char loop;
int size;

getsockopt(socket, IPPROTO_IP, IP_MULTICAST_LOOP, &loop, &size)
```

6.2 IP_MULTICAST_TTL.

If not otherwise specified, multicast datagrams are sent with a default value of 1, to prevent them to be forwarded beyond the local network. To change the TTL to the value you desire (from 0 to 255), put that value into a variable (here I name it "ttl") and write somewhere in your program:

```
u_char ttl;
setsockopt(socket, IPPROTO_IP, IP_MULTICAST_TTL, &ttl, sizeof(ttl));
```

The behavior with `getsockopt()` is similar to the one seen on IP_MULTICAST_LOOP.

6.3 IP_MULTICAST_IF.

Usually, the system administrator specifies the default interface multicast datagrams should be sent from. The programmer can override this and choose a concrete outgoing interface for a given socket with this option.

```
struct in_addr interface_addr;
setsockopt (socket, IPPROTO_IP, IP_MULTICAST_IF, &interface_addr,
sizeof(interface_addr));
```

>From now on, all multicast traffic generated in this socket will be output from the interface chosen. To revert to the original behavior and let the kernel choose the outgoing interface based on the system administrator's configuration, it is enough to call `setsockopt()` with this same option and `INADDR_ANY` in the interface field.

In determining or selecting outgoing interfaces, the following `ioctl`s might be useful: `SIOCGIFADDR` (to get an interface's address), `SIOCGIFCONF` (to get the list of all the interfaces) and `SIOCGIFFLAGS` (to get an interface's flags and, thus, determine whether the interface is multicast capable or not -the `IFF_MULTICAST` flag-).

If the host has more than one interface and the IP_MULTICAST_IF option is not set, multicast transmissions are sent from the default interface, although the remainding interfaces might be used for multicast *forwarding* if the host is acting as a multicast router.

TOC

6.4 IP_ADD_MEMBERSHIP.

Recall that you need to tell the kernel which multicast groups you are interested in. If no process is interested in a group, packets destined to it that arrive to the host are discarded. In order to inform the kernel of your interests and, thus, become a member of that group, you should first fill a `ip_mreq` structure which is passed later to the kernel in the `optval` field of the `setsockopt()` system call.

The ip_mreq structure (taken from `/usr/include/linux/in.h`) has the following members:

```
struct ip_mreq
{
        struct in_addr imr_multiaddr;   /* IP multicast address of group */
        struct in_addr imr_interface;   /* local IP address of interface */
};
```

(Note: the "physical" definition of the structure is in the file above specified. Nonetheless, you should not include `<linux/in.h>` if you want your code to be portable. Instead, include `<netinet/in.h>` which, in turn, includes `<linux/in.h>` itself).

The first member, `imr_multiaddr`, holds the group address you want to join. Remember that memberships are also associated with interfaces, not just groups. This is the reason you have to provide a value for the second member: `imr_interface`. This way, if you are in a multihomed host, you can join the same group in several interfaces. You can always fill this last member with the wildcard address (`INADDR_ANY`) and then the kernel will deal with the task of choosing the interface.

With this structure filled (say you defined it as: `struct ip_mreq mreq;`) you just have to call `setsockopt()` this way:

```
setsockopt (socket, IPPROTO_IP, IP_ADD_MEMBERSHIP, &mreq, sizeof(mreq));
```

Notice that you can join several groups to the same socket, not just one. The limit to this is `IP_MAX_MEMBERSHIPS` and, as of version 2.0.33, it has the value of 20.

6.5 IP_DROP_MEMBERSHIP.

The process is quite similar to joining a group:

```
struct ip_mreq mreq;
setsockopt (socket, IPPROTO_IP, IP_DROP_MEMBERSHIP, &mreq, sizeof(mreq));
```

where `mreq` is the same structure with the same data used when joining the group. If the `imr_interface` member is filled with `INADDR_ANY`, the first matching group is dropped.

If you have joined a lot of groups to the same socket, you don't need to drop memberships in all of them in order to terminate. When you close a socket, all memberships associated with it are dropped by the kernel. The same occurs if the process that opened the socket is killed.

Finally, keep in mind that a process dropping membership for a group does not imply that the host will stop receiving datagrams for that group. If another socket joined that group in that same interface previously to this `IP_DROP_MEMBERSHIP`, *the host* will keep being a member of that group.

Both ADD_MEMBERSHIP and DROP_MEMBERSHIP are nonblocking operations. They should return immediately indicating either success or failure.

---

7. The internals.

This section's aim is to provide some information, not needed to reach a basic understanding on how multicast works nor to be able to write multicast programs, but which is very interesting, gives some insight on the underlying multicast protocols and implementations, and may be useful to avoid common errors and misunderstandings.

7.1 IGMP.

When talking about `IP_ADD_MEMBERSHIP` and `IP_DROP_MEMBERSHIP`, we said that the information provided by this "commands" was used by the kernel to choose which multicast datagrams accept or discard. This is true, but it is not all the truth. Such a simplification would imply that multicast datagrams for *all* multicast groups around the world would be received by our host, and then it would check the memberships issued by processes running on it to decide whether to pass the traffic to them or to throw it out. As you can imagine, this is a complete bandwidth waste.

What actually happens is that hosts instruct their routers telling them which multicast groups they are interested in; then, those routers tell their up-stream routers they want to receive that traffic, and so on. Algorithms employed for making the decision of *when* to ask for a group's traffic or saying that it is not desired anymore, vary a lot. There's something, however, that never changes: *how* this information is transmitted. **IGMP** is used for that. It stands for Internet Group Management Protocol. It is a new protocol, similar in many aspects to ICMP, with a protocol number of 2, whose messages are carried in IP datagrams, and which all level 2-compliant host are required to implement.

As said before, it is used both by hosts giving membership information to its routers, and by routers to communicate between themselves. In the following I'll cover only the hosts-routers relationships, mainly because I was unable to find information describing router to router communication other than the mrouted source code (rfc 1075 describing the Distance Vector Multicast Routing Protocol is now obsoleted, and `mrouted` implements a modified DVMRP not yet documented).

IGMP version 0 is specified in RFC-988 which is now obsoleted. Almost no one uses version 0 now.

IGMP version 1 is described in RFC-1112 and, although it is updated by RFC-2236 (IGMP version 2) it is in wide use still. The Linux kernel implements the full IGMP version 1 and parts of version 2 requirements, but not all.

Now I'll try to give an informal description of the protocol. You can check RFC-2236 for an in-proof formal description, with lots of state diagrams and time-out boundaries.

All IGMP messages have the following structure:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     | Max Resp Time |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Group Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IGMP version 1 (hereinafter IGMPv1) labels the "Max Resp Time" as "Unused", zeroes it when sent, and ignores it when received. Also, it brakes the "Type" field in two 4-bits wide fields: "Version" and "Type". As IGMPv1 identifies a "Membership Query" message as 0x11 (version 1, type 1) and IGMPv2 as 0x11 too, the 8 bits have the same effective interpretation.

TOC

I think it is more instructive to give first the IGMPv1 description and next point out the IGMPv2 additions, as they a mainly that, additions.

For the following discussions it is important to remember that multicast routers receive *all* IP multicast datagrams.

IGMP version 1.

Routers periodically send *IGMP Host Membership Queries* to the all-hosts group (224.0.0.1) with a TTL of 1 (once every minute or two). All multicast-capable hosts hear them, but don't answer immediately to avoid an IGMP Host Membership Report storm. Instead, they start a random delay timer for each group they belong to *on the interface* they received the query.

Sooner or later, the timer expires in one of the hosts, and it sends an IGMP *Host Membership Report* (also with TTL 1) to the multicast address of the group being reported. As it is sent to the group, all hosts that joined the group -and which are currently waiting for their own timer to expire- receive it, too. Then, they stop their timers and don't generate any other report. Just one is generated -by the host that chose the smaller timeout-, and that is enough for the router. It only needs to know that there are members for that group in the subnet, not how many nor which.

When no reports are received for a given group after a certain number of queries, the router assumes that no members are left, and thus it doesn't have to forward traffic for that group on that subnet. Note that in IGMPv1 there are no "Leave Group messages".

When a host joins a *new* group, the kernel sends a report for that group, so that the respective process needs not to wait a minute or two until a new membership query is received. As you can see this IGMP packet is generated by the kernel as a response to the `IP_ADD_MEMBERSHIP` command, seen in section IP_ADD_MEMBERSHIP. Note the emphasis in the adjective "new": if a process issues an `IP_ADD_MEMBERSHIP` command for a group the host is already a member of, no IGMP packets are sent as we must already be receiving traffic for that group; instead, a counter for that group's use is incremented. `IP_DROP_MEMBERSHIP` generates no datagrams in IGMPv1.

Host Membership Queries are identified by Type 0x11, and Host Membership Reports by Type 0x12.

No reports are sent for the all-hosts group. Membership in this group is permanent.

IGMP version 2.

One important addition to the above is the inclusion of a *Leave Group* message (Type 0x17). The reason is to reduce the bandwidth waste between the time the last host in the subnet drops membership and the time the router times-out for its queries and decides there are no more members present for that group (leave latency). Leave Group messages should be addressed to the all-routers group (224.0.0.2) rather than to the group being left, as that information is of no use for other members (kernel versions up to 2.0.33 send them to the group; although it does no harm to the hosts, it's a waste of time as they have to process them, but don't gain useful information). There are certain subtle details regarding when and when-not to send Leave Messages; if interested, see the RFC.

When an IGMPv2 router receives a Leave Message for a group, it sends *Group-Specific Queries* to the group being left. This is another addition. IGMPv1 has no group-specific queries. All queries are sent to the all-hosts group. The Type in the IGMP header does not change (0x11, as before), but the "Group Address" is filled with the address of the multicast group being left.

The "Max Resp Time" field, which was set to 0 in transmission and ignored on reception in IGMPv1, is meaningful only in "Membership Query" messages. It gives the maximum time allowed before sending a report in units of 1/10 second. It is used as a tune mechanism.

IGMPv2 adds another message type: 0x16. It is a "Version 2 Membership Report" sent by IGMPv2 hosts if they detect an IGMPv2 router is present (an IGMPv2 host knows an IGMPv1 router is present when it receives a query with the "Max Response" field set to 0).

When more than one router claims to act as querier, IGMPv2 provides a mechanism to avoid "discussions": the router with

TOC

the lowest IP address is designed to be querier. The other routers keep timeouts. If the router with lower IP address c or is shutdown, the decision of who will be the querier is taken again after the timers expire.

7.2 Kernel corner.

This sub-section gives some start-points to study the multicast implementation of the Linux kernel. It does not explain that implementation. It just says where to find things.

The study was carried over version 2.0.32, so it could be a bit outdated by the time you read it (network code seems to have changed *A LOT* in 2.1.x releases, for instance).

Multicast code in the Linux kernel is always surrounded by `#ifdef CONFIG_IP_MULTICAST` / `#endif` pairs, so that you can include/ exclude it from your kernel based on your needs (this inclusion/exclusion is done at compile time, as you probably know if reading that section... `#ifdef`s are handled by the preprocessor. The decision is made based in what you selected when doing either a `make config`, `make menuconfig` or `make xconfig`).

You might want multicast features, but if your Linux box is not going to act as a multicast router you will probably not want multicast router features included in your new kernel. For this you have the multicast routing code surrounded by `#ifdef CONFIG_IP_MROUTE` / `#endif` pairs.

Kernel sources are usually placed in /usr/src/linux. However, the place may change so, both for accuracy and brevity, I will refer to the root directory of the kernel sources as just LINUX. Then, something like `LINUX/net/ipv4/udp.c` should be the same as `/usr/src/linux/net/ipv4/udp.c` if you unpacked the kernel sources in the `/usr/src/linux` directory.

All multicast interfaces with user programs shown in the section devoted to multicast programming were driven across the `setsockopt()`/ `getsockopt()` system calls. Both of them are implemented by means of functions that make some tests to verify the parameters passed to them and which, in turn, call another function that makes some additional tests, demultiplexes the call based on the `level` parameter to either system call, and then calls another function which... (if interested in all this jumps, you can follow them in `LINUX/net/socket.c` (functions `sys_socketcall()` and `sys_setsockopt()`, `LINUX/net/ipv4/af_inet.c` (function `inet_setsockopt()`) and `LINUX/net/ipv4/ip_sockglue.c` (function `ip_setsockopt()`)).

The one which interests us is `LINUX/net/ipv4/ip_sockglue.c`. Here we find `ip_setsockopt()` and `ip_getsockopt()` which are mainly a `switch` (after some error checking) verifying each possible value for `optname`. Along with unicast options, all multicast ones seen here are handled: `IP_MULTICAST_TTL`, `IP_MULTICAST_LOOP`, `IP_MULTICAST_IF`, `IP_ADD_MEMBERSHIP` and `IP_DROP_MEMBERSHIP`. Previously to the `switch`, a test is made to determine whether the options are multicast router specific, and if so, they are routed to the `ip_mroute_setsockopt()` and `ip_mroute_getsockopt()` functions (file `LINUX/net/ipv4/ipmr.c`).

In `LINUX/net/ipv4/af_inet.c` we can see the default values we talked about in previous sections (loopback enabled, TTL=1) provided when the socket is created (taken from function `inet_create()` in this file):

```
#ifdef CONFIG_IP_MULTICAST
        sk->ip_mc_loop=1;
        sk->ip_mc_ttl=1;
        *sk->ip_mc_name=0;
        sk->ip_mc_list=NULL;
#endif
```

Also, the assertion of "closing a socket makes the kernel drop all memberships this socket had" is corroborated by:

```
#ifdef CONFIG_IP_MULTICAST
                /* Applications forget to leave groups before exiting */
                ip_mc_drop_socket(sk);
#endif
```

taken from `inet_release()`, on the same file as before.

Device independent operations for the Link Layer are kept in `LINUX/net/core/dev_mcast.c`.

Two important functions are still missing: the processing of input and output multicast datagrams. As any other datagrams, incoming datagrams are passed from the device drivers to the `ip_rcv()` function (`LINUX/net/ipv4/ip_input.c`). In this function is where the perfect filtering is applied to multicast packets that crossed the devices layer (recall that lower layers only perform best-effort filtering and is IP who 100% knows whether we are interested in that multicast group or not). If the host is acting as a multicast router, this function decides too whether the datagram should be forwarded and calls `ipmr_forward()` appropriately. (`ipmr_forward()` is implemented in `LINUX/net/ipv4/ipmr.c`).

Code in charge of out-putting packets is kept in `LINUX/net/ipv4/ip_output.c`. Here is where the `IP_MULTICAST_LOOP` option takes effect, as it is checked to see whether to loop back the packets or not (function `ip_queue_xmit()`). Also the TTL of the outgoing packet is selected based on whether it is a multicast or unicast one. In the former case, the argument passed to the `IP_MULTICAST_TTL` option is used (function (`ip_build_xmit()`).

While working with `mrouted` (a program which gives the kernel information about how to route multicast datagrams), we detected that all multicast packets originated on the local network were properly routed..., except the ones from the Linux box that was acting as the multicast router!! ip_input.c was working OK, but it seemed ip_output.c wasn't. Reading the source code for the output functions, we found that outgoing datagrams were not being passed to `ipmr_forward()`, the function that had to decide whether they should be routed or not. The packets were outputed to the local network but, as network cards are usually unable to read their own transmissions, those datagrams were never routed. We added the necessary code to the `ip_build_xmit()` function and everything was OK again. (Having the sources for your kernel is not a luxury or pedantry; it's a need!)

`ipmr_forward()` has been mentioned a couple of times. It is an important function as it solves one important misunderstanding that appears to be widely expanded. When routing multicast traffic, it is *not* `mrouted` who makes the copies and sends them to the proper recipients. `mrouted` receives all multicast traffic and, based on that information, computes the multicast routing tables and *tells the kernel* how to route: "datagrams for this group coming from that interface should be forwarded to those interfaces". This information is passed to the kernel by calls to `setsockopt()` on a raw socket opened by the `mrouted` daemon (the protocol specified when the raw socket was created *must* be `IPPROTO_IGMP`). This options are handled in the `ip_mroute_setsockopt()` function from `LINUX/net/ipv4/ipmr.c`. The first option (would be better to call them commands rather than options) issued on that socket must be `MRT_INIT`. All other commands are ignored (returning `-EACCES`) if `MRT_INIT` is not issued first. Only one instance of `mrouted` can be running at the same time in the same host. To keep track of this, when the first `MRT_INIT` is received, an important variable, `struct sock* mroute_socket`, is pointed to the socket `MRT_INIT` was received on. If `mroute_socket` is not null when attending an `MRT_INIT` this means another mrouted is already running and `-EADDRINUSE` is returned. All resting commands (`MRT_DONE`, `MRT_ADD_VIF`, `MRT_DEL_VIF`, `MRT_ADD_MFC`, `MRT_DEL_MFC` and `MRT_ASSERT`) return `-EACCES` if they come from a socket different than `mroute_socket`.

As routed multicast datagrams can be received/sent across either physical interfaces or tunnels, a common abstraction for both was devised: VIFs, Virtual InterFaces. `mrouted` passes vif structures to the kernel, indicating physical or tunnel interfaces to add to its routing tables, and multicast forwarding entries saying where to forward datagrams.

VIFs are added with `MRT_ADD_VIF` and deleted with `MRT_DEL_VIF`. Both pass a `struct vifctl` to the kernel (defined in `/usr/include/linux/mroute.h`) with the following information:

```
struct vifctl {
```

TOC

```
        vifi_t  vifc_vifi;              /* Index of VIF */
        unsigned char vifc_flags;       /* VIFF_ flags */
        unsigned char vifc_threshold;   /* ttl limit */
        unsigned int vifc_rate_limit;   /* Rate limiter values (NI) */
        struct in_addr vifc_lcl_addr;   /* Our address */
        struct in_addr vifc_rmt_addr;   /* IPIP tunnel addr */
};
```

With this information a `vif_device` structure is built:

```
struct vif_device
{
        struct device   *dev;                   /* Device we are using */
        struct route    *rt_cache;              /* Tunnel route cache */
        unsigned long   bytes_in,bytes_out;
        unsigned long   pkt_in,pkt_out;         /* Statistics */
        unsigned long   rate_limit;             /* Traffic shaping (NI) */
        unsigned char   threshold;              /* TTL threshold */
        unsigned short  flags;                  /* Control flags */
        unsigned long   local,remote;           /* Addresses(remote for tunnels)*/
};
```

Note the `dev` entry in the structure. The `device` structure is defined in `/usr/include/linux/netdevice.h` file. It is a big structure, but the field that interests us is:

```
  struct ip_mc_list*    ip_mc_list;   /* IP multicast filter chain    */
```

The `ip_mc_list` structure -defined in `/usr/include/linux/igmp.h`- is as follows:

```
struct ip_mc_list
{
        struct device *interface;
        unsigned long multiaddr;
        struct ip_mc_list *next;
        struct timer_list timer;
        short tm_running;
        short reporter;
        int users;
};
```

So, the `ip_mc_list` member from the `dev` structure is a pointer to a linked list of `ip_mc_list` structures, each containing an entry for each multicast group the network interface is a member of. Here again we see membership is associated to interfaces. `LINUX/net/ipv4/ip_input.c` traverses this linked list to decide whether the received datagram is destined to any group the interface that received the datagram belongs to:

```
#ifdef CONFIG_IP_MULTICAST
                if(!(dev->flags&IFF_ALLMULTI) && brd==IS_MULTICAST
```

```
                && iph->daddr!=IGMP_ALL_HOSTS
                && !(dev->flags&IFF_LOOPBACK))
        {
                /*
                 *      Check it is for one of our groups
                 */
                struct ip_mc_list *ip_mc=dev->ip_mc_list;
                do
                {
                        if(ip_mc==NULL)
                        {
                                kfree_skb(skb, FREE_WRITE);
                                return 0;
                        }
                        if(ip_mc->multiaddr==iph->daddr)
                                break;
                        ip_mc=ip_mc->next;
                }
                while(1);
        }
#endif
```

The `users` field in the `ip_mc_list` structure is used to implement what was said in section IGMP version 1: if a process joins a group and the interface is already a member of that group (ie, another process joined that same group in that same interface before) only the count of members (`users`) is incremented. No IGMP messages are sent, as you can see in the following code (taken from `ip_mc_inc_group()`, called by `ip_mc_join_group()`, both in `LINUX/net/ipv4/igmp.c`):

```
        for(i=dev->ip_mc_list;i!=NULL;i=i->next)
        {
                if(i->multiaddr==addr)
                {
                        i->users++;
                        return;
                }
        }
```

When dropping memberships, the counter is decremented and additional operations are performed only when the count reaches 0 (`ip_mc_dec_group()`).

`MRT_ADD_MFC` and `MRT_DEL_MFC` set or delete forwarding entries in the multicast routing tables. Both pass a `struct mfcctl` to the kernel (also defined in `/usr/include/linux/mroute.h`) with this information:

```
struct mfcctl
{
        struct in_addr mfcc_origin;             /* Origin of mcast      */
        struct in_addr mfcc_mcastgrp;           /* Group in question    */
        vifi_t  mfcc_parent;                    /* Where it arrived     */
        unsigned char mfcc_ttls[MAXVIFS];       /* Where it is going    */
};
```

With all this information in hand, `ipmr_forward()` "walks" across the VIFs, and if a matching is found *it* duplicates the datagram and calls `ipmr_queue_xmit()` which, in turn, uses the output device specified by the routing table and the proper destination address if the packet is to be sent across a tunnel (ie, the unicast destination address of the other end of the tunnel).

Function `ip_rt_event()` (not directly related to output, but which is in ip_output.c too) receives events related to a network device, like the device going up. This function assures that then the device joins the ALL-HOSTS multicast group.

IGMP functions are implemented in `LINUX/net/ipv4/igmp.c`. Important information for that functions appears in `/usr/include/linux/igmp.h` and `/usr/include/linux/mroute.h`. The IGMP entry in the `/proc/net` directory is created with `ip_init()` in `LINUX/net/ipv4/ip_output.c`.

---

8. Routing Policies and Forwarding Techniques.

One trivial algorithm to make worldwide multicast traffic available everywhere could be to send it... everywhere, despite someone wants it or not. As this does not seem quite optimized, several routing algorithms and forwarding techniques have been implemented.

DVMRP (Distance Vector Multicast Routing Protocol) is, perhaps, the one most multicast routers use now. It is a *dense mode* routing protocol, that is, it performs well in environments with high bandwidth and densely distributed members. However, in *sparse mode* scenarios, it suffers from scalability problems.

Together with DVMRP we can find other dense mode routing protocols, such as **MOSPF** (Multicast Extensions to OSPF -Open Shortest Path First-) and PIM-DM (Protocol-Independent Multicast Dense Mode).

To perform routing in sparse mode environments, we have **PIM-SM** (Protocol Independent Multicast Sparse Mode) and **CBT** (Core Based Trees).

OSPF version 2 is explained in RFC 1583, and MOSPF in RFC 1584. PIM-SM and CBT specifications can be found in RFC 2117 and 2201, respectively.

All this routing protocols use some type of multicast forwarding, such as *flooding*, *Reverse Path Broadcasting* (RPB), *Truncated Reverse Path Broadcasting* (TRPB), *Reverse Path Multicasting* (RPM) or *Shared Trees*.

It would be too long to explain them here and, as short descriptions for them are publicly available, I'll just recommend reading the `draft-ietf-mboned-in.txt` text. You can find it in the same places RFCs are available, and it explains in some detail all the above techniques and policies.

9. Multicast Transport Protocols.

So far we have been talking about multicast transmissions using UDP. This is the usual practice, as it is impossible to do it with TCP. However, intense research is taking place since a couple of years in order to develop some new multicast transport protocols.

Several of these protocols have been implemented and are being tested. A good lesson from them is that it seems no multicast transport protocol is general and good enough for all types of multicast applications.

If transport protocols are complex and difficult to tune, imagine dealing with delays (in multimedia conferences), data loss, ordering, retransmissions, flow and congestion control, group management, etc, when the receiver is not one, but perhaps hundreds or thousands of sparse hosts. Here scalability is an issue, and new techniches are implemented, such as not giving acknowledges for every packet received but, instead, send *negative acknowledges* (NACKs) for data not received. RFC 1458 gives the proposed requirements for multicast protocols.

Giving descriptions of those multicast protocols is out of the scope of this section. Instead, I'll give you the names of some of them and point you to some sources of information: Real-Time Transport Protocol (RTP) is concerned with multi-partite multimedia conferences, **Scalable Reliable Multicast** (SRM) is used by the `wb` (the distributed White-Board tool, see section Multicast applications), **Uniform Reliable Group Communication Protocol** (URGC) enforces reliable and ordered transactions based in a centralized control, **Muse** was developed as an application specific protocol: to multicast news articles over the MBone, the **Multicast File Transfer Protocol** (MFTP) is quite descriptive by itself and people "join" to file transmission (previously announced) much in the same way they would join a conference, **Log-Based Receiver-reliable Multicast** (LBRM) is a curious protocol that keeps track of all packets sent in a logging server that tells the sender whether it has to retransmit the data or can drop it safely as all receivers got it. One protocol with a funny name -especially for a multicast protocol- is STORM (**STructure-Oriented Resilient Multicast**). Lots and lots of multicast protocols can be found searching the Web, along with some interesting papers proposing new activities for multicast (for instance, www page distribution using multicast).

A good page providing comparisons between reliable multicast protocols is

http://www.tascnets.com/mist/doc/mcpCompare.html.

A very good and up-to-date site, with lots of interesting links (Internet drafts, RFCs, papers, links to other sites) is:

http://research.ivv.nasa.gov/RMP/links.html.

http://hill.lut.ac.uk/DS-Archive/MTP.html is also a good source of information on the subject.

Katia Obraczka's "*Multicast Transport Protocols: A Survey and Taxonomy*" article gives short descriptions for each protocol and tries to classify them according to different features. You can read it in

the IEEE Communications magazine, January 1998, vol. 36, No. 1.

10. References.

10.1 RFCs.

- RFC 1112 "Host Extensions for IP Multicasting". Steve Deering. August 1989.
- RFC 2236 "Internet Group Management Protocol, version 2". W. Fenner. November 1997.
- RFC 1458 "Requirements for Multicast Protocols". Braudes, R and Zabele, S. May 1993.
- RFC 1469 "IP Multicast over Token-Ring Local Area Networks". T. Pusateri. June 1993.
- RFC 1390 "Transmission of IP and ARP over FDDI Networks". D. Katz. January 1993.
- RFC 1583 "OSPF Version 2". John Moy. March 1994.
- RFC 1584 "Multicast Extensions to OSPF". John Moy. March 1994.
- RFC 1585 "MOSPF: Analysis and Experience". John Moy. March 1994.
- RFC 1812 "Requirements for IP version 4 Routers". Fred Baker, Editor. June 1995
- RFC 2117 "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification". D. Estrin, D. Farinacci, A. Helmy, D. Thaler; S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. July 1997.
- RFC 2189 "Core Based Trees (CBT version 2) Multicast Routing". A. Ballardie. September 1997.
- RFC 2201 "Core Based Trees (CBT) Multicast Routing Architecture". A. Ballardie. September 1997.

10.2 Internet Drafts.

- "Introduction to IP Multicast Routing". `draft-ietf-mboned-intro- multicast-03.txt`. T. Maufer, C. Semeria. July 1997.
- "Administratively Scoped IP Multicast". `draft-ietf-mboned-admin-ip-space-03.txt`. D. Meyer. June 10, 1997.

10.3 Web pages.

- Linux Multicast Homepage. http://www.cs.virginia.edu/~mke2e/multicast.html
- Linux Multicast FAQ. http://andrew.triumf.ca/pub/linux/multicast-FAQ
- Multicast and MBONE on Linux. http://www.teksouth.com/linux/multicast/
- Christian Daudt's MBONE-Linux Page. http://www.microplex.com/~csd/linux/mbone.html
- Reliable Multicast Links http://research.ivv.nasa.gov/RMP/links.html
- Multicast Transport Protocols http://hill.lut.ac.uk/DS-Archive/MTP.html

10.4 Books.

- "TCP/IP Illustrated: Volume 1 The Protocols". Stevens, W. Richard. Addison Wesley Publishing Company, Reading MA, 1994
- "TCP/IP Illustrated: Volume 2, The Implementation". Wright, Gary and W. Richard Stevens. Addison Wesley Publishing Company, Reading MA, 1995
- "UNIX Network Programming Volume 1. Networking APIs: Sockets and XTI". Stevens, W. Richard. Second Edition, Prentice Hall, Inc. 1998.
- "Internetworking with TCP/IP Volume 1 Principles, Protocols, and Architecture". Comer, Douglas E. Second Edition, Prentice Hall, Inc. Englewood Cliffs, New Jersey, 1991

11. Copyright and Disclaimer.

Copyright 1998 Juan-Mariano de Goyeneche.

This HOWTO is free documentation; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details.

You can obtain a copy of the GNU General Public License by writing to the Free Software Foundation, 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

If you publish this document on a CD-ROM or in hardcopy form, a complimentary copy would be appreciated; mail me for my postal address. Also consider making a donation to the Linux Documentation Project or the Free Software Foundation to help support free documentation for GNU/Linux. Contact the Linux HOWTO coordinator, Tim Bynum linux-howto@sunsite.unc.edu, for more information.

12. Acknowledgements.

This is the best opportunity I've ever had to thank so *many* people I feel grateful to. So, I'm afraid this is going to be a large section... It is, in any case, the most important one of this paper (for me, at least...).

First, I want to thank Elena Apolinario Fernández de Sousa (yes, Elena is the first name; the REST is THE surname ;-)). I tried to reflect in this Howto all the knowledge I collected while working with her in connecting our Department to the MBone and debugging problems with locally generated CSCW software across multicast tunnels. She was of invaluable help in finding and correcting network problems, discovering and fixing kernel bugs that puzzled us for days, ... and keeping the sense of humor alive while problems appeared and appeared, but solutions didn't. She also read and corrected the drafts for this document and provided important ideas and suggestions. If this howto is here and is usefull for somebody, it will be, in many aspects, thanks to her. Thanks, Elena!

There is something I have been lucky enough to find all my (still-not-too-long) live, but, despite being repetitive, has never stopped amazing me. I'm talking about people that altruistically employ part of their time and/or resources to help other people learn new things; and, what is better, they enjoy doing it. This is not only (but also, too) explain things they already know, but lend their books, provide access to their sources and facilitate you the way to learn all things they know; sometimes, even more... I know quite a few of that people, and I'd like to thank them for all their help.

Pablo Basterrechea was my "first source of documentation" while I was in my pre-Internet stage. I learned assembly and advanced structured programming entirely from his books (well, the latter also from his programs...). Thanks for all, Pablo.

In my first course at the University that "primary source of documentation" moved to Pepe Mañas. He was teaching then Computer Programming there, and soon I became addict to his bookshelf. He lent me his books lots of times without asking for a minimum sign that could assure that I was going to return them back to him, not even my name! My first approach to TCP/IP was also by his hand: he lent me Comer's "Internetworking with TCP/IP, Volume 1" for the whole summer. He did not even know my name by then, but he lent me the book... That book influenced me a lot, and TCP/IP has become one of my primary fields of interest since that summer.

If there are two persons I must thank most, these are (in alphabetic order ;-)), José Manuel and Paco Moya. Nobody I asked more things more times (C, C++, Linux, security, Web, OSs, signals & systems, electronics, ... anything!) and, despite my persistence, I always got throughly and friendly responses and help. If I'm using GNU/Linux now, this is, again, thanks to them. I feel particularly lucky with friends like them. THANKS.

Iñigo Mascaraque also helped (from him I got my first System Administration book) and encouraged me in my beginnings, but never stopped reminding me that, although this was a fascinating world and an important part of my career, I should not forget the other, less-interesting, parts. (I don't forget, I$!).

As I am on the topic, I'd like to thank my parents, too. They always tried to make the best opportunities

available for me. Many thanks for all.

I also feel grateful to Joaquín Seoane, the first who trusted me enough to give me a root password in the time I was learning system administration by myself, and Santiago Pavón, the one who gave me my first opportunity here at DIT.

W. Richard Stevens' books have been a real revelation for me (it's a pity they are so expensive...). If he ever reads this paper, I'd like to thank him for them, and encourage him to keep on writing. Anything that comes out of his hands will -undoubtedly- be good for all of us.

Finally I'd like to thank Richard Stallman, Linus Torvalds, Alan Cox and all contributors to the Linux kernel and the free software in general, for giving us such a great OS.

I'm sure I'm forgetting someone here... Sorry. I'm certain they know I'm grateful to them too, so if they tell me, everybody will know it... :-)

Linux Security HOWTO

Kevin Fenzi, kevin@scrye.com & Dave Wreski, dave@nic.com

v1.0.2, 25 April 1999

This document is a general overview of security issues that face the administrator of Linux systems. It covers general security philosophy and a number of specific examples of how to better secure your Linux system from intruders. Also included are pointers to security-related material and programs. Improvements, constructive criticism, additions and corrections are gratefully accepted. Please mail your feedback to both authors, with "Security HOWTO" in the subject.

1. Introduction

- 1.1 New Versions of this Document
- 1.2 Feedback
- 1.3 Disclaimer
- 1.4 Copyright Information

2. Overview

- 2.1 Why Do We Need Security?
- 2.2 How Secure Is Secure?
- 2.3 What Are You Trying to Protect?
- 2.4 Developing A Security Policy
- 2.5 Means of Securing Your Site
- 2.6 Organization of This Document

3. Physical Security

- 3.1 Computer locks
- 3.2 BIOS Security

- 3.3 Boot Loader Security
- 3.4 xlock and vlock
- 3.5 Detecting Physical Security Compromises

4. Local Security

- 4.1 Creating New Accounts
- 4.2 Root Security

5. Files and Filesystem Security

- 5.1 Umask Settings
- 5.2 File Permissions
- 5.3 Integrity Checking with Tripwire `Tripwire`
- 5.4 Trojan Horses

6. Password Security and Encryption

- 6.1 PGP and Public-Key Cryptography
- 6.2 SSL, S-HTTP, HTTPS and S/MIME
- 6.3 Linux IPSEC Implementations
- 6.4 `ssh` (Secure Shell) and `stelnet`
- 6.5 PAM - Pluggable Authentication Modules
- 6.6 Cryptographic IP Encapsulation (CIPE)
- 6.7 Kerberos
- 6.8 Shadow Passwords.
- 6.9 "Crack" and "John the Ripper"
- 6.10 CFS - Cryptographic File System and TCFS - Transparent Cryptographic File System
- 6.11 X11, SVGA and display security

7. Kernel Security

- 7.1 2.0 Kernel Compile Options
- 7.2 2.2 Kernel Compile Options
- 7.3 Kernel Devices

TOC

8. Network Security

- 8.1 Packet Sniffers
- 8.2 System services and tcp_wrappers
- 8.3 Verify Your DNS Information
- 8.4 `identd`
- 8.5 SATAN, ISS, and Other Network Scanners
- 8.6 `sendmail`, `qmail` and MTA's
- 8.7 Denial of Service Attacks
- 8.8 NFS (Network File System) Security.
- 8.9 NIS (Network Information Service) (formerly YP).
- 8.10 Firewalls
- 8.11 IP Chains - Linux Kernel 2.2.x Firewalling
- 8.12 VPN's - Virtual Private Networks

9. Security Preparation (before you go on-line)

- 9.1 Make a Full Backup of Your Machine
- 9.2 Choosing a Good Backup Schedule
- 9.3 Backup Your RPM or Debian File Database
- 9.4 Keep Track of Your System Accounting Data
- 9.5 Apply All New System Updates.

10. What To Do During and After a Breakin

- 10.1 Security Compromise Underway.
- 10.2 Security Compromise has already happened

11. Security Sources

- 11.1 FTP Sites
- 11.2 Web Sites
- 11.3 Mailing Lists
- 11.4 Books - Printed Reading Material

12. Glossary

13. Frequently Asked Questions

14. Conclusion

15. Acknowledgements

1. Introduction

This document covers some of the main issues that affect Linux security. General philosophy and net-born resources are discussed.

A number of other HOWTO documents overlap with security issues, and those documents have been pointed to wherever appropriate.

This document is *not* meant to be a up to date exploits document. Large numbers of new exploits happen all the time. This document will tell you where to look for such up to date information, and will give some general methods to prevent such exploits from taking place.

1.1 New Versions of this Document

New versions of this document will be periodically posted to *comp.os.linux.answers*. They will also be added to the various anonymous FTP sites that archive such information, including:

`ftp://metalab.unc.edu/pub/Linux/docs/HOWTO`

In addition, you should generally be able to find this document on the Linux World Wide Web home page via:

`http://metalab.unc.edu/mdw/linux.html`

Finally, the very latest version of this document should also be available in various formats from:

`http://scrye.com/~kevin/lsh/`

1.2 Feedback

All comments, error reports, additional information and criticism of all sorts should be directed to:

`kevin@scrye.com`

and

`dave@nic.com`

Note: Please send your feedback to *both* authors. Also, be sure and include "Linux" "security", or "HOWTO" in your subject to avoid Kevin's spam filter.

1.3 Disclaimer

No liability for the contents of this document can be accepted. Use the concepts, examples and other content at your own risk. Additionally, this is an early version, possibly with many inaccuracies or errors.

A number of the examples and descriptions use the RedHat(tm) package layout and system setup. Your mileage may vary.

As far as we know, only programs that, under certain terms may be used or evaluated for personal purposes will be described. Most of the programs will be available, complete with source, under GNU terms.

1.4 Copyright Information

This document is copyrighted (c)1998,1999 Kevin Fenzi and Dave Wreski, and distributed under the following terms:

- Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium, physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the authors would like to be notified of any such distributions.
- All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.
- If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at

tjbynum@metalab.unc.edu

---

2. Overview

This document will attempt to explain some procedures and commonly-used software to help your Linux system be more secure. It is important to discuss some of the basic concepts first, and create a security foundation, before we get started.

2.1 Why Do We Need Security?

In the ever-changing world of global data communications, inexpensive Internet connections, and fast-paced software development, security is becoming more and more of an issue. Security is now a basic requirement because global computing is inherently insecure. As your data goes from point A to point B on the Internet, for example, it may pass through several other points along the way, giving other users the opportunity to intercept, and even alter, it. Even other users on your system may maliciously transform your data into something you did not intend. Unauthorized access to your system may be obtained by intruders, also known as "crackers", who then use advanced knowledge to impersonate you, steal information from you, or even deny you access to your own resources. If you're wondering what the difference is between a "Hacker" and a "Cracker", see Eric Raymond's document, "How to Become A Hacker", available at http://sagan.earthspace.net/~esr/faqs/hacker-howto.html.

2.2 How Secure Is Secure?

First, keep in mind that no computer system can ever be "completely secure". All you can do is make it increasingly difficult for someone to compromise your system. For the average home Linux user, not much is required to keep the casual cracker at bay. For high profile Linux users (banks, telecommunications companies, etc), much more work is required.

Another factor to take into account is that the more secure your system is, the more intrusive your security becomes. You need to decide where in this balancing act your system will still usable, and yet secure for your purposes. For instance, you could require everyone dialing into your system to use a call-back modem to call them back at their home number. This is more secure, but if someone is not at home, it makes it difficult for them to login. You could also setup your Linux system with no network or connection to the Internet, but this limits it's usefulness.

If you are a large to medium-sized site, you should establish a security policy stating how much security is required by your site and what auditing is in place to check it. You can find a well-known security policy example at http://ds.internic.net/rfc/rfc2196.txt. It has been recently updated, and contains a great framework for establishing a security policy for your company.

2.3 What Are You Trying to Protect?

TOC

Before you attempt to secure your system, you should determine what level of threat you have to protect against, what risks you should or should not take, and how vulnerable your system is as a result. You should analyze your system to know what you're protecting, why you're protecting it, what value it has, and who has responsibility for your data and other assets.

- Risk is the possibility that an intruder may be successful in attempting to access your computer. Can an intruder read or write files, or execute programs that could cause damage? Can they delete critical data? Can they prevent you or your company from getting important work done? Don't forget: someone gaining access to your account, or your system, can also impersonate you.

  Additionally, having one insecure account on your system can result in your entire network being compromised. If you allow a single user to login using a `.rhosts` file, or to use an insecure service, such as `tftp`, you risk an intruder getting 'his foot in the door'. Once the intruder has a user account on your system, or someone else's system, it can be used to gain access to another system, or another account.
- Threat is typically from someone with motivation to gain unauthorized access to your network or computer. You must decide who you trust to have access to your system, and what threat they could pose.

  There are several types of intruders, and it is useful to keep their different characteristics in mind as you are securing your systems.
  - **The Curious** - This type of intruder is basically interested in finding out what type of system and data you have.
  - **The Malicious** - This type of intruder is out to either bring down your systems, or deface your web page, or otherwise force you to spend time and money recovering from the damage he has caused.
  - **The High-Profile Intruder** - This type of intruder is trying to use your system to gain popularity and infamy. He might use your high-profile system to advertise his abilities.
  - **The Competition** - This type of intruder is interested in what data you have on your system. It might be someone who thinks you have something that could benefit him, financially or otherwise.
  - **The Borrowers** - This type of intruder is interested in setting up shop on your system and using it's resources for their own purposes. They typically will run chat or irc servers, porn archive sites, or even DNS servers.
  - **The Leapfrogger** - This type of intruder is only interested in your system to use it to get into other systems. If your system is well connected or a gateway to a number of internal hosts, you may well see this type trying to compromise your system.
- Vulnerability describes how well-protected your computer is from another network, and the potential for someone to gain unauthorized access.

  What's at stake if someone breaks into your system? Of course the concerns of a dynamic PPP home user will be different from those of a company connecting their machine to the Internet, or another large network.

How much time would it take to retrieve/recreate any data that was lost? An initial time investment now can save ten times more time later if you have to recreate data that was lost. Have you checked your backup strategy, and verified your data lately?

2.4 Developing A Security Policy

Create a simple, generic policy for your system that your users can readily understand and follow. It should protect the data you're safeguarding as well as the privacy of the users. Some things to consider adding are: who has access to the system (Can my friend use my account?), who's allowed to install software on the system, who owns what data, disaster recovery, and appropriate use of the system.

A generally accepted security policy starts with the phrase

That which is not permitted is prohibited

This means that unless you grant access to a service for a user, that user shouldn't be using that service until you do grant access. Make sure the policies work on your regular user account. Saying, "Ah, I can't figure this permissions problem out, I'll just do it as root" can lead to security holes that are very obvious, and even ones that haven't been exploited yet.

rfc1244 is a document that describes how to create your own network security polity.

rfc1281 is a document that shows an example security policy with detailed descriptions of each step.

Finally, you might want to look at the COAST policy archive at ftp://coast.cs.purdue.edu/pub/doc/policy to see what some real life security policies look like.

2.5 Means of Securing Your Site

This document will discuss various means with which you can secure the assets you have worked hard for: your local machine, your data, your users, your network, even your reputation. What would happen to your reputation if an intruder deleted some of your users' data? Or defaced your web site? Or published your company's corporate project plan for next quarter? If you are planning a network installation, there are many factors you must take into account before adding a single machine to your network.

Even if you have a single dialup PPP account, or just a small site, this does not mean intruders won't be interested in your systems. Large, high profile sites are not the only targets -- many intruders simply want to exploit as many sites as possible, regardless of their size. Additionally, they may use a security hole in your site to gain access to other sites you're connected to.

Intruders have a lot of time on their hands, and can avoid guessing how you've obscured your system just by trying all the possibilities. There are also a number of reasons an intruder may be interested in your systems, which we will discuss later.

TOC

Host Security

Perhaps the area of security on which administrators concentrate most is host-based security. This typically involves making sure your own system is secure, and hoping everyone else on your network does the same. Choosing good passwords, securing your host's local network services, keeping good accounting records, and upgrading programs with known security exploits are among the things the local security administrator is responsible for doing. Although this is absolutely necessary, it can become a daunting task once your network becomes larger than a few machines.

Network Security

Network security is also as necessary as local host security. With hundreds, thousands, or more computers on the same network, you can't rely on each one of those systems being secure. Ensuring that only authorized users can use your network, building firewalls, using strong encryption, and ensuring there are no "rogue" (that is, unsecured) machines on your network are all part of the network security administrator's duties.

This document will discuss some of the techniques used to secure your site, and hopefully show you some of the ways to prevent an intruder from gaining access to what you are trying to protect.

Security Through Obscurity

One type of security that must be discussed is "security through obscurity". This means, for example, moving a service that has known security vunerabilities to a non standard port in hopes that attackers won't notice it's there and thus won't exploit it. Rest assured that they can determine that its there and will exploit it. Security through obscurity is no security at all. Simply because you may have a small site, or a relatively low profile, does not mean an intruder won't be interested in what you have. We'll discuss what you're protecting in the next sections.

2.6 Organization of This Document

This document has been divided into a number of sections. They cover several broad security issues. The first, Physical Security, covers how you need to protect your physical machine from tampering. The second, Local Security, describes how to protect your system from tampering by local users. The third, Files and Filesystem Security, shows you how to setup your filesystems and premissions on your files. The next, Password Security and Encryption, discusses how to use encryption to better secure your machine and network. Kernel Security discusses what kernel options you should set or be aware of for a more secure system. Network Security, describes how to better secure your Linux system from network attacks. Security Preparation, discusses how to prepare your machine(s) before bringing them on-line. Next, What To Do During and After a Break-in, discusses what to do when you detect a system compromise in progress or detect one that has recently happened. In Security Resources, some primary security resources are enumerated. The Q and A section Frequently Asked Questions, answers some frequently asked questions, and finally a conclusion in Conclusion section.

The two main points to realize when reading this document are:

- Be aware of your system. Check system logs such as `/var/log/messages` and keep an eye on your system, and
- Keep your system up to date by making sure you have installed the current versions of software and have upgraded per security alerts. Just doing this will help make your system markedly more secure.

---

3. Physical Security

The first layer of security you need to take into account is the physical security of your computer systems. Who has direct physical access to your machine? Should they? Can you protect your machine from their tampering? Should you?

How much physical security you need on your system is very dependent on your situation, and/or budget.

If you are a home user, you probably don't need a lot (although you might need to protect your machine from tampering by children or annoying relatives). If you are in a Lab, you need considerably more, but users will still need to be able to get work done on the machines. Many of the following sections will help out. If you are in an office, you may or may not need to secure your machine off hours or while you are away. At some companies, leaving your console unsecured is a termination offense.

Obvious physical security methods such as locks on doors, cables, locked cabinets, and video surveillance are all good ideas, but beyond the scope of this document. :)

3.1 Computer locks

Many modern PC cases include a "locking" feature. Usually this will be a socket on the front of the case that allows you to turn an included key to a locked or unlocked position. Case locks can help prevent someone from stealing your PC, or opening up the case and directly manipulating/stealing your hardware. They can also sometimes prevent someone from rebooting your computer on their own floppy or other hardware.

These case locks do different things according to the support in the motherboard and how the case is constructed. On many PC's they make it so you have to break the case to get the case open. On some others, they make it so that it will not let you plug in new keyboards and mice. Check your motherboard or case instructions for more information. This can sometimes be a very useful feature, even though the locks are usually very low quality and can easily be defeated by attackers with locksmithing.

Some cases (most notably SPARCs and macs) have a dongle on the back that, if you put a cable through attackers would have to cut the cable or break the case to get into it. Just putting a padlock or combo lock through these can be a good deterrent to someone stealing your machine.

3.2 BIOS Security

The BIOS is the lowest level of software that configures or manipulates your x86-based hardware. LILO and other Linux boot methods access the BIOS to determine how to boot up your Linux machine. Other hardware that Linux runs on has similar software (OpenFirmware on Macs and new Suns, Sun boot PROM, etc...). You can use your BIOS to prevent attackers from rebooting your machine and manipulating your Linux system.

Many PC BIOSs let you set a boot password. This doesn't provide all that much security (the BIOS c be reset, or removed if someone can get into the case), but might be a good deterrent (i.e. it will take and leave traces of tampering). Similarly, on S/Linux (Linux for SPARC(tm) porcessor machines), your EEPROM can be set to require a boot-up password. This might slow attackers down.

Many x86 BIOSs also allow you to specify various other good security settings. Check your BIOS manual or look at it the next time you boot up. For example, some BIOSs disallow booting from floppy drives and some require passwords to access some BIOS features.

Note: If you have a server machine, and you set up a boot password, your machine will not boot up unattended. Keep in mind that you will need to come in and supply the password in the event of a power failure. ;(

3.3 Boot Loader Security

The various Linux boot loaders also can have a boot password set. LILO, for example, has `password` and `restricted` settings; `password` always requires password at boot time, whereas `restricted` requires a boot-time password only if you specify options (such as `single`) at the `LILO` prompt.

Keep in mind when setting all these passwords that you need to remember them. :) Also remember that these passwords will merely slow the determined attacker. They won't prevent someone from booting from a floppy, and mounting your root partition. If you are using security in conjunction with a boot loader, you might as well disable booting from a floppy in your computer's BIOS, and password-protect the BIOS.

If anyone has security-related information from a different boot loader, we would love to hear it. (`grub`, `silo`, `milo`, `linload`, etc).

Note: If you have a server machine, and you set up a boot password, your machine will *not* boot up unattended. Keep in mind that you will need to come in and supply the password in the event of a power failure. ;(

3.4 xlock and vlock

If you wander away from your machine from time to time, it is nice to be able to "lock" your console so that no one tampers with or looks at your work. Two programs that do this are: `xlock` and `vlock`.

`xlock` is a X display locker. It should be included in any Linux distributions that support X. Check out the man page for it for more options, but in general you can run `xlock` from any xterm on your console and it will lock the display and require your password to unlock.

`vlock` is a simple little program that allows you to lock some or all of the virtual consoles on your Linux box. You can lock just the one you are working in or all of them. If you just lock one, others can come in and use the console; they will just not be able to use your virtual console until you unlock it. `vlock` ships with redhat Linux, but your mileage may vary.

Of course locking your console will prevent someone from tampering with your work, but won't prevent

them from rebooting your machine or otherwise disrupting your work. It also does not prevent them accessing your machine from another machine on the network and causing problems.

More importantly, it does not prevent someone from switching out of the X Window System entirely, and going to a normal virtual console login prompt, or to the VC that X11 was started from, and suspending it, thus obtaining your priviledges. For this reason, you might consider only using it while under control of xdm.

3.5 Detecting Physical Security Compromises

The first thing to always note is when your machine was rebooted. Since Linux is a robust and stable OS, the only times your machine should reboot is when *you* take it down for OS upgrades, hardware swapping, or the like. If your machine has rebooted without you doing it, that may be a sign that an intruder has compromised it. Many of the ways that your machine can be compromised require the intruder to reboot or power off your machine.

Check for signs of tampering on the case and computer area. Although many intruders clean traces of their presence out of logs, it's a good idea to check through them all and note any discrepancy.

It is also a good idea to store log data at a secure location, such as a dedicated log server within your well-protected network. Once a machine has been compromised, log data becomes of little use as it most likely has also been modified by the intruder.

The syslog daemon can be configured to automatically send log data to a central syslog server, but this is typically sent in cleartext data, allowing an intruder to view data as it is being transferred. This may reveal information about your network that is not intended to be public. There are syslog daemons available that encrypt the data as it is being sent.

Also be aware that faking syslog messages is easy - with an exploit program having been published. Syslog even accepts net log entries claiming to come from the local host without indicating their true origin.

Some things to check for in your logs:

- Short or incomplete logs.
- Logs containing strange timestamps.
- Logs with incorrect permissions or ownership.
- Records of reboots or restarting of services.
- missing logs.
- su entries or logins from strange places.

We will discuss system log data later in the HOWTO.

---

4. Local Security

The next thing to take a look at is the security in your system against attacks from local users. Did we just say *local* users? Yes!

Getting access to a local user account is one of the first things that system intruders attempt while on their way to exploiting the root account. With lax local security, they can then "upgrade" their normal user access to root access using a variety of bugs and poorly setup local services. If you make sure your local security is tight, then the intruder will have another hurdle to jump.

Local users can also cause a lot of havoc with your system even (especially) if they really are who they say they are. Providing accounts to people you don't know or have no contact information for is a very bad idea.

4.1 Creating New Accounts

You should make sure to provide user accounts with only the minimal requirements for the task they need to do. If you provide your son (age 10) with an account, you might want him to only have access to a word processor or drawing program, but be unable to delete data that is not his.

Several good rules of thumb when allowing other people legitimate access to your Linux machine:

- Give them the minimal amount of privileges they need.
- Be aware when/where they login from, or should be logging in from.
- Make sure to remove inactive accounts
- The use of the same user-ID on all computers and networks is advisable to ease account maintence, as well as permit easier analysis of log data.
- The creation of group user-IDs should be absolutely prohibited. User accounts also provide accountability, and this is not possible with group accounts.

Many local user accounts that are used in security compromises are ones that have not been used in months or years. Since no one is using them they, provide the ideal attack vehicle.

4.2 Root Security

The most sought-after account on your machine is the root (superuser) account. This account has authority over the entire machine, which may also include authority over other machines on the network. Remember that you should only use the root account for very short, specific tasks, and should mostly run as a normal user. Even small mistakes made while logged in as the root user can cause problems. The less time you are on with root privledges, the safer you will be.

Several tricks to avoid messing up your own box as root:

TOC

- When doing some complex command, try running it first in a non-destructive way...especially commands that use globbing: e.g., if you want to do `"rm foo*.bak"`, first do `"ls foo*.bak"` and make sure you are going to delete the files you think you are. Using `echo` in place of destructive commands also sometimes works.
- Provide your users with a default alias to the rm command to ask for confirmation for deletion of files.
- Only become root to do single specific tasks. If you find yourself trying to figure out how to do something, go back to a normal user shell until you are *sure* what needs to be done by root.
- The command path for the root user is very important. The command path (that is, the `PATH` environment variable) specifies the directories in which the shell searches for programs. Try to limit the command path for the root user as much as possible, and never include `.` (which means "the current directory") in your PATH. Additionally, never have writable directories in your search path, as this can allow attackers to modify or place new binaries in your search path, allowing them to run as root the next time you run that command.
- Never use the rlogin/rsh/rexec suite of tools (called the r-utilities) as root. They are subject to many sorts of attacks, and are downright dangerous run as root. Never create a `.rhosts` file for root.
- The `/etc/securetty` file contains a list of terminals that root can login from. By default (on Red Hat Linux) this is set to only the local virtual consoles(vtys). Be very careful of adding anything else to this file. You should be able to login remotely as your regular user account and then `su` if you need to (hopefully over `ssh` or other encrypted channel), so there is no need to be able to login directly as root.
- Always be slow and deliberate running as root. Your actions could affect a lot of things. Think before you type!

If you absolutely positively need to allow someone (hopefully very trusted) to have root access to your machine, there are a few tools that can help. `sudo` allows users to use their password to access a limited set of commands as root. This would allow you to, for instance, let a user be able to eject and mount removable media on your Linux box, but have no other root privileges. `sudo` also keeps a log of all successful and unsuccessful sudo attempts, allowing you to track down who used what command to do what. For this reason `sudo` works well even in places where a number of people have root access, because it helps you keep track of changes made.

Although `sudo` can be used to give specific users specific privileges for specific tasks, it does have several shortcomings. It should be used only for a limited set of tasks, like restarting a server, or adding new users. Any program that offers a shell escape will give root access to a user invoking it via `sudo`. This includes most editors, for example. Also, a program as innocuous as `/bin/cat` can be used to overwrite files, which could allow root to be exploited. Consider `sudo` as a means for accountability, and don't expect it to replace the root user and still be secure.

---

5. Files and Filesystem Security

A few minutes of preparation and planning ahead before putting your systems online can help to protect them and the data stored on them.

- There should never be a reason for users' home directories to allow SUID/SGID programs to be run from there. Use the `nosuid` option in `/etc/fstab` for partitions that are writable by others than root. You may also wish to use `nodev` and `noexec` on users' home partitions, as well as `/var`, thus prohibiting execution of programs, and creation of character or block devices, which should never be necessary anyway.
- If you are exporting filesystems using NFS, be sure to configure `/etc/exports` with the most restrictive access possible. This means not using wildcards, not allowing root write access, and exporting read-only wherever possible.
- Configure your users' file-creation `umask` to be as restrictive as possible. See umask settings.
- If you are mounting filesystems using a network filesystem such as NFS, be sure to configure /etc/exports with suitable restrictions. Typically, using `nodev', `nosuid', and perhaps `noexec', are desireable.
- Set filesystem limits instead of allowing `unlimited` as is the default. You can control the per-user limits using the resource-limits PAM module and `/etc/pam.d/limits.conf`. For example, limits for group `users` might look like this:

  ```
  @users     hard  core    0
  @users     hard  nproc   50
  @users     hard  rss     5000
  ```

  This says to prohibit the creation of core files, restrict the number of processes to 50, and restrict memory usage per user to 5M.
- The `/var/log/wtmp` and `/var/run/utmp` files contain the login records for all users on your system. Their integrity must be maintained because it can be used to determine when and from where a user (or potential intruder) has entered your system. These files should also have `644` permissions, without affecting normal system operation.
- The immutable bit can be used to prevent accidentally deleting or overwriting a file that must be protected. It also prevents someone from creating a symbolic link to the file (such symbolic links have been the source of attacks involving deleting `/etc/passwd` or `/etc/shadow`). See the `chattr`(1) man page for information on the immutable bit.
- SUID and SGID files on your system are a potential security risk, and should be monitored closely. Because these programs grant special privileges to the user who is executing them, it is necessary to ensure that insecure programs are not installed. A favorite trick of crackers is to exploit SUID-root programs, then leave a SUID program as a backdoor to get in the next time, even if the original hole is plugged.

  Find all SUID/SGID programs on your system, and keep track of what they are, so you are aware of any changes which could indicate a potential intruder. Use the following command to find all SUID/SGID programs on your system:

  ```
  root#  find / -type f \( -perm -04000 -o -perm -02000 \)
  ```

  The Debian distribution runs a job each night to determine what SUID files exist. It then compairs this to the previous nights run. You can look in `/var/log/suid*` for this log.

  You can remove the SUID or SGID permissions on a suspicious program with `chmod`, then change it back if you

absolutely feel it is necessary.

- World-writable files, particularly system files, can be a security hole if a cracker gains access to your system modifies them. Additionally, world-writable directories are dangerous, since they allow a cracker to add or delete files as he wishes. To locate all world-writable files on your system, use the following command:

```
root# find / -perm -2 ! -type l -ls
```

  and be sure you know why those files are writable. In the normal course of operation, several files will be world-writable, including some from `/dev`, and symbolic links, thus the `! -type l` which excludes these from the previous `find` command.

- Unowned files may also be an indication an intruder has accessed your system. You can locate files on your system that have no owner, or belong to no group with the command:

```
root# find / -nouser -o -nogroup -print
```

- Finding `.rhosts` files should be a part of your regular system administration duties, as these files should not be permitted on your system. Remember, a cracker only needs one insecure account to potentially gain access to your entire network. You can locate all `.rhosts` files on your system with the following command:

```
root# find /home -name .rhosts -print
```

- Finally, before changing permissions on any system files, make sure you understand what you are doing. Never change permissions on a file because it seems like the easy way to get things working. Always determine why the file has that permission before changing it.

5.1 Umask Settings

The `umask` command can be used to determine the default file creation mode on your system. It is the octal complement of the desired file mode. If files are created without any regard to their permissions settings, the user could inadvertently give read or write permission to someone that should not have this permission. Typically `umask` settings include `022`, `027`, and `077` (which is the most restrictive). Normally the umask is set in `/etc/profile`, so it applies to all users on the system. The file creation mask can be calculated by subtracting the desired value from 777. In other words, a umask of 777 would cause newly-created files to contain no read, write or execute permission for anyone. A mask of 666 would cause newly-created files to have a mask of 111. For example, you may have a line that looks like this:

```
# Set the user's default umask
umask 033
```

Be sure to make root's umask `077`, which will disable read, write, and execute permission for other users, unless explicitly changed using `chmod`. In this case, newly-created directories would have 744 permissions, obtained by subtracting 033 from 777. Newly-created files using the 033 umask would have permissions of 644.

If you are using Red Hat, and adhere to their user and group ID creation scheme (User Private Groups), it is only necessary to use `002` for a `umask`. This is due to the fact that the default configuration is one user per group.

5.2 File Permissions

It's important to ensure that your system files are not open for casual editing by users and groups who shouldn't be doing such system maintenance.

Unix seperates access control on files and directories according to three characteristics: owner, group, and other. There

TOC

is always exactly one owner, any number of members of the group, and everyone else.

A quick explanation of Unix permissions:

Ownership - Which user(s) and group(s) retain(s) control of the permission settings of the node and parent of the node

Permissions - Bits capable of being set or reset to allow certain types of access to it. Permissions for directories may have a different meaning than the same set of permissions on files.

Read:

- To be able to view contents of a file
- To be able to read a directory

Write:

- To be able to add to or change a file
- To be able to delete or move files in a directory

Execute:

- To be able to run a binary program or shell script
- To be able to search in a directory, combined with read permission

Save Text Attribute: (For directories)

The "sticky bit" also has a different meaning when applied to directories than when applied to files. If the sticky bit is set on a directory, then a user may only delete files that the he owns or for which he has explicit write permission granted, even when he has write access to the directory. This is designed for directories like `/tmp`, which are world-writable, but where it may not be desirable to allow any user to delete files at will. The sticky bit is seen as a `t` in a long directory listing.

SUID Attribute: (For Files)

This describes set-user-id permissions on the file. When the set user ID access mode is set in the owner permissions, and the file is executable, processes which run it are granted access to system resources based on user who owns the file, as opposed to the user who created the process. This is the cause of many "buffer overflow" exploits.

SGID Attribute: (For Files)

If set in the group permissions, this bit controls the "set group id" status of a file. This behaves the same way as SUID, except the group is affected instead. The file must be executable for this to have any effect.

SGID Attribute: (For directories)

If you set the SGID bit on a directory (with `chmod g+s` *`directory`*), files created in that directory will have their group set to the directory's group.

You - The owner of the file

Group - The group you belong to

Everyone - Anyone on the system that is not the owner or a member of the group

File Example:

```
-rw-r--r--  1 kevin  users         114 Aug 28  1997 .zlogin
1st bit - directory?             (no)
 2nd bit - read by owner?         (yes, by kevin)
  3rd bit - write by owner?        (yes, by kevin)
   4th bit - execute by owner?      (no)
```

TOC

```
5th bit - read by group?         (yes, by users)
 6th bit - write by group?        (no)
  7th bit - execute by group?      (no)
   8th bit - read by everyone?      (yes, by everyone)
    9th bit - write by everyone?     (no)
     10th bit - execute by everyone?  (no)
```

The following lines are examples of the minimum sets of permissions that are required to perform the access described. You may want to give more permission than what's listed here, but this should describe what these minimum permissions on files do:

```
-r--------  Allow read access to the file by owner
--w-------  Allows the owner to modify or delete the file
            (Note that anyone with write permission to the directory
             the file is in can overwrite it and thus delete it)
---x------  The owner can execute this program, but not shell scripts,
             which still need read permission
---s------  Will execute with effective User ID = to owner
--------s-  Will execute with effective Group ID = to group
-rw------T  No update of "last modified time".  Usually used for swap
             files
---t------  No effect.  (formerly sticky bit)
```

Directory Example:

```
drwxr-xr-x  3 kevin  users         512 Sep 19 13:47 .public_html/
1st bit - directory?             (yes, it contains many files)
 2nd bit - read by owner?         (yes, by kevin)
  3rd bit - write by owner?        (yes, by kevin)
   4th bit - execute by owner?      (yes, by kevin)
    5th bit - read by group?         (yes, by users
     6th bit - write by group?        (no)
      7th bit - execute by group?      (yes, by users)
       8th bit - read by everyone?      (yes, by everyone)
        9th bit - write by everyone?     (no)
         10th bit - execute by everyone?  (yes, by everyone)
```

The following lines are examples of the minimum sets of permissions that are required to perform the access described. You may want to give more permission than what's listed, but this should describe what these minimum permissions on directories do:

```
dr--------  The contents can be listed, but file attributes can't be read
d--x------  The directory can be entered, and used in full execution
             paths
dr-x------  File attributes can be read by owner
d-wx------  Files can be created/deleted, even if the directory
             isn't the current one
d------x-t  Prevents files from deletion by others with write
             access. Used on /tmp
d---s--s--  No effect
```

System configuration files (usually in `/etc`) are usually mode `640` (`-rw-r-----`), and owned by root. Depending your sites security requirements, you might adjust this. Never leave any system files writable by a group or everyone. Some configuration files, including `/etc/shadow`, should only be readable by root, and directories in `/etc` should at least not be accessible by others.

SUID Shell Scripts

> SUID shell scripts are a serious security risk, and for this reason the kernel will not honor them. Regardless of how secure you think the shell script is, it can be exploited to give the cracker a root shell.

5.3 Integrity Checking with Tripwire `Tripwire`

Another very good way to detect local (and also network) attacks on your system is to run an integrity checker like `Tripwire`. `Tripwire` runs a number of checksums on all your important binaries and config files and compares them against a database of former, known-good values as a reference. Thus, any changes in the files will be flagged.

It's a good idea to install `Tripwire` onto a floppy, and then physically set the write protect on the floppy. This way intruders can't tamper with `Tripwire` itself or change the database. Once you have `Tripwire` setup, it's a good idea to run it as part of your normal security administration duties to see if anything has changed.

You can even add a `crontab` entry to run `Tripwire` from your floppy every night and mail you the results in the morning. Something like:

```
# set mailto
MAILTO=kevin
# run Tripwire
15 05 * * * root /usr/local/adm/tcheck/tripwire
```

will mail you a report each morning at 5:15am.

`Tripwire` can be a godsend to detecting intruders before you would otherwise notice them. Since a lot of files change on the average system, you have to be careful what is cracker activity and what is your own doing.

You can find `Tripwire` at http://www.tripwiresecurity.com, free of charge. Manuals and support can be purchased.

5.4 Trojan Horses

"Trojan Horses" are named after the fabled ploy in Homer's "Iliad". The idea is that a cracker distributes a program or binary that sounds great, and encourages other people to download it and run it as root. Then the program can compromise their system while they are not paying attention. While they think the binary they just pulled down does one thing (and it might very well), it also compromises their security.

You should take care of what programs you install on your machine. Redhat provides MD5 checksums and PGP signatures on it's RPM files so you can verify you are installing the real thing. Other distributions have similar methods. You should never run any unfamiliar binary, for which you don't have the source, as root! Few attackers are willing to release source code to public scrutiny.

Although it can be complex, make sure you are getting the source for a program from its real distribution site. If the program is going to run as root, make sure either you or someone you trust has looked over the source and verified it.

---

6. Password Security and Encryption

One of the most important security features used today are passwords. It is important for both you and all your users to have secure, unguessable passwords. Most of the more recent Linux distributions include `passwd` programs that do not allow you to set a easily guessable password. Make sure your `passwd` program is up to date and has these features.

In-depth discussion of encryption is beyond the scope of this document, but an introduction is in order. Encryption is very useful, possibly even necessary in this day and age. There are all sorts of methods of encrypting data, each with its own set of characteristics.

Most Unicies (and Linux is no exception) primarily use a one-way encryption algorithm, called DES (Data Encryption Standard) to encrypt your passwords. This encrypted password is then stored in (typically) `/etc/passwd` (or less commonly) `/etc/shadow`. When you attempt to login, the password you type in is encrypted again and compared with the entry in the file that stores your passwords. If they match, it must be the same password, and you are allowed access. Although DES is a two-way encryption algorithm (you can code and then decode a message, given the right keys), the variant that most unices use is one-way. This means that it should not be possible to reverse the encryption to get the password from the contents of `/etc/passwd` (or `/etc/shadow`).

Brute force attacks, such as "Crack" or "John the Ripper" (see Section crack) can often guess passwords unless your password is sufficiently random. PAM modules (see below) allow you to use a different encryption routine with your passwords (MD5 or the like). You can use Crack to your advantage, as well. Consider periodically running Crack against your own password database, to find insecure passwords. Then contact the offending user, and instruct him to change his password.

You can go to http://consult.cern.ch/writeup/security/security_3.html for information on how to choose a good password.

6.1 PGP and Public-Key Cryptography

Public-key cryptography, such as that used for PGP, uses one key for encryption, and one key for decryption. Traditional cryptography, however, uses the same key for encryption and decryption; this key must be known to both parties, and thus somehow transferred from one to the other securely.

To alleviate the need to securely transmit the encryption key, public-key encryption uses two separate keys: a public key and a private key. Each person's public key is available by anyone to do the encryption, while at the same time each person keeps his or her private key to decrypt messages encrypted with the correct public key.

There are advantages to both public key and private key cryptography, and you can read about those differences in the RSA Cryptography FAQ, listed at the end of this section.

PGP (Pretty Good Privacy) is well-supported on Linux. Versions 2.6.2 and 5.0 are known to work v
For a good primer on PGP and how to use it, take a look at the PGP FAQ:
http://www.pgp.com/service/export/faq/55faq.cgi

Be sure to use the version that is applicable to your country. Due to export restrictions by the US Government, strong-encryption is prohibited from being transferred in electronic form outside the country.

US export controls are now managed by EAR (Export Administration Regulations). They are no longer governed by ITAR.

There is also a step-by-step guide for configuring PGP on Linux available at http://mercury.chem.pitt.edu/~angel/LinuxFocus/English/November1997/article7.html. It was written for the international version of PGP, but is easily adaptable to the United States version. You may also need a patch for some of the latest versions of Linux; the patch is available at ftp://metalab.unc.edu/pub/Linux/apps/crypto.

There is a project working on a free re-implementation of pgp with open source. GnuPG is a complete and free replacement for PGP. Because it does not use IDEA or RSA it can be used without any restrictions. GnuPG is nearly in compliance with RFC2440 (OpenPGP). See the GNU Privacy Guard web page for more information: http://www.gpg.org/.

More information on cryptography can be found in the RSA cryptography FAQ, available at http://www.rsa.com/rsalabs/newfaq/. Here you will find information on such terms as "Diffie-Hellman", "public-key cryptography", "digital certificates", etc.

6.2 SSL, S-HTTP, HTTPS and S/MIME

Often users ask about the differences between the various security and encryption protocols, and how to use them. While this isn't an encryption document, it is a good idea to explain briefly what each protocol is, and where to find more information.

- **SSL:** - SSL, or Secure Sockets Layer, is an encryption method developed by Netscape to provide security over the Internet. It supports several different encryption protocols, and provides client and server authentication. SSL operates at the transport layer, creates a secure encrypted channel of data, and thus can seamlessly encrypt data of many types. This is most commonly seen when going to a secure site to view a secure online document with Communicator, and serves as the basis for secure communications with Communicator, as well as many other Netscape Communications data encryption. More information can be found at http://www.consensus.com/security/ssl-talk-faq.html. Information on Netscape's other security implementations, and a good starting point for these protocols is available at http://home.netscape.com/info/security-doc.html.
- **S-HTTP:** - S-HTTP is another protocol that provides security services across the Internet. It was designed to provide confidentiality, authentication, integrity, and non-repudiability [cannot be mistaken for someone else] while supporting multiple key-management mechanisms and cryptographic algorithms via option negotiation between the parties involved in each transaction.

S-HTTP is limited to the specific software that is implementing it, and encrypts each message individually. [From RSA Cryptography FAQ, page 138]

- **S/MIME:** - S/MIME, or Secure Multipurpose Internet Mail Extension, is an encryption standard used to encrypt electronic mail and other types of messages on the Internet. It is an open standard developed by RSA, so it is likely we will see it on Linux one day soon. More information on S/MIME can be found at http://home.netscape.com/assist/security/smime/overview.html.

6.3 Linux IPSEC Implementations

Along with CIPE, and other forms of data encryption, there is also several other implementations of IPSEC for Linux. IPSEC is an effort by the IETF to create cryptographically-secure communications at the IP network level, and to provide authentication, integrity, access control, and confidentiality. Information on IPSEC and Internet draft can be found at http://www.ietf.org/html.charters/ipsec-charter.html. You can also find links to other protocols involving key management, and an IPSEC mailing list and archives.

The x-kernel Linux implementation, which is being developed at the University of Arizona, uses an object-based framework for implementing network protocols called x-kernel, and can be found at http://www.cs.arizona.edu/xkernel/hpcc-blue/linux.html. Most simply, the x-kernel is a method of passing messages at the kernel level, which makes for an easier implementation.

Another freely-available IPSEC implementation is the Linux FreeS/WAN IPSEC. Their web page states,

> "These services allow you to build secure tunnels through untrusted networks. Everything passing through the untrusted net is encrypted by the IPSEC gateway machine and decrypted by the gateway at the other end. The result is Virtual Private Network or VPN. This is a network which is effectively private even though it includes machines at several different sites connected by the insecure Internet."

It's available for download from http://www.xs4all.nl/~freeswan/, and has just reached 1.0 at the time of this writing.

As with other forms of cryptography, it is not distributed with the kernel by default due to export restrictions.

6.4 `ssh` (Secure Shell) and `stelnet`

`ssh` and `stelnet` are programs that allow you to login to remote systems and have a encrypted connection.

`ssh` is a suite of programs used as a secure replacement for `rlogin`, `rsh` and `rcp`. It uses public-key cryptography to encrypt communications between two hosts, as well as to authenticate users. It can be used to securely login to a remote host or copy data between hosts, while preventing man-in-the-middle attacks (session hijacking) and DNS spoofing. It will perform data compression on your connections, and secure X11 communications between hosts. The `ssh` home page can be found at http://www.cs.hut.fi/ssh/

TOC

You can also use `ssh` from your Windows workstation to your Linux `ssh` server. There are several freely available Windows client implementations, including the one at http://guardian.htu.tuwien.ac.at/therapy/ssh/ as well as a commercial implementation from DataFellows, at http://www.datafellows.com. There is also a open source project to re-implement ssh called "psst...". For more information see: http://www.net.lut.ac.uk/psst/

SSLeay is a free implementation of Netscape's Secure Sockets Layer protocol, developed by Eric Young. It includes several applications, such as Secure telnet, a module for Apache, several databases, as well as several algorithms including DES, IDEA and Blowfish.

Using this library, a secure telnet replacement has been created that does encryption over a telnet connection. Unlike SSH, stelnet uses SSL, the Secure Sockets Layer protocol developed by Netscape. You can find Secure telnet and Secure FTP by starting with the SSLeay FAQ, available at http://www.psy.uq.oz.au/~ftp/Crypto/.

SRP is another secure telnet/ftp implementation. From their web page:

> "The SRP project is developing secure Internet software for free worldwide use. Starting with a fully-secure Telnet and FTP distribution, we hope to supplant weak networked authentication systems with strong replacements that do not sacrifice user-friendliness for security. Security should be the default, not an option!"

For more information, go to http://srp.stanford.edu/srp.

6.5 PAM - Pluggable Authentication Modules

Newer versions of the Red Hat Linux distribution ship with a unified authentication scheme called "PAM". PAM allows you to change your authentication methods and requirements on the fly, and encapsulate all local authentication methods without recompiling any of your binaries. Configuration of PAM is beyond the scope of this document, but be sure to take a look at the PAM web site for more information. http://www.kernel.org/pub/linux/libs/pam/index.html.

Just a few of the things you can do with PAM:

- Use encryption other than DES for your passwords. (Making them harder to brute-force decode)
- Set resource limits on all your users so they can't perform denial-of-service attacks (number of processes, amount of memory, etc)
- Enable shadow passwords (see below) on the fly
- allow specific users to login only at specific times from specific places

Within a few hours of installing and configuring your system, you can prevent many attacks before they even occur. For example, use PAM to disable the system-wide usage of `.rhosts` files in user's home directories by adding these lines to `/etc/pam.d/rlogin`:

```
#
# Disable rsh/rlogin/rexec for users
#
```

```
login auth required pam_rhosts_auth.so no_rhost
```


6.6 Cryptographic IP Encapsulation (CIPE)

The primary goal of this software is to provide a facility for secure (against eavesdropping, including traffic analysis, and faked message injection) subnetwork interconnection across an insecure packet network such as the Internet.

CIPE encrypts the data at the network level. Packets traveling between hosts on the network are encrypted. The encryption engine is placed near the driver which sends and receives packets.

This is unlike SSH, which encrypts the data by connection, at the socket level. A logical connection between programs running on different hosts is encrypted.

CIPE can be used in tunnelling, in order to create a Virtual Private Network. Low-level encryption has the advantage that it can be made to work transparently between the two networks connected in the VPN, without any change to application software.

Summarized from the CIPE documentation:

> The IPSEC standards define a set of protocols which can be used (among other things) to build encrypted VPNs. However, IPSEC is a rather heavyweight and complicated protocol set with a lot of options, implementations of the full protocol set are still rarely used and some issues (such as key management) are still not fully resolved. CIPE uses a simpler approach, in which many things which can be parameterized (such as the choice of the actual encryption algorithm used) are an install-time fixed choice. This limits flexibility, but allows for a simple (and therefore efficient, easy to debug...) implementation.

Further information can be found at http://www.inka.de/~bigred/devel/cipe.html

As with other forms of cryptography, it is not distributed with the kernel by default due to export restrictions.

6.7 Kerberos

Kerberos is an authentication system developed by the Athena Project at MIT. When a user logs in, Kerberos authenticates that user (using a password), and provides the user with a way to prove her identity to other servers and hosts scattered around the network.

This authentication is then used by programs such as `rlogin` to allow the user to login to other hosts without a password (in place of the `.rhosts` file). This authentication method can also used by the mail system in order to guarantee that mail is delivered to the correct person, as well as to guarantee that the sender is who he claims to be.

Kerberos and the other programs that come with it, prevent users from "spoofing" the system into believing they are someone else. Unfortunately, installing Kerberos is very intrusive, requiring the modification or replacement of numerous standard programs.

TOC

You can find more information about kerberos by looking at the kerberos FAQ, and the code can be found at http://nii.isi.edu/info/kerberos/.

[From: Stein, Jennifer G., Clifford Neuman, and Jeffrey L. Schiller. "Kerberos: An Authentication Service for Open Network Systems." USENIX Conference Proceedings, Dallas, Texas, Winter 1998.]

Kerberos should not be your first step in improving security of your host. It is quite involved, and not as widely used as, say, SSH.

6.8 Shadow Passwords.

Shadow passwords are a means of keeping your encrypted password information secret from normal users. Normally, this encrypted passwords are stored in `/etc/passwd` file for all to read. Anyone can then run password guesser programs on them and attempt to determine what they are. Shadow passwords, by contrast, are saved in `/etc/shadow`, which only privileged users can read. In order to use shadow passwords, you need to make sure all your utilities that need access to password information are recompiled to support them. PAM (above) also allows you to just plug in a shadow module; it doesn't require re-compilation of executables. You can refer to the Shadow-Password HOWTO for further information if necessary. It is available at http://metalab.unc.edu/LDP/HOWTO/Shadow-Password-HOWTO.html It is rather dated now, and will not be required for distributions supporting PAM.

6.9 "Crack" and "John the Ripper"

If for some reason your `passwd` program is not enforcing hard-to-guess passwords, you might want to run a password-cracking program and make sure your users' passwords are secure.

Password cracking programs work on a simple idea: they try every word in the dictionary, and then variations on those words, encrypting each one and checking it against your encrypted password. If they get a match they know what your password is.

There are a number of programs out there...the two most notable of which are "Crack" and "John the Ripper" (http://www.false.com/security/john/index.html) . They will take up a lot of your cpu time, but you should be able to tell if an attacker could get in using them by running them first yourself and notifying users with weak passwords. Note that an attacker would have to use some other hole first in order to read your `/etc/passwd` file, but such holes are more common than you might think.

Because security is only as strong as the most insecure host, it is worth mentioning that if you have any Windows machines on your network, you should check out L0phtCrack, a Crack implementation for Windows. It's available from http://www.l0pht.com

6.10 CFS - Cryptographic File System and TCFS - Transparent Cryptographic File System

TOC

CFS is a way of encrypting an entire directory trees and allowing users to store encrypted files on them. It uses a NFS server running on the local machine. RPMS are available at http://www.replay.com/redhat/, and more information on how it all works is at ftp://ftp.research.att.com/dist/mab/.

TCFS improves on CFS by adding more integration with the file system, so that it's transparent to users that the file system that is encrypted. more information at: http://edu-gw.dia.unisa.it/tcfs/.

It also need not be used on entire filesystems. It works on directories trees as well.

6.11 X11, SVGA and display security

X11

It's important for you to secure your graphical display to prevent attackers from grabbing your passwords as you type them, reading documents or information you are reading on your screen, or even using a hole to gain root access. Running remote X applications over a network also can be fraught with peril, allowing sniffers to see all your interaction with the remote system.

X has a number of access-control mechanisms. The simplest of them is host-based: you use `xhost` to specify what hosts are allowed access to your display. This is not very secure at all, because if someone has access to your machine, they can `xhost +` *`their machine`* and get in easily. Also, if you have to allow access from an untrusted machine, anyone there can compromise your display.

When using `xdm` (X Display Manager) to log in, you get a much better access method: MIT-MAGIC-COOKIE-1. A 128-bit "cookie" is generated and stored in your `.Xauthority` file. If you need to allow a remote machine access to your display, you can use the `xauth` command and the information in your `.Xauthority` file to provide access to only that connection. See the Remote-X-Apps mini-howto, available at http://metalab.unc.edu/LDP/HOWTO/mini/Remote-X-Apps.html.

You can also use `ssh` (see ssh , above) to allow secure X connections. This has the advantage of also being transparent to the end user, and means that no unencrypted data flows across the network.

Take a look at the `Xsecurity` man page for more information on X security. The safe bet is to use `xdm` to login to your console and then use `ssh` to go to remote sites on which you with to run X programs.

SVGA

SVGAlib programs are typically SUID-root in order to access all your Linux machine's video hardware. This makes them very dangerous. If they crash, you typically need to reboot your machine to get a usable console back. Make sure any SVGA programs you are running are authentic, and can at least be somewhat trusted. Even better, don't run them at all.

TOC

GGI (Generic Graphics Interface project)

The Linux GGI project is trying to solve several of the problems with video interfaces on Linux. GGI will move a small piece of the video code into the Linux kernel, and then control access to the video system. This means GGI will be able to restore your console at any time to a known good state. They will also allow a secure attention key, so you can be sure that there is no Trojan horse `login` program running on your console. http://synergy.caltech.edu/~ggi/

---

7. Kernel Security

This is a description of the kernel configuration options that relate to security, and an explanation of what they do, and how to use them.

As the kernel controls your computer's networking, it is important that it be very secure, and not be compromised. To prevent some of the latest networking attacks, you should try to keep your kernel version current. You can find new kernels at ftp://ftp.kernel.org or from your distribution vendor.

There is also a international group providing a single unified crypto patch to the mainstream linux kernel. This patch provides support for a number of cyrptographic subsystems and things that cannot be included in the mainstream kernel due to export restrictions. For more information, visit their web page at: http://www.kerneli.org

7.1 2.0 Kernel Compile Options

For 2.0.x kernels, the following options apply. You should see these options during the kernel configuration process. Many of the comments here are from `./linux/Documentation/Configure.help`, which is the same document that is referenced while using the Help facility during the `make config` stage of compiling the kernel.

- Network Firewalls (CONFIG_FIREWALL)

  This option should be on if you intend to run any firewalling or masquerading on your linux machine. If it's just going to be a regular client machine, it's safe to say no.

- IP: forwarding/gatewaying (CONFIG_IP_FORWARD)

  If you enable IP forwarding, your Linux box essentially becomes a router. If your machine is on a network, you could be forwarding data from one network to another, and perhaps subverting a firewall that was put there to prevent this from happening. Normal dial-up users will want to disable this, and other users should concentrate on the security implications of doing this. Firewall machines will want this enabled, and used in conjunction with firewall software.

  You can enable IP forwarding dynamically using the following command:

  ```
  root#  echo 1 > /proc/sys/net/ipv4/ip_forward
  ```

  and disable it with the command:

  ```
  root#  echo 0 > /proc/sys/net/ipv4/ip_forward
  ```

  Keep in mind the files, and their sizes, do not reflect their actual sizes, and despite being zero-length, may or may not be.

- IP: syn cookies (CONFIG_SYN_COOKIES)

  a "SYN Attack" is a denial of service (DoS) attack that consumes all the resources on your machine, forcing you to reboot. We can't think of a reason you wouldn't normally enable this. In the 2.1 kernel series this config option mearly allows syn cookies, but does not enable them. To enable them, you have to do:

```
root# echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

- IP: Firewalling (CONFIG_IP_FIREWALL)

  This option is necessary if you are going to configure your machine as a firewall, do masquerading, or wish to protect your dial-up workstation from someone entering via your PPP dial-up interface.
- IP: firewall packet logging (CONFIG_IP_FIREWALL_VERBOSE)

  This option gives you information about packets your firewall received, like sender, recipient, port, etc.
- IP: Drop source routed frames (CONFIG_IP_NOSR)

  This option should be enabled. Source routed frames contain the entire path to their destination inside of the packet. This means that routers through which the packet goes do not need to inspect it, and just forward it on. This could lead to data entering your system that may be a potential exploit.
- IP: masquerading (CONFIG_IP_MASQUERADE) If one of the computers on your local network for which your Linux box acts as a firewall wants to send something to the outside, your box can "masquerade" as that host, i.e., it forwards the traffice to the intended destination, but makes it look like it came from the firewall box itself. See http://www.indyramp.com/masq for more information.
- IP: ICMP masquerading (CONFIG_IP_MASQUERADE_ICMP) This option adds ICMP masquerading to the previous option of only masquerading TCP or UDP traffic.
- IP: transparent proxy support (CONFIG_IP_TRANSPARENT_PROXY) This enables your Linux firewall to transparently redirect any network traffice originating from the local network and destined for a remote host to a local server, called a "transparent proxy server". This makes the local computers think they are talking to the remote end, while in fact they are connected to the local proxy. See the IP-Masquerading HOWTO and http://www.indyramp.com/masq for more information.
- IP: always defragment (CONFIG_IP_ALWAYS_DEFRAG)

  Generally this option is disabled, but if you are building a firewall or a masquerading host, you will want to enable it. When data is sent from one host to another, it does not always get sent as a single packet of data, but rather it is fragmented into several pieces. The problem with this is that the port numbers are only stored in the first fragment. This means that someone can insert information into the remaining packets that isn't supposed to be there. It could also prevent a teardrop attack against an internal host that is not yet itself patched against it.
- Packet Signatures (CONFIG_NCPFS_PACKET_SIGNING)

  This is an option that is available in the 2.1 kernel series that will sign NCP packets for stronger security. Normally you can leave it off, but it is there if you do need it.
- IP: Firewall packet netlink device (CONFIG_IP_FIREWALL_NETLINK)

  This is a really neat option that allows you to analyze the first 128 bytes of the packets in a user-space program, to determine if you would like to accept or deny the packet, based on its validity.

7.2 2.2 Kernel Compile Options

For 2.2.x kernels, many of the options are the same, but a few new ones have been developed. Many of the comments here are from `./linux/Documentation/Configure.help`, which is the same document that is referenced while using the Help facility during the `make config` stage of compiling the kernel. Only the newly- added options are listed below. Consult the 2.0 description for a list of other necessary options. The most signficant change in the 2.2 kernel series is the IP firewalling code. The `ipchains` program is now used to install IP firewalling, instead of the `ipfwadm` program used in the 2.0 kernel.

- Socket Filtering (CONFIG_FILTER)

  For most people, it's safe to say no to this option. This option allows you to connect a userspace filter to any socket and determine if packets should be allowed or denied. Unless you have a very specific need and are capable of programming such a filter, you should say no. Also note that as of this writing, all protocols were supported except TCP.
- Port Forwarding Port Forwarding is an addition to IP Masquerading which allows some forwarding of packets from outside to inside a firewall on given ports. This could be useful if, for example, you want to run a web server behind the firewall or masquerading host and that web server should be accessible from the outside world. An external client sends a request to port 80 of the firewall, the firewall forwards this request to the web server, the web server handles the request and the results are sent through the firewall to the original client. The client thinks that the firewall machine itself is running the web server. This can also be used for load balancing if you have a farm of identical web servers behind the firewall. Information about this feature is available from http://www.monmouth.demon.co.uk/ipsubs/portforwarding.html (to browse the WWW, you need to have access to a machine on the Internet that has a program like lynx or netscape). For general info, please see ftp://ftp.compsoc.net/users/steve/ipportfw/linux21/
- Socket Filtering (CONFIG_FILTER) Using this option, user-space programs can attach a filter to any socket and thereby tell the kernel that it should allow or disallow certain types of data to get through the socket. Linux socket filtering works on all socket types except TCP for now. See the text file `./linux/Documentation/networking/filter.txt` for more information.
- IP: Masquerading The 2.2 kernel masquerading has been improved. It provides additional support for masquerading special protocols, etc. Be sure to read the IP Chains HOWTO for more information.

7.3 Kernel Devices

There are a few block and character devices available on Linux that will also help you with security.

The two devices `/dev/random` and `/dev/urandom` are provided by the kernel to provide random data at any time.

Both `/dev/random` and `/dev/urandom` should be secure enough to use in generating PGP keys, `ssh` challenges, and other applications where secure random numbers are requisite. Attackers should be unable to predict the next number given any initial sequence of numbers from these sources. There has been a lot of effort put in to ensuring that the numbers you get from these sources are random in every sense of the word.

The only difference is that `/dev/random` runs out of random bytes and it makes you wait for more to be accumulated. Note that on some systems, it can block for a long time waiting for new user-generated entry to be entered into the system. So you have to use care before using `/dev/random`. (Perhaps the best thing to do is to use it when you're generating sensitive keying information, and you tell the user to pound on the keyboard repeatedly until you print out "OK, enough".)

`/dev/random` is high quality entropy, generated from measuring the inter-interrupt times etc. It blocks until enough bits of random data are available.

`/dev/urandom` is similar, but when the store of entropy is running low, it'll return a cryptographically strong hash of what there is. This isn't as secure, but it's enough for most applications.

You might read from the devices using something like:

```
root#  head -c 6 /dev/urandom | mmencode
```

TOC

This will print six random characters on the console, suitable for password generation. You can find `mmencc` in the `metamail` package.

See `/usr/src/linux/drivers/char/random.c` for a description of the algorithm.

Thanks to Theodore Y. Ts'o, Jon Lewis, and others from Linux-kernel for helping me (Dave) with this.

---

8. Network Security

Network security is becoming more and more important as people spend more and more time connected. Compromising network security is often much easier than compromising physical or local, and is much more common.

There are a number of good tools to assist with network security, and more and more of them are shipping with Linux distributions.

8.1 Packet Sniffers

One of the most common ways intruders gain access to more systems on your network is by employing a packet sniffer on a already compromised host. This "sniffer" just listens on the Ethernet port for things like `passwd` and `login` and `su` in the packet stream and then logs the traffic after that. This way, attackers gain passwords for systems they are not even attempting to break into. Clear-text passwords are very vulnerable to this attack.

Example: Host A has been compromised. Attacker installs a sniffer. Sniffer picks up admin logging into Host B from Host C. It gets the admin's personal password as they login to B. Then, the admin does a `su` to fix a problem. They now have the root password for Host B. Later the admin lets someone `telnet` from his account to Host Z on another site. Now the attacker has a password/login on Host Z.

In this day and age, the attacker doesn't even need to compromise a system to do this: they could also bring a laptop or pc into a building and tap into your net.

Using `ssh` or other encrypted password methods thwarts this attack. Things like APOP for POP accounts also prevents this attack. (Normal POP logins are very vulnerable to this, as is anything that sends clear-text passwords over the network.)

8.2 System services and tcp_wrappers

Before you put your Linux system on *ANY* network the first thing to look at is what services you need to offer. Services that you do not need to offer should be disabled so that you have one less thing to worry about and attackers have one less place to look for a hole.

There are a number of ways to disable services under Linux. You can look at your `/etc/inetd.conf` file and see what services are being offered by your `inetd`. Disable any that you do not need by commenting them out (# at the beginning of the line), and then sending your inetd process a SIGHUP.

You can also remove (or comment out) services in your `/etc/services` file. This will mean that local clients will also be unable to find the service (i.e., if you remove `ftp`, and try and ftp to a remote site from that machine it will fail with an "unknown service" message). It's usually not worth the trouble to remove services, since it provides no additional security. If a local person wanted to use `ftp` even

TOC

though you had commented it out, they would make their own client that use the common FTP port a would still work fine.

Some of the services you might want to leave enabled are:

- `ftp`
- `telnet` (or `ssh`)
- mail, such as `pop-3` or `imap`
- `identd`

If you know you are not going to use some particular package, you can also delete it entirely. `rpm -e` *`packagename`* under the Red Hat distribution will erase an entire package. Under debian `dpkg --remove` does the same thing.

Additionally, you really want to disable the rsh/rlogin/rcp utilities, including login (used by `rlogin`), shell (used by `rcp`), and exec (used by `rsh`) from being started in `/etc/inetd.conf`. These protocols are extremely insecure and have been the cause of exploits in the past.

You should check your `/etc/rc.d/rcN.d`, (where `N` is your systems run level) and see if any of the servers started in that directory are not needed. The files in `/etc/rc.d/rcN.d` are actually symbolic links to the directory `/etc/rc.d/init.d`. Renaming the files in the `init.d` directory has the effect of disabling all the symbolic links in `/etc/rc.d/rcN.d`. If you only wish to disable a service for a particular run level, rename the appropriate file by replacing the upper-case `S` with a lower-case `s`, like this:

```
root#  cd /etc/rc6.d
root#  mv S45dhcpd s45dhcpd
```

If you have BSD style `rc` files, you will want to check `/etc/rc*` for programs you don't need.

Most Linux distributions ship with tcp_wrappers "wrapping" all your TCP services. A tcp_wrapper (`tcpd`) is invoked from `inetd` instead of the real server. `tcpd` then checks the host that is requesting the service, and either executes the real server, or denies access from that host. `tcpd` allows you to restrict access to your TCP services. You should make a `/etc/hosts.allow` and add in only those hosts that need to have access to your machine's services.

If you are a home dialup user, we suggest you deny ALL. `tcpd` also logs failed attempts to access services, so this can give alert you if you are under attack. If you add new services, you should be sure to configure them to use tcp_wrappers if they are TCP based. For example, a normal dial-up user can prevent outsiders from connecting to his machine, yet still have the ability to retrieve mail, and make network connections to the Internet. To do this, you might add the following to your `/etc/hosts.allow`:

ALL: 127.

And of course /etc/hosts.deny would contain:

ALL: ALL

TOC

which will prevent external connections to your machine, yet still allow you from the inside to conne servers on the Internet.

Keep in mind that tcp_wrappers only protect services executed from `inetd`, and a select few others. There very well may be other services running on your machine. You can use `netstat -ta` to find a list of all the services your machine is offering.

8.3 Verify Your DNS Information

Keeping up-to-date DNS information about all hosts on your network can help to increase security. If an unauthorized host becomes connected to your network, you can recognize it by its lack of a DNS entry. Many services can be configured to not accept connections from hosts that do not have valid DNS entries.

8.4 `identd`

`identd` is a small program that typically runs out of your `inetd` server. It keeps track of what user is running what TCP service, and then reports this to whoever requests it.

Many people misunderstand the usefulness of `identd`, and so disable it or block all off site requests for it. `identd` is not there to help out remote sites. There is no way of knowing if the data you get from the remote `identd` is correct or not. There is no authentication in `identd` requests.

Why would you want to run it then? Because it helps *you* out, and is another data-point in tracking. If your `identd` is un compromised, then you know it's telling remote sites the user-name or uid of people using TCP services. If the admin at a remote site comes back to you and tells you user so-and-so was trying to hack into their site, you can easily take action against that user. If you are not running `identd`, you will have to look at lots and lots of logs, figure out who was on at the time, and in general take a lot more time to track down the user.

The `identd` that ships with most distributions is more configurable than many people think. You can disable it for specific users (they can make a `.noident` file), you can log all `identd` requests (We recommend it), you can even have identd return a uid instead of a user name or even NO-USER.

8.5 SATAN, ISS, and Other Network Scanners

There are a number of different software packages out there that do port and service based scanning of machines or networks. SATAN, ISS, SAINT, and Nessus are some of the more well-known ones. This software connects to the target machine (or all the target machines on a network) on all the ports they can, and try to determine what service is running there. Based on this information, you can tell if the machine is vulnerable to a specific exploit on that server.

SATAN (Security Administrator's Tool for Analyzing Networks) is a port scanner with a web interface. It can be configured to do light, medium, or strong checks on a machine or a network of machines. It's a good idea to get SATAN and scan your machine or network, and fix the problems it finds. Make sure

you get the copy of SATAN from metalab or a reputable FTP or web site. There was a Trojan copy of SATAN that was distributed out on the net. http://www.trouble.org/~zen/satan/satan.html. Note that SATAN has not been updated in quite a while, and some of the other tools below might do a better job.

ISS (Internet Security Scanner) is another port-based scanner. It is faster than Satan, and thus might be better for large networks. However, SATAN tends to provide more information.

Abacus is a suite of tools to provide host based security and intrusion detection. look at it's home page on the web for more information. http://www.psionic.com/abacus/

SAINT is a updated version of SATAN. It is web based and has many more up to date tests than SATAN. You can find out more about it at: http://www.wwdsi.com/~saint

Nessus is a free security scanner. It has a GTK graphical interface for ease of use. It is also designed with a very nice plugin setup for new port scanning tests. For more information, take a look at: http://www.nessus.org

Detecting Port Scans

There are some tools designed to alert you to probes by SATAN and ISS and other scanning software. However, liberal use of tcp_wrappers, and make sure to look over your log files regularly, you should be able to notice such probes. Even on the lowest setting, SATAN still leaves traces in the logs on a stock Red Hat system.

There are also "stealth" port scanners. A packet with the TCP ACK bit set (as is done with established connections) will likely get through a packet-filtering firewall. The returned RST packet from a port that *_had no established session_* can be taken as proof of life on that port. I don't think TCP wrappers will detect this.

8.6 `sendmail`, `qmail` and MTA's

One of the most important services you can provide is a mail server. Unfortunately, it is also one of the most vulnerable to attack, simply due to the number of tasks it must perform and the privileges it typically needs.

If you are using `sendmail` it is very important to keep up on current versions. `sendmail` has a long long history of security exploits. Always make sure you are running the most recent version from http://www.sendmail.org.

Keep in mind that sendmail does not have to be running in order for you to send mail. If you are a home user, you can disable sendmail entirely, and simply use your mail client to send mail. You might also choose to remove the "-bd" flag from the sendmail startup file, thereby disabling incoming requests for mail. In other words, you can execute sendmail from your startup script using the following instead:

```
# /usr/lib/sendmail -q15m
```

This will cause sendmail to flush the mail queue every fifteen minutes for any messages that could not be

successfully delivered on the first attempt.

Many administrators choose not to use sendmail, and instead choose one of the other mail transport agents. You might consider switching over to `qmail`. `qmail` was designed with security in mind from the ground up. It's fast, stable, and secure. Qmail can be found at http://www.qmail.org

In direct competition to qmail is "postfix", written by Wietse Venema, the author of tcp_wrappers and other security tools. Formerly called vmailer, and sponsored by IBM, this is also a mail transport agent written from the ground up with security in mind. You can find more information about vmailer at http://www.postfix.org

8.7 Denial of Service Attacks

A "Denial of Service" (DoS) attack is one where the attacker tries to make some resource too busy to answer legitimate requests, or to deny legitimate users access to your machine.

Denial of service attacks have increased greatly in recent years. Some of the more popular and recent ones are listed below. Note that new ones show up all the time, so this is just a few examples. Read the Linux security lists and the bugtraq list and archives for more current information.

- **SYN Flooding** - SYN flooding is a network denial of service attack. It takes advantage of a "loophole" in the way TCP connections are created. The newer Linux kernels (2.0.30 and up) have several configurable options to prevent SYN flood attacks from denying people access to your machine or services. See Kernel Security for proper kernel protection options.
- **Pentium "F00F" Bug** - It was recently discovered that a series of assembly codes sent to a genuine Intel Pentium processor would reboot the machine. This affects every machine with a Pentium processor (not clones, not Pentium Pro or PII), no matter what operating system it's running. Linux kernels 2.0.32 and up contain a work around for this bug, preventing it from locking your machine. Kernel 2.0.33 has an improved version of the kernel fix, and is suggested over 2.0.32. If you are running on a Pentium, you should upgrade now!
- **Ping Flooding** - Ping flooding is a simple brute-force denial of service attack. The attacker sends a "flood" of ICMP packets to your machine. If they are doing this from a host with better bandwidth than yours, your machine will be unable to send anything on the network. A variation on this attack, called "smurfing", sends ICMP packets to a host with *your* machine's return IP, allowing them to flood you less detectably. You can find more information about the "smurf" attack at http://www.quadrunner.com/~chuegen/smurf.txt

  If you are ever under a ping flood attack, use a tool like `tcpdump` to determine where the packets are coming from (or appear to be coming from), then contact your provider with this information. Ping floods can most easily be stopped at the router level or by using a firewall.
- **Ping o' Death** - The Ping o' Death attack sends ICMP ECHO REQUEST packets that are too large to fit in the kernel data structures intended to store them. Because sending a single, large (65,510 bytes) "ping" packet to many systems will cause them to hang or even crash, this problem was quickly dubbed the "Ping o' Death." This one has long been fixed, and is no longer anything to worry about.
- **Teardrop / New Tear** - One of the most recent exploits involves a bug present in the IP

fragmentation code on Linux and Windows platforms. It is fixed in kernel version 2.0.33, and not require selecting any kernel compile-time options to utilize the fix. Linux is apparently not vulnerable to the "newtear" exploit.

TOC

You can find code for most exploits, and a more in-depth description of how they work, at http://www.rootshell.com using their search engine.

8.8 NFS (Network File System) Security.

NFS is a very widely-used file sharing protocol. It allows servers running `nfsd` and `mountd` to "export" entire filesystems to other machines using NFS filesystem support built in to their kernels (or some other client support if they are not Linux machines). `mountd` keeps track of mounted filesystems in `/etc/mtab`, and can display them with `showmount`.

Many sites use NFS to serve home directories to users, so that no matter what machine in the cluster they login to, they will have all their home files.

There is some small amount of security allowed in exporting filesystems. You can make your `nfsd` map the remote root user (uid=0) to the `nobody` user, denying them total access to the files exported. However, since individual users have access to their own (or at least the same uid) files, the remote root user can login or `su` to their account and have total access to their files. This is only a small hindrance to an attacker that has access to mount your remote filesystems.

If you must use NFS, make sure you export to only those machines that you really need to. Never export your entire root directory; export only directories you need to export.

See the NFS HOWTO for more information on NFS, available at http://metalab.unc.edu/mdw/HOWTO/NFS-HOWTO.html

8.9 NIS (Network Information Service) (formerly YP).

Network Information service (formerly YP) is a means of distributing information to a group of machines. The NIS master holds the information tables and converts them into NIS map files. These maps are then served over the network, allowing NIS client machines to get login, password, home directory and shell information (all the information in a standard `/etc/passwd` file). This allows users to change their password once and have it take effect on all the machines in the NIS domain.

NIS is not at all secure. It was never meant to be. It was meant to be handy and useful. Anyone that can guess the name of your NIS domain (anywhere on the net) can get a copy of your passwd file, and use "crack" and "John the Ripper" against your users' passwords. Also, it is possible to spoof NIS and do all sorts of nasty tricks. If you must use NIS, make sure you are aware of the dangers.

There is a much more secure replacement for NIS, called NIS+. Check out the NIS HOWTO for more information: http://metalab.unc.edu/mdw/HOWTO/NIS-HOWTO.html

8.10 Firewalls

TOC

Firewalls are a means of controlling what information is allowed into and out of your local network. Typically the firewall host is connected to the Internet and your local LAN, and the only access from your LAN to the Internet is through the firewall. This way the firewall can control what passes back and forth from the Internet and your lan.

There are a number of types of firewalls and methods of setting them up. Linux machines make pretty good firewalls. Firewall code can be built right into 2.0 and higher kernels. The `ipfwadm` for 2.0 kernels, or `ipchains` for 2.2 kernels, user-space tools allows you to change, on the fly, the types of network traffic you allow. You can also log particular types of network traffic.

Firewalls are a very useful and important technique in securing your network. However, never think that because you have a firewall, you don't need to secure the machines behind it. This is a fatal mistake. Check out the very good `Firewall-HOWTO` at your latest metalab archive for more information on firewalls and Linux. http://metalab.unc.edu/mdw/HOWTO/Firewall-HOWTO.html

More information can also be found in the IP-Masquerade mini-howto: http://metalab.unc.edu/mdw/HOWTO/mini/IP-Masquerade.html

More information on `ipfwadm` (The tool that lets you change settings on your firewall, can be found at it's home page: http://www.xos.nl/linux/ipfwadm/

If you have no experience with firewalls, and plan to set up one for more than just a simple security policy, the Firewalls book by O'Reilly and Associates or other online firewall document is mandatory reading. Check out http://www.ora.com for more information. The National Institute of Standards and Technology have put together an excellent document on firewalls. Although dated 1995, it is still quite good. You can find it at http://csrc.nist.gov/nistpubs/800-10/main.html. Also of interest includes:

- The Freefire Project -- a list of freely-available firewall tools, available at http://sites.inka.de/sites/lina/freefire-l/index_en.html
- SunWorld Firewall Design -- written by the authors of the O'Reilly book, this provides a rough introduction to the different firewall types. It's available at http://www.sunworld.com/swol-01-1996/swol-01-firewall.html

8.11 IP Chains - Linux Kernel 2.2.x Firewalling

Linux IP Firewalling Chains is an update to the 2.0 Linux firewalling code for the 2.2 kernel. It has a great deal more features than previous implementations, including:

- More flexible packet manipulations
- More complex accounting
- Simple policy changes possible atomically
- Fragments can be explicitly blocked, denied, etc.
- Logs suspicious packets.

- Can handle protocols other than ICMP/TCP/UDP.

If you are currently using ipfwadm on your 2.0 kernel, there are scripts available to convert the ipfwadm command format to the format ipchains uses.

Be sure to read the IP Chains HOWTO for further information. It is avilable at http://www.rustcorp.com/linux/ipchains/HOWTO.html

8.12 VPN's - Virtual Private Networks

VPN's are a way to establish a "virtual" network on top of some already existing network. This virtual network often is encrypted and passes traffic only to and from some known entities that have joined the network. VPN's are often used to connect someone working at home over the public internet to a internal company network by using a encrypted virtual network.

If you are running a linux masquerading firewall and need to pass MS PPTP (Microsoft's VPN point to point product) packets, there is a linux kernel patch out to do just that. See: ip-masq-vpn.

There are several linux VPN solutions available:

- vpnd. See the http://www.crosswinds.net/nuremberg/~anstein/unix/vpnd.html.
- Free S/Wan, available at http://www.xs4all.nl/~freeswan/
- ssh can be used to construct a VPN. See the VPN mini-howto for more information.
- vps (virtual private server) at http://www.strongcrypto.com.

See also the section on IPSEC for pointers and more information.

---

9. Security Preparation (before you go on-line)

Ok, so you have checked over your system, and determined it's as secure as feasible, and you're ready to put it online. There are a few things you should now do in order to prepare for an intrusion, so you can quickly disable the intruder, and get back up and running.

9.1 Make a Full Backup of Your Machine

Discussion of backup methods and storage is beyond the scope of this document, but here are a few words relating to backups and security:

If you have less than 650mb of data to store on a partition, a CD-R copy of your data is a good way to go (as it's hard to tamper with later, and if stored properly can last a long time). Tapes and other re-writable media should be write-protected as soon as your backup is complete, and then verified to prevent tampering. Make sure you store your backups in a secure off-line area. A good backup will ensure that you have a known good point to restore your system from.

9.2 Choosing a Good Backup Schedule

A six-tape cycle is easy to maintain. This includes four tapes for during the week, one tape for even Fridays, and one tape for odd Fridays. Perform an incremental backup every day, and a full backup on the appropriate Friday tape. If you make some particularly important changes or add some important data to your system, a full backup might well be in order.

9.3 Backup Your RPM or Debian File Database

In the event of an intrusion, you can use your RPM database like you would use `tripwire`, but only if you can be sure it too hasn't been modified. You should copy the RPM database to a floppy, and keep this copy off-line at all times. The Debian distribution likely has something similar.

The files `/var/lib/rpm/fileindex.rpm` and `/var/lib/rpm/packages.rpm` most likely won't fit on a single floppy. But if Compressed, each should fit on a seperate floppy.

Now, when your system is compromised, you can use the command:

```
root#  rpm -Va
```

to verify each file on the system. See the `rpm` man page, as there are a few other options that can be included to make it less verbose. Keep in mind you must also be sure your RPM binary has not been compromised.

This means that every time a new RPM is added to the system, the RPM database will need to be

rearchived. You will have to decide the advantages versus drawbacks.

9.4 Keep Track of Your System Accounting Data

It is very important that the information that comes from `syslog` has not been compromised. Making the files in `/var/log` readable and writable by only a limited number of users is a good start.

Be sure to keep an eye on what gets written there, especially under the `auth` facility. Multiple login failures, for example, can indicate an attempted break-in.

Where to look for your log file will depend on your distribution. In a Linux system that conforms to the "Linux Filesystem Standard", such as Red Hat, you will want to look in `/var/log` and check `messages`, `mail.log`, and others.

You can find out where your distribution is logging to by looking at your `/etc/syslog.conf` file. This is the file that tells `syslogd` (the system logging daemon) where to log various messages.

You might also want to configure your log-rotating script or daemon to keep logs around longer so you have time to examine them. Take a look at the `logrotate` package on recent Red Hat distributions. Other distributions likely have a similar process.

If your log files have been tampered with, see if you can determine when the tampering started, and what sort of things appeared to be tampered with. Are there large periods of time that cannot be accounted for? Checking backup tapes (if you have any) for untampered log files is a good idea.

Log files are typically modified by the intruder in order to cover his tracks, but they should still be checked for strange happenings. You may notice the intruder attempting to gain entrance, or exploit a program in order to obtain the root account. You might see log entries before the intruder has time to modify them.

You should also be sure to seperate the `auth` facility from other log data, including attempts to switch users using `su`, login attempts, and other user accounting information.

If possible, configure `syslog` to send a copy of the most important data to a secure system. This will prevent an intruder from covering his tracks by deleting his login/su/ftp/etc attempts. See the `syslog.conf` man page, and refer to the `@` option.

There are several more advanced `syslogd` programs out there. Take a look at http://www.core-sdi.com/ssyslog/ for Secure Syslog. Secure Syslog allows you to encrypt your syslog entries and make sure no one has tampered with them.

Another `syslogd` with more features is syslog-ng. It allows you a lot more flexability in your logging and also can has your remote syslog streams to prevent tampering.

Finally, log files are much less useful when no one is reading them. Take some time out every once in a while to look over your log files, and get a feeling for what they look like on a normal day. Knowing this can help make unusual things stand out.

9.5 Apply All New System Updates.

TOC

Most Linux users install from a CD-ROM. Due to the fast-paced nature of security fixes, new (fixed) programs are always being released. Before you connect your machine to the network, it's a good idea to check with your distribution's ftp site and get all the updated packages since you received your distribution CD-ROM. Many times these packages contain important security fixes, so it's a good idea to get them installed.

---

10. What To Do During and After a Breakin

So you have followed some of the advice here (or elsewhere) and have detected a break-in? The first thing to do is to remain calm. Hasty actions can cause more harm than the attacker would have.

10.1 Security Compromise Underway.

Spotting a security compromise under way can be a tense undertaking. How you react can have large consequences.

If the compromise you are seeing is a physical one, odds are you have spotted someone who has broken into your home, office or lab. You should notify your local authorities. In a lab, you might have spotted someone trying to open a case or reboot a machine. Depending on your authority and procedures, you might ask them to stop, or contact your local security people.

If you have detected a local user trying to compromise your security, the first thing to do is confirm they are in fact who you think they are. Check the site they are logging in from. Is it the site they normally log in from? No? Then use a non-electronic means of getting in touch. For instance, call them on the phone or walk over to their office/house and talk to them. If they agree that they are on, you can ask them to explain what they were doing or tell them to cease doing it. If they are not on, and have no idea what you are talking about, odds are this incident requires further investigation. Look into such incidents , and have lots of information before making any accusations.

If you have detected a network compromise, the first thing to do (if you are able) is to disconnect your network. If they are connected via modem, unplug the modem cable; if they are connected via ethernet, unplug the Ethernet cable. This will prevent them from doing any further damage, and they will probably see it as a network problem rather than detection.

If you are unable to disconnect the network (if you have a busy site, or you do not have physical control of your machines), the next best step is to use something like `tcp_wrappers` or `ipfwadm` to deny access from the intruder's site.

If you can't deny all people from the same site as the intruder, locking the user's account will have to do. Note that locking an account is not an easy thing. You have to keep in mind `.rhosts` files, FTP access, and a host of possible backdoors).

After you have done one of the above (disconnected the network, denied access from their site, and/or disabled their account), you need to kill all their user processes and log them off.

You should monitor your site well for the next few minutes, as the attacker will try to get back in. Perhaps using a different account, and/or from a different network address.

10.2 Security Compromise has already happened

TOC

So you have either detected a compromise that has already happened or you have detected it and locked (hopefully) the offending attacker out of your system. Now what?

Closing the Hole

If you are able to determine what means the attacker used to get into your system, you should try to close that hole. For instance, perhaps you see several FTP entries just before the user logged in. Disable the FTP service and check and see if there is an updated version, or if any of the lists know of a fix.

Check all your log files, and make a visit to your security lists and pages and see if there are any new common exploits you can fix. You can find Caldera security fixes at http://www.caldera.com/tech-ref/security/. Red Hat has not yet seperated their security fixes from bug fixes, but their distribution errata is available at http://www.redhat.com/errata

Debian now has a security mailing list and web page. See: http://www.debian.com/security/ for more information.

It is very likely that if one vendor has released a security update, that most other Linux vendors will as well.

There is now a linux security auditing project. They are methodically going through all the user space utilities and looking for possible security exploits and overflows. From their announcement:

> "We are attempting a systematic audit of Linux sources with a view to being as secure as OpenBSD. We have already uncovered (and fixed) some problems, but more help is welcome. The list is unmoderated and also a useful resource for general security discussions. The list address is: security-audit@ferret.lmh.ox.ac.uk To subscribe, send a mail to: security-audit-subscribe@ferret.lmh.ox.ac.uk"

If you don't lock the attacker out, they will likely be back. Not just back on your machine, but back somewhere on your network. If they were running a packet sniffer, odds are good they have access to other local machines.

Assessing the Damage

The first thing is to assess the damage. What has been compromised? If you are running an Integrity Checker like `Tripwire`, you can use it to perform an integrity check, and should help to tell you. If not, you will have to look around at all your important data.

Since Linux systems are getting easier and easier to install, you might consider saving your config files and then wiping your disk(s) and reinstalling, then restoring your user files from backups and your config files. This will ensure that you have a new, clean system. If you have to backup files from the compromised system, be especially cautious of any binaries that you restore, as they may be Trojan horses placed there by the intruder.

Re-installation should be considered mandatory upon an intruder obtaining root access. Additionally,

you'd like to keep any evidence there is, so having a spare disk in the safe may make sense.

Then you have to worry about how long ago the compromise happened, and whether the backups hold any damaged work. More on backups later.

Backups, Backups, Backups!

Having regular backups is a godsend for security matters. If your system is compromised, you can restore the data you need from backups. Of course, some data is valuable to the attacker too, and they will not only destroy it, they will steal it and have their own copies; but at least you will still have the data.

You should check several backups back into the past before restoring a file that has been tampered with. The intruder could have compromised your files long ago, and you could have made many successful backups of the compromised file!!!

Of course, there are also a raft of security concerns with backups. Make sure you are storing them in a secure place. Know who has access to them. (If an attacker can get your backups, they can have access to all your data without you ever knowing it.)

Tracking Down the Intruder.

Ok, you have locked the intruder out, and recovered your system, but you're not quite done yet. While it is unlikely that most intruders will ever be caught, you should report the attack.

You should report the attack to the admin contact at the site where the attacker attacked your system. You can look up this contact with `whois` or the Internic database. You might send them an email with all applicable log entries and dates and times. If you spotted anything else distinctive about your intruder, you might mention that too. After sending the email, you should (if you are so inclined) follow up with a phone call. If that admin in turn spots your attacker, they might be able to talk to the admin of the site where they are coming from and so on.

Good crackers often use many intermediate systems, some (or many) of which may not even know they have been compromised. Trying to track a cracker back to their home system can be difficult. Being polite to the admins you talk to can go a long way to getting help from them.

You should also notify any security organizations you are a part of (CERT or similar), as well as your Linux system vendor.

---

11. Security Sources

There are a LOT of good sites out there for Unix security in general and Linux security specifically. It's very important to subscribe to one (or more) of the security mailing lists and keep current on security fixes. Most of these lists are very low volume, and very informative.

11.1 FTP Sites

CERT is the Computer Emergency Response Team. They often send out alerts of current attacks and fixes. See ftp://ftp.cert.org for more information.

Replay (http://www.replay.com) has archives of many security programs. Since they are outside the US, they don't need to obey US crypto restrictions.

Matt Blaze is the author of CFS and a great security advocate. Matt's archive is available at ftp://ftp.research.att.com/pub/mab

`tue.nl` is a great security FTP site in the Netherlands. ftp.win.tue.nl

11.2 Web Sites

- The Hacker FAQ is a FAQ about hackers: The Hacker FAQ
- The COAST archive has a large number of Unix security programs and information: COAST
- SuSe Security Page: http://www.suse.de/security/
- Rootshell.com is a great site for seeing what exploits are currently being used by crackers: http://www.rootshell.com/
- BUGTRAQ puts out advisories on security issues: BUGTRAQ archives
- CERT, the Computer Emergency Response Team, puts out advisories on common attacks on unix platforms: CERT home
- Dan Farmer is the author of SATAN and many other security tools. His home site has some interesting security survey information, as well as security tools: http://www.trouble.org
- The Linux security WWW is a good site for Linux security information: Linux Security WWW
- Infilsec has a vulnerability engine that can tell you what vunerabilities affect a specific platform: http://www.infilsec.com/vulnerabilities/
- CIAC sends out periodic security bulletins on common exploits: http://ciac.llnl.gov/cgi-bin/index/bulletins
- A good starting point for Linux Pluggable Authentication modules can be found at

http://www.kernel.org/pub/linux/libs/pam/.

- The debian project has a web page for their security fixes and information. It is at http://www.debian.com/security/.
- WWW Security FAQ, written by Lincoln Stein, is a great web security reference. Find it at http://www.w3.org/Security/Faq/www-security-faq.html

11.3 Mailing Lists

Bugtraq: To subscribe to bugtraq, send mail to listserv@netspace.org containing the message body subscribe bugtraq. (see links above for archives).

CIAC: Send e-mail to majordomo@tholia.llnl.gov. In the BODY (not subject) of the message put (either or both): subscribe ciac-bulletin

Red Hat has a number of mailing lists, the most important of which is the redhat-announce list. You can read about security (and other) fixes as soon as they come out. Send email to majordomo@redhat.com and put subscribe redhat-announce.

The Debian project has a security mailing list that covers their security fixes. see http://www.debian.com/security/ for more information.

11.4 Books - Printed Reading Material

There are a number of good security books out there. This section lists a few of them. In addition to the security specific books, security is covered in a number of other books on system administration.

Building Internet Firewalls By D. Brent Chapman & Elizabeth D. Zwicky

1st Edition September 1995

ISBN: 1-56592-124-0

Practical UNIX & Internet Security, 2nd Edition By Simson Garfinkel & Gene Spafford

2nd Edition April 1996

ISBN: 1-56592-148-8

Computer Security Basics By Deborah Russell & G.T. Gangemi, Sr.

1st Edition July 1991

ISBN: 0-937175-71-4

Linux Network Administrator's Guide By Olaf Kirch

1st Edition January 1995

ISBN: 1-56592-087-2

PGP: Pretty Good Privacy By Simson Garfinkel

1st Edition December 1994

ISBN: 1-56592-098-8

Computer Crime A Crimefighter's Handbook By David Icove, Karl Seger & William VonStorch (Consulting Editor Eugene H. Spafford)

1st Edition August 1995

ISBN: 1-56592-086-4

---

12. Glossary

- **authentication:** The property of knowing that the data received is the same as the data that was sent, and that the claimed sender is in fact the actual sender.
- **bastion Host:** A computer system that must be highly secured because it is vulnerable to attack, usually because it is exposed to the Internet and is a main point of contact for users of internal networks. It gets its name from the highly fortified projects on the outer walls of medieval castles. Bastions overlook critical areas of defense, usually having strong walls, room for extra troops, and the occasional useful tub of boiling hot oil for discouraging attackers.
- **buffer overflow:** Common coding style is to never allocate large enough buffers, and to not check for overflows. When such buffers overflow, the executing program (daemon or set-uid program) can be tricked in doing some other things. Generally this works by overwriting a function's return address on the stack to point to another location.
- **denial of service:** A denial of service attack is when an attacker consumes the resources on your computer for things it was not intended to be doing, thus preventing normal use of your network resources for legimite purposes.
- **dual-homed Host:** A general-purpose computer system that has at least two network interfaces.
- **firewall:** A component or set of components that restricts access between a protected network and the Internet, or between other sets of networks.
- **host:** A computer system attached to a network.
- **IP spoofing:** IP Spoofing is a complex technical attack that is made up of several components. It is a security exploit that works by tricking computers in a trust-relationship that you are someone that you really aren't. There is an extensive paper written by daemon9, route, and infinity in the Volume Seven, Issue fourty-Eight issue of Phrack Magazine.
- **non-repudiation:** The property of a receiver being able to prove that the sender of some data did in fact send the data even though the sender might later deny ever having sent it.
- **packet:** The fundamental unit of communication on the Internet.
- **packet filtering:** The action a device takes to selectively control the flow of data to and from a network. Packet filters allow or block packets, usually while routing them from one network to another (most often from the Internet to an internal network, and vice-versa). To accomplish packet filtering, you set up rules that specify what types of packets (those to or from a particular IP address or port) are to be allowed and what types are to be blocked.
- **perimeter network:** A network added between a protected network and an external network, in order to provide an additional layer of security. A perimeter network is sometimes called a DMZ.
- **proxy server:** A program that deals with external servers on behalf of internal clients. Proxy clients talk to proxy servers, which relay approved client requests to real servers, and relay answers back to clients.
- **superuser:** An informal name for `root`.

13. Frequently Asked Questions

1. Is it more secure to compile driver support directly into the kernel, instead of making it a module?

   Answer: Some people think it is better to disable the ability to load device drivers using modules, because an intruder could load a Trojan module or a module that could affect system security.

   However, in order to load modules, you must be root. The module object files are also only writable by root. This means the intruder would need root access to insert a module. If the intruder gains root access, there are more serious things to worry about than whether he will load a module.

   Modules are for dynamically loading support for a particular device that may be infrequently used. On server machines, or firewalls for instance, this is very unlikely to happen. For this reason, it would make more sense to compile support directly into the kernel for machines acting as a server. Modules are also slower than support compiled directly in the kernel.
2. Why does logging in as root from a remote machine always fail?

   Answer: See Root Security. This is done intentionally to prevent remote users from attempting to connect via `telnet` to your machine as `root`, which is a serious security vulnerability. Don't forget: potential intruders have time on their side, and can run automated programs to find your password.
3. How do I enable shadow passwords on my Red Hat 4.2 or 5.x Linux box?

   Answer: Shadow passwords is a mechanism for storing your password in a file other than the normal `/etc/passwd` file. This has several advantages. The first one is that the shadow file, `/etc/shadow`, is only readable by root, unlike `/etc/passwd`, which must remain readable by everyone. The other advantage is that as the administrator, you can enable or disable accounts without everyone knowing the status of other users' accounts.

   The `/etc/passwd` file is then used to store user and group names, used by programs like `/bin/ls` to map the user ID to the proper username in a directory listing.

   The `/etc/shadow` file then only contains the username and his/her password, and perhaps accounting information, like when the account expires, etc.

   To enable shadow passwords, run `pwconv` as root, and `/etc/shadow` should now exist, and be used by applications. Since you are using RH 4.2 or above, the PAM modules will automatically adapt to the change from using normal `/etc/passwd` to shadow passwords without any other change.

   Since you're interested in securing your passwords, perhaps you would also be interested in generating good passwords to begin with. For this you can use the `pam_cracklib` module, which is part of PAM. It runs your password against the Crack libraries to help you decide if it is too easily guessable by password cracking programs.

TOC

4. How can I enable the Apache SSL extensions?

   Answer:

   1.Get SSLeay 0.8.0 or later from ftp://ftp.psy.uq.oz.au/pub/Crypto/SSL

   2.Build and test and install it!

   3.Get Apache 1.2.5 source

   4.Get Apache SSLeay extensions from here

   5.Unpack it in the apache-1.2.5 source directory and patch Apache as per the README.

   6.Configure and build it.

   You might also try Replay Associates which has many pre-built packages, and is located outside of the United States.

5. How can I manipulate user accounts, and still retain security?

   Answer: The Red Hat distribution, especially RH5.0, contains a great number of tools to change the properties of user accounts.

   - The `pwconv` and `unpwconv` programs can be used to convert between shadow and non-shadowed passwords.
   - The `pwck` and `grpck` programs can be used to verify proper organization of the `passwd` and `group` files.
   - The `useradd`, `usermod`, and `userdel` programs can be used to add, delete and modify user accounts. The `groupadd`, `groupmod`, and `groupdel` programs will do the same for groups.
   - Group passwords can be created using `gpasswd`.

   All these programs are "shadow-aware" -- that is, if you enable shadow they will use `/etc/shadow` for password information, otherwise it won't.

   See the respective man pages for further information.

6. How can I password protect specific HTML documents using Apache?

   I bet you didn't know about http://www.apacheweek.org, did you?

   You can find information on user Authentication at http://www.apacheweek.com/features/userauth as well as other web server security tips from http://www.apache.org/docs/misc/security_tips.html

---

14. Conclusion

By subscribing to the security alert mailing lists, and keeping current, you can do a lot towards securing your machine. If you pay attention to your log files and run something like `tripwire` regularly, you can do even more.

A reasonable level of computer security is not difficult to maintain on a home machine. More effort is required on business machines, but Linux can indeed be a secure platform. Due to the nature of Linux development, security fixes often come out much faster than they do on commercial operating systems, making Linux an ideal platform when security is a requirement.

15. Acknowledgements

Information here is collected from many sources. Thanks to the following that either indirectly or directly have contributed: following who either indirectly or directly have contributed:

Rob Riggs rob@DevilsThumb.com

S. Coffin scoffin@netcom.com

Viktor Przebinda viktor@CRYSTAL.MATH.ou.edu

Roelof Osinga roelof@eboa.com

Kyle Hasselbacher kyle@carefree.quux.soltc.net

David S. Jackson dsj@dsj.net

Todd G. Ruskell ruskell@boulder.nist.gov

Rogier Wolff R.E.Wolff@BitWizard.nl

Antonomasia ant@notatla.demon.co.uk

Nic Bellamy sky@wibble.net

Eric Hanchrow offby1@blarg.net

Robert J. Berger rberger@ibd.com

Ulrich Alpers lurchi@cdrom.uni-stuttgart.de

David Noha dave@c-c-s.com

The following have translated this HOWTO into various other languages!

A special thank you to all of them for help spreading the linux word...

Polish: Ziemek Borowski ziembor@FAQ-bot.ZiemBor.Waw.PL

Japanese: FUJIWARA Teruyoshi fjwr@mtj.biglobe.ne.jp

Indonesian: Tedi Heriyanto 22941219@students.ukdw.ac.id

Linux IR HOWTO

Werner Heuser, < wehe@snafu.de >

v2.4, 9 February 1999

An introduction to Linux and infrared devices and how to use the software provided by the Linux/IrDA project. This package uses IrDA(TM) compliant standards. IrDA(TM) is an industrial standard for infrared wireless communication, and most laptops made after January 1996 are equipped with an IrDA(TM) compliant infrared transceiver. Infrared ports let you communicate with printers, modems, fax machines, LANs, and other laptops. Speed ranges from 2400bps to 4Mbps. The Linux/IrDA stack supports IrLAP, IrLMP, IrIAS, IrIAP, IrLPT, IrCOMM, IrOBEX, and IrLAN. Several of the protocols are implemented as both clients and servers. There is also support for multiple IrLAP connections, via several IrDA(TM) devices at once. The Linux/IrDA project started at the end of 1997 and its status is still experimental, so please don't expect every feature working straight. AFAIK Linux/IrDA is the _only_ open source IrDA implementation currently available. Remote Control (RC) via infrared is not the aim of the project, though partly treated in this HOWTO.

1. Introduction

2. Prerequisites

3. Kernel

- 3.1 General Parameters
- 3.2 IrDA Specific Parameters

4. Linux/IrDA-Utils

5. Configuration

- 5.1 General Configuration
- 5.2 IrManager
- 5.3 Low Level Drivers

TOC

6. Specific Connections and Protocols

- 6.1 Printer Connection - IrLPT
- 6.2 LAN Connection - IrLAN
- 6.3 Palm III Connection - IrOBEX
- 6.4 Cellular Phone Connection
- 6.5 Digital Camera Connection
- 6.6 Window$95 and Linux/IrDA
- 6.7 Linux to Linux Connection
- 6.8 Multiple Instances

7. Hardware Supported by Linux/IrDA

- 7.1 Obtaining Information about the Infrared Port in Laptops
- 7.2 Hardware Overviews

8. GUI

9. Power Saving

10. Troubleshooting, Mailing List

- 10.1 General Information
- 10.2 Troubleshooting Techniques

11. Known Bugs

12. FAQ

13. Infrared Remote Control

- 13.1 Resources
- 13.2 Infrared Remote Control - IrDA

14. Infrared and Eye Safety

15. Credits

16. Revision History

17. Copyright and Disclaimer

1. Introduction

Better red, than dead. - Unknown AuthorEss

Since 2.1.131 and 2.2.0 Linux/IrDA is part of the kernel. Please note that the status of the project is still experimental. If you need information about Linux/IrDA support for 2.0.x kernels please look into the former version of this HOWTO at http://www.snafu.de/~wehe/index_li.html. But AFAIK this package is no longer maintained.

Companies and developers which are interested in joining these efforts should contact the Linux/IrDA project at http://www.cs.uit.no/~dagb/irda or me at < wehe@snafu.de >.

Some history about Linux/IrDA. The project started at the end of 1997 with the name Linux/IrDA. Due to some troubles with the name IrDA, which is trademarked by the Infrared Data Association IrDA http://www.irda.org/, the name was changed to Linux/IR. At the end of 1998 the the relationship between both became better and the name was changed to Linux/IrDA again.

This document is based on the "How to use" part of the Linux/IrDA project homepage http://www.cs.uit.no/~dagb/irda/howto.html. I also included material provided by the Linux/IrDA core team, the Linux/IrDA mailing list and other sources.

The document is included in the LINUX DOCUMENTATION PROJECT http://metalab.unc.edu/LDP.

The latest version of this document is available at http://www.snafu.de/~wehe/index_li.html

I tried to check all information but I don't have all the necessary infrared hardware yet, so if something doesn't work for you, please don't blame me.

Please feel free to contact me for comments or questions. I know this material is not finished or perfect, but I hope you find it useful anyway.

<Werner Heuser>

2. Prerequisites

- BIOS

  - Make sure your infrared port is enabled in the BIOS and check what interrupt and port address it uses.
- Infrared Controller Chip

  - Make sure your infrared port is detected by the Linux kernel. For detailed information see the "Hardware Overview" section below.
- modutils

  - Make sure you use modutils 2.1.x by `insmod --version`. I use version 2.1.121.
- Shared Library

  - The shared library libc.so.5 and the loader ld-linux.so.1 has to be available.

  - But glibc2 aka libc6 should work also.

  - I am not sure wether you need the zlib library if you use the data compression features.
- GUI

  Currently there are two GraphicalUserInterface's for Linux/IrDA under development:

  - GNOBEX A GNOME application developed by Dag Brattli http://www.cs.uit.no/~dagb/irda/irda.html with support for drag'n drop from the GNOME file manager (gmc). It will also show the progress of the file transfer and give some better error messages when something goes wrong. The GUI isn't finished yet, but if you want to try the GUI you will need the Perl-GTK+ module.
  - KDE A KDE application developed by Thomas Davis. Look at his page http://www.jps.net/tadavis/irda.

  For all you lurkers out - here's your chance to contribute! Both GUI's need some icons. Any icons need to be: a) set size (48x48 pixels seems to be a common size, I think) b) large & mini (ask about size for that; mini's are for docking and such) c) 16 colors d) free for use. e) please, don't blatently copy MS icons! Please contact the developers.
- Security

  - Most important, you must `sync` your disks!!! Maybe you have to reboot your machine. Have you read the disclaimer?
- Miscellaneous

  - Other useful progs: APSFILTER, EZ-Magic, MagicFilter or something similar for the printer configuration.

3. Kernel

Please read the Kernel-HOWTO to get more information about the compilation process. You'll find the Linux/IrDA code in:

/usr/src/linux/net/irda (protocol stuff)

/usr/src/linux/drivers/net/irda (device drivers)

/usr/src/linux/include/net/irda (header files)

3.1 General Parameters

- Make sure you use *kernel 2.2.x* sources. If unsure about your kernel version try `uname -r`.

- Get the latest kernel patch from the Linux/IrDA project http://www.cs.uit.no/~dagb/irda/snapshots/. Put it into /usr/src or where else your kernel sources live and apply something like (replace `patch-2_2.0-irdaXXX` with the actual file name):

```
cd /usr/src
tar xvzf patch-2_2.0-irdaXXX.tar.gz
cd linux
patch -p1 -l < ./patch-2_2.0-irdaXXX
```

- Experimental support has to be enabled (CONFIG_EXPERIMENTAL).

- Enable sysctl in "General Setup" (CONFIG_SYSCTL).

- You should have *proc file system support* (CONFIG_PROC_FS).

- Also *serial support* for the SIR features (CONFIG_SERIAL).

- I am not sure wether there has to be *printer support* for using a printer with Linux/IrDA (CONFIG_PRINTER). But I assume this feature is not necessary.

- *Networking support* must be enabled (CONFIG_NET).

- Make sure you have *module support* (CONFIG_MODULES) in your kernel! Test it e.g. with `lsmod`.

- Also `kerneld` support is recommended (CONFIG_KERNELD). But `kmod` (CONFIG_KMOD) also works. A monolithic kernel seems to work, too.

- Matt Francis wrote: "I notice that some of the modules appear to want not only module support (as noted on the webpage), but also *misc user device support*." (CONFIG_UMISC).

If you only apply the Linux/IrDA patch, you should not have to do a `make clean`, so that should save you some time. I suggest you do something like this:

```
make dep && make all && make modules && make install && make modules_ins
```

If you get really strange errors, then try to rebuild from scratch after a `make clean`.

3.2 IrDA Specific Parameters

The following is my suggestion for ../linux-2.2.x/Documentation/Configure.help, parts are from Dag Brattli and Andreas Butz:

IrDA subsystem support

CONFIG_IRDA

IrDA(TM) is an industrial standard for infrared wireless communication. Infrared ports let you communicate with printers, modems, fax machines, LANs, and laptops. Speed ranges from 2400bps to 4Mbps. To use this features you need the irda_utils provided by the Linux/IrDA project http://www.cs.uit.no/ dagb/irda/ Further information you may find there and in the Linux/IR-HOWTO at http://www.snafu.de/ wehe/index_li.html Currently it is recommended to build IrDA support as modules only. Please see Documentation/modules.txt. Please note the status of Linux/IrDA is still experimental.

IrDA protocols

- IrLAN protocol

  CONFIG_IRLAN

  Builds the IrDA network device. Use ``ifconfig eth0 <IP-NUMBER>" to configure it. - Just say Y
- IrLAN client support

  CONFIG_IRLAN_CLIENT

  If you connect to infrared devices via IrLAN one has to be the server and the other the client. You can use both the client and the server at the same time. The first one to connect becomes the client. - Just say Y Note: The latest patch includes peer-to-peer support instead.
- IrLAN server support

  CONFIG_IRLAN_SERVER

  If you connect to infrared devices via IrLAN one has to be the server and the other the client. You can use both the client and the server at the same time. The first one to connect becomes the client. - Just say Y Note: The latest patch includes peer-to-peer support instead.
- IrOBEX protocol

  CONFIG_IROBEX

  IrOBEX is a protocol for exchanging objects (files, vcards, etc.) over an infrared connection. You can use it to exchange files between linux and a PALM III. IrOBEX can also be used between two Linux boxes, Linux and Windows95, etc. - Just say Y
- IrCOMM protocol

CONFIG_IRCOMM

Over IrCOMM you may communicate with cellular phones, etc. To use this service you have to build a new device with ``mknod /dev/irnine c 60 64", which works like /dev/ttySx. - Just say Y ..Note: major and minor number are still not the official ones yet. For latest improvements (IrSocket is on the way!), please look at the page of Takahide Higuchi http://www.pluto.dti.ne.jp/ thiguchi/irda/

..Note: At the moment IrCOMM seems to crash your kernel easily, you should probably wait for the next patch.

- IrLPT client support

  CONFIG_IRLPT_CLIENT

  Say Y here if you want to build support for the IrLPT client protocol. If you want to compile it as a module, say M here and read Documentation/modules.txt. The IrLPT client protocol can be used to print documents to IrDA compatible printers like the HP-5MP, or IrLPT printer adapters like the ACTiSYS IR-100M. - Just say Y

- IrLPT server support

  CONFIG_IRLPT_SERVER

  Say Y here if you want to build support for the IrLPT server protocol. If you want to compile it as a module, say M here and read Documentation/modules.txt. The IrLPT server protocol makes it possible to use a Linux machine as an infrared printer server for other laptops. So if your Linux machine has a cable connection to a printer, then other laptops can use the Linux machine to print out documents using infrared communication. - Just say Y

IrDA protocol options

CONFIG_IRDA_OPTIONS

You may define some IrDA protocol options.

- Cache last

  LSAP CONFIG_IRDA_CACHE_LAST_LSAP

  Say Y here if you want IrLMP to cache the last LSAP used. This makes sense since most frames will be sent/received on the same connection. Enabling this option will save a hash-lookup per frame. If unsure, say Y.

- FAST RRs

  CONFIG_IRDA_FAST_RR

  Use this option if you want to send faster RR (Receive Ready) frames if the transmit queue is empty. This will give you much better latencies but will consume more power, because of the bouncing RR frame.

- Recycle RRs

  CONFIG_IRDA_RECYCLE_RR

  In the normal life of the IrLAP protocol, it sends a lot of small RR (Receive Ready) frames over the link (at least when it has nothing else to do). Saying Y to this option will make IrLAP recycle these frames thus avoiding many alloc_skb's and kfree_skb's. To do this it will only buffer one of these frame which is enough for the usual case.

TOC

- Debug information

  CONFIG_IRDA_DEBUG

  Say Y here if you want the IrDA subsystem to write debug information to your syslog. You can change the debug level in /proc/sys/net/irda/debug. If unsure, say Y (since it makes it easier to find the bugs).

IrDA compressors

CONFIG_IRDA_COMPRESSION

You may use the compression methods BZIP2 and BSD. These are not IrDA standard. This will allow two linux boxes to handshake compression. It should be compatible with other IrDA devices, although communication will not be compressed then.

- Deflate compression (experimental)

  CONFIG_IRDA_DEFLATE

  Say Y here if you want to build support for the Deflate compression protocol. If you want to compile it as a module, say M here and read Documentation/modules.txt. The deflate compression (GZIP) is exactly the same as used by the PPP protocol. Enabling this option will build a module called irda_deflate.o.
- BZIP2 compression

  CONFIG_IRDA_BZIP2

  Help not available yet.
- BSD compression

  CONFIG_IRDA_BSD

  Help not available yet.

Infrared-port device drivers

Three sorts of low level infrared drivers are available: serial, dongle and FIR. They will show up in /proc/net/dev (irda0) after initialisation.

IrTTY (uses serial driver)

Most IrDA chips support StandardInfraRed (SIR), which works up to 115200bps and emulates a serial port (16550A UART). On many laptops this port is detected by the serial support of the kernel, see ``dmesg''. IrTTY connects the Linux/IrDA services to this port. - You should say Y here.

- Serial dongle support

  CONFIG_IRTTY_SIR

  Say Y here if you want to build support for the IrTTY line discipline. If you want to compile it as a module, say M here and read Documentation/modules.txt. IrTTY makes it possible to use Linux's own serial driver for all IrDA ports that are 16550 compatible. Most IrDA chips are 16550 compatible so you should probably say Y to this option. Using IrTTY will however limit the speed of the connection to 115200 bps (IrDA SIR mode). If unsure, say Y.

TOC

Dongle support

CONFIG_DONGLE

Currently four dongles (infrared adapters for the serial port) are supported. The dongle is an infrared device which may be connected to serial port, if you don't have built-in infrared support for your machine. If you use a dongle together with a laptop you maybe have to disable the IrDA support in the BIOS.

- ESI JetEye PC dongle

  CONFIG_ESI_DONGLE

  Say Y here if you want to build support for the Extended Systems JetEye PC dongle. If you want to compile it as a module, say M here and read Documentation/modules.txt. The ESI dongle attaches to the normal 9-pin serial port connector, and can currently only be used by IrTTY. To activate support for ESI dongles you will have to insert ``irattach -d esi'' in the /etc/irda/drivers script. http://www.extendsys.com/support/ftp/infrared.html
- ACTiSYS IR-220L and IR220L+ dongle

  CONFIG_ACTISYS_DONGLE

  Say Y here if you want to build support for the ACTiSYS IR-220L and IR220L+ dongles. If you want to compile it as a module, say M here and read Documentation/modules.txt. The ACTiSYS dongles attaches to the normal 9-pin serial port connector, and can currently only be used by IrTTY. To activate support for ACTiSYS dongles you will have to insert ``irattach -d actisys'' or ``irattach -d actisys_plus'' in the/etc/irda/drivers script. http://www.actisys.com
- Tekram IrMate 210B dongle

  CONFIG_TEKRAM_DONGLE

  Say Y here if you want to build support for the Tekram IrMate 210B dongle. If you want to compile it as a module, say M here and read Documentation/modules.txt. The Tekram dongle attaches to the normal 9-pin serial port connector, and can currently only be used by IrTTY. To activate support for Tekram dongles you will have to insert ``irattach -d tekram'' in the /etc/irda/drivers script. http://www.tekram.de/

- CONFIG_GIRBIL_DONGLE

  Say Y here if you want to build support for the Greenwich Instruments GirBIL dongle. If you want to compile it as a module, say M here and read Documentation/modules.txt. The Greenwich dongle attaches to the normal 9-pin serial port connector, and can currently only be used by IrTTY. To activate support for Greenwich dongles you will have to insert ``irattach -d girbil'' in the /etc/irda/drivers script. http://www.greenwichinst.com/

FIR support

FastInfraredSupport (FIR) needs a specific controller chip, which supports up to 4Mps. - Just say Y

- NSC PC87108

  CONFIG_NSC_FIR

  NationalSemiConductor NSC PC87108 FIR chip e.g. used in the IBM Thinkpad 560X and ACTiSYS IR2000 dongle. Probably the NSC PC87338 FIR chip is also supported. The driver supports SIR, MIR and FIR (4Mbps) speeds. - Just say Y

- Winbond W83977AF (IR)

  CONFIG_WINBOND_FIR

  Winbond W83977AF (IR) FIR chip e.g. used in the Corel Netwinder PC. The driver supports SIR, MIR and FIR (4Mbps) speeds. - Just say Y
- Sharp UIRCC

  CONFIG_SHARP_FIR

  Say Y here if you want to build support for the Sharp UIRCC IrDA chipset. If you want to compile it as a module, say M here and read Documentation/modules.txt. This chipset is used by the Toshiba Tecra laptops.

---

4. Linux/IrDA-Utils

- Use the latest source snapshot of `irda-utils` available at http://www.cs.uit.no/~dagb/irda/irda-utils/
- Untar the package with `tar xvzf irda-utils<VERSION>`. I recommend to do this in /usr/src.
- Do a `make depend`.
- Do a `make clean` (not necessary if you compile the package for the first time).
- Do a `make all` to build the binaries.
- Do a `make install`, this brings `irattach` and `irmanager` into the right place and installs some config files in /etc/irda.

A recommendation from Bjoern Hansson <Bjorn.Hansson@signal.uu.se>: If `make depend` fails on `stdef.h` and `stdarg.h` just add -I/usr/lib/gcc-lib/i586-linux/egcs-2.90.29/include/ or the according path for your system to the `SYS_INCLUDES` line in Makefile.

5. Configuration

5.1 General Configuration

- First you should put your IR devices in range. Though it might be possible that the Linux/IrDA service detects every new device automagically I only have good experience with the devices in range during the configuration process.
- Keep your infrared devices together in a range below one meter and an angle of 30 degree. There has to be a direct line of sight between them. If this is not possible, you may use a mirror (an unused M$ CD should work quite good).
- Add the following lines to your /etc/conf.modules file:

```
alias tty-ldisc-11 irtty          #
alias char-major-60 ircomm_tty    # if you want IrCOMM support
```

  I'm really not sure about those parameters anymore. There is conflicting information on Dag Brattli's pages and on the list. Though they had to be used with former versions of the Linux/IrDA package, this behaviour probably changed with the version for 2.2.x kernels.
- Have a look into the files in /etc/irda. They are similar to the files in /etc/pcmcia. Edit /etc/irda/drivers to reflect your setup. Most people will use `irattach` from that file. The files are:

```
Makefile
network*
network.redhat*
serial
drivers
network.opts
obex
printer
```

- Run `depmod -a`.

5.2 IrManager

Dag Brattli wrote: " *IrManager* [...].is a user-space daemon that is inspired and quite similar to the *cardmgr* used in the PCMCIA distribution.

The *IrManager* will receive events from the kernel level side of the protocol stack. When the *IrManager* receives an event it can execute shell commands and scripts, so I have added the /etc/irda directory which will contain such scripts. [...]

For example, if IrLMP discovers a remote device with IrLAN provider capabilities and no local IrLAN client has registered, then IrLMP will send an event to the IrManager and make it "modprobe" the module required. [...]

When application level clients are ready for communication and user-space configuration, they can also notify IrManager about this, so that it can execute the right script. For example IrLAN will send the event EVENT_IRLAN_START when the data channel is ready for exchanging Ethernet frames. When IrManager receives this event, it will execute `/etc/irda/network start <devname>` to configure the network interface. This network script is actually the same as used by the PCMCIA code and since I'm using the Redhat variant of it, it will in turn execute `/sbin/ifup <devname>`.

So by using the IrManager, I "only" have to do this when I start the stack:

```
irattach /dev/ttyS2 &
irmanager -d 1           # -d 1 means: start discovery process
```

and then when my laptop discovers the IrLAN provider (HP Netbeamer in my case) it will ask *IrManager* to load the module `irlan_client`. When the connection is up and ready, it will ask it to execute /etc/irda/network start eth0. When the connection is broken, it will again ask it to take down the interface using `/etc/irda/network stop eth0`.[...]

That's all to get it working if you are using Redhat. If you are using some other distribution which doesn't have /sbin/ifup, then you better copy /etc/pcmcia/network.opts to /etc/irda/network.opts or configure the file yourself.

If you want to use the IrLAN server, you will still have to `modprobe irlan_server` before you start the `irmanager` _without_ `-d 1`.

And just like the `cardmgr`, you will (if you want to) get the beeps when the connection is up and running and when it is disconnected!!!

I hope that we can add such scripts for all the other clients/services that need user level configuration. It would be really cool to have a /etc/irda/printer script for configuring IrDA(TM) capable printers. So if you get in range of an IrDA(TM) capable printer, then IrManager should load the `irlpt_client` module, and also configure the other stuff that needs to be done for using this printer.

I also hope that we can use the `config` file for configuring IrDA(TM) ports and device drivers. Something like:

```
Device Drivers
  module "irtty" script="irattach /dev/ttyS2"
  module "smc_ircc" irq=11 port=0x34f
```

So that IrManager can load and start all these when it is executed. In this way we would only have to start IrManager in /etc/rc.d/init.d/irda and the rest would be plug and play. There would be no need for manually starting programs and configuring devices. When `irmanager` receives the following events for a device <dev> it will currently do:

EVENT_IRLAN_START, start and configure the device using `/sbin/ifup <dev>`

EVENT_IRLAN_STOP, close the device using `/sbin/ifdown <dev>`

This can however be easily changed by the user, if this is not what is the prefered behaviour.

5.3 Low Level Drivers

There are three sorts of low level drivers: SIR, dongle and FIR. If the right driver is detected by the kernel you get a message like:

```
IrDA irda_device irda0 registered.
```

SIR

- Try to find out which serial port is used by the IR device. You may do so by watching the output of `dmesg`. If serial support is modularized do an `insmod serial` first. Look for an entry like:

  ```
  Serial driver version 4.25 with no serial options enabled
  ttyS00 at 0x03f8 (irq = 4) is a 16550A     #first serial port /dev/ttyS0
  ```

```
ttyS01 at 0x3000 (irq = 10) is a 16550A    #e.g. infrared port
ttyS02 at 0x0300 (irq = 3) is a 16550A     #e.g. PCMCIA modem port
```

If this is not the case, you either don't have infrared support enabled in the BIOS or your infrared device is not detected by the kernel. Currently I know only two laptop models with this effect, the HP OmniBook 800 and the Toshiba Libretto models. I am not sure whether PnP support effects the detection of the IR port. If you are unsure try it out and let me know the results.

- In some situations you may have to use `setserial /dev/ttyS<0-2> port 0xNNNN irq M` to set the values for your infrared serial port, especially if the infrared port is a separate serial line. You usually don't need to change the values! For further information look into the FAQ section below.
- If you don't use `kerneld` or `kmod` insert the irda module with `modprobe irda`.
- Do `lsmod`. It should show the modules `irda` and `irtty` now.
- A look into /var/log/messages should show the entry "`Serial connection established`" now.
- Say `irmanager -d1`, which will start the necessary programs, such as `irattach`.
- Give `irattach` some time, e.g. seven seconds, to detect other IR devices. Then watch the output from the kernel that you will hopefully get in /var/log/messages. It should look like the following (I removed some lines, which were not related to Linux/IrDA):

```
Jan  2 12:57:26 japh kernel: ttyS00 at 0x03f8 (irq = 4) is a 16550A
Jan  2 12:57:26 japh kernel: ttyS02 at 0x03e8 (irq = 4) is a 16550A
Jan  2 12:57:26 japh kernel: Linux Support for the IrDA (tm) protocols (Dag Brattli)
Jan  2 12:59:09 japh syslog: executing: 'echo 1 > /proc/sys/net/irda/discovery'
Jan  2 12:59:09 japh syslog: Setting discovery to 1 exited with status 1
Jan  2 12:59:09 japh syslog: + 0.1 Fri Jul 25 11:45:26 1997 Dag Brattli
Jan  2 12:59:09 japh syslog: + 0.1 Fri Jul 25 11:45:26 1997 Dag Brattli
Jan  2 12:59:09 japh syslog: Serial connection established.
Jan  2 12:59:09 japh kernel: IrDA irda_device irda0 registered.
Jan  2 13:01:22 japh syslog: executing: './drivers start '
Jan  2 13:01:22 japh syslog: Serial connection established.
Jan  2 13:01:42 japh syslogd: Printing partial message
Jan  2 13:01:42 japh 0.1 Fri Jul 25 11:45:26 1997 Dag Brattli
Jan  2 13:02:49 japh kernel: IrDA Discovered: japh
Jan  2 13:02:49 japh kernel:     Services: Computer
```

Dongle Connection - Infrared Adapters for the Serial Port

The currently supported dongles are the Extended Systems Inc. ESI-9680 JetEye, the Tekram IRmate 210B, the ACTiSYS IR220L and 220L+, the Greenwich GIrBIL. dongle.

Dag Brattli wrote (modified by wh): "To use dongles you have to do something like this:

```
modprobe tekram         # or esi or actisys
irmanager -d 1          #
irattach -d tekram      # or -d esi or -d actisys
```

As you can see, you must still use the `-d` option with `irattach` since it is possible to have two serial ports using different dongles at the same time (so the tty you are binding must know which dongle it is supposed to use). So if you have two dongles and two serial ports, you could do something like this:

```
modprobe tekram
modprobe esi
irattach /dev/ttyS0 -d esi &
```

```
irattach /dev/ttyS1 -d tekram &
```

PS: I would not try to turn the two dongles against each other, since I really don't know how the stack would react :-) ... Since I don't have any of these new ACTiSYS 220L+ dongles, I'm not able to test it. Since the new dongle has support for one extra speed (38400bps), you must specify the dongles differently with `irattach` so that the kernel knows which dongle you are using (and what QoS can be used):

```
irattach /dev/ttyS0 -d actisys     # for the 220L dongle
irattach /dev/ttyS0 -d actisys+    # for the 220L+ dongle
```

The current implementation of dongle support does not have any state associated with it, so its not possible to use both ACTiSYS dongles (220L and 220L+) at the same time (connected to two serial ports) for now. If someone needs to be able to do so, please mail me (Dag Brattli) and I will think about it!"

Note: When I tried to use an infrared modem (Swissmod 56Ki, manufactured by Telelink AG) connected to my laptop (IrDA works with Window$95 only, due to non standard hardware) I had to remove the infrared support in the BIOS to get it working!

Dongle Connection - Infrared Motherboard Adapter

Support for the ACTiSYS IR2000 dongle has been implemented in a file called pc87108 which you can either compile into the kernel or `insmod/modprobe` to insert the module:

```
irmanager -d 1
modprobe pc87108
```

or insert `modprobe pc87108` into the /etc/irda/drivers file (I think).

Fast InfraRed (FIR)

The IrDA(TM) standard knows *three* kinds of speeds:

1. SIR = Standard IrDA, up to 115kbps IrDA,
2. MIR = Medium Speed IrDA,
3. FIR = Fast IrDA (4Mbps),
4. VFIR = Very Fast IrDA(16Mbps), seems to become a future standard

Up to 115.200bps many infrared controllers work like a serial port. Up to 4Mbps they work in "FIR mode". Currently there are two chips supported: NationalSemiConductor NSC PC87108 e.g. used in IBM Thinkpad 560X and Winbond W83977AF (IR) FIR chip e.g. used in the Corel Netwinder PC. You may start the FIR service by loading the according module. Linux/IrDA will probe your hardware then.

6. Specific Connections and Protocols

6.1 Printer Connection - IrLPT

IrLPT is under heavy construction at the moment. The client side should work, but the server side may crash.

- Remove any current print jobs with `lprm "*"`.
- If you don't use `kerneld` do a `modprobe irtty`.
- Do a `modprobe irlpt_client`.
- Check the modules with `lsmod`. This should show: `irda`, `irtty` and `irlpt_client`
- `cat /proc/misc`. Gives you the *minor device-number* . It is the first number in the line with irlpt0.
- `su` to root, and do `mknod /dev/irlpt0 c 10 <minor device-number>`. Note: Something like `./MAKEDEV irlpt0` is not possible yet. But maybe `load_misc irlpt` works, though I couldn't test this yet.
- Try to write a small file to /dev/irlpt0 by `cat FILE >/dev/irlpt0` (do not wonder about a bad format this is just a first check). For me this didn't always work, but I couldn't find out why not.
- The better way is to change your /etc/printcap to use /dev/irlpt0 in addition or instead of /dev/lp1. See *Printing-HOWTO* for detailed information.
- For easy printer setup you may use a printing software like APSFILTER, MagicFilter EZ-Magic (with RedHat there should also be a GUI for this purpose). Make a copy of /etc/printcap before.
- Example for APSFILTER with a HP 6P (non-postscript, HP 6MP is with postscript). The two relevant questions are:

  "Do you have a (s)serial or a (p)arallel printer interface?" Answer "p"

  "What's the device name for your parallel printer interface?" Answer "/dev/irlpt0"
- Restart the print daemon with `kill -HUP <PID of lpd>`. If you use another print daemon choose the according command.
- Watch whether the connection indicator of your printer shows activity, e.g. the green light above the IR port of a HP 6P/MP comes on (lower left hand corner, near the paper tray).
- I couldn't get to manage printjobs larger than approximately 10 pages yet. But maybe this depends on the memory size of my hardware, which is 16MB. There seems to be a problem with the software too, Thomas Davis wrote: "I will ... limit the irlpt, so it won't eat memory when you send a large print file.".

Takahide Higuchi reported: " I have been debugging IrCOMM with a printer (Canon BJC-80v) with IrDA port and IrCOMM protocol (not IrLPT). I can print a short e-mail text though, it easily causes dead lock when I try to print a postscript with gs."

From the page of Thomas Davis http://www.jps.net/tadavis/irda : To use the IrLPT server, you need to perform the following steps:

```
/sbin/insmod irlpt_server
/sbin/mknod /dev/irlptd c 10 `grep irlptd /proc/misc|cut -f 1`
```

At this point, the IrLPT server is ready to recieve print jobs; now; all you need is this simple shell script

```
#/bin/sh
#
while (true)
do
cat /dev/irlptd | lpr
done
```

6.2 LAN Connection - IrLAN

- You might connect your Linux box using IrLAN to another network device such as a Linux box with IrLAn, a HP NetBeamer or a Window$95 box with Inrared Network Device support.
- Dag Brattli wrote: "If you want to use IrLAN you must `modprobe irlan_client` before `ifup eth0`. I had to remove the request_module() stuff since that needed a process context which I don't have in the kernel. "
- Run `ifconfig eth0 up <ip_address> netmask <ip_netmask>` to configure it with IP-address and other parameters. If the protocol is still running you may start communicating. It is possible to use RedHat's netcfg to do this, since it makes it very easy. Next time you only need to do `/sbin/ifup eth0`.
- Test the network device by pinging to it. For detailed information about further setup see the *NET3-HOWTO*.
- Do not forget to add a route, e.g. `route add default gw <ip_gateway>` or `route add -host <target host> dev eth0`.
- Ping to another IP now, to test the connection.
- For testing reasons I recommend only to use one laptop and one IR ethernet device in the same room. If there are problems look which different modes for the IR ethernet device are possible. Try them.

For an ACTiSYS FIR board and dongle you may do:

```
irmanager -d1
/sbin/modprobe pc87108  # remove irattach from /etc/irda/drivers, or
                        # substitute irattach with the modprobe!
```

On machine 1:

```
modprobe irlan_client # not really necessary since irmanager should do this!
```

On machine 2 (if you don't have an access-point)

```
modprobe irlan_server
```

Do not compile `irlan_server` into the kernel, since it currently does not like that! You should have configured /etc/sysconfig/network-scripts/ircfg-eth0 with a proper ad-hoc network if you are using two machines. If you have an access-point, then the normal setup should be fine.

Notice that in the latest patch (2.2.0-irda1) irlan_client will call the device `irlan0` by default, but you can change this by giving `eth=1` as an option to `irlan_client` (`modprobe irlan_client eth=1` or `options irlan_client eth=1` in /etc/conf.modules). The next release of IrLAN will be only one module, so you don't need to think about if you need to have the client and/or the server installed.

It's possible to do `ifconfig irlan0 -broadcast` to stop the AP from flooding you with broadcast frames! That can be a problem if you are connected to a very large Ethernet segment. The only problem is that your machine will then have to initiate all communications and can therefore not function as a server (well, you could probably make a stationary machine somewhere answer ARP requestes on your behalf).

6.3 Palm III Connection - IrOBEX

The IrOBEX stuff seems under rapidly improving changing development. So the applications change too. Therefore I just can't give quite exact information. Please see also the report by Dag Brattli at http://www.cdpubs.com/hhsys/archives/66/10brattl.pdf .

The /etc/irda script is really only good for configuration of the devices, making the right mknod for /dev/irobex etc, not for starting applications.

- Palm III -> Linux

  1) Terminal 1> `irattach /dev/ttyS<x>`

  2) Terminal 2> `load_misc irobex`

  3) Terminal 3> Start `irobex_app` in the irobex directory. I suppose `irobex_app` is not working anymore. Now you should use the `gtk/irobex` program instead! You need to have the gtk library installed to use this program. A command line frontend should be programmed by someone. Maybe the programm to use is `irobex_receive`.

  4) Beam something from your Palm III.

  5) If everything is successful, you can take a look at a new file that has been created in the directory in which you started irobex_app (or in /tmp for irobex_receive). This file will be named after the object you just transfered.
- Linux -> Palm III

  This should also be possible, but I don't have any further information right now.
- PPP

  Rui Oliveira wrote: "This is just to let you know that with the latest IrCOMM patch (050998) of Takahide Higuchi, I managed to HotSync and establish a PPP connection between my Palm III and my Linux box. I'm using IRLink (from IsComplete) to redirect the serial port to ir. Communication with pilot-xfer works flawlessly. Although I was able to establish a PPP connection, I'm still unable to fetch mail and do Web browsing. This is probably due to connection time-outs. I am checking this out.". Please see the *PPP-HOWTO* for further information about PPP.
- IrCOMM

Jon Howell wrote: "I thought I'd try IrCOMM, since the Palm III can be made to reroute serial info to IR port (using IrLink from IS/Complete, available at www.palmcentral.com), and then you can run a terminal program (like "PalmTelnet" in serial mode) over IrDA. I can only assume it's using the IrCOMM protocol. I've tested this configuration between two palm pilots, but of course I can't know what the protocol running over the IR is.

I couldn't figure out what to do with the IrCOMM code. I see from the status page that there's "Client support (should start to get usable), " but I can't find any docs that indicate how it has to be used. "

6.4 Cellular Phone Connection

As far as I know some cellular phones use the IrCOMM standard, e.g. Ericsson SH888 and NOKIA 8110. Benny Amorsen wrote: "I have used minicom with the SH888, and I actually got it to establish a connection in minicom (CONNECT 9600...). I could not get it to do anything after that, and hanging up by dropping carrier does not work. +++ worked, so I could hang up that way. I would say that the SH888 is very close to working with linux-irda., and actually it might work in 2.0.x kernels. I only use 2.1 kernels.".

Maybe other cellular phones use the IrOBEX standard, see the Palm III section for information about setting up a connection.

Carlos Vidal wrote: Correct me if I'm wrong, but it seems to me that Nokia telephones do not contain a genuine hardware modem, but something which is similar in principle to WinModems for PC. Whenever Nokia writes about modem communication, they use the name "Windows software modem" (or something similar). Which is actually backed up by the need to use special Nokia software for Windows (called Nokia Cellular Data Suite).

Joonas Lehtinen wrote: This is true with 61xx models. Models: 8810, 9000(i) and 9110 should work fine. (They have inbuilt modem). My N9000 reports IrCOMM with linux.

To start a communication session with /dev/irnine, for instance, say:

```
dip -t
> port irnine
> term
```

Probably you may use `cu` instead of `dip`, too. There are also reports about some efforts with the Ericsson GF768 and IR Modem DI 27.

Benny Amorsen wrote: The SH888 emulates an IRDA-port when you connect it using the serial cable. Why someone would think up something weird like that is beyond me, but that is the way you get it to work in Windows. Not that I ever managed to make it work in Windows, though.

6.5 Digital Camera Connection

Markus Schill wrote: "Great that there are also other people who are interested in using the SONY DSC-F1 IR adapter under linux. Up to now I have only toyed around with the linux-irda software and the serial IR adapter from PuMa Technologies that came with the camera. This is the status. I am using linux 2.0.33 and the latest linux-irda... If I use:

```
insmod irda
```

```
insmod irtty
irattach /dev/cua0
```

the adapter starts talking to the camera. /var/log/messages says that SONY-DSC-F1 was found, but no service is started. (Please note, this probably doesn't apply to the 2.2.x kernel versions of Linux/IrDA, wh).

There are two programs for linux available that can be used for the communication with the camera via cable: (1) `chotplay` and (2) `stillgrab`. They both take a tty as commandline option, so I guess that they should work if the irtty layer of the protocol stack works correctly ... I have not looked at anything in the linux-irda code, yet!). I am not sure whether I understand the stack but shouldn't the irtty make the thing look like a normal tty? What service should be started. "

Dag Brattli wrote: "I'm not sure which application level protocol the camera uses, but it is possible that it implements the IrDA(TM) Infrared Transfer Picture Specification (IrTran-P). If you take a look at http://www.irda.org/standards/pubs/IrTran-P_10.pdf, you will see that it is a protocol which is implemented above IrCOMM (not IrTTY!). IrTTY is something we use just to be able to talk to the Linux serial driver. "

6.6 Window$95 and Linux/IrDA

Introduction

Why this? Unfortunately Linux users are not always supported with the necessary hardware information. Sometimes it is possible to look at this informations in Window$95.

Where to get it from? At http://www.microsoft.com/windows95/info/irda.htm you will find a support pack "Infrared Transfer 2.0". It is a self-extracting archive `W95IR.EXE` with 331KB.

With some machines e.g. a HP Omnibook 800 it is necessary to use a vendor specific version of this package (e.g. for the HP Omnibook 800 you may find it on the recovery CD).

Especially the `..\windows\inf\*.inf` files and the device manager are of interest to look for configuration details.

As far as I know Window$NT doesn't support IrDA(TM). About Window$98 I have heard there is no IrDA(TM) support yet.

There are also some non M$ products available. Note: Some of them use proprietary infrared protocols:

- CounterPoint: QuickBeam 1.15
- LapLink 7.5
- CarbonCopy 32 4.0
- pc ANYWHERE 7.5
- Puma Technology: TRANXIT pro 4.0

Connection between Linux/IrDA and Window$95 IrDA(TM)

I suppose there are *three* ways to connect Linux/IrDA and Window$95:

A *network connection* between two PC's. If you have set up *Infrared Transfer 2.0*, you will find an IrDA(TM) network device in the <*Network Device Section*>. But I couldn't get a working connection yet.

Maybe it is also possible to use the *IrOBEX protocol*. But I don't know which software to use and where to get it. I supposed the necessary software comes with a Palm III, but this seems not to be true.

TOC

Takahide Higuchi <thiguchi@pluto.dti.ne.jp> provided *IrCOMM support*. From his page at http://www.pluto.dti.ne.jp/~thiguchi/irda/ I have taken the following description (I have modified it at little): "With IrCOMM support you can send or receive short messages between a linux box and a terminal program on a win95 laptop! Please add this line to /etc/conf.modules:

```
alias char-major-60 ircomm_tty
```

Next, make a device file `mknod /dev/irnine c 60 64`. Now Linux/IrDA services can be started as usual with `irattach /dev/ttyS? &`. /dev/irnine can be used as a serial device. `ircomm` and `ircomm_tty` modules will be loaded automatically by `kerneld/kmod` when a program uses /dev/irnine. NOTE: I think "setserial" utility will not work on /dev/irnine. Tips:

- To accept login via IrCOMM, use this as a root: First, please enable IrDA and IrCOMM. Then edit /etc/inittab and add a line like this:

  ```
  T1:23:respawn:/sbin/getty -L -w irnine 38400 vt100
  ```

  and do this as a root: `init q`. And `init` will start waiting for incoming IrCOMM connection. You will see your favorite Linux's login prompt from terminal emulator on Win95!
- If you try `pppd`, please consider the -crtscts option to disable flow-control. I implemented some flow-control emulation but it is not tested.
- Now my patch reports what kind of features is needed by the peer infrared device. Messages like this will be written in syslog:

  ```
  Sep 4 10:01:02 monolith kernel: parse_control:instruction(0x12)
  Sep 4 10:01:02 monolith kernel: data:03
  ```
- I especially want to know what message SH888 (or other infrared devices except for win95 PC) says. So please mail me your syslog generated during IrCOMM connection! If you have a copy of the IrCOMM specification written by IrDA(TM), please see page 34 or 38, and you will know what these messages mean."

6.7 Linux to Linux Connection

Connection Methods

There should be *three* ways to get two Linux machines connected via Linux/IrDA.

- Dag Brattli wrote about the *IrOBEX support*: "The awakened reader may wonder what prevents the beaming of files from Linux to Linux? Well, nothing!! (but I haven't tried that yet). This means that we now have a "simple" way of beaming files between Linux laptops. I think that this may be the "killer app" we all have been waiting for!" Try to "`load_misc irobex` at both ends, and then try `iroabex_app get` on one of the machines and `irobex put <file>` on the other.".
- Via *Linux/IrDA network* connection. I suppose you have to load the module `irlan_client` at one machine and `irlan_server` at the other one.
- With *IrCOMM* support, in other words over a serial line, which could mean `minicom`, `pppd`, etc.

Compression

Please note this feature is still quite experimental! Dag Brattli wrote: "Just wanted you to know I have just added COMPRESSION support to IrLAP! As you may know, this is _not_ part of the IrDA(TM) standard, but Linux can now negotiate with its peer and check if it has the same compression capabilities). So obviously if you are talking to Win95, Palm III or whatever, you will _not_ get compression!!! This is something which is exclusive for Linux as far as I know! The IrDA(TM) standard says that devices should ignore unknown field in the negotiation header, so we are still "compatible" with IrDA(TM) (have just borrowed an unused header value).

If you want to try using the compression code (Linux <-> Linux) you will have to insert the `irda_deflate` module some time before you actually make the connection. I do it before `irattach`.

The compression standard I have added is the deflate format used by the zlib library which is described by RFCs (Request for Comments) 1950 to 1952 in the files ftp://ds.internic.net/rfc/rfc1950.txt (*zlib format*), rfc1951.txt (*deflate format*) and rfc1952.txt (*gzip format*).

The compression interface is similar to PPP, so you can add as many different compressors as you want. Currently there is only support for GZIP, but BSD compression will be added later."

6.8 Multiple Instances

Dag Brattli wrote: "The IrLAP layer has been enhanced to allow more than one instance (so I can use IrLAN on my built-in ir-port, and communicate with the Pilot over the IrDA dongle at the same time) ... So how do you make two Linux/IrDA connections? Well, you just fire up `irattach` for each of the IR ports you have like this: "

```
irattach /dev/ttyS0 &    (my ESI dongle)
irattach /dev/ttyS2 &    (my builtin IrDA port)

insmod irlan_client
insmod irobex
```

---

7. Hardware Supported by Linux/IrDA

7.1 Obtaining Information about the Infrared Port in Laptops

To get the IrDA port of your laptop working with Linux/IrDA you may use StandardInfraRed (SIR) or FastInfraRed (FIR).

SIR

Up to 115.200bps the infrared port emulates a serial port like the 16550A UART. This will be detected by the kernel serial driver at boot time or when you load the serial module. If infrared support is enabled in the BIOS, for most laptops you will get a kernel message like:

```
Serial driver version 4.25 with no serial options enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A     #first serial port /dev/ttyS0
ttyS01 at 0x3000 (irq = 10) is a 16550A    #e.g. infrared port
ttyS02 at 0x0300 (irq = 3) is a 16550A     #e.g. PCMCIA modem port
```

FIR

If you want to use up to 4Mbps, your machine has to be equipped with a certain FIR chip. You need a certain Linux/IrDA driver to support this chip. Therefore you need exact information about your FIR chip. You may get this information in one of the following ways:

1. Read the specification of your machine, though it is very rare that you will find enough information there.
2. Do a `cat /proc/pci`, though often the PCI information is incomplete. You may find the latest information about PCI device and vendor numbers at the page of Craig Hart http://members.hyperlink.net.au/~chart . From kernel 2.1.82 on, you may use `lspci` from the `pci-utils` package, too. The according files for 2.2.x kernels are in /proc/bus/pci.
3. If you have installed the Linux/IrDA software load the FIR modules and watch the output of `dmesg`, wether FIR is detected or not.
4. Though I didn't use them for this purpose yet the `isapnp` tools, could be useful.
5. Another way how to figure it out explained by Thomas Davis: "Dig through the FTP site (of the vendor, wh), find the FIR drivers, and they have (for a SMC chip, wh):

```
-rw-rw-r--   1 ratbert  ratbert       743 Apr  3  1997 smcirlap.inf
-rw-rw-r--   1 ratbert  ratbert     17021 Mar 24  1997 smcirlap.vxd
```

```
-rw-rw-r--   1 ratbert  ratbert      1903 Jul 18  1997 smcser.inf
-rw-rw-r--   1 ratbert  ratbert     31350 Jun  7  1997 smcser.vxd
```

TOC

If in doubt, always look for the .inf/.vxd drivers for Win95; Win95 doesn't ship with _ANY_ FIR drivers. (they are all third party, mostly from Counterpoint, who was assimilated by ESI)."

6. Use the DOS tool `CTPCI330.EXE` provided in ZIP format by the german computer magazine CT http://www.heise.de . The information provided by this program is sometimes better than with the Linux tools.
7. There is also a small DOS utility made by SMC, called "FindChip". The package also includes a "FirSetup" utility that is supposed to be able to set all values except the chip address. Look at http://www.smsc.com/ftppub/chips/appnote/ir_utils.zip Warning: The package is not intended for the end user, and some of the utilities could be harmful. The only documentation in the package is in M$ Word format. Linux users may read this with `catdoc`, available at http://www.fe.msk.ru/~vitus/catdoc/ .
8. Use the `Device Manager` of Window$95/98/NT.
9. You may also use the hardware overviews mentioned below.

7.2 Hardware Overviews

There are some overviews about Linux and infrared capable devices in the WWW:

- The Linux/IrDA Project - Hardware Survey at http://www.cs.uit.no/~dagb/irda/hardware.html
- Takahide Higuchi at http://www.pluto.dti.ne.jp/~thiguchi/ir/product.html.This page is in japanese.
- I have also set up a hardware overview at http://www.snafu.de/~wehe/index_li.html. This list also contains information about infrared capable devices which are not mentioned here (mice, printers, remote control, transceivers, etc.).

  To make this hardware overview list more valuable it is necessary to collect more information about the infrared devices in different hardware. You can help by sending me a short e-mail containing the exact name of the hardware you have and which type of infrared controller is used.

  Please let me also know how well Linux/IrDA worked, at which tty, port and interrupt it works and the corresponding infrared device (e.g. printer, cellular phone) you use.

  You can also help by contributing detailed technological information about some infrared devices, which is necessary to develope an according driver for Linux.

8. GUI

If you want to try out a little GUI, you can try to run `ircp.pl`. You will need the Perl-GTK+ module (http://www.gnome.org) to make it run. The GUI is far from finished, so don't expect too much :-)

9. Power Saving

In the specifications of my HP OmniBook 800 it is recommended to turn off the IR port, if it is not in use, because it may consume up to 10 percent of the battery time.

If necessary, you may also try to disable the `Fast RRs` feature in the IrDA section of the kernel. This option will give you much better latencies but will consume more power.

10. Troubleshooting, Mailing List

10.1 General Information

If you encounter problems. Try the following:

- Read the FAQ section below.
- Look at /var/log/messages and/or /var/log/kern.
- Do a `dmesg`.
- Look at the different files in /proc/irda.
- Look at the *mailing list archiv* at http://www.ita.chalmers.se/~svinto/hypermail/irda/, whether your problem is already known.
- Ask in the *Linux/IrDA mailing list*. To join the Linux-IrDA mailing list, send a mail to linux-irda-request@list.uit.no with "subscribe" in the email body. Use linux-irda@list.uit.no to post a message. You are welcome to use this mailing list for posting questions, answers, bug-reports, patches, suggestions and comments. To avoid spam, the list is now moderated, so there may be some time before your posting is distributed to the other list members.

10.2 Troubleshooting Techniques

Although I'm not much of a hacker I collected some tricks to track errors or bugs in the Linux/IrDA software.

- You may set the debug level in /proc/sys/net/irda/debug to 1, 2, 3, 4.
- Use the files in /proc/sys/net/irda to try different parameters like `echo 0 > /proc/sys/net/irda/discovery`. The /proc/*/irda files are:

```
root@duckman:~# ls /proc/sys/net/irda/* /proc/net/irda/*
/proc/net/irda/discovery        /proc/net/irda/irlmp
/proc/sys/net/irda/devname
/proc/net/irda/irda_device      /proc/net/irda/irttp
/proc/sys/net/irda/discovery
/proc/net/irda/irias            /proc/sys/net/irda/compression
/proc/net/irda/irlap            /proc/sys/net/irda/debug
```

- It is also possible to debug the code. But I don't know how to do this. If you want to use SKB debug code, you may edit irda.h and change `/include/linux/skbuff.h` (see revision history of snapshot 10-2-98).
- For problems with the `irda` module a utility from the *modules package* `kdstat` might be

helpful. But I was not able to try this.

- "You can now alter the number of discovery packets used (1, 6, 8 or 16) and the timeout between sending them (2-8 * 10 ms) in /proc/sys/net/irda. Please experiment if you have problems discovering your device. My Palm III seems to like 16 `discovery_slots` and 8 (*10 ms) for `slot_timeout`. " ... "The absolute minimum for reliable discovery of the IR-610 seems to be 9."
- If anybody gets a kernel Oops, then please feed it to the ../linux/scripts/ksymoops/ksymoops program, so that we can find out where it went wrong. Just cut out the Oops lines from the syslog, save them to a file, and then run `ksymoops <file>`
- Dag Brattli wrote: I found out that the cs4232 sound card was giving me several hundred interrupts per second! I removed the sound stuff from my kernel, and the machine is now generally about 4 times faster! Linux/IrDA may get problems if you are running the esound server (esd) on your machine. Both my machines, a 166Mhz Pentium laptop and a 200Mhz Pentium Pro cannot run Linux/IrDA when esd is running. The reason is that esd makes the soundcard give interrups over 300 times/second which makes the serial driver overrun when receiving. This is because the serial driver now uses slow interrupts in Linux-2.2 (everything is slow interrupts in 2.2), so the interrupt-handler schedules on its way out. The good thing about slow interrupts is that packets are delivered much faster, since you don't need to wait for the next timer-tick. The only exception for this is the pc87108 driver which works fine since it uses DMA and will only give a couple of interrupts per packet.

11. Known Bugs

If you find a bug, please send a bug report to the mailing list, including `dmesg` output, and which Linux version, and hardware you are using. Thank you!

Sometimes IrCOMM fails to connect (especially when both devices discover each other. You can disable discovering with `echo 0 >/proc/sys/net/irda/discovery`)

A CR (carriage return) character cannot be transfered between two linux boxes via IrCOMM with `cat file >/dev/irnine` and `cat /dev/irnine`. It causes a strange thing and freezes your Linux box. Compiling the pc87108 device driver non modular crashes the kernel on boot. Temporary solution: compile the driver as a module

IrOBEX may eat some data on receive. The bug is most probably in the user-space side of IrOBEX.

12. FAQ

- Q1 - Question: I do not know anything about ports and irqs. What should I do?
- Answer:

  PART A: Hardware settings

  - 1 Have a look at your hardware specs!!! If not available look at the support page of your vendor, or contact the support hotline. You might also find the information in one of the hardware overviews mentioned above.

  - 2 Use a current BIOS. Usually available at the support page of your vendor.

  - 3 Try `setserial /dev/ttyS? -g -a | egrep 16550A`. One of the shown devices is probably the one you are looking for. Usually it is the second one, but with no guarantee.

  - 4 Note: What seems like an UART is physically the IrDA controller. For my HP Omnibook 800 this is the VLSI VL82C147 PCI - IrDA controller. These controllers should behave up to 115 200 bps like UART's. But sometimes it is very difficult to get the right configuration.

  PART B: How to tell the kernel about the hardware settings

  -4 `cat /dev/ioports` to see which ports are already in use.

  -5 `cat /dev/interrupts` to see which interrupts are already in use.

  -6 Make ports and interrupts available for use with the IR device, e.g. stop the pcmcia service or include a line like this in /etc/sysconfig/pcmcia:
  `PCIC_OPTS="irq_list=3,4,5,7,9,10,12,14,15"`

  -7 Now try to guess what the right interrupt and port is. Use `setserial /dev/ttySx irq M port 0xNNNN` to tell the kernel. If there is more then one possible chance try them all (Note: As mentioned in the *Serial-HOWTO* you should not try irq 0, 1, 6, 8, 13, 14).

  -8 If you were successful please send the useful parameters to the author, because I would like to include them in the hardware overview.

  -9 Good luck.

  It might also be necessary to fine tune the IR serial port with setserial, e.g., `setserial /dev/ttyS0 spd_vhi` (speed rate 115200).
- Q2 - Question: For me, `irattach` hangs, but recognizes the printer. /var/log/messages shows that irattach found my HP LaserJet 6P.
- Answer: The "hang" is normal for irattach. Everything is working right if you see the HP Laserjet show up in the log. "hang" means irattach is polling the IrDA-Devices for incoming connections. If you kill it with <CTRL C> the irattach program crashes and /dev/ttySx does not work anymore.

The problem is within the irda module, and not with the irattach program. Rebooting is the only thing to do! Next time put irattach in the background by using `irattach &`. Stop it if necessary with `killall irattach`. Recommendation by Andreas Butz: To my knowledge, <CTRL Z> `bg` should work, too, but I haven't tried it in this specific case. Normally it has the exact same effect as appending & to a command.

- Q3 - Question: I get a message like tcsetattr read/write error in /var/log/messages.
- Answer: Caused probably by wrong /dev/ttyS* or wrong irq or port.
- Q4 - Question: Every setting seems alright, because I get the appropriate messages. But it still does not work.
- Answer: Move the devices to within 0.5 meter (1.5 feet). Check that only one application is using the infrared port. Check that both devices are using the same protocol, such as IrOBEX or IrCOMM.
- Q5 - Question: I have downloaded the latest snapshot, and compiled it successfully under Linux 2.0.33 running on an IBM Thinkpad 560E. In the absence of any other IrDA machines to test with, is it safe to assume that once the module has been inserted and the syslog reports "irattach: Serial connection established.", is the IR really working, and will it start to respond once there is another machine with which to talk?
- Answer by Dag Brattli: Sorry, this only means that irattach has done its part of the job, which is just to start the irda-tty. Maybe the message should have been different, but as I said, it tells that the serial connection between the irda-chip and the irda-driver is established.

  Note: Support for IrDA on 2.0.x kernels has been discontinued. You are encouraged to switch to 2.2.x kernels and use the newest IrDA patches available at http://www.cs.uit.no/~dagb/irda/snapshots/.
- Q6 - Question: At startup `modprobe -a` checks /lib/modules/<KERNEL-VERSION>/net/irda.o and causes the messages: "IrLAP; Missing IrTTY /IrLMP Error no IrLAP connection" (in /var/log/messages and on the console).
- Answer by Werner Heuser: Workaround for SYSTEM V style systems: Put a script named for example "ir_rmmod" containing

  ---

  ```
  #!/bin/sh
  echo "$0 : remove irda module"
  rmmod irport.o
  rmmod irtty.o
  rmmod irda.o
  ```

  ---

  in the startup process (/etc/init.d and a symbolic link name for example "S100ir_rmmod" in /etc/rc3.d to "ir_rmmod"). (Verify the path for "sh"). For BSD style systems try the corresponding approach.
- Q7 - Question by Ho Chin Keong: Is there other way of setting up communication between the 2 laptops besides setting up a LAN route between the two?
- Answer by Dag Brattli: Yes and no! One of the IrDA standard, IrCOMM permits you to emulate a

serial cable between two laptops, so you can use any application written for serial ports (termi PPP, slip, etc.). This is however not yet implemented in Linux/IrDA. The IrLPT (printer) sup is actually a subset of IrCOMM, so some of it is working!

- Q8 - Question by Ho Chin Keong: If I block the infrared path deliberately for more than 10 seconds, the connection could not re-establish. I have to kill the irattach and restart the whole procedure to start the infrared route. The connection could be maintained, however, if the blocking is less than 10 seconds. Is this part of the design or a bug? Is there any way whereby we can lengthen this time limit from 10 s to longer or infinitely?
- Answer by Thomas Davis: This seems to be a bug in the primary side of the IrLAP/IrLMP code. It appears not to send the reset/disconnect notice all the way back up the stack. You'll notice it when IrLPT gets stuck in the query mode while you were trying to talk to a printer, and disconnected/interrupted it when it was handshaking. (and now, it shows up in the IrLAN portion)
- Q9 - Pierre-Guillaume Raverdy asked: Should I update to IR lib on my palm and update the system to version 3.0.2?
- Answer by Dag Brattli: You should not need to update your Pilot, but it should not do any harm to do so. It is however required if you want to use the IrCOMM library from IsComplete
- Q10 - Pierre-Guillaume Raverdy asked: Also any simple source code (especially on the palm side) would be greatly appreciated.
- Answer by Dag Brattli: Get the Pilot SDK from Palm. Unzip the examples.zip and take a look at the beamer application.
- Q11 - Kai Poehlmann wrote: I have a gsm-phone from Ericsson SH 888 and a linux-computer without an irda-port. I have heard that Ericsson wants to talk with the phone the IrDA-protocol also when using the serial cable... :-/ Is it now possible to use this phone with this IrDA-port for linux and the cable?
- Answer by Matthias Lemke: Yes its possible. I tried it 6 weeks ago. But i had same problems like with real-irda. I can connect my phone. i can reset it by ATZ but after ATDT nothing happens. Same with cable or without. But i think you should test newer versions of the port.
- Q12 - Jonah Peskin wrote: Has anyone had any success extending the range of an IrDA transceiver? I'm using a Dell Inspiron 7000 notebook, and it seems to have a receiving range of about 1 meter. I realize this fits the IrDA spec, but are there any laptops or devices that can do better than a meter? Why not just get a 1 meter cable?
- Answer by tzeruch@ceddec.com:Because manual connection (until things like the Palm Cradle) is cumbersome and time consuming and often creates problems (accessing the back side of my desktop requires contortions). The main problem is that you generally have to make the receiver more sensitive. Basic physics has the inverse square law: the intensity drops with the SQUARE of the distance, so going from 1 to 5 meters requires 25x the power (and battery drain on a portable device), or 25x the sensitivity (and dynamic range - it still has to be able to work at 3 inches). And if you want to do it on the other end, it doesn't simply have to be 25x more sensitive, it must pick up the tiny IrDA pulse needle in a haystack of florescent lights, screen savers, moving shadows... Someone tried it with a Palm III upgrade board: http://home.t-online.de/home/PSPilot/ppppiii.htm

13. Infrared Remote Control

13.1 Resources

Remote control via infrared is not the aim of the Linux/IrDA project but is included in this HOWTO to cover "Linux and Infrared" more completely. I found *three projects* which worked on this topic. You may find some links to current information at http:// www.snafu.de/~wehe/index_li.html.

- LIRC LInux Remote Control

  LIRC is a package that supports receiving and sending IR signals of the most common IR remote controls. It contains a device driver for hardware connected to the serial port, a daemon that decodes and sends IR signals using this device driver, a mouse daemon that translates IR signals to mouse movements and a couple of user programs that allow to control your computer with a remote control. Takahide Higuchi wrote about LIRC: "It's great, and it seems almost complete solution, but it seems there is almost nothing supporting hardware on the market (or need to solder some special circuit ... it is hard work for many people to do so). I believe that LIRC will be more popular if consumer IR support is implemented in FastIR drivers and some common API (for example, a raw IrSocket and common ioctls) is made!". You may find LIRC at http://www.thp.uni-koeln.de/~rjkm/lirc/

- Serial Infrared Remote Controller

  This is a simple, cheap device that can be connected to any serial port to control most components that have infrared remote controls. It was designed and built on a solderless breadboard and is finally designed as a PC board. You may find this package at http://www.armory.com/~spcecdt/remote/remote.html

- Infrared Tools for the COREL Netwinder PC

  Ryan Shillington wrote some tools to control the COREL Netwinder via infrared, for example:

  Server Side for the Corel Palm Administrator (deamon). It depends on having ir-simple installed and up and running. With this you can check and change IP addresses, Gateway addresses, setup eth1, etc. You can also run simple commands AND you can check the Temperature, Memory, Load averages, etc.

  Client Side for the Corel Palm Administrator. You can also run simple commands AND you can check the Temperature, Memory, Load averages, etc.

  A very basic Infra Red device driver. This does not support IrDA (only unreliable transfers). It looks specifically for Remote Control signals (and Keyboard, etc.). It blocks and passes data up very differently.

  You may find the tools at http://www.netwinder.org/~ryansh/

13.2 Infrared Remote Control - IrDA

TOC

Two of the above mentioned projects use some kind of selfmade dongle for infrared remote control. There is also a description to build a serial IrDA dongle by yourself in the german ELEKTOR 5/97 p. 28 magazine. Maybe someone can merge these two kind of dongles together.

For a discussion of the relation between Infrared Remote Control and IrDA I quote from the Linux/IrDA mailing list (shortend and modified by wh):

Ryan Shillington wrote: "Remote IR and ASK-IR are very different from FIR or MIR or SIR.

Remote IR and ASK-IR are very low speed and low frequency (but very long range) uses for IR. They operate around 2400 baud.

SIR operates at higher rates, and is meant for long range transmission where you need more than a few characters pass through (unlike a remote control).

MIR is a little faster (less range), but with speeds up to 1.15 Mbps, and FIR (where the devices have to be practically touching) is 4Mbps. The range is inversely proportional to the speed you can send data at.

I'm working on drivers for Remote-IR, but you should know that your IR stuff has to support it. Look for protocols like NEC, RC-5 or RC-0 (those are the most common ones).

You can use SIR to receive Remote Control signals. Set your baud rate nice and low and data will come through. BUT, from my experience, it's not the RIGHT data. It's not being analyzed in the right way, and as such, you can't compute the checksums or check it with its complement.

I have managed to get data in (using SIR) with remote controls. I have been told that SIR will read the remote control stuff differently depending on temperature (although I have never had that experience). "

Lichen Wang <lwang1@ix.netcom.com> wrote in response: "The so-called ASKIR in most laptops etc. is not meant for remote IR devices. ASKIR is meant for Sharp Wizard and Zauaus PDAs and some of Sharp's notebook PCs. Sharp stated this long before IrDA was established and is still supporting it to maintain backward compatibility. Apple's Newton had this capability at one time, too.

Briefly, ASKIR uses 9.6 Kbps (19.2 and 38.4 Kbps are also possible) asynchronous data format of 8 data bits, 1 stop bit, and odd parity. The "start" bit as well as all 0 bit in data/parity are transmitted as IR square wave at 500 KHz (DASK sub-carrier). The "stop" bit as well as all 1 bit in data/parity are represented by the absence of any IR transmission.

As you can see, this is totally incompatible with exiting IR remote control.

[..]

True. Not only can you use SIR hardware to *receive*, you can transmit, too. Of course, there are some limitations.

Most IR remote controls use 38 KHz sub-carrier. 3 times 38 is 114, very close to 115.2. You can set the UART to operate at 115.2 Kbps, 7 data bits, no parity, and 1 stop bit - a total of 9 bits. Each 3 cycles of the 38 KHz sub-carrier can be *received* or *transmitted* as a byte of 0x5B.

There are some physical limitations in addition to the fact that the sub-carrier must be 38 KHz. The SIR *receiver* is not as sensitive to 38 KHz as the IR remote receiver designed for that. The SIR *transmitter* has a much lower duty cycle and thus can not emit a strong sub-carrier either.

IR remote encodes the control signal by turning on and off the sub-carrier at certain specific patterns. Now that you can *transmit* and *receive* the sub-carrier, what remains is all in timing.

For *transmit*, you have to know how many consecutive bytes of 0x5B to send for each burst of the sub-carrier, and how long to be quiet between the bursts.

For *receive*, you have to know how many of the 0x5Bs you received are consecutive, and how long the gaps were between these groups of consecutive bytes.

[..]

My experience with the IrDA link distance of SIR, MIR and FIR is somewhat different from what Ryan said.

[..]

SIR, MIR and FIR should all work from 0 to 100 cm but in practice:

(a) Some devices may have problems at *LONG* distances.

When possible, place the two communicating devices no more than 50 cm apart. Low power devices, such as Pagers, Phones, etc. may have even shorter ranges despite the fact that they use SIR instead of MIR or FIR.

(b) Some devices may have problems at *SHORT* distances.

Place the two devices at least a few cm apart. Putting the two devices too close to each other can cause troubles.

It is somewhat intuitive that when the link is not reliable we put the two devices closer together. But it is counterintuitive that too close is not good either. The reason is that the light intensity at 1 cm is 10.000 times brighter than that at 100 cm. At 0.5 cm, it is 40.000 times, etc. The IR receiver manufacturers have difficulties to cover this huge dynamic range. We all have problems reading under a 10 W light bulb, but imagine how it feels under a 100.000 W light!

[..]

The IrDA Physical Layer is totally incompatible with the DASK modulation used in IR remote controls. Thus it is not possible to use the same controller function for both FIR and remote control. However, practically all FIR controller chips do include some additional functions to support remote control. National, SMC, and Winbond (just to name a few) all have such I/O chips.

The IR transmitter for FIR and remote control are very similar. I have tried a standard FIR transmitter. It can reach 10 meters for remote control purpose. Thus it performs just as good as transmitters designed for remote control.

The IR receiver for FIR and remote control are somewhat different. A FIR receiver can receive remote

control signals but can reach only 1 meter whereas receivers designed for remote control typically can reach 10 meters.

I have an ISA bus adapter with a National I/O chip that supports both FIR and remote control. I also have IR Dongles that include both FIR and remote control receivers. (Plus a transmitter for both modes.) I cannot find any software to support remote control functions. I did my own experiments in DOS (I cannot run Linux yet.) Anybody interest in this? "

Benny Amorsen wrote: "I have a laptop that is supposed to support ASKIR. The mode of the infrared port can be switched to ASKIR in the BIOS. Having to reboot to switch the mode in the BIOS makes it useless, though, so someone would have to find a way to switch on the fly. "

Dag Brattli wrote: It should be possible to use IrControl (formerly IrBus) for IrDA compliant remote controls. I currently don't know about any remote controls using IrControl standard, but there should be some out there (anyone else who knows better?). You should go to the IrDA site (http://www.irda.org) and get the physical layer standard (which includes IrControl I think).

"Normal" IrDA (using IrLAP) is _not_ well suited for remote control because of the connection oriented nature (and just supports 9600bps for connectionless use). The reason for the limited range is eye-safety they say (but I currently don't know why CIR works better using the same power). I have however seen laptops connect at 4-5 meters (but I don't think that any high speed communication would be possible).

Most IrDA chipsets are capable of CIR operation, and it is quite easy to modify the drivers so they talk CIR. Takahide Higuchi has started to look at IrSockets and it would be great if we could open a "raw" Ir(DA) socket which then could send and receive CIR packets. Then all the CIR applications could live in userspace.

I know that Corel is interested in using CIR for controlling the NetWinder (and they actually have running code). Take a look at http://www.slashdot.org/articles/98/12/05/0916216.shtml or http://www.netwinder.org/~ryansh

---

14. Infrared and Eye Safety

This section summarizes some ideas and thoughts that were exchanged on the Linux/IrDA mailing list. It is not medically wellfounded, and whoever has better evidence or some more wellfounded source of information is encouraged to contribute it to this HOWTO.

The IrDA spec says that the range of IrDA devices has been limited to 1m for reasons of eye safety. Another plausible assumption is that power consumption and IR pollution/crosstalk were reasons for this limitation. In principle there could be danger for the eye, because infrared light is not registered by the eye, and thus the pupil won't close in order to protect the retina from bright IR light sources. This is the same situation as with UV light, which will cause snow blindness eventually, but in contrast to UV light, IR light contains much less harmful energy due to its longer wavelength.

The only legal restrictions and medical advices we were able to find on the web were concerned with infrared emissions of heat lamps or in the welding process. This suggests that IR light as emitted by IrDA devices will be harmless, since even the peak power emitted by strong IR LEDs (ca. 300mW) is several orders of magnitude below the power emitted by medical IR heat lamps (up to 500W). For these, however, you are supposed to wear protective goggles, so maybe if you are looking straight into 1.000 infrared LEDs flashing at once, you should do so, too. The effect of infrared light is mostly heat, though, and not an alteration or destruction of the biological cell structure, such as caused by UV light. Though in the specs for the HP OmniBook 800 Hewlett-Packard recommends not to look directly into the IR LED.

As stated above, this discussion is only based on guesswork and common sense assumptions about the data found in IR LED and heat lamp specs. If anybody with a better medical knowledge can comment on this, please do so!!!

15. Credits

Thanks to:

- Dag Brattli - Linux/IrDA core team
- Thomas Davis - Linux/IrDA core team
- Takahide Higuchi - Linux/IrDA core team
- Ralf Zabka
- Benny Amorsen
- Lichen Wang
- Ryan Shillington
- Richard Titmuss
- Fons Botman
- Rui Oliveira
- Jon Howell
- Carlos Vidal
- Joonas Lehtinen
- Markus Schill
- Bjoern Hansson
- Pawel Machek
- Ho Chin Keong
- Bjoern Mork
- Andreas Butz
- Tang Ning
- The members of the Linux/IrDA mailing list.
- The writers of the other HOWTOs which gave me many inspirations.
- The developers of the SGML-Tools which provided some means to write a HOWTO.

Sorry I didn't start to follow the credits when starting the HOWTO, so probably I forgot somebody.

16. Revision History

- v0.1 to v0.4a, 19 March 1998 to 4 August 1998, drafts, not included in the LDP
- v1.0, 14 August 1998, release to the LDP
- v1.1, 18 August 1998, added info about IrCOMM patch by Takahide Higuchi, minor changes
- v1.2, 24 August 1998, updated to `linux-irda-1998-08-20` snapshot, added FIR section and revision history, minor changes
- v1.3, 27 September 1998, added sections about multiple instances, cellular phones, digital cameras,Linux to Linux connection, the cutting edge - CVS, power saving; some changes in general configuration section, changes in hardware overview section, minor changes
- v1.4, 11 October 1998, better description of IrCOMM support, changes in dongle connection section, changes in Palm III section, minor changes
- v1.5, 12 October 1998, minor changes
- v1.6, 26 October 1998, section about IrManager added, updated to the `linux-irda-1998-10-21` snapshot, changed dongle connection section, minor changes
- v1.7, 1 November 1998, added remote control section, changed dongle connection section, minor changes
- v2.0, 9 January 1999, nearly complete rewrite and rearrangement according to the new structure of Linux/IR which is included into the kernel since 2.1.131, added info about BIOS support into dongle connection section, configuration tool section and CVS section removed
- v2.1, 13 January 1999, minor changes
- v2.2, 26 January 1999, project name changed from Linux/IR to Linux/IrDA, extended the Troubleshooting chapter, changed the order of the Known Bugs chapter after the Troubleshooting chapter, removed some lint
- v2.3, 4 February 1999, added chapter about Eye Safety written by Andreas Butz; spell checking, reworking of Kernel Parameters chapter and additional information by Andreas Butz; minor changes
- v2.4, 9 February 1999, changed information about applying a patch file

17. Copyright and Disclaimer

Copyright © 1998, 1999 by Werner Heuser. This document may be distributed under the terms set forth in the LDP license at http://metalab.unc.edu/LDP/COPYRIGHT.html.

The information in this document is correct to the best of my knowledge, but there's a always a chance I've made some mistakes, so don't follow everything too blindly, especially if it seems wrong. Nothing here should have a detrimental effect on your computer, but just in case I take no responsibility for any damages incurred from the use of the information contained herein.

ISP-Hookup-HOWTO

Egil Kvaleberg, egil@kvaleberg.no

v1.26, 5 March 1998

This document describes how to use Linux to connect to an Internet Service Provider via a dial-up modem TCP/IP connection. As well as the basic dial-up procedure and IP establishment, email and news handling is covered.

1. Introduction

- 1.1 Copyright
- 1.2 Disclaimer
- 1.3 Translations
- 1.4 New versions of this document
- 1.5 Feedback

2. How do I connect to the rest of the world?

- 2.1 The basic configuration

3. How do I *surf*?

4. How do I send and receive email

5. News

- 5.1 How do I set up an online news-reader?
- 5.2 How do I set up an offline news-reader?
- 5.3 How do I set up C News?
- 5.4 How do I set up Leafnode?

6. How do I automate the connection procedure?

TOC

7. Final words

- 7.1 Other things I should know about?

8. ISP specific information

- 8.1 How do I learn more?
- 8.2 Thanks to

---

1. Introduction

This description has been made to answer a few questions about how dial-up ISP (Internet Service Provider) subscribers may configure and use Linux.

To aid those who will connect their Linux based machines to an ISP for the first time, an attempt has been made to cover most issues encountered. This quite unavoidably will create a certain degree of overlap with other Linux Howto-documents and LDP books. Reference should be made to these documents to provide better understanding and detail.

Much of the existing documentation is targeted towards users with a certain degree of experience, and first time users will often have trouble sorting out the relevant information.

To simplify, the examples given will assume the following:

- User name: `dirk`
- Password: `PrettySecret`
- Internet service provider: `acme.xz`
- Email server: `mail.acme.xz`
- News server: `news.acme.xz`
- Name server: `193.212.1.0`
- Phone number: `12345678`

Our `dirk` will be calling his machine `roderick`.

All references in the table above should naturally be replaced by whatever is valid for the ISP one is using. Often, just a minimum of changes will otherwise be required for users with different ISPs. I would like to be informed about what problems you encounter on this account.

1.1 Copyright

Copyright (c) 1996,1997,1998 by Egil Kvaleberg. This document may be distributed under the terms set forth in the LDP license at

http://sunsite.unc.edu/LDP/COPYRIGHT.html

1.2 Disclaimer

No liability for the contents of this documents can be accepted. Use the concepts, examples and other content at your own risk.

One of many possible setups will be described. In the Linux world, there is usually a number of ways in which to accomplish things. Paragraphs containing hints to alternatives are marked by **ALT:** Please also note that FTP-references often will change slightly as new versions of programs arrive.

As far as I know, only programs that under certain terms may be used or evaluated for personal purposes will be described. Most of the programs will be available complete with source under GNU-like terms.

1.3 Translations

This document has been translated to the following languages:

German, by Cristoph Seibert:

http://tech.appl-opt.physik.uni-essen.de/LinuX/german-howto/DE-ISP-Verbindung-HOWTO.html

Korean, by Kidong Lee:

http://www.postech.ac.kr/~cessi/howto/ISP-Hookup/ISP-Hookup-HOWTO.html

French, by Eric Jacoboni:

http://www.minet.net/linux/HOWTO-fr/ISP-Hookup-HOWTO.html

Polish, by Piotr Pogorzelski: **URL TBA**

Norwegian:

http://www.kvaleberg.com/no-linux.html

1.4 New versions of this document

New versions of this document will be periodically posted to *comp.os.linux.answers*. They will also be added to the various anonymous FTP sites who archive such information, including:

ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO

In addition, you should generally be able to find this document on the Linux Documentation Project page via:

http://sunsite.unc.edu/LDP/

The very latest version of this document should also be available in various formats from:

ftp://ftp.sol.no/user/egilk/ISP-Hookup-HOWTO.txt

ftp://ftp.sol.no/user/egilk/ISP-Hookup-HOWTO.ps.gz

ftp://ftp.sol.no/user/egilk/ISP-Hookup-HOWTO.tar.gz

http://www.kvaleberg.com/ISP-Hookup-HOWTO.html

1.5 Feedback

All comments, error reports, additional information and criticism of all sorts should be directed to:

egil@kvaleberg.no

http://www.kvaleberg.com/

2. How do I connect to the rest of the world?

It will be assumed that we have installed the essential networking software modules from your Linux distribution, and that you have set up which serial port that is to be used for the /dev/modem.

The default configuration will usually only allow direct access to /dev/modem as user `root`.

To connect to ISP shell accounts directly, and to experiment with connection sequences, you may use the minicom program. It is pretty straight forward to use.

2.1 The basic configuration

Configuration of the machine for use on the net should be done as user `root`. Before proceeding any further, ensure that the file /etc/hosts.deny contains the following line:

```
ALL: ALL
```

You would normally want to allow yourself, so add the following line to /etc/hosts.allow:

```
ALL: LOCAL
```

Or if you insist:

```
ALL: 127.0.0.1
```

For the following, note that it is meant for those connected via PPP and with a dynamic IP address. If you have the benefit of a fixed connection, there will be some differences.

It is nice to have a name connected to the machine, a name that the dynamic IP user really can select as he or she pleases. Put the name in /etc/HOSTNAME:

```
roderick
```

The next step is to set up the name server in /etc/resolv.conf:

```
search .
nameserver 193.212.1.0
<nidx>nameserver
```

The name server must be specified by a numeric IP address, and will be different from ISP to ISP. If required, you can have up to three different servers, each on a separate line. They will be requested in the sequence in which they are listed.

If you want to be able to use names like `somemachine` as an abbreviation for `somemachine.acme.xz`, you must replace the first line with:

TOC

```
search acme.xz
```

A certain minimum of configuration will also be required in /etc/hosts. Most users will be able to manage with:

```
127.0.0.1       localhost
0.0.0.0         roderick
```

Those with a fixed IP-address will obviously replace 0.0.0.0 with that.

Likewise, a minimum /etc/networks is:

```
loopback        127.0.0.0
localnet        0.0.0.0
```

You should also set your external mail domain in /etc/mailname:

```
acme.xz
```

The username and password at the ISP must be specified in /etc/ppp/pap-secrets

```
dirk * PrettySecret
```

For those ISPs using CHAP instead of PAP the filename is /etc/ppp/chap-secrets.

Finally, the nitty gritty regarding the connection procedure itself must be specified before PPP can be initiated. This is done in /etc/ppp/chatscript:

```
TIMEOUT 5
"" ATZ
OK ATDT12345678
ABORT "NO CARRIER"
ABORT BUSY
ABORT "NO DIALTONE"
ABORT WAITING
TIMEOUT 45
CONNECT ""
TIMEOUT 5
"name:" ppp
```

Details here may have to be tuned somewhat. The phone number in the third line must of course be set as required. Some users may need to replace the `ATZ` modem initialization string with something more tailored for the modem being used. The last line specifies that one is expecting a prompt that ends with `name:`, and that the response should be `ppp` when it arrives. Other systems may have other login procedures.

To actually initiate a call, the PPP protocol may be initiated by issuing the following command:

TOC

```
exec pppd connect \
     'chat -v -f /etc/ppp/chatscript' \
      -detach crtscts modem defaultroute \
      user dirk \
      /dev/modem 38400
```

We should now be on-air, and stay up until the program is killed by typing a ctrl-C. Any messages concerning the connection will be appended to the system logs. To read them, try:

```
tail /var/adm/messages
```

or

```
dmesg
```

As long as PPP is up, you will have direct access to the Internet and may use programs like ftp, ncftp, rlogin, telnet, finger etc. All these programs should be part of the network package.

Further information concerning PPP is also available from:

/usr/lib/ppp/README.linux

/usr/lib/ppp/README.linux-chat

Finally, an additional word about security The file /etc/inetd.conf lists all services that your machine will offer externally. With the /etc/hosts.deny file we have made, no external access will be allowed. For those who need it, access must be allowed explicitly in /etc/hosts.allow. Local traffic may be allowed by:

```
ALL: LOCAL
```

See also `man 5 hosts_access`.

A final minor issue: A certain confusion exists regarding the names of the POP protocols. A definition in /etc/services compatible with just about everything is:

```
pop2            109/tcp         pop-2           # PostOffice V.2
pop3            110/tcp         pop-3 pop       # PostOffice V.3
```

ALT: Instead of chatscript, one might use the much more flexible dip. But not in connection with diald.

ALT: Those fortunate enough to have a permanent TCP/IP connection via e.g. an Ethernet may safely ignore anything about PPP and rather start concentrating about setting up their network card.

ALT: Others may not have the possibility of using PPP, but may be able to use SLIP instead, for which there is support in much the same manner as for PPP. Another possibility is UUCP Others again may have to rely on exchange of news and email be means of SOUP A description for the latter case may be found in:

ftp://ftp.sol.no/user/bjorn/Linux-offline.tgz

The TERM program is also an option. Refer to the *Term-HOWTO*.

3. How do I *surf*?

If you think that text is the most important, you might want to use the Lynx web browser. It is available from:

ftp://sunsite.unc.edu/pub/Linux/system/Network/info-systems/lynx-2.3.bin2.tar.gz

If you have installed the X window system, you can also use one of the many graphical browsers. Chimera may be found at:

ftp://sunsite.unc.edu/pub/Linux/system/Network/info-systems/chimera-1.65.bin.ELF.tar.gz

http://www.unlv.edu/chimera/

Mosaic

ftp://sunsite.unc.edu/pub/Linux/system/Network/info-systems/Mosaic-2.7b1-aout.tgz

ftp://ftp.NCSA.uiuc.edu/Web/Mosaic/Unix/binaries/2.6

Netscape (aka. Mozilla

ftp://sunsite.unc.edu/pub/Linux/system/Network/info-systems/netscape-v11b3.tar.gz

ftp://ftp.cs.uit.no/pub/www/netscape

These browsers are constantly available in new and in various ways *exciting* versions.

Use and evaluation of these programs is subject to certain terms. Please observe them.

4. How do I send and receive email

First of all, ensure that sendmail is installed. Sendmail sorts internal and out-bound mail, and will buffer out-bound mail until such time it is possible to forward it.

Sendmail is based on a configuration found in /etc/sendmail.cf. An example suitable for ISP users can be found in:

ftp://ftp.sol.no/user/egilk/sendmail.cf This is based on procmail as a delivery agent but may easily be changed to use deliver.

It is if course required to have an **official** domain address for out-bound mail, something which is specified in sendmail.cf:

```
# who I masquerade as (null for no masquerading)
DMacme.xz
```

This assumes that you have the same user name locally as you have at your ISP. If it is different, just specify the full name instead:

```
DMdick@acme.xz
```

Sendmail is now configured for sending **directly** to the recipient. To avoid long and repeated connections in those cases where the connection to the receiving end is slow and irregular, is is usually nice to use ones ISP as a buffer store. This can be specified by the DS specification:

```
# "Smart" relay host (may be null)
DSmail.acme.xz
```

Beware that sendmail is somewhat sensitive to handling of tab stop characters in sendmail.cf. You might want to use the vi editor to ensure that these tab characters are retained unchanged.

Email reception can often be performed via the POP3 protocol, which can be initiated every time the connection is brought up. A script for testing this is:

```
sendmail -q
popclient -3 -v mail.acme.xz -u dirk -p "PrettySecret" \
       -k -o /usr/spool/mail/dirk
```

This script may be started after PPP connection has been established. Beware that this script is just for testing, so ensure that the local mailbox is left untouched while it runs. The `-k` option means that the mail is **kept** in the ISP mailbox, and you are simply given a copy of the mail. You would of course want to remove this option once you are confident that your setup is working.

Beware that the password will show on the command line. Also note that popclient is getting old fashioned, and that you should consider using fetchmail instead.

A more secure and better version of this script may be found at:

ftp://ftp.sol.no/user/egilk/pop-script.tar.gz

This version of the script requires that procmail is installed, but that is something you'll never regret anyway. Most distributions include it, otherwise you may try:

ftp://sunsite.unc.edu/pub/Linux/system/Mail/mailhandlers/procmail-3.10-2.ta

Procmail is a simple and flexible tool that can sort incoming email based on a large range of criteria. In addition to being able to handle automated tasks like vacation messages and such.

Note that when we use procmail directly as in this case, the situation is somewhat different from what is described in the procmail documentation. A .forward is **not** required, and we also don't need a .procmailrc. The latter is only required if we want to sort the mail.

The user interface for reading and sending of email can be found in programs like pine or elm.

ALT: Fetchmail has recently become an improved alternative to popclient. The latest version is available from:

ftp://ftp.ccil.org/pub/esr/fetchmail/fetchmail-3.3.tar.gz

ALT: For an ordinary dial-up ISP user it is not really necessary to have the sendmail daemon active. To reduce resource usage, and possibly other problems, one may thus comment out any startup of sendmail, as is usually found in /etc/rc.d/rc.M (this varies from distribution to distribution).

ALT: In place of sendmail one might use the simpler smail. You'll find a good description of it (as well as most other things mentioned here) in the *Linux Network Administrator's Guide*.

ALT: There is also an m4 macro package for making a fresh /etc/sendmail.cf. For a simple installation it might be just as well to modify an existing configuration.

ALT: There are also simpler although less flexible alternatives for handling email. Pine may run stand-alone as long as it is configured properly, for instance. It might even be possible to use newer versions of some web-browsers.

ALT: Many are very enthusiastic regarding the Emacs companion Gnus as an email and news handler. Further information can be found at:

http://www.ifi.uio.no/~larsi/

ALT: An alternative to popclient is pop-perl5. It is available from:

ftp://sunsite.unc.edu/pub/Linux/System/Mail/pop-perl5-1.1.tar.gz

---

5. News

5.1 How do I set up an online news-reader?

As long as PPP is active, it will be possible to read news **online**. There are lots of available programs, two simple alternatives being rtin and trn.

To start reading news, the only thing required in terms of configuration in most cases is to set NNTPSERVER (usually once and for all in the file .profile):

```
export NNTPSERVER=news.acme.xz
```

To get the *From*-address correct in postings, some programs *may* require:

```
export NNTP_INEWS_DOMAIN=acme.xz
```

5.2 How do I set up an offline news-reader?

To be able to read news while offline and thus reduce phone bills and give greater flexibility, one must set up a local news-spool of one sort or the other. This requires some configuration, and there will also be a certain amount of disk space involved. After initial setup, things should run more or less by themselves, with only some attention needed from time to time.

Two different solutions will be described here.

5.3 How do I set up C News?

The solution described here is based on the news-server **C News** and the NNTP protocol. C News was originally targeted towards another sort of configuration, but is flexible enough to handle our situation too. One might also use the more recent **INN** news server but it might require a bit more in terms of resources. Either way, be careful **not** to install both; they don't live together easily.

It is crucial that all maintenance of news is done while logged in as user `news`, and that all configuration files is placed in /usr/lib/news. One way of handling this is, while logged in as `root` to write `su news; cd`.

The most important files in the configuration are:

- active is an overview over active newsgroups. It is updated as required by the command addgroup, e.g. `addgroup comp.os.linux.networking y`.
- organization should simply contain whatever you want in the *Organization:* header field, in our case:

  ```
  Dirk Gently's Holistic Detective Agency
  ```
- mailname should in our case be set to `acme.xz`.
- whoami is set to the name of your `site` in the Path: thread. In a setup as described here, using newsx, this name will never leave the machine, so you can set this to whatever you like as long as you are pretty sure it

is unique. In this case `roderick`.

- The file sys controls fetching and further distribution of news. We will assume the ISP in our case adds `acme.xz` to the Path, and that this is the only news source we have. The example given really tells that we will accept everything that arrives, and that we will only post news to `acme.xz` that it hasn't seen before, and is originally posted at our own site. In this simplified setup we assume that the all groups will come from a single source. `/all` specifies the distribution, and **must** be included. The letter `F` says that (pointers to) outgoing news articles will be collected in a file.

```
ME:all/all::
acme/acme.xz:all,!junk/all:FL:
```

- A subdirectory for the outgoing news must be created, in our case:

```
mkdir /var/spool/news/out.going/acme
```

- mailpaths controls posting in moderated groups, although this task may usually be left to the ISP.

C News needs a certain degree of daily maintenance, but this can be specified once and for all via the command `crontab -e` issued as user `news`. A suggested setup follows; it can be tuned as required:

```
# maintain incoming and outgoing batches
10,40 *  * * * /usr/lib/newsbin/input/newsrun

# expire C News, once a day
30 0  * * * /usr/lib/newsbin/expire/doexpire

# monitor and report if needed
00 2  * * sat /usr/lib/newsbin/maint/addmissing
40 3  * * * /usr/lib/newsbin/maint/newswatch
50 3  * * * /usr/lib/newsbin/maint/newsdaily
```

newsrun moves articles in and out (twice every hour), doexpire will delete articles as they get old (every night at 00:30), and the three last commands does various supervisory and error correcting tasks.

One should also ensure that things are cleaned up when starting the machine. As user root, add the following line to /etc/rc.d/rc.local:

```
su news -c /usr/lib/newsbin/maint/newsboot
```

News may be collected via the program **NewsX**, picking news from an NNTP-server The program can be found at:

ftp://sunsite.unc.edu/pub/Linux/system/news/transport/newsx-0.9.tar.gz Or:

ftp://ftp.sol.no/user/egilk/newsx-0.9.tar.gz

Setting up `NewsX` is quite simple. Installation is a classic case of:

```
make
su
make install
exit
```

TOC

With the setup outlined here, all you have to do is to create the groups you want to read using the addgroup command.

To fetch articles, user `news` issues the following commands (assuming communication via PPP or similar is up):

```
newsrun
newsx acme news.acme.xz
newsrun
```

The option `-d` gives continuous printout to the screen. Refer to the NewsX documentation for further information.

NewsX will also take care of posting of outgoing news.

To control disposal of articles as they get old, a file `explist` is required. The comments in this example should explain what we want to do:

```
# hold onto history lines 14 days, nobody gets >120 days
/expired/                        x       14      -
/bounds/                         x       0-1-120 -

# retain these for 2 months
comp.sources,comp.os.linux.all   x       60      -

# noise gets thrown away fast
junk,control                     x       2       -

# default:  14 days, no archive
all                              x       14      -
```

ALT: In a small news-spool, one will usually not need the newsgroup `control`. The traffic is **huge** compared to the possible usefulness. The main point is that articles will be canceled, and that groups may be created automatically. To ensure that control messages containing `newgroup` not shall mess up things for us, a file called newgroupperm specifies what we will allow:

```
comp.os.linux   tale@uunet.com  yv
all             any             nq
```

In this example, all proper groups under comp.os.linux will be created (y), and the user `news` will be notified (v). Everything else will be silently (q) ignored (n). The last line is sufficient if you want to create all groups manually.

ALT: An alternative to NewsX is suck.

5.4 How do I set up Leafnode?

A different solution altogether is to install the integrated package leafnode. This will handle all tasks required for a personal news spool, and is easy to configure. It is available via:

http://www.troll.no/freebies/leafnode.html

TOC

As for C News, all news maintenance really should be performed as user `news`.

The home directory for leafnode is in /usr/lib/leafnode. To install, write:

```
cd /usr/lib/leafnode
tar -xzvf leafnode-0.8.tgz
cd leafnode-0.8
make
su
make install
```

Note in the following that the prefix /usr/local/sbin should be replaced with /usr/sbin if you installed leafnode from a package.

While still being logged in as `root`, change the line that controls NNTP in /etc/inetd.conf:

```
nntp  stream  tcp  nowait  news  /usr/sbin/tcpd /usr/local/sbin/leafnode
```

Activate it by:

```
killall -HUP inetd
```

Return to user `news` by writing exit. In /usr/lib/leafnode/config change the line that defines the NNTP server. In our case:

```
server = news.acme.xz
```

Leafnode will look after itself by adding the following command via `crontab -e` as user `news`:

```
# expire Leafnode, once a day
0 4 * * * /usr/local/sbin/texpire
```

News exchange is also done as user `news` by the following command (assuming PPP is up and running):

```
/usr/local/sbin/fetch
```

Users who wants to read news should then use the recipe in *How do I set up an online news-reader?*, except that they configure for the local machine, i.e:

```
export NNTPSERVER=localhost
```

That should be all there is to it. The first `fetch` will transfer a list of available newsgroups. Leafnode will then monitor what groups the users are requesting, and adapt to this the **next** time it is activated.

Note that leafnode does not seems to work in cases where NNTP authorization is required. +.LP

ALT: An alternative to leafnode is nntpcache, available from:

`ftp://ftp.suburbia.net/pub/nntpcache/nntpcache.tgz` **ALT:** Another alternative is to use the newsreader slrn together with the slrn-pull package. The newsreader must be compiled with the `spool` feature set.

---

6. How do I automate the connection procedure?

Automated handling of news and email is quite easy to implement in Linux.

First and foremost one should make a /usr/lib/ppp/ppp-on that initiates the ISP connection. Often, this file will simply contain the following:

```
/usr/sbin/pppd
```

Further specification will be performed in /etc/ppp/options:

```
connect "/usr/lib/ppp/chat -v -f /etc/ppp/chatscript"
crtscts
modem
defaultroute
asyncmap 00000000
user dirk
/dev/modem 38400
```

To end a connection, use the supplied version of /usr/lib/ppp/ppp-off.

Having tested the functionality of these two scripts, one must then write scripts that perform the various tasks. The script to collect email has been described before, and we will here assume it is located at /home/dirk/pop.

A script for exchange of email can then be produced in /root/mail:

```
#! /bin/sh
#
# exchange mail
# 10 minutes timeout:
TIMEOUT=600
DT=10

# kick sendmail:
sendmail -q &

# retrieve mail:
su dirk -c /home/dirk/pop

# wait for sendmail to terminate:
t=0
while ! mailq | grep -q "Mail queue is empty"; do
    t=$[$t+$DT]
```

```
        if [ $t -gt $TIMEOUT ] ; then
         echo "sendmail -q timeout ($TIMEOUT).."
         exit 1
        fi
        sleep $DT
    done

    exit 0
```

The script to exchange news may be placed in /usr/lib/news/news:

```
    #!/bin/sh
    #
    # exchange news
    # must be run as news:
    cd /usr/lib/news

    #update the outgoing batch (C News):
    /usr/lib/newsbin/input/newsrun < /dev/null

    #exchange news:
    /usr/lib/newsbin/newsx acme news.acme.xz

    #and flush the incoming batch:
    /usr/lib/newsbin/input/newsrun < /dev/null
```

A script to connect the various bits and pieces remains, and can be placed in /root/news+mail:

```
    #!/bin/sh
    #
    # exchange news and email
    # must be run as root
    #
    if ! /usr/lib/ppp/ppp-on; then
        exit 1
    fi
    trap "/usr/lib/ppp/ppp-off" 1 2 3 15

    #exchange news+mail:
    /root/mail &
    su news -c ~news/news
    wait

    #disconnect..
    /usr/lib/ppp/ppp-off
```

```
#update the incoming batch (C News):
su news -c /usr/lib/newsbin/input/newsrun < /dev/null &

exit 0
```

It is quite easy to make an extension to the above that only will establish a connection if outgoing email and news is present. Lets call it /root/news+mail.cond, and keep in mind that the name of the outgoing news-spool must be updated to suit:

```
#!/bin/sh
#
# exchange news and email, only if outgoing news or mail
# (C News spool)
if [ -s /var/spool/news/out.going/acme/togo ] ||
    ! ( mailq | grep -q "Mail queue is empty"); then
     /root/news+mail
fi
```

The only thing remaining is to specify when all this is going to happen. This is done using the command `crontab -e` as root. Let us assume that we always want to exchange news and mail at 07:00 in the morning, and after that every 4th hour assuming there are outgoing email and news:

```
00 7            * * *   /root/news+mail
00 11,15,19,23  * * *   /root/news+mail.cond
```

Ensure that every component is tested well before you connect them together. One may later add several other tasks, such as adjustment of the time of day (using ntpdate), and automatic update (mirroring) of locally maintained WWW and FTP files up to the ISP (using make and ftp).

ALT: Depending on ones preferences, it is also possible to turn the process upside down. Every time a PPP link is initiated, the script /etc/ppp/ip-up will be started. One may here add whatever magic is required to start exchange of email and news. See `man pppd` for further detail.

ALT: It is also possible to automatically connect PPP whenever network traffic is detected. This is in many ways the more elegant solution, but it is quite dependent on a good configuration to avoid frequent (and costly) connections being made. More information can be found at:

http://www.dna.lth.se/~erics/diald.html

The `diald` utility is available from:

ftp://sunsite.unc.edu/pub/Linux/system/network/serial/diald-0.16.tar.gz

At the same location one will also find other variations on the theme PPP connections.

---

7. Final words

7.1 Other things I should know about?

- Various error messages in the system will normally be issued as internal email. To ensure that these will actually be read, one should create an /etc/aliases. Remember the command `newaliases` every time you change this. An example that should cover most eventualities is:

  ```
  PostMaster: root
  ftp: root
  news: root
  usenet: root
  FaxMaster: root
  fax: root
  WebMaster: root
  MAILER.DAEMON: root
  ```

- Many programs for Linux may be found at **Sunsite**, which is usually quite busy. But there are many mirrors and every time there is a reference to ftp://sunsite.unc.edu/pub/Linux/.. one should try to use a mirror close to home, e.g. ftp://ftp.nvg.unit.no/pub/linux/sunsite/...
- If you happen to be migrating from Yarn it should be possible to convert these to standard folders using the yarn2mf available at:

ftp://ftp.sol.no/user/egilk/yarn2mf.zip

8. ISP specific information

More specific information for certain ISPs is available from a variety of sources:

Demon Internet

ftp://ftp.demon.co.uk/pub/unix/linux/Demon/slack3.0.help.tgz

Netcom

http://www.netcom.com/bin/webtech/NetCruiser/Operating_Systems/Linux/linux.cfg.html

PowerTech, Telenor Online, Telia

http://www.kvaleberg.com/no-isp.html

Stanford

http://www-leland.stanford.edu/~wkn/Linux/network/network.html

MCI

http://www.kvaleberg.com/linux-mci.html

SISCOM

http://www.siscom.net/support/linux_setup.htm

AOL is not possible since AOL uses a proprietary protocol.

If you can supply ISP specific information not listed here, please get in touch.

8.1 How do I learn more?

The Linux Documentation Project book called Linux Network Administrator's Guide by Olaf Kirch is pretty mandatory for anyone that will set up and maintain anything involving TCP/IP and Internet

ftp://sunsite.unc.edu/pub/Linux/docs/linux-doc-project/network-guide/nag-1.0.ascii.tar.gz

The documentation that follows each software package will normally give you all the detailed information you need, if not always the overview. The man-pages will be the first place to look. Try for instance:

```
man pppd
```

You will also find some documentation about certain programs in the /usr/doc tree, although this is not always well structured.

The following HOWTOs will be highly relevant:

- **Installation-HOWTO** will get the basics sorted.
- **NET-2-HOWTO** is a very thorough description of installation and setup of the NET code. Much of this should already have been done if you use a standard Linux distribution (e.g. Slackware, Red Hat, Debian). But many sections on setup and troubleshooting will be very worthwhile.
- **Mail-HOWTO** explains how to configure various tools. Again, much of this will already have been done for you when you install a standard Linux distribution.
- **News-HOWTO** is for setting up a (conventional) news spool.
- **Tiny-News** covers yet another alternative for collecting news.
- **PPP-HOWTO** is a good description of problems you may encounter when setting up a PPP connection.
- **Serial-HOWTO** contains everything you need to know about setting up serial ports.

- **Mail-Queue** tells you how to send up *sendmail* to always queue remote mail but deliver local mail at once.

Red Hat has a mailing list for PPP issues; to join send an email to

redhat-ppp-list-request with the subject line

```
subscribe
```

8.2 Thanks to

Information here is collected from many sources. Thanks to the following that either indirectly or directly have contributed:

```
Adam Holt <holt@graphics.lcs.mit.edu>
Arne Coucheron <arneco@oslonett.no>
Arne Riiber <riiber@oslonett.no>
Arnt Gulbrandsen <agulbra@troll.no>
Bjorn Steensrud <bjornst@powertech.no>
Gisle Hannemyr <gisle@a.sn.no>
Hans Amund Rosbach <haro@sesam.dnv.no>
Hans Peter Verne <hpv@ulrik.uio.no>
Harald T Alvestrand <Harald.T.Alvestrand@uninett.no>
Harald Terkelsen  <Harald.Terkelsen@adm.hioslo.no>
Haavard Engum <hobbes@interlink.no>
James Youngman <JYoungman@vggas.com>
Johan S. Seland <johanss@sn.no>
John Phillips <john@linux.demon.co.uk>
Jorn Lokoy <jorn@oslonett.no>
Kenneth Tjostheim <kenneth.tjostheim@asplanviak.no>
Kjell M. Myksvoll <kjell.myksvoll@fou.telenor.no>
Kjetil T. Homme <kjetilho@math.uio.no>
Michael Meissner <meissner@cygnus.com>
N J Bailey <N.J.Bailey@leeds.ac.uk>
Nicolai Langfeldt <janl@math.uio.no>
Ove Ruben R Olsen <Ove.R.Olsen@ub.uib.no>
R. Bardarson <ronb@powernet.net>
Steinar Fremme <steinar@fremme.no>
Sverre H. Huseby <sverrehu@ifi.uio.no>
Trond Eivind Glomsrod <teg@stud.imf.unit.no>
Tommy Larsen <tommy@mix.hive.no>
Yves Bellefeuille <yan@storm.ca>
```

The Linux Installation HOWTO

by Eric S. Raymond

v4.15, 20 November 1998

This document describes how to obtain and install Linux software. It is the first document which a new Linux user should read to get started.

1. Introduction

- 1.1 Purpose of this document
- 1.2 Other sources of information
- 1.3 New versions of this document
- 1.4 Feedback and Corrections

2. Recent Changes

3. The Easiest Option: Buy, Don't Build

4. Before You Begin

- 4.1 Hardware requirements
- 4.2 Space requirements and coexistence
- 4.3 Choosing a Linux distribution

5. Installation Overview

- 5.1 First Installation Steps: The Easy Way
- 5.2 First Installation Steps: The Hard Way
- 5.3 Continuing the Installation
- 5.4 Basic Parts of an Installation Kit

TOC

6. Installation In Detail

- 6.1 Getting prepared for installation
- 6.2 Creating the boot and root floppies
- 6.3 Repartitioning your DOS/Windows drives
- 6.4 Creating partitions for Linux
- 6.5 Booting the installation disk
- 6.6 Using the rootdisk
- 6.7 Installing software packages
- 6.8 After package installations

7. Booting Your New System

8. After Your First Boot

- 8.1 Beginning System Administratration
- 8.2 Custom LILO Configuration

9. Administrivia

- 9.1 Terms of Use
- 9.2 Acknowledgements

---

1. Introduction

1.1 Purpose of this document

Linux is a freely-distributable implementation of Unix for inexpensive personal machines (it was developed on 386s, and now supports 486, 586, Pentium, PowerPC, Sun Sparc and DEC Alpha hardware). It supports a wide range of software, including X Windows, Emacs, TCP/IP networking (including SLIP), and many applications.

This document assumes that you have heard of and know about Linux, and now want to get it running. It focuses on the Intel base version, which is the most popular, but much of the advice applies on Power PCs, Sparcs and Alphas as well.

1.2 Other sources of information

If you are new to Linux, there are several sources of basic information about the system. The best place to find these is at the Linux Documentation Project home page at http://sunsite.unc.edu/LDP/linux.html. You can find the latest, up-to-date version of this document there, as http://sunsite.unc.edu/LDP/HOWTO/Installation-HOWTO.html

You should probably start by browsing the resources under General Linux Information; the Linux INFO-SHEET and the Linux META-FAQ. The `Linux Frequently Asked Questions' document contains many common questions (and answers!) about Linux---it is a ``must read'' for new users.

You can find help for common problems on the USENET newsgroups comp.os.linux.help and comp.os.linux.announce.

The Linux Documentation Project is writing a set of manuals and books about Linux, all of which are freely distributable on the net and available from the LDP home page.

The book ``*Linux Installation and Getting Started*'' is a complete guide to getting and installing Linux, as well as how to use the system once you've installed it. It contains a complete tutorial to using and running the system, and much more information than is contained here. You can browse it, or download a copy, from the LDP home page.

Finally, there is a rather technical Guide to x86 Bootstrapping. This document is NetBSD- rather than Linux-oriented, but contains useful material on disk configuration and boot managers for multi-OS setups.

TOC

1.3 New versions of this document

New versions of the Linux Installation HOWTO will be periodically posted to comp.os.linux.help and news:comp.os.linux.announce and news.answers. They will also be uploaded to various Linux WWW and FTP sites, including the LDP home page.

You can also view the latest version of this on the World Wide Web via the URL http://sunsite.unc.edu/LDP/HOWTO/Installation-HOWTO.html.

1.4 Feedback and Corrections

If you have questions or comments about this document, please feel free to mail Eric S. Raymond, at esr@thyrsus.com. I welcome any suggestions or criticisms. If you find a mistake with this document, please let me know so I can correct it in the next version. Thanks.

Please do *not* mail me questions about how to solve hardware problems encountered during installation. Consult ``*Linux Installation and Getting Started*'', bug your vendor, or consult the Linux newsgroup comp.os.linux.setup. This HOWTO is intended to be rapid, painless guide to *normal* installation -- a separate HOWTO on hardware problems and diagnosis is in preparation.

---

2. Recent Changes

- Added the `Buy, Don't Build'.
- Added the material on booting from CD-ROM.

3. The Easiest Option: Buy, Don't Build

Linux has now matured enough that there are now system integrators who will assemble a workstation for you, install and configure a Linux, and do an intensive burn-in to test it before it's shipped to you. If you have more money than time, or you have stringent reliability or performance requirements, these integrators provide a valuable service by making sure you won't get hardware that's flaky or dies two days out of the box.

There are several firms of this kind (and I'll list them here as I learn more about them). The only such outfit I know about personally is VA Research. These good people build high-end, high quality Linux workstations with a nifty Tux-the-penguin logo on the front. They have intimate ties to the Linux community (the Debian project lives on a machine in their back room, Linus owns one of their boxes, and they even throw resources at your humble HOWTO maintainer occasionally).

For those of us without a champagne budget, the rest of this HOWTO is about how to install Linux yourself.

4. Before You Begin

Before you can install Linux, you'll need to be sure your machine is Linux-capable, and choose a Linux to install. The Linux Pre-installation checklist may help you organize configuration data before you begin.

4.1 Hardware requirements

What kind of system is needed to run Linux? This is a good question; the actual hardware requirements for the system change periodically. The Linux Hardware-HOWTO, http://sunsite.unc.edu/LDP/HOWTO/Hardware-HOWTO.html, gives a (more or less) complete listing of hardware supported by Linux. The Linux INFO-SHEET, http://sunsite.unc.edu/LDP/HOWTO/INFO-SHEET.html, provides another list.

For the Intel versions, a hardware configuration that looks like the following is required:

Any 80386, 80486, Pentium or Pentium II processor will do. Non-Intel clones of the 80386 and up will generally work. You do not need a math coprocessor, although it is nice to have one.

The ISA, EISA, VESA Local Bus and PCI bus architectures are supported. The MCA bus architecture (found on IBM PS/2 machines) is supported in the newest development (2.1.x) kernels, but may not be ready for prime time yet.

You need at least 4 megabytes of memory in your machine. Technically, Linux will run with only 2 megs, but most installations and software require 4. The more memory you have, the happier you'll be. I suggest 8 or 16 megabytes if you're planning to use X-Windows.

Of course, you'll need a hard drive and an AT-standard drive controller. All MFM, RLL, and IDE drives and controllers should work. Many SCSI drives and adaptors are supported as well; the Linux SCSI-HOWTO contains more information on SCSI. If you are assembling a system from scratch to run Linux, the small additional cost of SCSI is well worth it for the extra performance and reliability it brings.

You will need a 3.5" floppy drive. While 5.25" floppies are supported under Linux, they are little-enough used that you should not count on disk images necessarily fitting on them. (A stripped-down Linux can actually run on a single floppy, but that's only useful for installation and certain troubleshooting tasks.)

You also need an MDA, Hercules, CGA, EGA, VGA, or Super VGA video card and monitor. In general, if your video card and monitor work under MS-DOS then it should work under Linux. However, if you wish to run X Windows, there are other restrictions on the supported video hardware. The Linux XFree86-HOWTO, http://sunsite.unc.edu/LDP/HOWTO/XFree86-HOWTO.html, contains more information about running X and its requirements.

You'll want a CD-ROM drive. If it's ATAPI, SCSI, or true IDE you should have no problem making it

work (but watch for cheap drives advertising "IDE" interfaces that aren't true IDE). If your CD-ROM uses a proprietary interface card, it's possible the installation kernel you're going to boot from floppy won't be able to see it -- and an inaccessible CD-ROM is a installation show-stopper. Also, CD-ROMs that attach to your parallel port won't work at all. If you're in doubt, consult the Linux CD-ROM HOWTO, http://sunsite.unc.edu/LDP/HOWTO/CDROM-HOWTO.html for a list and details of supported hardware.

So-called "Plug'n'Play" jumperless cards can be a problem. Support for these is under active development, but not there yet in the 2.0.25 kernel. Fortunately this is only likely to be a problem with sound or Ethernet cards.

If you're running on a box that uses one of the Motorola 68K processors (including Amiga, Atari, or VMEbus machines), see the Linux/m68k FAQ at http://www.clark.net/pub/lawrencc/linux/faq/faq.html for information on minimum requirements and the state of the port. The FAQ now says m68k Linux is as stable and usable as the Intel versions.

4.2 Space requirements and coexistence

You'll need free space for Linux on your hard drive. The amount of space needed depends on how much software you plan to install. Most installations require somewhere in the ballpark of 200 to 500 megs. This includes space for the software, swap space (used as virtual RAM on your machine), and free space for users, and so on.

It's conceivable that you could run a minimal Linux system in 80 megs or less (this used to be common when Linux distributions were smaller), and it's conceivable that you could use well over 500 megs or more for all of your Linux software. The amount varies greatly depending on the amount of software you install and how much space you require. More about this later.

Linux will co-exist with other operating systems, such as MS-DOS, Microsoft Windows, or OS/2, on your hard drive. (In fact you can even access MS-DOS files and run some MS-DOS programs from Linux.) In other words, when partitioning your drive for Linux, MS-DOS or OS/2 live on their own partitions, and Linux exists on its own. We'll go into more detail about such ``dual-boot'' systems later.

You do NOT need to be running MS-DOS, OS/2, or any other operating system to use Linux. Linux is a completely different, stand-alone operating system and does not rely on other OSs for installation and use.

In all, the minimal setup for Linux is not much more than is required for most MS-DOS or Windows 3.1 systems sold today (and it's a good deal less than the minimum for Windows 95!). If you have a 386 or 486 with at least 4 megs of RAM, then you'll be happy running Linux. Linux does not require huge amounts of diskspace, memory, or processor speed. Matt Welsh, the originator of this HOWTO, used to run Linux on a 386/16 MHz (the slowest machine you can get) with 4 megs of RAM, and was quite happy. The more you want to do, the more memory (and faster processor) you'll need. In our experience a 486 with 16 megabytes of RAM running Linux outdoes several models of expensive workstations.

4.3 Choosing a Linux distribution

TOC

Before you can install Linux, you need to decide on one of the ``distributions'' of Linux which are available. There is no single, standard release of the Linux software---there are many such releases. Each release has its own documentation and installation instructions.

Linux distributions are available both via anonymous FTP and via mail order on diskette, tape, and CD-ROM. The Linux Distribution HOWTO, http://sunsite.unc.edu/LDP/HOWTO/Distribution-HOWTO.html, includes descriptions of many Linux distributions available via FTP and mail order.

In the dim and ancient past when this HOWTO was first written (1992-93), most people got Linux by tortuous means involving long downloads off the Internet or a BBS onto their DOS machines, followed by an elaborate procedure which transferred the downloads onto multiple floppy disks. One of these disks would then be booted and used to install the other dozen. With luck (and no media failures) you'd finish your installation many hours later with a working Linux. Or maybe not.

While this path is still possible (and you can download any one of several distributions from http://sunsite.unc.edu/pub/Linux/distributions/), there are now much less strenuous ways. The easiest is to buy one of the high-quality commercial Linux distributions distributed on CD-ROM, such as Red Hat, Debian, Linux Pro, or WGS. These are typically available for less than $50 at your local bookstore or computer shop, and will save you many hours of aggravation.

You can also buy anthology CD-ROMs such as the InfoMagic Linux Developer's Resource set. These typically include several Linux distributions and a recent dump of major Linux archive sites, such as sunsite or tsx-11.

In the remainder of this HOWTO we will focus on the steps needed to install from an anthology CD-ROM, or one of the lower-end commercial Linuxes that doesn't include a printed installation manual. If your Linux includes a paper manual some of this HOWTO may provide useful background, but you should consult the manual for detailed installation instructions.

---

5. Installation Overview

It's wise to collect configuration information on your hardware before installing. Know the vendor and model number of each card in your machine; collect the IRQs and DMA channel numbers. You probably won't need this information -- but if it turns out you do, you'll need it very badly.

If you want to run a "dual-boot" system (Linux and DOS or Windows both), rearrange (repartition) your disk to make room for Linux. If you're wise, you'll *back up everything first!*

5.1 First Installation Steps: The Easy Way

If you have an EIDE/ATAPI CDROM (normal these days), check your machine's BIOS settings to see if it has the capability to boot from CD-ROM. Most machines made after mid-1997 can do this.

If yours is among them, change the settings so that the CD-ROM is checked first. This is often in a 'BIOS FEATURES' submenu of the BIOS configuration menus.

Then insert the installation CD-ROM. Reboot. You're started.

If you have a SCSI CDROM you can often still boot from it, but it gets a little more motherboard/BIOS dependent. Those who know enough to spend the extra dollars on a SCSI CDROM drive probably know enough to figure it out.

5.2 First Installation Steps: The Hard Way

1. Make installation floppies.
2. Boot an installation mini-Linux from the floppies in order to get access to the CD-ROM.

5.3 Continuing the Installation

1. Prepare the Linux filesystems. (If you didn't edit the disk partition table earlier, you will at this stage.)
2. Install a basic production Linux from the CD-ROM.
3. Boot Linux from the hard drive.
4. (Optional) Install more packages from CD-ROM.

5.4 Basic Parts of an Installation Kit

TOC

Here are the basic parts of an installable distribution:

1. The README and FAQ files. These will usually be located in the top-level directory of your CD-ROM and be readable once the hard disk has been mounted under Linux. (Depending on how the CD-ROM was generated, they may even be visible under DOS/Windows.) It is a good idea to read these files as soon as you have access to them, to become aware of important updates or changes.
2. A number of bootdisk images (often in a subdirectory). If your CD-ROM is not bootable, one of these is is the file that you will write to a floppy to create the boot disk. You'll select **one** of the above bootdisk images, depending on the type hardware that you have in your system.

The issue here is that some hardware drivers conflict with each other in strange ways, and instead of attempting to debug hardware problems on your system it's easier to use a boot floppy image with only the drivers you need enabled. (This will have the nice side effect of making your kernel smaller.)

- A rootdisk image (or perhaps two). If your CD-ROM is not bootable, you will write one of these to a floppy to create the installation disk(s). Nowadays the root disk or disks is generally independent of your hardware type; it will assume an EGA or better color screen.
- A rescue disk image. This is a disk containing a basic kernel and tools for disaster recovery in case something steps on the kernel or boot block of your hard disk.
- RAWRITE.EXE. This is an MS-DOS program that will write the contents of a file (such as a boot or rootdisk image) directly to a floppy, without regard to format.

You only need RAWRITE.EXE if you plan to create your boot and root floppies from an MS-DOS system. If you have access to a UNIX workstation with a floppy drive instead, you can create the floppies from there, using the `dd' command. or possibly a vendor-provided build script. See the man page for dd(1) and ask your local UNIX gurus for assistance.

- The CD-ROM itself. The purpose of the boot disk is to get your machine ready to load the root or installation disks, which in turn are just devices for preparing your hard disk and copying portions of the CD-ROM to it. If your CD-ROM is bootable, you can boot it and skip right to preparing your disk.

---

6. Installation In Detail

6.1 Getting prepared for installation

Linux makes more effective use of PC hardware than MS-DOS, Windows or NT, and is accordingly less tolerant of misconfigured hardware. There are a few things you can do before you start that will lessen your chances of being stopped by this kind of problem.

First, collect any manuals you have on your hardware -- motherboard, video card, monitor, modem, etc. -- and put them within easy reach.

Second, gather detailed information on your hardware configuration. One easy way to do this, if you're running MS-DOS 5.0, or up, is to print a report from the Microsoft diagnostic utility msd.exe (you can leave out the TSR, driver, memory-map, environment-strings and OS-version parts). Among other things, this will guarantee you full and correct information on your video card and mouse type, which will be helpful in configuring X later on.

Third, check your machine for configuration problems with supported hardware that could cause an un-recoverable lockup during Linux installation.

- It is possible for a DOS/Windows system using IDE hard drive(s) and CD ROM to be functional even with the master/slave jumpers on the drives incorrectly set. Linux won't fly this way. If in doubt, check your master-slave jumpers!
- Is any of your peripheral hardware designed with neither configuration jumpers nor non-volatile configuration memory? If so, it may require boot-time initialization via an MS-DOS utility to start up, and may not be easily accessible from Linux. CD-ROMs, sound cards, Ethernet cards and low-end tape drives can have this problem. If so, you may be able to work around this with an argument to the boot prompt; see the Linux Boot Prompt HOWTO, http://sunsite.unc.edu/LDP/HOWTO/BootPrompt-HOWTO.html for details).
- Some other operating systems will allow a bus mouse to share an IRQ with other devices. Linux doesn't support this; in fact, trying it may lock up your machine. If you are using a bus mouse, see the Linux Bus Mouse HOWTO, http://sunsite.unc.edu/LDP/HOWTO/Busmouse-HOWTO.html, for details.

If possible, get the telephone number of an experienced Linux user you can call in case of emergency. Nine times out of ten you won't need it, but it's comforting to have.

Budget time for installation. That will be about one hour on a bare system or one being converted to all-Linux operation. Or up to three hours for a dual-boot system (they have a much higher incidence of false starts and hangups).

6.2 Creating the boot and root floppies

(This step is only needed if you can't boot from a CD-ROM.)

Your Linux CD-ROM may come with installation aids that will take you through the process of building boot,

root, and rescue disks with interactive prompts. These may be an MS-DOS installation program (such as t Red Hat `redhat.exe` program) or a Unix script, or both.

If you have such a program and can use it, you should read the rest of this subsection for information only. Run the program to do actual installation -- its authors certainly knew more about the specific distribution than I, and you'll avoid many error-prone hand-entry steps.

More detailed information on making boot and root disks, see the Linux Bootdisk HOWTO at http://sunsite.unc.edu/LDP/HOWTO/Bootdisk-HOWTO.html.

Your first step will be to select a boot-disk image to fit your hardware. If you must do this by hand, you'll generally find that either (a) the bootdisk images on your CD-ROM are named in a way that willl help you pick a correct one, or (b) there's an index file nearby describing each image.

Next, you must create floppies from the bootdisk image you selected, and from the root and rescue disk images. This is where the MS-DOS program RAWRITE.EXE comes into play.

Next, you must have two or three *high-density* MS-DOS formatted floppies. (They must be of the same type; that is, if your boot floppy drive is a 3.5" drive, both floppies must be high-density 3.5" disks.) You will use RAWRITE.EXE to write the boot and rootdisk images to the floppies.

Invoke it with no arguments, like this:

```
C:\> RAWRITE
```

Answer the prompts for the name of the file to write and the floppy to write it to (such as `A:`). RAWRITE will copy the file, block-by-block, directly to the floppy. Also use RAWRITE for the root disk image (such as COLOR144). When you're done, you'll have two floppies: one containing the boot disk, the other containing the root disk. Note that these two floppies will no longer be readable by MS-DOS (they are ``Linux format'' floppies, in some sense).

You can use the dd(1) commands on a UNIX system to do the same job. (For this, you will need a UNIX workstation with a floppy drive, of course.) For example, on a Sun workstation with the floppy drive on device */dev/rfd0*, you can use the command:

```
$ dd if=bare of=/dev/rfd0 obs=18k
```

You must provide the appropriate output block size argument (the `obs' argument) on some workstations (e.g., Suns) or this will fail. If you have problems the man page for dd(1) may be be instructive.

Be sure that you're using brand-new, error-free floppies. The floppies must have no bad blocks on them.

Note that you do not need to be running Linux or MS-DOS in order to install Linux. However, running Linux or MS-DOS makes it easier to create the boot and root floppies from your CD-ROM. If you don't have an operating system on your machine, you can use someone else's Linux or MS-DOS just to create the floppies, and install from there.

6.3 Repartitioning your DOS/Windows drives

On most used systems, the hard drive is already dedicated to partitions for MS-DOS, OS/2, and so on. You'll need to resize these partitions in order to make space for Linux. If you're going to run a dual-boot system, it's strongly recommended that you read one or more of the following mini-HOWTOS, which describe different

TOC

dual-boot configurations.

- The DOS-Win95-OS2-Linux mini-HOWTO, http://sunsite.unc.edu/LDP/HOWTO/mini/Linux+DOS+Win95+OS2.html.
- The Linux+Win95 mini-HOWTO, http://sunsite.unc.edu/LDP/HOWTO/mini/Linux+Win95.html
- The Linux+NT-Loader mini-HOWTO, http://sunsite.unc.edu/LDP/HOWTO/mini/Linux+NT-Loader.html

Even if they are not directly applicable to your system, they will help you understand the issues involved.

NOTE: Some Linuxes will install to a directory on your MS-DOS partition. (This is different than installing FROM an MS-DOS partition.) Instead, you use the ``UMSDOS filesystem'', which allows you to treat a directory of your MS-DOS partition as a Linux filesystem. In this way, you don't have to repartition your drive.

I only suggest using this method if your drive already has four partitions (the maximum supported by DOS) and repartitioning would be more trouble than it's worth (it slows down your Linux due to filename translation overhead). Or, if you want to try out Linux before repartitioning, this is a good way to do so. But in most cases you should re-partition, as described here. If you do plan to use UMSDOS, you are on your own---it is not documented in detail here. From now on, we assume that you are NOT using UMSDOS, and that you will be repartitioning.

A *partition* is just a section of the hard drive set aside for a particular operating system to use. If you only have MS-DOS installed, your hard drive probably has just one partition, entirely for MS-DOS. To use Linux, however, you'll need to repartition the drive, so that you have one partition for MS-DOS, and one (or more) for Linux.

Partitions come in three flavors: *primary*, *extended*, and *logical*. Briefly, primary partitions are one of the four main partitions on your drive. However, if you wish to have more than four partitions per drive, you need to create an extended partition, which can contain many logical partitions. You don't store data directly on an extended partition---it is used only as a container for logical partitions. Data is stored only on either primary or logical partitions.

To put this another way, most people use only primary partitions. However, if you need more than four partitions on a drive, you create an extended partition. Logical partitions are then created on top of the extended partition, and there you have it---more than four partitions per drive.

Note that you can easily install Linux on the second drive on your system (known as `D:` to MS-DOS). You simply specify the appropriate device name when creating Linux partitions. This is described in detail below.

Back to repartitioning your drive: The problem with resizing partitions is that there is no way to do it (easily) without deleting the data on those partitions. Therefore, you will need to make a full backup of your system before repartitioning. In order to resize a partition, we simply delete the partition(s), and re-create them with smaller sizes.

NOTE: There is a non-destructive disk repartitioner available for MS-DOS, called FIPS. Look at http://sunsite.unc.edu/pub/Linux/system/install. With FIPS, a disk optimizer (such as Norton Speed Disk), and a little bit of luck, you should be able to resize MS-DOS partitions without destroying the data on them. It's still suggested that you make a full backup before attempting this.

If you're not using FIPS, however, the classic way to modify partitions is with the program FDISK. For

example, let's say that you have an 80 meg hard drive, dedicated to MS-DOS. You'd like to split it in half-megs for MS-DOS and 40 megs for Linux. In order to do this, you run FDISK under MS-DOS, delete the meg MS-DOS partition, and re-create a 40 meg MS-DOS partition in its place. You can then format the new partition and reinstall your MS-DOS software from backups. 40 megabytes of the drive is left empty. Later, you create Linux partitions on the unused portion of the drive.

In short, you should do the following to resize MS-DOS partitions with FDISK:

1. Make a full backup of your system.
2. Create an MS-DOS bootable floppy, using a command such as

   ```
   FORMAT /S A:
   ```

3. Copy the files `FDISK.EXE` and `FORMAT.COM` to this floppy, as well as any other utilities that you need. (For example, utilities to recover your system from backup.)
4. Boot the MS-DOS system floppy.
5. Run FDISK, possibly specifying the drive to modify (such as `C:` or `D:`).
6. Use the FDISK menu options to delete the partitions which you wish to resize. **This will destroy all data on the affected partitions.**
7. Use the FDISK menu options to re-create those partitions, with smaller sizes.
8. Exit FDISK and re-format the new partitions with the `FORMAT` command.
9. Restore the original files from backup.

Note that MS-DOS FDISK will give you an option to create a ``logical DOS drive''. A logical DOS drive is just a logical partition on your hard drive. You can install Linux on a logical partition, but you don't want to create that logical partition with MS-DOS fdisk. So, if you're currently using a logical DOS drive, and want to install Linux in its place, you should delete the logical drive with MS-DOS FDISK, and (later) create a logical partition for Linux in its place.

The mechanism used to repartition for OS/2 and other operating systems is similar. See the documentation for those operating systems for details.

6.4 Creating partitions for Linux

After repartitioning your drive, you need to create partitions for Linux. Before describing how to do that, we'll talk about partitions and filesystems under Linux.

Partition basics

Linux requires at least one partition, for the *root filesystem*, which will hold the Linux kernel itself.

You can think of a *filesystem* as a partition formatted for Linux. Filesystems are used to hold files. Every system must have a root filesystem, at least. However, many users prefer to use multiple filesystems---one for each major part of the directory tree. For example, you may wish to create a separate filesystem to hold all files under the */usr* directory. (Note that on UNIX systems, forward slashes are used to delimit directories, not backslashes as with MS-DOS.) In this case you have both a root filesystem, and a */usr* filesystem.

Each filesystem requires its own partition. Therefore, if you're using both root and */usr* filesystems, you'll need to create two Linux partitions.

TOC

In addition, most users create a *swap partition*, which is used for virtual RAM. If you have, say, 4 megabytes of memory on your machine, and a 10-megabyte swap partition, as far as Linux is concerned you have 14 megabytes of virtual memory.

When using swap space, Linux moves unused pages of memory out to disk, allowing you to run more applications at once on your system. However, because swapping is often slow, it's no replacement for real physical RAM. But applications that require a great deal of memory (such as the X Window System) often rely on swap space if you don't have enough physical RAM.

Nearly all Linux users employ a swap partition. If you have 4 megabytes of RAM or less, a swap partition is required to install the software. It is strongly recommended that you have a swap partition anyway, unless you have a great amount of physical RAM.

The size of your swap partition depends on how much virtual memory you need. It's often suggested that you have at least 16 megabytes of virtual memory total. Therefore, if you have 8 megs of physical RAM, you might want to create an 8-megabyte swap partition. Note that swap partitions can be no larger than 128 megabytes in size. Therefore, if you need more than 128 megs of swap, you must create multiple swap partitions. You may have up to 16 swap partitions in all.

You can find more on the theory of swap space layout and disk partitioning in the Linux Partition mini-HOWTO (http://sunsite.unc.edu/LDP/HOWTO/mini/Partition.html).

Note: it is possible, though a bit tricky, to share swap partitions between Linux and Windows 95 in a dual-boot system. For details, see the Linux Swap Space Mini-HOWTO, http://sunsite.unc.edu/LDP/HOWTO/unmaintained/Swap-Space.

Gotcha #1: If you have an EIDE drive with a partition that goes above 504MB, your BIOS may not allow you to boot to a Linux installed there. So keep your root partition below 504MB. This shouldn't be a problem for SCSI drive controllers, which normally have their own drive BIOS firmware. For technical details, see the Large Disk Mini-HOWTO, http://sunsite.unc.edu/LDP/HOWTO/mini/Large-Disk.html.

Gotcha #2: Are you mixing IDE and SCSI drives? Then watch out. Your BIOS may not allow you to boot directly to a SCSI drive.

Sizing partitions

Besides your root and swap partitions, you'll want to set up one or more partitions to hold your software and home directories.

While, in theory, you could run everything off a single huge root partition, almost nobody does this. Having multiple partitions has several advantages:

- It often cuts down the time required for boot-time file-system checks.
- Files can't grow across partition boundaries. Therefore you can use partition boundaries as firebreaks against programs (like Usenet news) that want to eat huge amounts of disk, to prevent them from crowding out file space needed by your kernel and the rest of your applications.
- If you ever develop a bad spot on your disk, formatting and restoring a single partition is less painful than having to redo everything from scratch.

On today's large disks, a good basic setup is to have a small root partition (less than 80 meg), a medium-sized /usr partition (up to 300 meg or so) to hold system software, and a /home partition occupying the rest of your

available space for home directories.

You can get more elaborate. If you know you're going to run Usenet news, for example, you may want to give it a partition of its own to control its maximum possible disk usage. Or create a /var partition for mail, news, and temporary files all together. But in today's regime of very cheap, very large hard disks these complications seem less and less necessary for your first Linux installation. For your first time, especially, keep it simple.

6.5 Booting the installation disk

The first step is to boot the bootdisk you generated. Normally you'll be able to boot hands-off; the boot kernel prompt will fill itself in after 10 seconds. This is how you'll normally boot from an IDE disk.

By giving arguments after the kernel name, you can specify various hardware parameters, such as your SCSI controller IRQ and address, or drive geometry, before booting the Linux kernel. This may be necessary if Linux does not detect your SCSI controller or hard drive geometry, for example.

In particular, many BIOS-less SCSI controllers require you to specify the port address and IRQ at boot time. Likewise, IBM PS/1, ThinkPad, and ValuePoint machines do not store drive geometry in the CMOS, and you must specify it at boot time. (Later on, you'll be able to configure your production system to supply such parameters itself.)

Watch the messages as the system boots. They will list and describe the hardware your installation Linux detects. In particular, f you have a SCSI controller, you should see a listing of the SCSI hosts detected. If you see the message

```
SCSI: 0 hosts
```

Then your SCSI controller was not detected, and you will have to figure out how to tell the kernel where it is.

Also, the system will display information on the drive partitions and devices detected. If any of this information is incorrect or missing, you will have to force hardware detection.

On the other hand, if all goes well and you hardware seems to be detected, you can skip to the following section, ``Loading the root disk."

To force hardware detection, you must enter the appropriate parameters at the boot prompt, using the following syntax:

```
linux <parameters...>
```

There are a number of such parameters available; we list some of the most common below. Modern Linux boot disks will often give you the option to look at help screen describing kernel parameters before you boot.

- *hd=cylinders,heads,sectors* Specify the drive geometry. Required for systems such as the IBM PS/1, ValuePoint, and ThinkPad. For example, if your drive has 683 cylinders, 16 heads, and 32 sectors per track, enter

  ```
  linux hd=683,16,32
  ```

- *tmc8xx=memaddr,irq* Specify address and IRQ for BIOS-less Future Domain TMC-8xx SCSI controller. For example,

```
linux tmc8xx=0xca000,5
```

Note that the *0x* prefix must be used for all values given in hex. This is true for all of the following options.

- *st0x=memaddr,irq* Specify address and IRQ for BIOS-less Seagate ST02 controller.
- *t128=memaddr,irq* Specify address and IRQ for BIOS-less Trantor T128B controller.
- *ncr5380=port,irq,dma* Specify port, IRQ, and DMA channel for generic NCR5380 controller.
- *aha152x=port,irq,scsi_id,1* Specify port, IRQ, and SCSI ID for BIOS-less AIC-6260 controllers. This includes Adaptec 1510, 152x, and Soundblaster-SCSI controllers.

If you have questions about these boot-time options, please read the Linux *SCSI HOWTO*, which should be available on any Linux FTP archive site (or from wherever you obtained this document). The *SCSI HOWTO* explains Linux SCSI compatibility in much more detail.

6.6 Using the rootdisk

After booting the bootdisk, you will be prompted to enter the root disk or disks. At this point you should remove the bootdisk from the drive and insert the rootdisk. Then press *enter* to go on. You may have to load a second root disk.

What's actually happening here is this: the boot disk provides a miniature operating system which (because the hard drive isn't prepared) uses a portion of your RAM as a virtual disk (called, logically enough, a `ramdisk').

The root disks loads onto the ramdisk a small set of files and installation tools which you'll use to prepare your hard drive and install a production Linux on it from your CD-ROM.

Choosing EGA or X installation

Older Linuxes (including Slackware) gave you a shell at this point and required you to enter installation commands by hand in a a prescribed sequence. This is still possible, but newer ones start by running a screen-oriented installation program which tries to interactively walk you through these steps, giving lots of help.

You will probably get the option to try to configure X right away so the installation program can go graphical. If you choose this route, the installation program will quiz you about your mouse and monitor type before getting to the installation proper. Once you get your production Linux installed, these settings will be saved for you. You will be able to tune your monitor's performance later, so at this stage it makes sense to settle for a basic 640x480 SVGA mode.

X isn't necessary for installation, but (assuming you can get past the mouse and monitor configuration) many people find the graphical interface easier to use. And you're going to want to bring up X anyway, so trying it early makes some sense.

Just follow the prompts in the program. It will take you through the steps necessary to prepare your disk, create initial user accounts, and install software packages off the CD-ROM.

In the following subsections we'll describe some of the tricky areas in the installation sequence as if you were doing them by hand. This should help you understand what the installation program is doing, and why.

TOC

Using `fdisk` and `cfdisk`

Your first installation step once the root-disk Linux is booted will be to create or edit the partition tables on your disks. Even if you used FDISK to set up partitions earlier, you'll need to go back to the partition table now and insert some Linux-specific information now.

To create or edit Linux partitions, we'll use the Linux version of the `fdisk` program, or its screen-oriented sibling `cfdisk`.

Generally the installation program will look for a preexisting partition table and offer to run `fdisk` or `cfdisk` on it for you. Of the two, `cfdisk` is definitely easier to use, but current versions of it are also less tolerant of a nonexistent or garbled partition table.

Therefore you may find (especially if you're installing on virgin hardware) that you need to start with `fdisk` to get to a state that `cfdisk` can deal with. Try running `cfdisk`; if it complains, run `fdisk`. (A good way to proceed if you're building an all-Linux system and `cfdisk` complains is to use `fdisk` to delete all the existing partions and then fire up `cfdisk` to edit the empty table.)

A few notes apply to both `fdisk` and `cfdisk`. Both take an argument which is the name of the drive that you wish to create Linux partitions on. Hard drive device names are:

- */dev/hda* First IDE drive
- */dev/hdb* Second IDE drive
- */dev/sda* First SCSI drive
- */dev/sdb* Second SCSI drive

For example, to create Linux partitions on the first SCSI drive in your system, you will use (or your installation program might generate from a menu choice) the command:

```
cfdisk /dev/sda
```

If you use `fdisk` or `cfdisk` without an argument, it will assume */dev/hda*.

To create Linux partitions on the second drive on your system, simply specify either */dev/hdb* (for IDE drives) or */dev/sdb* (for SCSI drives) when running `fdisk`.

Your Linux partitions don't all have to be on the same drive. You might want to create your root filesystem partition on */dev/hda* and your swap partition on */dev/hdb*, for example. In order to do so just run `fdisk` or `cfdisk` once for each drive.

In Linux, partitions are given a name based on the drive which they belong to. For example, the first partition on the drive */dev/hda* is */dev/hda1*, the second is */dev/hda2*, and so on. If you have any logical partitions, they are numbered starting with */dev/hda5*, */dev/hda6* and so on up.

NOTE: You should not create or delete partitions for operating systems other than Linux with Linux `fdisk` or `cfdisk`. That is, don't create or delete MS-DOS partitions with this version of `fdisk`; use MS-DOS's version of `FDISK` instead. If you try to create MS-DOS partitions with Linux `fdisk`, chances are MS-DOS will not recognize the partition and not boot correctly.

Here's an example of using `fdisk`. Here, we have a single MS-DOS partition using 61693 blocks on the drive, and the rest of the drive is free for Linux. (Under Linux, one block is 1024 bytes. Therefore, 61693 blocks is about 61 megabytes.) We will create just two partitions in this tutorial example, swap and root. You should probably extend this to four Linux partitions in line with the recommendations above: one for swap,

one for the root filesystem, one for system software, and a home directory area.

First, we use the ``p'' command to display the current partition table. As you can see, */dev/hda1* (the first partition on */dev/hda*) is a DOS partition of 61693 blocks.

```
Command (m for help):   p
Disk /dev/hda: 16 heads, 38 sectors, 683 cylinders
Units = cylinders of 608 * 512 bytes

     Device Boot  Begin   Start     End  Blocks   Id  System
  /dev/hda1   *       1       1     203   61693    6  DOS 16-bit >=32M

Command (m for help):
```

Next, we use the ``n'' command to create a new partition. The Linux root partition will be 80 megs in size.

```
Command (m for help):  n
Command action
    e   extended
    p   primary partition (1-4)
p
```

Here we're being asked if we want to create an extended or primary partition. In most cases you want to use primary partitions, unless you need more than four partitions on a drive. See the section ``Repartitioning'', above, for more information.

```
Partition number (1-4): 2
First cylinder (204-683):  204
Last cylinder or +size or +sizeM or +sizeK (204-683): +80M
```

The first cylinder should be the cylinder AFTER where the last partition left off. In this case, */dev/hda1* ended on cylinder 203, so we start the new partition at cylinder 204.

As you can see, if we use the notation ``+80M'', it specifies a partition of 80 megs in size. Likewise, the notation ``+80K'' would specify an 80 kilobyte partition, and ``+80'' would specify just an 80 byte partition.

```
Warning: Linux cannot currently use 33090 sectors of this partition
```

If you see this warning, you can ignore it. It is left over from an old restriction that Linux filesystems could only be 64 megs in size. However, with newer filesystem types, that is no longer the case... partitions can now be up to 4 terabytes in size.

Next, we create our 10 megabyte swap partition, */dev/hda3*.

```
Command (m for help): n
Command action
    e   extended
    p   primary partition (1-4)
p
```

```
Partition number (1-4): 3
First cylinder (474-683):  474
Last cylinder or +size or +sizeM or +sizeK (474-683):  +10M
```

Again, we display the contents of the partition table. Be sure to write down the information here, especially the size of each partition in blocks. You need this information later.

```
Command (m for help): p
Disk /dev/hda: 16 heads, 38 sectors, 683 cylinders
Units = cylinders of 608 * 512 bytes

   Device Boot  Begin   Start     End  Blocks   Id  System
/dev/hda1   *       1       1     203   61693    6  DOS 16-bit >=32M
/dev/hda2         204     204     473   82080   83  Linux native
/dev/hda3         474     474     507   10336   83  Linux native
```

Note that the Linux swap partition (here, */dev/hda3*) has type ``Linux native''. We need to change the type of the swap partition to ``Linux swap'' so that the installation program will recognize it as such. In order to do this, use the `fdisk` ```t`'' command:

```
Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 82
```

If you use ```L`'' to list the type codes, you'll find that 82 is the type corresponding to Linux swap.

To quit `fdisk` and save the changes to the partition table, use the ```w`'' command. To quit `fdisk` WITHOUT saving changes, use the ```q`'' command.

After quitting `fdisk`, the system may tell you to reboot to make sure that the changes took effect. In general there is no reason to reboot after using `fdisk`---modern versions of `fdisk` and `cfdisk` are smart enough to update the partitions without rebooting.

Post-partition steps

After you've edited the partition tables, your installation program should look at them and offer to enable your swap partition for you. Tell it yes.

(This is made a question, rather than done automatically, on the off chance that you're running a dual-boot system and one of your non-Linux partitions might happen to look like a swap volume.)

Next the program will ask you to associate Linux filesystem names (such as /, /usr, /var, /tmp, /home, /home2, etc.) with each of the non-swap partitions you're going to use.

There is only one hard and fast rule for this. There must be a root filesystem, named /, and it must be bootable. You can name your other Linux partitions anything you like. But there are some conventions about how to name them which will probably simplify your life later on.

Earlier on I recommended a basic three-partition setup including a small root, a medium-sized system-software partition, and a large home-directory partition. Traditionally, these would be called /, /usr, and /home. The

TOC

counterintuitive `/usr' name is a historical carryover from the days when (much smaller) Unix systems ca system software and user home directories on a single non-root partition. Some software depends on it.

If you have more than one home-directory area, it's conventional to name them /home, /home2, /home3, etc. This may come up if you have two physical disks. On my personal system, for example, the layout currently looks like this:

```
Filesystem          1024-blocks  Used Available Capacity Mounted on
/dev/sda1               30719   22337     6796     77%   /
/dev/sda3              595663  327608   237284     58%   /usr
/dev/sda4             1371370    1174  1299336      0%   /home
/dev/sdb1             1000949  643108   306130     68%   /home2
```

The second disk (sdb1) isn't really all /home2; the swap partitions on sda and sdb aren't shown in this display. But you can see that /home is the large free area on sda and /home2 is the user area of sdb.

If you want to create an partition for scratch, spool, temporary, mail, and news files, call it /var. Otherwise you'll probably want to create a /usr/var and create a symbolic link named /var that points back to it (the installation program may offer to do this for you).

6.7 Installing software packages

Once you've gotten past preparing your partitions, the remainder of the installation should be almost automatic. Your installation program (whether EGA or X-based) will guide you through a series of menus which allow you to specify the CD-ROM to install from, the partitions to use, and so forth.

Here we're not going to document many of the specifics of this stage of installation. It's one of the parts that varies most between Linux distributions (vendors traditionally compete to add value here), but also the simplest part. And the installation programs are pretty much self-explanatory, with good on-screen help.

6.8 After package installations

After installation is complete, and if all goes well, the installation program will walk you through a few options for configuring your system before its first boot from hard drive.

LILO, the LInux LOader

LILO (which stands for LInux LOader) is a program that will allow you to boot Linux (as well as other operating systems, such as MS-DOS) from your hard drive.

You may be given the option of installing LILO on your hard drive. Unless you're running OS/2, answer `yes'. OS/2 has special requirements; see Custom LILO Configuration below.

Installing LILO as your primary loader makes a separate boot diskette unnecessary; instead, you can tell LILO at each boot time which OS to boot.

TOC

Making a production boot disk (optional)

You may also be given the chance to create a ``standard boot disk'', which you can use to boot your newly-installed Linux system. (This is an older and slightly less convenient method which assumes that you will normally boot DOS, but use the boot disk to start Linux.)

For this you will need a blank, high-density MS-DOS formatted diskette of the type that you boot with on your system. Simply insert the disk when prompted and a boot diskette will be created. (This is not the same as an installation bootdisk, and you can't substitute one for the other!)

Miscellaneous system configuration

The post-installation procedure may also take you through several menu items allowing you to configure your system. This includes specifying your modem and mouse device, as well as your time zone. Follow the menu options.

It may also prompt you to create user accounts or put a password on the root (administration) account. This is not complicated and you can usually just walk through the screen instructions.

---

7. Booting Your New System

If everything went as planned, you should now be able to boot Linux from the hard drive using LILO. Alternatively, you should be able to boot your Linux boot floppy (not the original bootdisk floppy, but the floppy created after installing the software). After booting, login as *root*. Congratulations! You have your very own Linux system.

If you are booting using LILO, try holding down *shift* or *control* during boot. This will present you with a boot prompt; press *tab* to see a list of options. In this way you can boot Linux, MS-DOS, or whatever directly from LILO.

8. After Your First Boot

You should now be looking at the login prompt of a new Linux, just booted from your hard drive. Congratulations!

8.1 Beginning System Administratration

Depending on how the installation phase went, you may need to create accounts, change your hostname, or (re)configure X at this stage. There are many more things you could set up and configure, including backup devices, SLIP/PPP links to an Internet Service Provider, etc.

A good book on UNIX systems administration should help. (I suggest *Essential Systems Administration* from O'Reilly and Associates.) You will pick these things up as time goes by. You should read various other Linux HOWTOs, such as the *NET-3-HOWTO* and *Printing-HOWTO*, for information on other configuration tasks.

8.2 Custom LILO Configuration

LILO is a boot loader, which can be used to select either Linux, MS-DOS, or some other operating system at boot time. Chances are your distribution automatically configured LILO for you during the installation phase (unless you're using OS/2, this is what you should have done). If so, you can skip the rest of this section.

If you installed LILO as the *primary* boot loader, it will handle the first-stage booting process for all operating systems on your drive. This works well if MS-DOS is the only other operating system that you have installed. However, you might be running OS/2, which has its own Boot Manager. In this case, you want OS/2's Boot Manager to be the primary boot loader, and use LILO just to boot Linux (as the *secondary* boot loader).

An important gotcha for people using EIDE systems: due to a BIOS limitation, your boot sectors for any OS have to live on one of the first two physical disks. Otherwise LILO will hang after writing "LI", no matter where you run it from.

If you have to configure LILO manually, this will involve editing the file */etc/lilo.conf*. Below we present an example of a LILO configuration file, where the Linux root partition is on */dev/hda2*, and MS-DOS is installed on */dev/hdb1* (on the second hard drive).

```
# Tell LILO to install itself as the primary boot loader on /dev/hda.
boot = /dev/hda
# The boot image to install; you probably shouldn't change this
install = /boot/boot.b

# The stanza for booting Linux.
image = /vmlinuz       # The kernel is in /vmlinuz
  label = linux        # Give it the name "linux"
```

```
    root = /dev/hda2     # Use /dev/hda2 as the root filesystem
    vga = ask            # Prompt for VGA mode
    append = "aha152x=0x340,11,7,1"  # Add this to the boot options,
     # for detecting the SCSI controller

  # The stanza for booting MS-DOS
  other = /dev/hdb1      # This is the MS-DOS partition
    label = msdos        # Give it the name "msdos"
    table = /dev/hdb     # The partition table for the second drive
```

Once you have edited the */etc/lilo.conf* file, run */sbin/lilo* as *root*. This will install LILO on your drive. Note that you must rerun */sbin/lilo* anytime that you recompile your kernel in order to point the boot loader at it properly (something that you don't need to worry about just now, but keep it in mind).

Note how we use the *append* option in */etc/lilo.conf* to specify boot parameters as we did when booting the bootdisk.

You can now reboot your system from the hard drive. By default LILO will boot the operating system listed first in the configuration file, which in this case is Linux. In order to bring up a boot menu, in order to select another operating system, hold down *shift* or *ctrl* while the system boots; you should see a prompt such as

```
Boot:
```

Here, enter either the name of the operating system to boot (given by the *label* line in the configuration file; in this case, either *linux* or *msdos*), or press *tab* to get a list.

Now let's say that you want to use LILO as the secondary boot loader; if you want to boot Linux from OS/2's Boot Manager, for example. In order to boot a Linux partition from OS/2 Boot Manager, unfortunately, you must create the partition using OS/2's *FDISK* (not Linux's), and format the partition as FAT or HPFS, so that OS/2 knows about it. (That's IBM for you.)

In order to have LILO boot Linux from OS/2 Boot Manager, you only want to install LILO on your Linux root filesystem (in the above example, */dev/hda2*). In this case, your LILO config file should look something like:

```
boot = /dev/hda2
install = /boot/boot.b
compact

image = /vmlinuz
  label = linux
  root = /dev/hda2
  vga = ask
```

Note the change in the *boot* line. After running */sbin/lilo* you should be able to add the Linux partition to Boot Manager. This mechanism should work for boot loaders used by other operating systems as well.

9. Administrivia

9.1 Terms of Use

This document is copyright 1998 by Eric S. Raymond. You may use, disseminate, and reproduce it freely, provided you:

- Do not omit or alter this copyright notice.
- Do not omit or alter or omit the version number and date.
- Do not omit or alter the document's pointer to the current WWW version.
- Clearly mark any condensed, altered or versions as such.

These restrictions are intended to protect potential readers from stale or mangled versions. If you think you have a good case for an exception, ask me.

9.2 Acknowledgements

My grateful acknowledgement to Matt D. Welsh, who originated this HOWTO. I removed much of the Slackware-specific content and refocused the remainder of the document on CD-ROM installation, but a substantial part of the content is still his.

The 4.1 version was substantially improved by some suggestions from David Shao <dshao@best.com>.

The Linux Intranet Server HOWTO

Pramod Karnad, karnad@indiamail.com

v2.11, 7 August 1997

This document describes how to setup an Intranet using Linux as the server which binds Unix, Netware, NT and Windows together. Hence by just establishing the connection to the Linux box you are provided transparent access to all the various platforms. Detailed explanations are provided for setting up HTTP using the NCSA server and connect to it using TCP/IP clients from Novell, Microsoft under Windows3.1, WFWG,Win95 and WinNT and MacTCP on the Apple PowerMac.

1. Introduction

- 1.1 What is required
- 1.2 New versions of this document
- 1.3 Feedback

2. Install the HTTP server

- 2.1 Preparation before downloading
- 2.2 Compiling HTTPd

3. Testing HTTPd

4. Connecting to the Linux Server

- 4.1 Setup the Linux server
- 4.2 Setup the Netware server
- 4.3 Setup the Netware Client
- 4.4 Setup Microsoft Client
- 4.5 Setup TCP/IP on Macintosh

TOC

5. Setting up the Intranet

- 5.1 NCPFS
- 5.2 SMBFS
- 5.3 NFS

6. Accessing the Web

- 6.1 Accessing the mounted filesystems
- 6.2 Connecting to the Internet
- 6.3 Other uses

7. More things to do

8. Credits and Legalities

- 8.1 Thanks
- 8.2 Copyright information

---

1. Introduction

In simple terms, the **Intranet** is the descriptive term being used for the implementation of Internet technologies within a corporate organisation, rather than for external connection to the global Internet. This implementation is performed in such a way as to transparently deliver the immense informational resources of an organisation to each individuals desktop with minimal cost, time and effort. This document attempts to explain in simple terms how to setup an Intranet using tools which are readily available and are generally costing little or are free.

This document assumes that you already know how to install TCP/IP on your Linux server and connect it physically to your LAN using an Ethernet network card. This also assumes you have some basic knowledge of Netware, WinNT and Mac systems. The configuration of the Netware server has been shown using version 3.1x as the basis. You can also use INETCFG to achieve the same result. On the client side the discussion is with respect to Windows 3.1x, Windows for Workgroups and Win95, WinNT and the Apple PowerMac.

I am using the private network addresses (RFC-1918) of 172.16.0.0 and 172.17.0.0 only as examples. You may choose suitable addresses depending on your configuration.

```
    Linux        Netware       WFWG/WinNT
    Server       Server        Server
   172.16.0.1   172.16.0.2    172.16.0.3
      |            |             |    172.16.0.0
------+-----+-----+--------+--+--------------
            |               | 172.16.0.254
         W/S 1           Router
       172.16.0.5          | 172.17.0.254
                           |
                 ----------+-------+--------
            172.17.0.0             |
                                 W/S 2
                               172.17.0.5
```

1.1 What is required

You will need the following software before attempting the installation.

- the HTTP server software which can be downloaded from OneStep NCSA HTTPd Downloader at http://hoohoo.ncsa.uiuc.edu/docs/setup/OneStep.html page.
- The Novell Netware Client available from HTTP://support.novell.com/ (The TCP/IP files are included with the client).

TOC

- The Microsoft TCP/IP client available from HTTP://www.microsoft.com/
- The Apple MacTCP client available from HTTP://www.apple.com/
- WWW Browsers like Netscape at HTTP://home.netscape.com/ or MS Internet Explorer at HTTP://www.microsoft.com/ or NCSA Mosaic from http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html

1.2 New versions of this document

New versions of the Linux Intranet Server HOWTO will be periodically posted to comp.os.linux.announce and comp.os.linux.help. They will also be uploaded to various Linux FTP sites, including sunsite.unc.edu.

The Latest version of this document is available in HTML format at http://www.inet.co.th/cyberclub/karnadp/http.html

1.3 Feedback

If you have questions or comments about this document, please feel free to mail Pramod Karnad, at karnad@indiamail.com. Suggestions, criticism and mail are always welcome. If you find a mistake with this document, please let me know so I can correct it in the next version. Thanx.

---

2. Install the HTTP server

When you download the server you have two options: To get the source and compile it yourself, or get the precompiled binaries. The precompiled binaries for Linux (ELF) version are available at NCSA but not the older versions.

2.1 Preparation before downloading

The server at NCSA will guide you through the steps for configuration options and prepare the various files for you. But before you attempt to download HTTPd be prepared with answers to the following questions

The Operating System

First, you must choose whether to download the source or a pre-compiled version of the software. If your particular system doesn't appear in the menu, then you will have to get the default source, and compile it yourself.

To check the version of your Linux go to the command prompt on your Linux machine and type

```
linux:~$  uname -a
```

which will respond with a line which looks similar to this

```
linux:~$  uname -a
Linux linux 2.0.29 #4 Tue Sep 13 04:05:51 CDT 1994 i586
linux:~$
```

The version of Linux is 2.0.29.

The remaining parameters can be specified before downloading or configured later by modifying the file `srm.conf` in the `/usr/local/etc/httpd/conf` directory. The names of the actual directives that appear in the file `httpd.conf` are shown in brackets. The only exception is the directive DocumentRoot which appears in the file `srm.conf`

Process type (ServerType)

This specifies how your machine will run your HTTPd server. The preferred method is "standalone". This makes the HTTP daemon to be running constantly. If you choose to load HTTPd under "inetd", the server binary will be reloaded into memory for every request, which may slow your server down.

TOC

Binding Port (Port)

This specifies which port of your machine that the HTTPd daemon will bind to and listen for HTTP requests. If you can login as "root", use the default setting of 80. Otherwise choose a setting between 1025 and 65535.

Server user identity (User)

This is the user id the server will change to when answering requests and acting on files.This question needs to be answered only if you are running the server as "standalone". If you are someone without root permissions, just use your own login name. If you are system administrator, you might want to create a special user so you can control file permissions.

Server group identity (Group)

This is the group id the server will change to when answering requests and acting on files. This is similar to Server User identity and is applicable only if you are running the server as standalone.

If you do not have root permissions, just use the name of your primary group. You can find out your group by typing **groups** at the Linux command prompt.

Server administrator email address (ServerAdmin)

This is the email address that the user should send an email message to when reporting a problem with the server. You can put your personal e-mail address.

Location of server directory (ServerRoot)

This is where the server resides on your system. If you have root permissions leave it in its recommended location `/usr/local/etc/httpd`. If you cannot login as root, choose a subdirectory in your home path. You can find out the path of your home directory with the **pwd** command.

Location of HTML files (DocumentRoot)

This is where the HTML files to be served are located. The default location is `/usr/local/etc/httpd/htdocs`. You could however set it to be the home directory of the special user you chose in Server user identity, or a subdirectory in your home directory if you can't login as root.

When in doubt, use the default settings. Now that you have answers to the above questions you can Download NCSA HTTPd at http://hoohoo.ncsa.uiuc.edu/docs/setup/OneStep.html. You should read the HTTPd Documentation at http://hoohoo.ncsa.uiuc.edu/docs/ before you attempt installation. If you are planning to compile the code then you need to modify the makefiles in each of the th ree directories `support, src, cgi-src`. If your version of Linux is already supported then you just have to type **make linux** at the top level directory (i.e. `/usr/local/etc/httpd`)

2.2 Compiling HTTPd

TOC

Compiling is simple, just type `make linux` at the prompt in the server root directory. **Note:** Users of pre-ELF Linux have to uncomment `#define NO_PASS` in file `portability.h` and set `DBM_LIBS= -ldbm` in the `Makefile` before compiling HTTPd.

---

3. Testing HTTPd

After you have installed HTTPd, login as root and start it by typing **httpd &** . (assuming you have installed as standalone) You should now be able to see it in the list generated by **ps**. The simplest way to test HTTPd is by Telnet. At the Linux command prompt type

```
linux:~$  telnet 172.16.0.1 80
```

where 80 is the default port for HTTP. If you have configured "Port" as something different then type that number instead. You should get a response which looks like this

```
Trying 172.16.0.1...
Connected to linux.mydomain.
Escape character is '^]'.
```

Now if you type in any character and press Enter you should get a response similar to the one shown below.

```
HTTP/1.0 400 Bad Request
Date: Wed, 10 Jan 1996 10:24:37 GMT
Server: NCSA/1.5
Content-type: text/html

<HEAD><TITLE>400 Bad Request < /TITLE> < /HEAD>
<BODY><H1>400 Bad Request < /H1>
Your client sent a query that this server could
not understand.<P>
Reason: Invalid or unsupported method.<P>
< /BODY>
```

Now we are ready to connect to this server using another PC and a WWW Browser.

4. Connecting to the Linux Server

Please refer to the diagram shown in the chapter Introduction for the addressing scheme used. Workstation 1 (W/S1) is on network 172.16.0.0 and can access the Linux server directly whereas Workstation 2 (W/S2) is on network 172.17.0.0 and needs to use the gateway (router) 172.17.0.254 to access the Linux box. This gateway information needs to be provided while configuring the clients only on W/S2. Netware refers to the gateway as 'ip_router'.

I am using W/S2 to illustrate the client setup. To setup W/S1 just change the address 172.17.0.5 to 172.16.0.5 and ignore all references to the gateway/router.

If you do not have a router you can skip the next section and proceed to

- Setup Netware Server if you use a Netware server.
- Setup MS Windows Client if you use the Microsoft Client.

4.1 Setup the Linux server

You may skip this section if you do not have a router.

You have to configure the Linux server to recognise the router thus allowing Workstation 2 to connect to the Web server. In order to setup the Linux server you should login as root. At the server prompt type

```
route add gw default 172.16.0.254
```

To use this gateway everytime you boot the Linux server edit the file `/etc/rc.d/rc.inet1` and change the line containing the gateway definition to `GATEWAY = "172.16.0.254"`. Make sure the line for adding the gateway is not commented out.

ALT: You can add routes to the networks on the other side of the router. This would be done as

```
route add -net 172.17.0.0 gw 172.16.0.254
```

To add this route everytime you boot Linux add the command to your `/etc/rc.d/rc.local` file.

4.2 Setup the Netware server

In order to setup the Netware server you should have Supervisor permissions or atleast Console operator permissions. If these cannot be got, try asking your Network Administrator to help you with the setup. At the Server enable the Ethernet_II frame type on the LAN by typing these commands or include them in the AUTOEXEC.ncf file.

```
load NE2000 frame=Ethernet_II name=IPNET
load TCPIP
bind IP to IPNET addr=172.16.0.2 mask=FF.FF.FF.0
```

You might have to specify the slot or board number while loading the NE2000 driver depending on your machine configuration. (eg: load NE2000 slot=3 frame=.....)

4.3 Setup the Netware Client

TOC

On the PC you have the choice of Win3.1,WFWG or Win95. The installation procedure differs between Win95 and the older windows if you are using the 32bit client from Microsoft or Novell. If you are going to use the 16bit client, the procedure is the same and you can refer to the Windows 3.x installation instructions. For installing the 32bit client for Win95 skip to Windows 95 installation.

Windows 3.x

If you are using Win3.1 or WFWG you can install the Netware Client (VLMs) and some additional files which are provided with the TCP/IP diskette, namely

TCPIP.exe, VTCPIP.386, WINSOCK.dll and WLIBSOCK.dll

Note that the WINSOCK.dll file is different from the ones provided with Win95 and Trumpet. Install the Netware Client with the support for windows. Copy VTCPIP.386, WINSOCK.dll and WLIBSOCK.dll to the SYSTEM directory and TCPIP.exe to the NWCLIENT directory. Now modify the STARTNET.bat in the NWCLIENT directory to

```
lsl
ne2000     ---> your network card driver
c:\windows\odihlp.exe     ---->if you are using WFWG
ipxodi
tcpip        ---> add this line
nwip         ---> if you use Netware/IP
vlm
```

Create a subdirectory (say) `\NET\TCP` and copy the files HOSTS, NETWORKS, PROTOCOLS and SERVICES from `/etc` on your Linux server or the directory `SYS:ETC` on your Netware server. Edit the copied HOSTS file to add the line for your new Linux server. This will enable you to refer to the Linux server as `http://linux.mydomain/` instead of `http://172.16.0.1/`in your WWW browser

```
127.0.0.1      localhost
172.16.0.1      linux.mydomain
```

Edit the NET.cfg file in NWCLIENT directory

```
Link Driver NE2000
    port 300
    int 3
    MEM D0000
    FRAME Ethernet_802.2

; ---- add these lines ----

   FRAME Ethernet_II

Protocol TCPIP
   PATH TCP_CFG C:\NET\TCP
   ip_address  172.17.0.5
   ip_netmask  255.255.255.0
   ip_router   172.17.0.254     ---> add the address of your gateway only
                                ---> if you have to use this
```

```
                                        ---> gateway to reach your HTTP server

      Link Support
         MemPool 6192       ---> the minimum is 1024. Try with different values
         Buffers 10 1580    ---> this again can be fine tuned

   ;-----------------------------------
   ; You may need to add lines like these if you are using Netware/IP
   ;
      NWIP
         NWIP_DOMAIN_NAME  mydomain
         NSQ_BROADCAST   ON
         NWIP1_1 COMPATIBILITY   OFF
         AUTORETRIES     1
         AUTORETRY SECS  10
```

Edit the SYSTEM.ini file in the WINDOWS directory and add this entry for VTCPIP.386

```
      [386Enh]
      .....
      network=*vnetbios, vipx.386, vnetware.386, VTCPIP.386
      .....
```

Reboot your PC, run STARTNET.bat and you can now use your favorite WWW browser to access your Web pages. You need not login to Netware and you don't have to run TCPMAN (if you use Trumpet Winsock).

Windows 95

This section explains how to install the 32bit client on Win95. Firstly you must install the following

```
      Client for Netware Networks (from Microsoft or Novell)
      Microsoft TCP/IP Protocol
      Network Adapter
```

To install these items, click on My Computer, Control Panel, Networks. Click Add. You will now be in a window that displays Client, Adapter, Protocol and Service. To install the Client for Netware Networks:

```
     1. DoubleClick on Client
     2. Click on Microsoft or Novell
     3. DoubleClick on Client for Netware Networks
```

To install the TCP/IP Protocol:

```
     1. DoubleClick on Protocol
     2. Click on Microsoft
     3. DoubleClick on TCP/IP
```

Windows 95 by default installs several other protocols automatically. Remove them by clicking on them and clicking the Remove button. Typically Win95 installs the Microsoft NetBeui protocol, and IPX/SPX compatible protocol. You can delete the NetBEUI protocol, but you will need the IPX/SPX protocol if you wish to login to the Netware Server.

To setup TCP/IP click on TCP/IP, click on Properties, click on the tab IP address

```
Enter your IP address in the "Specify an IP address "
                      box as 172.17.0.5
In the Subnet Mask box enter 255.255.255.0
```

select the tab Gateway

```
Enter your gateway (router) address in the box New gateway
   as 172.17.0.254
Click the Add button
```

The gateway address should now appear under the installed gateways box. Now Click OK.

You should get a message to reboot. Do so. You should now be able to use the Browser to connect to your HTTP Server.

4.4 Setup Microsoft Client

If you are using the Microsoft Client for accessing your network, then this section details how to install TCP/IP for

- Windows for Workgroups (WFWG)
- Windows 95
- Windows NT 4.0

Note: To enable you to refer to the Linux server as `http://linux.mydomain/` instead of `http://172.16.0.1/` in the WWW browser and all your intranet commands you need to edit the `hosts` file. You can add more entries for each of your other hosts (Netware, Unix, WinNT) as well. The Windows family keeps its HOSTS file in `\WINDOWS` or in `\WINDOWS\SYSTEM` depending on the version. Edit this file and add a line for your Linux server as:

```
127.0.0.1       localhost
172.16.0.1      linux.mydomain

172.16.0.2      netware.mydomain
172.16.0.3      winNT.mydomain
172.16.0.5      ws_1
```

Windows for Workgroups

This section explains how to install the 32bit client on WFWG. Firstly you must download the TCP/IP drivers for Windows from Microsoft. The current version is 3.11b and is available at ftp://ftp.microsoft.com or other sites as `tcp32b.exe`. Make sure that you have load Win32s before trying to load the TCP/IP-32bit driver.

Having expanded the TCP/IP files into a temporary directory (say `C:\TEMP`), check your `\WINDOWS\SYSTEM` directory for copies of `OEMSETUP.INF`. If there are any, rename them. Now copy the `OEMSETUP.INF` file from the TEMP directory to the `\WINDOWS\SYSTEM` directory. If you have loaded any other TCP/IP stacks on your system, please remove them before you proceed.

Start Network Setup or Windows Setup/Change Network settings

```
Click the Networks button
Click Install Microsoft Windows Network.
  Choose support for additional networks (if required)
Click OK
```

TOC

You should be prompted for your network adapter - select the appropriate one. If you are not prompted, then

```
Click the Adapter button
    select an adapter (say NE2000)
    Click OK
Click the Protocol button
    select the MS TCP/IP-32 protocol
    click OK
```

You will now be prompted to configure the TCP/IP protocol stack. You can always reconfigure this by highlighting the TCP/IP protocol shown in the box Adapters and clicking the Setup button.

```
In the IP address box enter 172.17.0.5
In the Subnet Mask box enter 255.255.255.0
Enter your gateway (router) address in the box default gateway
  as 172.17.0.254
```

Click OK. The computer will ask you to restart. Do so. You should now be able to use the Browser to connect to your HTTP Server.

Windows 95

This section explains how to install the 32bit client for Microsoft on Win95. Firstly you must install the following

```
Client for Microsoft Networks
Microsoft TCP/IP Protocol
Network Adapter
```

To install these items, click on My Computer, Control Panel, Networks. Click Add. You will now be in a window that displays Client, Adapter, Protocol and Service. To install the Client for Microsoft Networks:

```
1. DoubleClick on Client
2. Click on Microsoft
3. DoubleClick on Client for Microsoft Networks
```

To install the TCP/IP Protocol:

```
1. DoubleClick on Protocol
2. Click on Microsoft
3. DoubleClick on TCP/IP
```

Windows 95 by default installs several protocols automatically. Remove them by clicking on them and clicking the Remove button. Typically Win95 installs the Microsoft NetBeui protocol.

To setup TCP/IP click on TCP/IP, click on Properties, click on the tab IP address

```
Enter your IP address in the "Specify an IP address "
                      box as 172.17.0.5
In the Subnet Mask box enter 255.255.255.0
```

select the tab Gateway

```
Enter your gateway (router) address in the box New gateway
   as 172.17.0.254
```

```
Click the Add button
```

The gateway address should now appear under the installed gateways box. Now Click OK.

You should get a message to reboot. Do so. You should now be able to use the Browser to connect to your HTTP Server.

Windows NT

This section details how to Install the TCP/IP client for WinNT 4.0. Start Control Panel/ Network

```
Select the Adapter tab.
    Click Add to add a new adapter (if you don't have one)
```

You should be prompted for your network adapter - select the appropriate one. To add the protocols.

```
Select the Protocols tab
    Click Add
    Select the TCP/IP protocol
    Click OK
```

You will now be prompted to configure the TCP/IP protocol stack. You can always reconfigure this by highlighting the TCP/IP protocol and clicking the Properties button.

```
Select the tab IP Address
   Mark the checkbox 'Specify an IP address'
   In the IP address box enter 172.17.0.5
   In the Subnet Mask box enter 255.255.255.0
   Enter your gateway (router) address in the box Default Gateway
       as 172.17.0.254
```

Click OK. The computer will ask you to restart. You can now use any Browser to connect to your HTTP Server.

4.5 Setup TCP/IP on Macintosh

If you are using the Macintosh for accessing your network, then this section details how to install MacTCP for the PowerMacs.

Note: To enable you to refer to the Linux server as `http://linux.mydomain/` instead of `http://172.16.0.1/` in the WWW browser and all your intranet commands you need to edit the `hosts` file. The format of the hosts file is different from the one used in Unix. The Mac hosts file is based on RFC-1035. You can add more entries for each of your other hosts (Netware, Unix, WinNT) as well. The MacOS keeps its HOSTS file in the `Preferences folder` under the `System folder`. Edit this file and add a line for your Linux server as:

```
linux.mydomain    A  172.16.0.1

netware.mydomain  A  172.16.0.2
winNT.mydomain    A  172.16.0.3
ws_1              A  172.16.0.5
```

TOC

MacTCP

This section explains how to install MacTCP. Firstly you must download the MacTCP files from Apple or install it from the Internet Connection CD. To configure MacTCP, click the Apple Menu/ Control Panels/ TCP/IP. In the screen change the setting for 'Connect via:' to 'Ethernet'

Change the 'Configure' setting to 'Manually'

```
In the IP address box enter 172.17.0.5
In the Subnet Mask box enter 255.255.255.0
Enter your gateway (router) address in the box
  Router address as 172.17.0.254
```

Click OK. You should now be able to use the Browser to connect to your HTTP Server.

---

5. Setting up the Intranet

An Intranet cannot be complete without sharing the resources on the different platforms. You will need support for other filesystems, so that you can access the data available on them. This document provides instructions to connect Linux to the following popular filesystems.

- NCP filesystem for Netware
- SMB filesystem for Windows
- NFS filesystem for Unix

These filesystems can be compiled into the Linux kernel or added as modules, depending on the version of Linux. If you are not familiar with compiling the kernel you can refer to the Kernel HOWTO http://sunsite.unc.edu/mdw/HOWTO/Kernel-HOWTO.html and the Module HOWTO http://sunsite.unc.edu/mdw/HOWTO/Module-HOWTO.html for compiling the kernel with modules.

5.1 NCPFS

To share the files on the Netware server you will need support for NCP (ncpfs). NCPFS works with kernel version 1.2.x and 1.3.71 upwards. It does not work with any earlier 1.3.x kernel. It cannot access the NDS database in Netware 4.x, but can make use of the bindery. If you are using Netware 4.x you can enable bindery support for specific containers using the command `Set Bindery Context` at the console as:

```
set Bindery Context = CORP.MYDOM;WEBUSER.MYDOM
```

In the above example two containers have bindery support enabled.

You will need to download the NCP filesystem utilities using the URL ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/ncpfs.tgz (currently ncpfs-2.0.10) from Sunsite.

Installation

To install the ncpfs utilities, type

```
zcat ncpfs.tgz | tar xvf -
```

to expand the files into its own directory. In this case you will get a directory `ncpfs-2.0.10` Change your directory to this ncpfs directory before proceeding with the installation. Read the README and edit the Makefile if necessary.

The installation of ncpfs depends on the kernel version you are using. For kernel 1.2, you should simply type 'make'. Subsequently typing 'make install' will install the executables and man pages.

If you use Kernel 1.3.71 or later, you might have to recompile your kernel. With these kernels, the kernel part of ncpfs is already included in the main source tree. To check if the kernel needs to be recompiled type

```
cat /proc/filesystems
```

It should show you a line saying that the kernel knows ncpfs.

If ncpfs is not there, you can either recompile the kernel or add ncpfs as a module. For recompiling the kernel you should type 'make config' and when it asks you for

```
The IPX protocol (CONFIG_IPX) [N/y/?]
```

simply answer 'y'. Probably you do not need the full internal net that you are asked for next. Once the kernel is successfully installed, reboot, check `/proc/filesystems` and if everything is OK proceed with the installation of the ncpfs utilities. Change directory to the location holding your downloaded ncpfs files, and type 'make'. After the compilation is finished type 'make install' to install the various utilities and man pages.

Mounting NCPFS

To check the installation type

```
        ipx_configure --auto_interface=on --auto_primary=on

    ....wait for 10 seconds and type

        slist
```

You should be able to see a list of your Netware servers. Now we are ready to share files from the Netware server.

Suppose we need to access HTML files from directory `\home\htmldocs` on volume VOL1: on the server MYDOM_NW, I recommend that you create a new user (say) 'EXPORT' with password 'EXP123' on this server to whom you grant appropriate access rights to this directory using SYSCON or NWADMIN.

On the Linux machine create a new directory `/mnt/MYDOM_NW`. Now type the command

```
        ncpmount -S MYDOM_NW -U EXPORT -P EXP123 /mnt/MYDOM_NW
```

to mount the netware file system. Typing the command

```
        ls /mnt/MYDOM_NW/vol1/home/htmldocs
```

will show you a list of all the files in `MYDOM_NW/VOL1:\HOME\HTMLDOCS` (using Netware file notation). If you have any problems please read the IPX HOWTO at http://sunsite.unc.edu/mdw/HOWTO/IPX-HOWTO.html for more insights into the IPX system.

5.2 SMBFS

To share the files on the Windows server you will need support for SMB (smbfs).

You will need to download the SMB filesystem utilities from ftp://sunsite.unc.edu/pub/Linux/system/filesystems/smbfs/smbfs.tgz (currently smbfs-2.0.1) from Sunsite.

Installation

To install the smbfs utilities, type

```
        zcat smbfs.tgz | tar xvf -
```

to expand the files into its own directory. In this case you will get a directory `smbfs-2.0.1` Change your directory to this smbfs directory before proceeding with the installation. Read the README and edit the Makefile if

necessary.

The installation of smbfs depends on the kernel version you are using. For kernel 1.2, you should simply type 'make'. Subsequently typing 'make install' will install the executables and man pages.

If you use Kernel 2.0 or later, you might have to recompile your kernel. With these kernels, the kernel part of smbfs is already included in the main source tree. To check if the kernel needs to be recompiled type

```
cat /proc/filesystems
```

It should show you a line saying that the kernel knows smbfs.

If smbfs is not there, you can either recompile the kernel or add smbfs as a module. For recompiling the kernel you should type 'make config' and when it asks you for adding SMB filesystem support simply answer yes. Once the kernel is successfully installed, reboot, check `/proc/filesystems` and if everything is OK proceed with the installation of the smbfs utilities. Change directory to the location holding your downloaded smbfs files, and type 'make'. After the compilation is finished type 'make install' to install the various utilities and man pages.

Mounting SMBFS

In our example let us assume that the WinNT server is called 'MYDOM_NT' and is sharing its directory `C:\PUB\HTMLDOCS` with a share name of 'HTMLDOCS' without a password. On the Linux machine create a new directory `/mnt/MYDOM_NT`. Now type the command

```
smbmount //MYDOM_NT/HTMLDOCS /mnt/MYDOM_NT -n
```

to mount the SMB (windows share) file system. If this does not work try

```
smbmount //MYDOM_NT/COMMON /mnt/MYDOM_NT -n -I 172.16.0.3
```

Typing the command

```
ls /mnt/MYDOM_NT
```

will show you a list of all the files in `bsol;bsol;MYDOM_NT\PUB\HTMLDOCS` (using Windows file notation).

5.3 NFS

First you will need a kernel with the NFS file system either compiled in or available as a module.

Suppose you have a Unix host running NFS with the name MYDOM_UNIX and an IP address of 172.16.0.4. You can check the directories that are being exported (shared) by this host by typing the command

```
showmount -e 172.16.0.4
```

Once we know the exported directories you can mount them by entering a appropriate mount command. I recommend that you create a subdirectory under '/mnt' (say) 'MYDOM_UNIX' and use that as your mount point.

```
mount -o rsize=1024,wsize=1024 172.16.0.4:/pub/htmldocs /mnt/MYDOM_UNIX
```

The rsize and wsize may have to be changed depending on your environment.

If you have any problems please read the NFS HOWTO at http://sunsite.unc.edu/mdw/HOWTO/NFS-HOWTO for more insights into the NFS system.

6. Accessing the Web

Now that we have setup the HTTP server, the clients and interconnected the Linux server with the other servers, we need to make some small adjustments on the Linux server to be able to access these mounted filesystems from the Web Browser.

6.1 Accessing the mounted filesystems

To access the mounted directories in your HTML pages you have two methods:

- Create a link in DocumentRoot (`/usr/local/etc/httpd/htdocs`) to refer to the mounted directory as

```
ln -s /mnt/MYDOM_NW/vol1/home/htmldocs netware
                  or

ln -s /mnt/MYDOM_NT    winNT
                  or

ln -s /mnt/MYDOM_UNIX    unix
```

- to edit the file `srm.conf` in your `/usr/local/etc/httpd/conf` directory and add a new alias.

```
# Alias fakename realname
Alias /icons/     /usr/local/etc/httpd/icons/

# alias for netware server
Alias /netware/   /mnt/MYDOM_NW/vol1/home/htmldocs/
Alias /winNT/     /mnt/MYDOM_NT/
Alias /unix/      /mnt/MYDOM_UNIX
```

And restart your HTTPd. You can access the documents on the netware server by referring to them as `http://linux.mydomain/netware/index.htm` for the netware files and similar notations for the others.

6.2 Connecting to the Internet

You can finally connect your Intranet to the Internet to access E-Mail and all the wonderful information out there. I propose to write a brief note on how to do this in a future revision. Detailed explanations are available in the ISP Hookup HOWTO from http://sunsite.unc.edu/mdw/HOWTO/ISP-Hookup-HOWTO.html and Diald mini HOWTO at

http://sunsite.unc.edu/mdw/HOWTO/mini/Diald for setting up these connections.

6.3 Other uses

The HTTP server can be used in the office to provide transparent access to information residing on different servers, at several locations and directories. The data can be simple documents in Word, Lotus spreadsheets, or complex databases.

The application of this technology is being typically used as follows:-

- Publishing corporate documents

  These documents can include newsletters, annual reports, maps, company facilities, price lists, product information literature, and any document which is of value within the corporate entity.
- Access into searchable directories

  Rapid access to corporate phone books and the like. This data can be mirrored at a Web site or, via CGI scripts, the Web server can serve as a gateway to back-end pre-existing or new applications. This means that, using the same standard access mechanisms, information can be made more widely available and in a simpler manner. This means that it can be used to create an interface with RDBMS like ORACLE and SYBASE for generating real-time information. Here is a list of links to such sites on the Web.

  - Web Access - http://cscsun1.larc.nasa.gov/~beowulf/db/web_access.html - CGI gateways - HTTP://www.w3.org/hypertext/WWW/RDBGate/Overview.html
- Corporate/Department/Individual pages

  As cultures change within organistions to the point where even each department moves towards their own individual mission statements, the Intranet technology provides the ideal medium to communicate current information to the Department or Individual. Powerful search engines provide the means for people to find the group or individual who has the answers to the continuous questions which arise in the normal day-to-day course of doing business.
- Simple Groupware applications

  With HTML forms support, sites can provide sign-up sheets, surveys and simple scheduling.
- Software distribution

  Administrators can use the Intranet to deliver software and up-dates on-demand to users across the corporate network . This can be done with 'Java' which allows the creation and transparent distribution of objects on-demand rather than just data or applications. This is indeed possible more easily with the newer versions of Linux which has builtin support for Java.
- Mail

  With the move to the use of Intranet mail products with standard and simple methods for attachment of documents, sound, vision and other multimedia between individuals, mail is being pushed further forward as a simple, de facto communications method. Mail is essentially individual to individual, or individual to small group, communication. Several utilities are

TOC

available on the Linux platform to setup an E-mail system like **sendmail, pop3d, imapd**.

- User Interface

  The Intranet technology is evolving so rapidly that the tools available, in particular HTML, can be used to dramatically change the way we interface with systems. With HTML you can build an Interface which is only limited by the creators imagination. The beauty about using Intranet technologies for this is that it is so simple. Clicking a hyperlink from HTML can take you to another page, it could ring an alarm, run a yearend procedure or anything else that a computer program can do.

---

7. More things to do

Here is a list of other interesting things to do with your Linux Intranet server. All the software mentioned below is freeware or shareware.

- Browse the Linux server using Network Neighbourhood in Win95/ NT; Setup a WINS like NBT server. Check out the SAMBA Web page at http://lake.canberra.edu.au/pub/samba/samba.html
- Implement a search engine on your Intranet. Connect to ht://Dig at http://htdig.sdsu.edu/
- Use CUSeeMe by setting up a local reflector. Refer to their page at Cornell http://cu-seeme.cornell.edu/
- Setup Web Conferencing. Use COW from http://thecity.sfsu.edu/COW/
- Deploy a SQL database. Refer to the mSQL Home page at http://Hughes.com.au/
- Setup FTP,Gopher,Finger,Bootp servers on the Netware server. Get them at http://mft.ucs.ed.ac.uk/
- Emulate a Netware server. Check out the NCP Utilities at ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/

If you find other interesting things to do with your Linux Intranet server, please feel free to mail me.

8. Credits and Legalities

8.1 Thanks

Thanks to the people at NCSA for providing such excellent documentation, David Anderson and all others for trying out this HOWTO and sending in their comments. The details on Netware/IP are courtesy Romel Flores (rom@mnl.sequel.net).

8.2 Copyright information

This document is copyrighted © 1996,1997 Pramod Karnad and distributed under the following terms:

- Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.
- All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.
- If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu Finger for phone number and snail mail address.

Linux Italian-HOWTO

Marco Gaiarin (gaio@pluto.linux.it)

v1.2.0, 3 November 1998

Questo HOWTO descrive in alcuni semplici passi come configurare il proprio sistema Linux per supportare pienamente la tastiera e la localizzazione italiana. Inoltre si vuole in questo testo dare tutta una serie di informazioni sul mondo Linux in Italia.

1. Introduzione

- 1.1 Dove potete trovare sempre l'ultima versione...
- 1.2 Convenzioni tipografiche
- 1.3 Note sui Link
- 1.4 Perché Debian?!
- 1.5 Ringraziamenti
- 1.6 Traccia delle modifiche
- 1.7 Copyright

2. La tastiera

- 2.1 Console
- 2.2 XFree86
- 2.3 Per maggiori informazioni

3. National Language Support (NLS)

- 3.1 La localizzazione in una riga...
- 3.2 ...o quasi!
- 3.3 Manpage
- 3.4 Catalogue

- 3.5 Per maggiori informazioni

4. Altre localizzazioni

- 4.1 Font in console
- 4.2 Font in X
- 4.3 Stampante e DOS
- 4.4 libpaper
- 4.5 Per maggiori informazioni

5. Ispell

- 5.1 Ispell e Emacs
- 5.2 Per maggiori informazioni

6. TeX

- 6.1 Un po' di teoria...
- 6.2 L'installazione TeX di Linux
- 6.3 Produzione di elaborati TeX ben localizzati
- 6.4 Per maggiori informazioni

7. Siamo in Europa, bellezza!

- 7.1 Per maggiori informazioni

8. TimeZone

- 8.1 Q: ma perché Linux mi ha sbagliato il passaggio di fuso?!
- 8.2 Q: ma perché ho l'ora impostata a CEST?

9. Documentazione

- 9.1 Pubblicazioni offline (su carta... ;)
- 9.2 Pubblicazioni online
- 9.3 Distribuzioni in Italiano

- 9.4 Altro materiale in Italiano

10. Linux in Italia

- 10.1 Associazioni, gruppi di utenti
- 10.2 Aree di discussione pubbliche
- 10.3 Siti Linux

---

1. Introduzione

> ``Perché scegliere UNIX? A chi utilizza già un PC o un Macintosh potrebbe sorgere il dubbio che il computer a sua disposizione sia già sufficiente per tutte le esigenze descritte in precedenza e che, in fondo, UNIX non sia altro che uno dei tanti dinosauri destinati a morte certa. Probabilmente questo avverrà (in fondo nulla è eterno), ma solamente nel momento in cui tutti i personal computer saranno in grado di svolgere i compiti che, almeno per ora, sono ad esclusivo appannaggio di UNIX."
>
> (da ``UNIX Flash", ed. Apogeo)

Questo testo **non** è una guida all'installazione, né un testo di introduzione a Linux. Se questo è il vostro primo approccio a Linux, perché lo avete appena installato o vi accingete a farlo, qui potete trovare solamente alcuni riferimenti ad altre pubblicazioni che vi possono aiutare. In particolare consiglio gli AppuntiLinux come testo introduttivo a Linux.

Quindi, in questo testo si presume che:

- abbiate Linux installato e funzionante. Quello che di sicuro non troverete qui è una guida alla risoluzione di problemi hardware
- abbiate una versione di Linux ``recente", ovvero un kernel 2.0.0 o successivi, e tutte le nuove versioni delle utilità che occorrono per queste versioni di kernel
- abbiate una minima conoscenza della organizzazione del filesystem e la padronanza dei principali comandi, nonché un'idea dei concetti di utente e utente ``root"
- abbiate una minima conoscenza dell'uso dei programmi fondamentali, almeno un editor di testi.

Se siete alle prime armi con Linux, lo avete appena installato, il consiglio che vi do è quello di mettere da parte questo HOWTO e farvi una certa cultura su UNIX in generale e su Linux.

Potete trovare una miriade di informazioni (libri e documenti da leggere, forum di discussione, ``persone" da incontrare) negli ultimi due capitoli di questo HOWTO.

1.1 Dove potete trovare sempre l'ultima versione...

Dalla passata versione l'Italian HOWTO è stato ``snellito"; ormai non ci sono piú grossi problemi di localizzazione ed ho quindi preferito spostare buona parte del materiale (la piú ``volatile") sotto forma di contributi in un URL a parte.

Trovate quindi l'ultimissima versione di questo testo, assieme a contributi e qualche curiosità, all'url http://www.pluto.linux.it/~gaio/Italian-HOWTO; in ogni caso l'Italian HOWTO è un documento ``ufficiale" e quindi fa parte sia di ILDP che di LDP, con la conseguenza pratica che lo trovate ormai dappertutto, in tutti i formati disponibili (testo, html per leggere a video e PostScript per stamparlo comodamente), solitamente anche preinstallato dalla vostra distribuzione preferita.

TOC

Inoltre l'autore è reperibile all'indirizzo gaio@pluto.linux.it. Oppure agli indirizzi 2:333/1016 (FidoNet), 61:3917/1 (PeaceLink), 91:13/1 (PNet), 1907:395/101.20 (ScoutNet).

Sono contentissimo di ricevere commenti e suggerimenti per questo testo, e invito tutti a segnalarmi eventuali errori, dimenticanze o ad inviarmi proposte e contributi.

1.2 Convenzioni tipografiche

- Questo è un **programma(n)** o file di configurazione del sistema. Il numero tra parentesi non è messo a caso. Vuol dire che quel programma è dotato di una *man page*, consultabile con il comando `man n programma`; per saperne di piú sulle manpage potete consultare la sezione Manpage. Il fatto che scriva **esplicitamente** quel numero tra parentesi vuole essere un invito **esplicito** a leggere quella man page.
- Questo è un `comando` da impartire o un nome di file o directory.
- Questo è un **[Tasto]** sulla tastiera. Questa è una combinazione di tasti **[Tasto1+Tasto2]**. Di solito, per comodità, eviterò di scrivere le combinazioni di tasti con shift, perciò al posto di **[Shift+a]** scriverò il piú semplice **[A]**.
- Questo è un *termine nuovo*, che va quindi ``metabolizzato''. Di solito tendo a essere chiaro, nel senso che dal contesto o dalle righe immediatamente successive se ne dovrebbe capire il significato. Se non è cosí, fatemelo sapere! ;)
- Ormai molto spesso parlo di **pacchetti**, ovvero di quegli archivi di programmi, file di configurazione e documentazione usati dalle moderne distribuzioni che permettono una agevole manutenzione (installazione, disinstallazione, configurazione) di un sistema Linux; il pacchetto ha un nome, che riporto, e delle informazioni di versione e tipo (`.deb` per Debian, `.rpm` per RedHat ad esempio) che invece solitamente non riporto perché implicite o inessenziali.

1.3 Note sui Link

In questo testo ho cercato di porre molta attenzione ai link, di modo che la versione html sia piacevolmente fruibile. Ma buona parte della documentazione a cui questo testo fa riferimento molto probabilmente è già presente nel vostro disco o nel CD della vostra distribuzione Linux: prendete quindi tutti questi link con un po' di sale in zucca, e fate una ricerca locale prima di lanciarvi in estenuanti riferimenti a siti d'oltreoceano.

Ad esempio tutti gli HOWTO sono solitamente già presenti nella directory `/usr/doc/HOWTO`, compreso questo che state leggendo e le traduzioni in Italiano (almeno in Debian); non saranno le ultime versioni, chiaro, ma solitamente sono piú che ottime.

Uomo avvisato, mezzo salvato! ;)

TOC

1.4 Perché Debian?!

L'Italian-HOWTO, pur cercando di essere il più indipendente possibile dalla distribuzione, fa riferimento pesantemente a Debian.

Questa vuole essere una scelta ``tecnica'', perché ha una init SYSV, perché ha un sistema di pacchettizzazione e configurazione serio e funzionale, perché rispetta tutti gli standard che si sono via via creati all'interno della comunità Linux, perché è già ora parzialmente in italiano e in futuro lo sarà (si spera) totalmente, perché è l'unica ad avere come obiettivo l'internazionalizzazione.

Ma vuole essere anche una scelta ``politica'', visto che Debian è l'unica distribuzione non commerciale e OpenSource, e credo quindi che meriti i miei due centesimi... di Euro! ;)

Debian e RedHat, in fondo, sono molto simili, quindi gli utenti RH (e distribuzioni RH-like come SuSE e Caldera) non dovrebbero trovarsi troppo a disagio, se non per la localizzazione quasi assente di queste distribuzioni.

1.5 Ringraziamenti

Come sempre ringrazio Anna per l'affetto, mia mamma perché mi sopporta, il papà perché non si è ancora pentito di avermi dato il numero di carta di credito, Jacopo perché sopravvive alla mia arte (?!) culinaria, Luca perché si lascia sfruttare ;), il PLUTO e PeaceLink che mi sopportano nonostante non faccia quasi niente, Leandes di Pâs e il Coordinamento Telematico di Azione Cattolica (e in particolare Lindo) che sopportano la mia continua rottura di scatole, l' Azione Cattolica di Cimpello/Praturlone che deve tenersi un segretario che si dimentica tutto, rete FidoNet che probabilmente si è beccata un bell'osso di sysop; e una caterva di altra gente che non ringrazio per decenza. ;)

Ma una dedica particolare va a Maria Luisa, Stefano e il nuovo arrivato Alessandro (il mio primo nipotino! ;). Ora spetta anche a me fare in modo che Alessandro possa vivere, da grande, in un mondo solidale, giusto, nonviolento, pulito. In fondo la scelta è semplice... pensare solo a noi stessi, oppure lasciare questo mondo un po' meglio di come l'abbiamo trovato! ;)

Dedico inoltre questo HOWTO a quattro pazzi di amici, che hanno scelto di fare la cosa piú difficile: cercare di vivere con la propria passione... sembra quasi una poesia ma è invero una ProSA! ;)

1.6 Traccia delle modifiche

0.1b

Prima versione, niente di più che una traduzione/copiatura di altro materiale simile reperito in rete

0.2b

Aggiunto il capitolo su Linux in Italia, qualche bugfix.

0.4b

TOC

Risistemazione generale, aggiunta del capitolo su ispell

0.6b

Profondo lifting (ho tolto una marea di roba che si trova semplicemente in altri HOWTO), riscrittura del capitolo sulla localizzazione con aggiunta di informazioni sui vari progetti, aggiornamento del capitolo su ispell. Questa vuole essere l'ultima versione beta (è un buon proposito ;).

0.8b

Niente di meno che una correzione della precedente versione, con qualche aggiunta. Questa è di sicuro l'ultima versione beta.

1.0.0

Il testo dovrebbe aver preso la sua struttura definitiva; è stato riorganizzato tutto ed in particolare sono state aggiunte sezioni per libpaper, fortune in italiano e distribuzioni in italiano, è stata tolta la bibliografia in inglese; sono state inoltre riviste completamente le parti riguardanti la tastiera (aggiunta di xkbd), TeX (da NTeX a teTeX) e la locale; è stato aggiunto il crossreferencing all'interno del documento.

1.2.0

Versione riveduta e corretta (soprattutto gli URL, che in un anno cambiano parecchio! ;) del precedente; inizia il lavoro di ``potatura'' di buona parte del materiale, man mano che i problemi piú grossi (locale e tastiera) vengono risolti. È stato aggiunto il paragrafo (praticamente vuoto) per le future euroquestioni.

1.7 Copyright

Per motivi spudoratamente legali, riporto come messaggio di copyright la versione ``originale'' in inglese. Trovate subito sotto la traduzione.

This Howto is copyrighted by Marco Gaiarin and distributed as other Linux HOWTOs under the terms described below.

Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu. You may finger this address for phone number and additional contact information.

Questo HOWTO è di Marco Gaiarin ed è distribuito come gli altri HOWTO di Linux sotto i termini descritti sotto.

I documenti degli HOWTO di Linux possono essere riprodotti e distribuiti in tutto o in parte, con ogni mezzo fisico o elettronico, purchè questo avviso di copyright sia mantenuto su tutte le copie. La distribuzione commerciale è permessa e incoraggiata; comunque all'autore piacerebbe essere avvisato di ogni distribuzione di questo tipo.

Ogni traduzione, lavoro derivato o comprendente ogni documento degli HOWTO di Linux deve essere coperto sotto questo avviso di copyright. Cioè non potete produrre un lavoro derivato da un HOWTO e imporre restrizioni aggiuntive sulla sua distribuzione. Eccezioni a queste regole possono essere garantite sotto certe condizioni; contattate il coordinatore degli HOWTO di Linux all'indirizzo indicato sotto.

Se avete domande contattate Tim Bynum, il coordinatore degli HOWTO di Linux, a linux-howto@sunsite.unc.edu. Potete fare finger a questo indirizzo per il numero di telefono e altre informazioni per contattarlo.

---

2. La tastiera

La gestione della tastiera di Linux è organizzata su piú ``strati'' o livelli di astrazione il cui numero, tre, non ci è di grande interesse. Quello che interessa a noi è il risultato, ovvero una gestione della tastiera indipendente dall'hardware (ricordo che Linux è un sistema operativo multipiattaforma e non funziona solo sui PC compatibili...) e facilmente riconfigurabile.

Infatti è sufficiente modificare l'interfaccia tra uno strato e l'altro per ottenere gli effetti desiderati: nel nostro caso il perfetto funzionamento della tastiera italiana.

2.1 Console

Per ottenere la configurazione della tastiera italiana in console è sufficiente impartire il comando `loadkeys it.map`; questo comando non fa altro che caricare la tabella di descrizione della tastiera (*keymap*, appunto) `it.map` al posto di quella di default ``inserita'' nel kernel.

Solitamente è la procedura di installazione che si preoccupa di chiedere all'utente quale tastiera usare e a inserire un comando come quello qui sopra in uno degli script di avvio del sistema.

In Debian esiste il comando **kbdconfig** che permette di riconfigurare la tastiera senza editare a mano gli script di avvio, le altre distribuzioni avranno di sicuro sistemi simili...

Questa Keymap inoltre supporta alcune aggiunte essenziali per i programmatori, come le parentesi graffe, il carattere tilde e il carattere apice inverso ottenuti premendo rispettivamente [AltGr+8], [AltGr+9], [AltGr+0] e [AltGr+'].

Se si vuole indagare ulteriormente nella gestione della tastiera nella console Linux, si può far riferimento ai comandi (e alle rispettive manpage) **showkey(1)** è **dumpkey(1)**, oltre ovviamente al già citato **loadkey(1)**.

Keymap alternative

Nonostante questa keymap sia già molto buona, si sta cercando di fare meglio; in particolare Lorenzo Maria Catucci (lorenzo@argon.roma2.infn.it) sta lavorando a una serie di keymap per console e per X congruenti e possibilmente senza problemi. Trovate le bozze di queste keymap nel sito dell'Italian HOWTO http://www.pluto.linux.it/~gaio/Italian-HOWTO.

Nello stesso sito potete trovare inoltre delle keymap alternative che permettono di ``emulare'' la tastiera Amiga e cose simili...

2.2 XFree86

Di recente (dalla versione 3.2 in poi...) XFree ha completamente cambiato la gestione della tastiera, aggiungendo una estensione che si chiama ``XKeyboard'' (o XKBD), e qualche astrusa voce in più nel file di configurazione. Il nuovo sistema è effettivamente migliore, ma talmente configurabile che c'è voluto un annetto perché venisse metabolizzato e quindi venissero prodotte le keymap secondo queste nuove specifiche...

TOC

Il problema vero è che **xf86config(1)** non riesce più a gestire in fase di configurazione automatica la tastiera italiana (o meglio, per trovare uno straccio di riferimento all'Italia occorre muoversi per un po' tra complicate opzioni...), ma niente paura.

Si può benissimo completare la configurazione scegliendo una tastiera predefinita (come quella americana) e poi editare il file `/etc/X11/XF86Config` aggiungendo o modificando le informazioni:

```
Section "Keyboard"

[...]

LeftAlt      Meta
RightAlt     ModeShift

# ovviamente se avete una tastiera W95
# al posto di pc102 ci va pc104 (o pc105)!
#
Protocol     "Standard"
XkbRules     "xfree86"
XkbModel     "pc102"
XkbLayout    "it"

EndSection
```

Le prime due righe, in particolare, servono ad avvisare X che abbiamo intenzione di usare [AltGr] come modificatore ModeShift per poter utilizzare, come siamo abituati, parentesi quadre, at e diesis. (Non vi ricordate? Era una di quelle imbarazzanti domande che vi ha posto **xf86config(1)**...)

Questa definizione di tastiera è leggermente diversa (come disposizione dei tasti ``nascosti") di quella in console (ad esempio le parentesi graffe sono ora in [AltGr+7], [AltGr+0]).

XF86Setup(1)

In alternativa è possibile usare il nuovo programma di configurazione di XFree, presente dalla versione 3.2 e seguenti, **XF86Setup(1)**; la configurazione di X11 è molto piú intuitiva con XF86Setup che con xf86config: per un novellino (ma anche non) può essere utile anche perché permette di controllare al volo se le impostazioni sono corrette (da Giovanni Bortolozzo).

Per poter utilizzare XF86Setup occorre installare il server VGA16 in aggiunta a quello appropriato per la propria scheda video.

Backspace e Delete

Un problema noioso, che però non riguarda solo la tastiera italiana, è che in X gli effetti della pressione dei tasti BackSpace e Canc coincidono (ambedue fanno Del, ovvero cancellano il carattere sul cursore; o ambedue fanno BackSpace, ovvero cancellano il carattere alla sinistra del cursore, dipende di solito dal tasso di umidità dell'aria... ;-).

Questo è un problema comune a tutti gli utenti Linux, ed inoltre esistono ottimi documenti che spiegano come sistemare (o mettere una pezza) a questo problema; per puntatori vedi alla fine di questa sezione.

Questo comportamento è dovuto in maggior parte da incompatibilità ``storiche'' tra librerie/applicazioni/metodi di programmazione, e anche in distribuzioni come Debian che hanno forzato una unica politica di accesso ai terminali (con il database **terminfo(5)**) il problema con alcune (vecchie) applicazioni rimane.

Insomma, è un problema **delle applicazioni** e non di Linux.

Per programmi (o sistemi) che usano ancora **termcap(5)** è molto probabile che occorra una correzione della definizione di xterm. Non è proprio il caso di inserire una cosa del genere in questo HOWTO, mal che vada nel sito dell'Italian-HOWTO http://www.pluto.linux.it/~gaio/Italian-HOWTO; è comunque piú corretto cercare una versione dell'applicazione compilata con terminfo oppure... cambiare applicazione! ;)

Per comprendere meglio queste cose sono letture consigliate le manpage di **termcap(5)** e **terminfo(5)**, oltre a quanto riportato alla fine di questa sezione.

MetroX e tastiera italiana

Ho visto che piú persone hanno problemi con la tastiera sotto Metro-X, non so se intanto il problema è stato risolto, eventualmente segnalatemelo che elimino questa sezione.

Per risolvere in modo pulito e definitivo il problema modificate questa sezione del file `/etc/X11/XMetroconfig`

```
Section "Keyboard"
        LeftAlt         Meta
        RightAlt        ModeShift
        Protocol        "Standard"
        XkbKeymap       "xfree86(it)"   # al posto di "metro(it)"
        XkbModel        "pc105"         # solo se avete i tasti W95
        XkbRules        "xfree86"
        XkbLayout       "it"
EndSection
```

Se prima di farlo guardate alla fine del file `/var/log/Xerrors`, scoprirete quale è il problema: mancano i file kbd della Metro (da Andrea Tuccia).

Keymap alternative

Anche qui sono state raccolte una serie di keymap alternative a quelle ufficiali presenti nelle distribuzioni; trovate anche queste al sito http://www.pluto.linux.it/~gaio/Italian-HOWTO.

Inoltre Dino Paganello mi segnala il programma **xrus(1)** che permette di fare lo switch al volo tra diverse descrizioni di tastiera (lui lo usa per passare la volo dalla tastiera americana a quella italiana).

2.3 Per maggiori informazioni

Ulteriori informazioni, specialmente se si vuole migliorare la gestione della tastiera (aggiungere i tasti cursore in determinate applicazioni, macro ai tasti funzione, ...), si possono trovare in:

- Keyboard and Console HOWTO.
- XFree86 HOWTO (anche in versione Italiana).
- Esiste inoltre la *kbd.FAQ* contenuta nel pacchetto del software di gestione della tastiera, e che dovrebbe essere già stata installata da qualche parte nel vostro disco (di solito `/usr/doc/kbd`).
- Anne Baretta (anne@ibbnet.nl) ha creato delle pagine per aiutare la configurazione di BS/DEL; non le ho guardate, ma potrebbero essere un buon punto di partenza. Le trovate su http://www.ibbnet.nl/~anne/keyboard.html
- Esistono anche queste pagine (http://electron.et.tudelft.nl/~jdegoede/fixkeys.txt) per il problema BS/DEL; riporto questo URL solo per ribadire che questi **non** sono modi con cui è possibile affrontare (e tantomeno risolvere) il problema; testi del genere, che non hanno alla base un serio studio e soprattutto l'uso di terminfo, sono come una lobotomia come rimedio per il mal di testa. Magari funziona ma...

---

3. National Language Support (NLS)

Di certo vi stupirà sapere che lo standard *POSIX*, e quindi anche Linux, prevede il supporto delle regole nazionali per la gestione di set di caratteri, modo di scrivere data, ora, numeri in generale e quindi tutto quello che può (informaticamente parlando) distinguere una paese dall'altro.

Dalla versione 5.2.18 delle GNU libc, **tutta** la localizzazione è supportata anche da Linux. Ma non basta avere installate delle moderne libc, occorre anche che questa sia stata compilata con determinate opzioni e occorre installare tutta una serie di file e directory che descrivono la ``locale'' Italia. Insomma, compito non facile se la vostra distribuzione non ha già il supporto per la localizzazione...

Per poter influenzare Linux di modo che reagisca secondo la locale scelta, occorre modificare delle variabili d'ambiente. In ordine:

LANG

È la variabile ``generale'' del gruppo, serve a definire una regola generale (il paese) con cui poi assegnare le variabili che vengono sotto. Una definizione delle variabili sottostanti annulla però quella di default definita da LANG. Inoltre LANG influenza anche il programma **man(1)**, ci torneremo poi.

LC_COLLATE

Influisce sulle regole di ordinamento alfabetico.

LC_CTYPE

Definisce il set di caratteri usato dal sistema.

LC_MONETARY

Definisce come devono essere scritte le valute, ovvero quale, tra virgola e punto, è il separatore di decimali e migliaia e viceversa; definisce inoltre il simbolo di valuta.

LC_NUMERIC

Separatore di decimali e migliaia, formattazione dei numeri.

LC_MESSAGES

Definisce i valori ``si'' e ``no''.

LC_TIME

Come stampare data e ora (questo influenza date(1) e programmi vari).

LC_ALL

Come **LANG**, solo che questa ignora i valori definiti per ogni singola variabile (forza tutto al suo valore). Come spiegherò in seguito, questa variabile **non** va usata a meno di non sapere perfettamente quello che si sta facendo.

3.1 La localizzazione in una riga...

TOC

Piú semplicemente di quello che si crede, basta mettere nel proprio `~/.profile` o `/etc/profile` per *sh-type shells*:

```
export LANG=it_IT
```

Oppure se avete *c-type shells*:

```
setenv LANG it_IT
```

Perché ``it_IT''?! È solo una convenzione, e il significato è ``lingua_PAESE''. Per fare un esempio potrebbe essere it_CH (Svizzera italiana).

Come sempre la man page di **locale(7)** è lettura molto consigliata, come un giro per `/usr/share/locale` per vedere le locali disponibili.

Bug delle localizzazioni

Solitamente si manifestano con errate definizioni degli alias di locale, quindi definizioni che dovrebbero essere equivalenti danno risultati completamente diversi; per rimediare a questo occorre prelevare le definizioni di locale aggiornate, o installarle (molte distribuzioni non le installano proprio).

Può anche essere che alcune distribuzioni non abbiano proprio il supporto per la locale, al che conviene cambiare distribuzione (l'alternativa è ricompilare a mano la libreria di sistema libc).

Bug delle applicazioni

Anche qui, come per il problema BS/DEL, se qualcosa non funziona è colpa, quasi certamente, della applicazione.

C'è qui una specie di ``caverna degli orrori''... ;)))

Netscape

Netscape, se vengono definite le variabili LC_NUMERIC e LC_MONETARY, si mette a stampare e salvare PostScript con punti al posto delle virgole e viceversa, con il risultato che **gs(1)** va in palla...

Se capita, ricontrollate la vostra definizione di locale, e se avete fretta di stampare basta che definiate le variabili seguenti:

```
LC_NUMERIC=posix
LC_MONETARY=posix
```


che forzano i valori di default (e permettono di stampare).

Notare che se si è definita LC_ALL, queste due righe qui sopra vengono ignorate: per questo è bene **NON** definire la variabile LC_ALL.

È probabile che questo problema non sia solo di Netscape: se vi capita un problema simile, sapete ora come ``mettere una pezza'', ma avvisate l'autore del programma in questione della presenza di questo bug.

perl(1)

Spesso **perl(1)** si lamenta della mancanza della locale; questo è dovuto sia a una non completa implementazione da parte della distribuzione, sia in parte a perl stesso che non riesce a gestire gli alias delle varie locali, e quindi riesce a ``seguire'' la variabile LANG solo se punta a una locale con una reale directory sul disco.

Io sulla mia Debian non sono riuscito a riprodurre il problema (se non settando LANG a una locale inesistente ;) e quindi sospetto che tutto nasca da una difficile interazione tra localizzazione e perl stesso, che verrà risolta sicuramente dalle prossime versioni di perl.

RedHat 5.0 soffre di questo problema, che è stato corretto nella 5.1.

3.2 ...o quasi!

Resta il problema di far capire alle applicazioni ``cattive'' che vogliamo avere i caratteri accentati. Questi hanno l'interessante proprietà di essere lunghi 8 bit, mentre molti programmi ne considerano solo 7, perché è da 7 bit il codice ASCII internazionalmente riconosciuto.

Queste note che scrivo dovrebbero presto finire in soffitta, visto che ormai le applicazioni che non gestiscono bene gli 8 bit o che non sano reagire alla variabile LANG sono in via di estinzione.

Editor

Emacs 19

Come sempre emacs è molto particolare. Avvisiamolo che vogliamo i caratteri accentati con :

```
(standard-display-european t)
(set-input-mode (car(current-input-mode))
(nth 1 (current-input-mode)) 0)
```

Aggiunto al nostro bravo `~/.emacs`

joe(1)

Basta editare il file di configurazione `/etc/joe/joerc` o nel proprio `~/.joerc` di modo che comprenda la riga (con il ``-'' sulla colonna 1):

```
-asis           Characters 128 - 255 shown as-is
```

Mail/News

pine(1)

Basta scegliere il set di caratteri ISO-8859-1 dal menu di configurazione, seguendo le voci ``Setup'', poi ``Configure'' e infine ``character-set''; questo fa uscire le mail in codifica ``quoted-printable'' se contengono caratteri accentati (il che è bene ;).

tin(1)

Tin è perfettamente localizzato, occorre solo scegliere il charset ISO-8859-1 per la visualizzazione nel menú di configurazione (attraverso [M]).

nn(1)

Aggiungere la seguente riga nel file `~/.nn/init`:

```
set data-bits 8
```

readline(3)

Readline, che è una libreria standard GNU per la gestione dell'input, nonostante sia giovane, pimpante e ben mantenuta, non arriva per definizione con il supporto per i caratteri accentati. Questa libreria è usata dalla shell di default di linux **bash(1)**, per citare una applicazione di larghissimo uso. Non è da considerare un bug (questo comportamento forse è voluto per garantire il funzionamento anche sui vecchi terminali a 7 bit...) ma è chiaro che l'uso dei caratteri accentati deve essere forzato, ad esempio, se si vogliono usare nomi di file con lettere accentate.

Per fortuna basta solo aggiungere nel proprio `~/.inputrc`

```
set meta-flag on
set convert-meta off
set output-meta on
```


3.3 Manpage

Le *manpage* sono quelle pagine testuali che si richiamano con il comando **man(1)**, e che rappresentano la documentazione online di ogni buon sistema UNIX, non solo di Linux.

Il comando man è abbastanza intelligente da reagire se è definita la variabile LANG. In pratica man va a cercare le manpages prima in $MANPATH/$LANG e poi in $MANPATH se configurato correttamente, permettendo di leggere le pagine nella lingua preferita.

Chiaramente c'è già chi sta traducendo le pagine; se vi sentite dei traduttori nati non vi resta che saltare al progetto ILDP per saperne di piú e collaborare.

Le pagine già tradotte possono essere tranquillamente installate manualmente, prelevando l'ultima versione da ftp://ftp.pluto.linux.it/pub/pluto/ildp/man e leggendo attentamente le istruzioni allegate. Gli utenti Debian possono trovare nella loro distribuzione il pacchetto **manpages-it** già pronto.

3.4 Catalogue

Oltre tutto questo, c'è una terza e importantissima funzione della NLS: i messaggi dei programmi nella lingua locale.

Per farsi un'idea, agli utenti Linux non tocca editare i sorgenti di un programma, modificare a mano tutte le stringhe e poi ricompilare perché questo si presenti in corretto italiano. Esiste una ``prassi'' di programmazione che permette di specificare un linguaggio di default ``compilato'' nell'eseguibile (di solito inglese, per compatibilità) e invece definire delle *catalog*, ovvero dei cataloghi di messaggi in un formato particolare, detto *portable object*, o più brevemente *po*, che messi in una determinata directory vengono usati semplicemente al posto dei messaggi presenti nel programma.

Linux International

Non c'è da meravigliarsi che ci abbia pensato mamma GNU stessa a creare una organizzazione nonprofit, Linux International appunto, dedicata a questo, dobbiamo ammetterlo, immane lavoro di traduzione e aggiornamento.

In particolare in questo sito vengono messe a disposizione della comunità UNIX internazionale una serie di liste in cui poter coordinarsi per la traduzione, una per ogni paese piú una serie di liste generiche di ``intracoordinamento''.

Visto che tradurre i messaggi (catalogues) e tradurre manpage e HOWTO sono compiti molto simili, per non creare fastidiose sovrapposizioni i compiti sono stati divisi tra due ambiti. Linux International in cui si definiscono le regole generali di traduzione e si traducono catalogues e programmi GNU ``ufficiali'', e

ILDP che invece è piú legata al mondo Linux e maggiormente dedicata alla traduzione di documentazione (HOWTO e manpage) e anche alla sua produzione direttamente in Italiano.

Piú avanti elenco le liste a cui ti puoi iscrivere per collaborare.

Ti prego di **NON**, ripeto **NON iniziare nessuna localizzazione senza prima aver sentito i ``colleghi'' di queste mailing list**, il tuo lavoro potrebbe venir vanificato in un secondo per un banale doppione.

3.5 Per maggiori informazioni

Ulteriori informazioni possono essere recuperate nei seguenti testi:

- Linux Danish/International HOWTO È stato il primo HOWTO che ha trattato il tema della nazionalizzazione di Linux, ed è tuttora un riferimento. A dispetto del nome è piú ``International'' che ``Danish'' visto che è in inglese...
- Un giro per ftp://sunsite.unc.edu/pub/Linux/utils/nls è consigliato ai programmatori. In particolare consiglio la lettura del file `locale-tutorial-0.8.txt.gz`.
- Come già detto, il sito ufficiale GNU dove potete trovare informazioni sulla localizzazione GNU e le definizioni di locale più aggiornate è ftp://dkuug.dk/i18n. In particolare se volete tentare l'installazione manuale delle definizioni di locale potete prendere il materiale che trovate a ftp://dkuug.dk/i18n/WG15-collection e, ovviamente, leggere attentamente la documentazione allegata o i succitati testi.
- Maggiori informazioni sul formato dei Portable Object e sullo stato delle traduzioni possono essere trovate al sito ftp://alpha.gnu.ai.mit.edu/gnu/po/ oppure ftp://svpop.com.dist.unige.it:/pub/Linux/gnu/po che ne è un mirror.
- Un URL mitico al riguardo, ottimo punto di partenza per ricerche sulla internazionalizzazione, è http://www.vlsivie.tuwien.ac.at/mike/i18n.html. Le pagine sono curate da Michael Gschwind, e sono una miniera di link per approfondire.

---

4. Altre localizzazioni

Qui si vogliono trattare quelle localizzazioni che sono un po' piú hardware-dipendenti, meno standard e in particolar modo che riguardano i PC compatibili. Molto probabilmente molte di queste cose non si applicano alle altre piattaforme su cui gira Linux, ne chiedo venia. Ma o mi regalate un PPC, una Sparc, un Alpha, un A4000, un MAC, ... oppure se avete altre note per queste piattaforme da aggiungere fatemelo sapere. ;-)

4.1 Font in console

Linux permette di cambiare il font standard presente nella *ROM* della scheda video, in modo da avere il set di caratteri ISO-8859-1 completo a disposizione. La cosa può essere semplicemente ottenuta con uno script del genere, da eseguire al boot (in Debian basta copiarlo nella directory `/etc/rc.boot` e renderlo eseguibile):

```
#!/bin/sh

# qualche variabile...
#
# font Latin1 (ISO-8859-1)
CONSOLE_FONT=lat1u-16.psf
# mappa nulla
CONSOLE_MAP=trivial

if [ -x /usr/bin/setfont ]; then
        echo -n "Setting up ISO 8859-1 font"
        setfont /usr/share/consolefonts/$CONSOLE_FONT
        echo -n "."
        mapscrn /usr/share/consoletrans/$CONSOLE_MAP
        echo -n "."
        for do
                echo -ne '\033(B' > /dev/tty$N
        done
        echo ". done."
fi
```

Il primo comando carica un nuovo set di caratteri, il secondo permette di eseguire una traduzione ``al volo'' tra caratteri richiesti e voluti e il terzo rende attiva la nuova coppia tabella-tavola di traslazione in ogni console.

Ad esempio di default non viene caricata nessuna tabella e viene eseguita la traduzione da Latin1 (qu che Linux vuole) a CP437 (quello che il PC ha). Ovviamente le manpages di **setfont(1)** e **mapscrn(1)** sono utili letture.

Debian

Se si ha un sistema Debian 2.0 semplicemente il file `/etc/kbd/config` è il posto dove mettere la mappa dei caratteri ed il font da utilizzare: per l'italiano, la scelta giusta è:

```
CONSOLE_FONT=lat1u-16.psf
CONSOLE_MAP=trivial
```

(da Lorenzo Maria Catucci).

SVGATextMode

Ma c'è una soluzione piú elegante. Esiste un programma che si chiama **SVGATextMode(8)** che fa tutta una serie di operazioni sulla console testo molto simpatiche.

Permette di definire dimensioni ``non standard'' per la console (diverse da 80x25, insomma), permette di alzare la frequenza di refresh del video di modo da stancare meno gli occhi, gestisce chipset avanzati...

Ha un file di configurazione, `/etc/TextConfig` che ha una sintassi molto simile al file di configurazione di XFree e che permette in un sol colpo di definire il font da caricare. Ci penserà poi il programma stesso ad attivare il font e renderlo disponibile.

Basta modificare leggermente il file, facendo in modo che si presenti all'incirca come:

```
[...]

Option "LoadFont"
FontProg "/usr/bin/setfont"
FontPath "/usr/share/consolefonts"
FontSelect "lat1u-16.psf"   8x16 9x16 8x15 9x15
FontSelect "lat1u-14.psf"   8x14 9x14 8x13 9x13
FontSelect "lat1u-12.psf"   8x12 9x12 8x11 9x11
FontSelect "lat1u-08.psf"   8x8  9x8  8x7  9x7
FontSelect "lat1u-08.psf"   8x32 9x32 8x31 9x31

[...]
```

e si ottengono magicamente i caratteri Latin1, e molto di piú.

TOC

4.2 Font in X

Buona parte dei caratteri installati con X seguono lo standard Latin1, e idem i font PostScript normalmente reperibili (come quelli dell'ATM).

Inoltre sta facendo la sua comparsa il primo font server TrueType; I font TrueType sono organizzati secondo un set di caratteri che è un superset del Latin1 (MicroSoft chiaramente fa sempre di testa sua, e mette caratteri dove lo standard lo vieta esplicitamente...), e quindi non dovrebbero dare problemi neanche loro.

4.3 Stampante e DOS

Se stampate file DVI, PostScript o comunque in grafica, non ci sono problemi. Ci penserà il vostro programma per la stampa a convertire il vostro file nel formato matrice-di-punti più consono alla vostra stampante. Ma se volete stampare in puro testo, senza formattazioni? Resta, soprattutto per le stampanti ad aghi, il metodo più veloce. Ma quanto è standard il set ISO-8859-1?

ISO-8859-1 è il set di caratteri di Unix in generale, di Windows, Amiga, OS/2. Mancano all'appello il DOS e Macintosh. Ma in DOS la CP850 è in pratica il set Latin1, con i caratteri rimescolati un po' per essere compatibile verso il basso con la CP437. Sentitevi liberi quindi da qualsiasi ``sindrome da 8 bit'', e usate tranquillamente questo set di caratteri. Se dovete importare dei testi Linux in Macintosh, semplicemente usate il filtro per ``puro testo Windows'' o cose simili.

Se volete stampare puro testo con le accentate da Linux, a questo punto non vi resta che provare una delle soluzioni seguenti:

- Se la vostra stampante supporta il set di caratteri ISO-8859-1, mettetela in quella modalità, e fatela finita una volta per tutte con le diatribe tra set di caratteri.
- Se la vostra stampante supporta la CP819, siete a cavallo visto che ``CP819'' è il nome DOS del set di caratteri Latin1.
- Se la stampante supporta la CP850, mettetela in quella modalità e cercate il programma **recode(1)**. Si può usare come `recode latin1:cp850 file` dove `file` è il file da convertire, che viene sovrascritto. Per automatizzare la procedura si può anche installare come filtro per la stampa.
- Se la stampante supporta solo CP437, o non volete/potete modificarla, non preoccupatevi, tanto comunque riuscite a stampare le cose ``normali'', come tutte le lettere accentate che trovate sulla tastiera e la lettera ``È'' (questo perché, come già detto CP850 e CP437 coincidono in corrispondenza di quei caratteri).

Chiaramente, per le ultime tre soluzioni, è il caso di utilizzare anche da DOS (se avete un sistema dual boot) la CP819 o in mancanza d'altro la CP850; per configurare la gestione del set di caratteri DOS, leggete le manpage, hem (la forza dell'abitudine! O;) il manuale dei comandi **COUNTRY** e **MODE CON**.

Inoltre se la confusione attorno alle sigle dei vari set di caratteri sale, potete anche leggere le man page **ascii(7)** e **latin1(7)**.

Un esempio di filtro

TOC

Per avere automaticamente la conversione, si può utilizzare questo filtro:

```
#!/bin/sh
if /usr/local/bin/recode latin1:cp850
then
        exit 0
else
        exit -1
fi
```

potete salvare questo filtro come `/usr/local/bin/iso2850` e poi dargli i permessi di esecuzione (`chmod ugo+x /usr/local/bin/iso2850`). ora non vi resta che aggiungere nel file di configurazione delle stampanti `/etc/printcap`:

```
# esempio di filtro, sostituire ``lp2'' con il device corretto
# i file log e acct vanno creati se utilizzati, nei contesti
# normali non serve sapere chi usa e quante pagine stampa ogni
# utente! ;)
#
cp850|dos:\
:lp=/dev/lp2:\
:sd=/var/spool/lp2:\
:lf=/var/spool/lp2/log:\
:af=/var/spool/lp2/acct:\
:if=/var/local/bin/iso2850:\
:sh:
```

Ora, se la stampante è impostata per usare la codepage 850, basta un bel `lpr -Pcp850 nomefile` per stampare un file latin1 con tutte le accentate al loro posto.

4.4 libpaper

Libpaper è una piccola libreria e un programmino che dovrebbe ``standardizzare'' la gestione dei formati di carta sotto linux. In pratica fornisce delle funzioni da linkare al proprio programma che permettono di usare delle informazioni sul formato di carta centralizzate (prese da `/etc/papersize` o dalla variabile d'ambiente PAPERSIZE.

In Debian è installata di default e gs la usa. Potrebbe essere una cosa utile, e va incentivata.

TOC

Maggiori informazioni le potete trovare alla manpage di **paperconf(1)**, ovviamente se avete libpaper installata nel vostro sistema.

4.5 Per maggiori informazioni

- Tocca ripetermi, e citare nuovamente le pagine di Michael Gschwind, ovvero http://www.vlsivie.tuwien.ac.at/mike/i18n.html.

---

5. Ispell

Esiste un vocabolario italiano per **ispell(1)** che trovate all'URL ftp://ftp.pluto.linux.it/pub/pluto/ildp/ispell/. Anche se non è ancora perfetto, è già più che funzionante e può essere tranquillamente usato.

Nella distribuzione Debian è già presente il pacchetto **iitalian** che installa e configura correttamente il vocabolario, pronto all'uso.

Se vuoi collaborare a migliorare il vocabolario o la tavola degli affissi (per sapere cos'è `man 5 ispell`), esiste una mailing list (che scherzosamente chiamo IspellIT!), che coordina gli sforzi. Trovi maggiori informazioni nella sezione dedicata alle liste.

5.1 Ispell e Emacs

Per forzare il riconoscimento del nuovo vocabolario in emacs, occorre aggiungere questo nel proprio `~/.emacs`

```
(setq ispell-dictionary-alist
        (append ispell-dictionary-alist
        `(,(list
          "italiano"
          "[A-Za-z]"
          "[^A-Za-z]"
          "[---']" t nil "~list"))))
```

---

dopodichè, M-x ispell-change-dictionary Invio Spazio ti mostra un completion buffer che contiene la magica parola, ``italiano'' (da Giacomo Boffi).

Questo però non è perfetto nel senso che ho idea che Emacs legga prima il `~/.emacs` e poi crei i menú; infatti la voce aggiunta appare in testa al sotto menú e non alla fine come vorrei io. Secondo me piú robusta sarebbe la soluzione di far modificare l'ispell-mode.el per fargli aggiungere anche l'italiano fra le lingue supportate (da Giovanni Bortolozzo).

5.2 Per maggiori informazioni

- Responsabile del progetto IspellIT! è Sandro Dentella (sandro@crcc.it), puoi contattare lui per ulteriori informazioni.
- Sembra che qualcun altro si stia divertendo con vocabolari e tabelle degli affissi, infatti c'è un

secondo vocabolario sulla cui qualità però non mi sbilancio, non conoscendolo. È possibile recuperarlo all'URL http://www.mrg.dist.unige.it/~marco/ispell.html.

- Inoltre anche un giretto per la Ispell Home Page non è una cattiva idea.

6. TeX

TeX non è una applicazione specifica per Linux, anzi, forse è una delle applicazioni utilizzabili nel maggior numero di sistemi... ma d'altro canto TeX è una delle applicazioni piú usate, e ho quindi ritenuto argomento ``critico'' il produrre documenti TeX correttamente localizzati.

6.1 Un po' di teoria...

Il TeX è in pratica un linguaggio di programmazione. Scritto un file in un determinato formato, si può compilare e quello che si ottiene è un file ``oggetto'', ovvero un file intermedio, in un formato detto *DVI*, ovvero DeVice Indipendent format, da cui si ottiene un file in linguaggio per le stampanti, di solito un file PostScript.

TeX è un ottimo programma per l'impaginazione di libri. La sua nascita nel grande mondo del free software lo ha velocemente indirizzato verso le pubblicazioni tecniche, scientifiche ma anche quelle multilingue. Inoltre permette una potente gestione degli indici e dei riferimenti, nonchè estensioni per la gestione dei riferimenti bibliografici.

Un file sorgente di TeX è un file testo (come tutti i linguaggi di programmazione) con delle istruzioni aggiuntive (che iniziano per ``\'') che spiegano al programma come organizzare l'output.

Come tutti i buoni compilatori, piano piano è stato dotato di librerie, chiamate *package* che ne estendono le funzionalità. Le più note sono il LaTeX, che definisce un insieme di stili tipografici per fare libri, articoli e lettere, con tutti gli elementi importanti di questi stili (titoli per libri e articoli, indirizzi per le lettere) già predefiniti e pronti all'uso. Esistono poi parecchie estensioni al LaTeX, le principali sono l'*AMSLaTeX*, che permette di ampliare di molto la già completa gestione di equazioni matematiche del TeX (per la cronaca, quasi tutti i libri di matematica in commercio sono fatti in TeX) e Babel, che definisce la gestione di diverse lingue, anche contemporaneamente sullo stesso documento.

Come per Linux stesso, di TeX ce n'è uno solo, ma visto che ha bisogno comunque di librerie di font, programmi accessori... di distribuzioni ce ne sono tante. Si prende in considerazione qui la distribuzione teTeX.

Il formato del file

Un file TeX è composto da un *preambolo* che definisce lo ``stile'' con cui è impaginato il testo e da un *corpo*, che comprende sia il testo che le formattazioni. Il corpo è quello che va dal comando `\begin{document}` al comando `\end{document}`, quello che c'è sopra è il preambolo, mentre quello che c'è dopo viene ignorato.

Nel preambolo vengono definite:

Classe

(o **Stile**, in LaTeX2.09) Definisce lo stile del documento, ovvero se deve essere un libro (book), lettera (letter), articolo (article), ...

Opzioni

definisce sia le opzioni *interne*, ovvero quelle delle varie classi (dimensione dei caratteri in punti, dimensione della carta, ...), sia i pacchetti (*package*) o stili aggiuntivi esterni, come il Babel stesso e il supporto per le accentate.

Altro

altre cose fanno parte del preambolo, come definizioni aggiuntive e strutture particolari, ma che ai fini del nostro studio sono inessenziali.

La definizione è quindi del tipo:

```
% Questo è il preambolo

% Questa è la definizione della classe e le sue opzioni
\documentclass[opz1, opz2, ...]{classe}

% Questo è il caricamento dei pacchetti e le loro opzioni
\usepackage[opz1, opz2, ...]{pacchetto1}
\usepackage[opz1, opz2, ...]{pacchetto2}
        .
        .
        .

% Qui inizia il testo
\begin{document}

[...]

\end{document}
```

Si vede come nel LaTeX2e ci sia separazione logica tra le opzioni dello stile, o classe, e dei pacchetti aggiuntivi, cosa preferibile visto il proliferare di nuovi pacchetti e estensioni.

6.2 L'installazione TeX di Linux

Tutta l'installazione del TeX di Linux si trova in `/usr/lib/texmf` e subdirectory.

Configurazione di teTeX

Per la configurazione di teTeX è sufficiente eseguire il programma di utilità **texconfig(1)**. Le voci da (eventualmente) modificare sono:

HYPEN

permette di modificare le tabelle di sillabazione attive: selezionata questa opzione si entra automaticamente in un editor (di solito **vi(1)**) ed occorre ``scommentare'' (eliminare il carattere di commento ``%'') di fronte alla riga `italian ithyph.tex`; dopo aver salvato il file ed essere usciti (con [Esc] [:] [w] [q] [Invio] se si tratta di vi) vengono automaticamente rigenerati i file di formato.

MODE

permette di scegliere la configurazione per la propria stampante: **ATTENZIONE** che questo non configura la stampa vera e propria, ma solo il modo di default con il quale vengono generati i font da dvips; quindi scegliete come valore la risoluzione maggiore che riuscite ad ottenere dalla vostra stampante o da tutte le stampanti che usate con il vostro computer

XDVI

permette di scegliere la dimensione di carta di default e fare delle prove di stampa e di resa; è sufficiente scegliere l'opzione ``a4''

DVIPS

permette di definire anche qui il formato della pagina, gli offset sulla pagina stessa (zone di non stampabilità), il modo e la stampante di default; permette di fare la stesa operazione per un certo numero di stampanti addizionali

Alla fine di questa semplice operazione avrete la vostra installazione di TeX perfettamente funzionante.

Configurazione delle altre distribuzioni di TeX

Per le altre distribuzioni di TeX presenti nelle vecchie distribuzioni di Linux occorre al contrario del teTeX, editare e ricompilare a mano i file di formato (piú o meno ;). Non è una operazione semplice, troverete maggiori informazioni presso la documentazione allegata alla vostra distribuzione di TeX.

NTeX

Una nota è da fare sulla distribuzione NTeX, che dalla versione 2.X.X e successive (non quindi quella contenuta nelle vecchie slackware...) contiene un programma di configurazione basato su una bella interfaccia grafica in Tcl-Tk. All'uscita dall'interfaccia grafica si ha un sistema TeX perfettamente funzionante e configurato, sillabazione compresa.

C'è da dire che le distribuzioni non usano più NTeX, probabilmente perché NTeX è cresciuta oltre modo (più di 160 MB, senza supporto per il giapponese!). Il riassunto è che, almeno fino all'uscita di teTeX 1.0 (che non ho idea come sarà strutturato), NTeX rappresenta una alternativa per esperti, che hanno bisogno di supporto per lingue poco diffuse od oscure (da Stefano Miccoli).

teTeX di Slackware 3.3

Attenzione che il teTeX fornito con la Slack 3.3 ha un'imprecisione: nel file `italian.ldf` di Babel devi definire a mano la traduzione per la parola ``Dimostrazione'' che è lasciata all'inglese ``Proof'' (se ti serve l'ambiente proof nei tuoi documenti) (da Enrico Cavalli).

La sindrome di Zio Paperone

La tabella di sillabazione italiana presente nelle distribuzioni diverse da teTeX può essere vecchia o sbagliata. Cercate nelle sottodirectory di `/usr/lib/texmf` il file `hyphen.italian` o `ithyph.tex`. Se è già presente, controllate (è un file di testo, usate pure **less(1)**!!!) che NON sia la tabella di sillabazione di G. Patergnani. Questa soffre di quella che io scherzosamente chiamo ``sindrome di Zio Paperone'', ovvero tende a mandare a capo malamente i suoi nipoti (qu-i, qu-o, qu-a).

Altrimenti, se la tabella non esiste o è quella del Patergnani, prendete il file aggiornati dai siti CTAN, uno per tutti ftp://ftp.dante.de/.

6.3 Produzione di elaborati TeX ben localizzati

Ora, dopo tutte queste disquisizioni, abbiamo capito che c'è bisogno di:

A4

Ovvero per stampare decentemente sulla nostra stampante.

InputEnc

Ovvero il supporto per le lettere accentate direttamente all'interno del file LaTeX, per evitare di ricorrere a pesanti sequenze di comandi.

Babel

Per il supporto della sillabazione italiana e di altre cose simpatiche senza stravolgere completamente la configurazione del TeX.

Stampa di testi in inglese

Piacerebbe avere l'opzione della dimensione del foglio A4 definita direttamente a livello di TeX, visto che è il nostro standard. Questa operazione, anche se possibile, è sconsigliata. In TeX la formattazione di certi documenti dipende fortemente da margini e dimensione della carta. Cambiare uno di questi parametri senza rivedere il layout del documento potrebbe portare a delle emerite schifezze (figure sovrapposte, ...). Quindi, se vogliamo stampare dei testi in inglese, dobbiamo reperire i sorgenti in TeX, editare il file, aggiungendo nel preambolo l'opzione:

```
\documentclass[opz1,opz2,...,a4paper]{classe}
```

---

e magari controllare con un previewer come **ghostview**, prima di stampare, che l'output sia accettabi

TOC

Composizione di testi in italiano

Se invece stiamo costruendo un nuovo testo, e vogliamo che tutto funzioni alla perfezione, è sufficiente che il preambolo sia del tipo:

```
\documentclass[opt1,opt2,...,italian,a4paper]{classe}
\usepackage{babel}
\usepackage[latin1]{inputenc}
```

È bene mettere l'opzione ``italian'' come globale, visto che potrebbe essere utilizzata da altri pacchetti.

Inoltre per un maggiore ordine nella sillabazione, e molti altri buoni motivi (tra cui le guiellmots «») è bene adoperare anche un:

```
\usepackage[T1]{fontenc}
```

che permette di utilizzare i nuovi font ``ec'' in luogo dei vecchi ``cm'' di Knuth (da Lorenzo Maria Catucci).

6.4 Per maggiori informazioni

Se avete intenzione di stampare semplicemente dei file TeX, queste conoscenze sono sufficienti.

Ulteriori informazioni possono essere reperite nella (nutrita!) documentazione che viene allegata alla distribuzione teTeX, e che potete trovare nella directory `/usr/lib/texmf/doc` oppure facendo riferimento al teTeX HOWTO, anche in versione italiana.

Se invece avete intenzione di proseguire nella conoscenza del TeX, sono consigliabili, in particolare, i libri:

- *Donald Knuth*, **The TeXbook**, Addison-Wesley, 1992
- *Leslie Lamport*, **LaTeX User's Guide and Reference Manual**, Addison-Wesley, 1994
- *Helmut Kopka*, *Patrick Daly*, **LaTeX Guide**, Addison-Wesley, 1994
- **AMS-LaTeX Version 1.1 User Guide**, American Mathematical Society, 1991
- *Norman Walsh*, **Making TeX Work**, O'Reilly & Associates
- *Tobias Oetiker*, **The not so short Introduction to LaTeX2e (or LaTeX2e in 69 minutes)** ovvero il file `/usr/lib/texmf/doc/latex/general/lshort2e.dvi`, da cui si impara piú che dal Lamport, risparmiando giorni di studio (da Lorenzo Maria Catucci).

E in particolare in italiano...

- *Gianni Gilardi*, **Il TeX, Introduzione al linguaggio e complementi avanzati**, Zanichelli/Dec
- *Claudio Beccari*, **LaTeX, Guida ad un sistema di editoria elettronica**, Hoepli Milano
- *Marc Baudoin*, **Impara LaTeX!**, http://yamaha.cefriel.it:8080/~cannarsi/documents.html Si tratta di una traduzione di ``Apprends LaTeX!", realizzato da Marc Baudoin e in uso in numerose scuole francesi nonchè in vari laboratori di ricerca. Ha il pregio, oltre all'essere di libero dominio e in italiano, di permettere un apprendimento molto rapido di Latex e cominciare da subito a scrivere i propri documenti, come l'autore ed io abbiamo avuto modo di constatare dal 94 (prima versione) ad oggi presso l'Ecole nationale superieure de techniques avancees. È anche possibile scaricare la versione PostScript al sito: ftp://ftp.pluto.linux.it/pub/ildp/misc/impara_latex/.
- *Davide G. M. Salvetti*, **TEX.ITA FAQ**, http://space.tin.it/io/dsalvett/TeX-FAQ/TeX-FAQ.html Sono le FAQ dell'area UNIX.ITA di rete FidoNet, una ottima FAQ e un buon punto per iniziare successive ricerche.
- Esiste anche questo link, http://www.fi.infn.it/pub/tex/doc/html/latex.html, anche se non ho avuto modo di vederlo e valutarne la qualità.

---

7. Siamo in Europa, bellezza!

Il che comporta che a breve dovremmo ``scalciare'' via dalla tastiera e dai nostri font alcune simboli per fare spazio ai nuovi arrivati (il simbolo dell'euro e dei centesimi di euro, in particolare).

Già che c'erano, sembra si stiano mettendo d'accordo (a livello di ISO, International Standard Organization) per fare un unico font che vada bene a tutte le nazioni europee, comprese quelle che dovrebbero entrare a breve. Si vocifera di un nuovo font ISO-8859-15 (soprannominato Latin0), e quindi è probabile che tra un po' toccherà cambiare tutto...

Ma noi aspettiamo fiduciosi! ;)

7.1 Per maggiori informazioni

Questa sezione dovrebbe espandersi sempre piú, nei prossimi anni; per ora non mi sbilancio e lascio solo dei riferimenti ad altri documenti.

- Si possono trovare informazioni sul nuovo font ``europeo'' all'URL http://www.iso.ch/cate/d29505.html.
- Altre informazioni sui set di caratteri e affini possono essere recuperate all'URL http://czyborra.com/charsets.
- Per il TeX appaiono i primi font con il simbolo, basta dare una occhiata a ftp://ftp.dante.de/tex-archive/fonts/euro.

8. TimeZone

La *TimeZone*, ovvero il fuso orario corretto per l'Italia è GMT-1 d'inverno e GMT-2 d'estate, ovvero GMT-1 per l'ora **solare** e GMT-2 per l'ora **legale**. Oppure MET o CET (Middle o Central Europe Time), che in aggiunta fanno anche il passaggio automatico ora legale/ora solare. Notare che MET è il vecchio nome caduto in disuso, e non dovrebbe essere usato.

Ogni distribuzione ha il suo programmino per definire la timezone in maniera automatica. In Debian solitamente viene chiesto in fase di installazione, e non si può sbagliare (l'unica opzione utilizzabile è Europe/Rome, evidentemente, e per fortuna, in Debian sono impermeabili alle ultime mode secessioniste ;).

Di solito negli ambienti UNIX si preferisce impostare l'ora del CMOS a GMT e quindi avvisare **clock(1)** di questo con l'opzione ``-u "; questo però di solito entra in conflitto con la gestione dell'ora di Windows, che invece insiste a modificare brutalmente l'ora CMOS.

Vedete voi cosa è meglio fare, Debian in fase di installazione chiede quale delle due modalità usare di default e configura di conseguenza. Come sempre `man clock` per saperne di piú.

8.1 Q: ma perché Linux mi ha sbagliato il passaggio di fuso?!

Faccio notare che nel '96 sono cambiate le regole per la determinazione del passaggio ora legale/ora solare. Le nuove regole dovrebbero essere presenti nelle nuove distribuzioni, o comunque reperibili in rete.

8.2 Q: ma perché ho l'ora impostata a CEST?

Perché è estate, CEST sta per ``Central Europe Summer Time". O meglio, è in vigore l'ora legale, ma è piú bello pensare all'estate!

9. Documentazione

9.1 Pubblicazioni offline (su carta... ;)

Questa vuole essere una piccola bibliografia per chi inizia a conoscere Linux.

Per l'utente medio sono più che sufficienti i testi liberamente reperibili in rete, come i libri del LDP e gli HOWTO. Per un utente più attento può essere utile un approfondimento, anche perché buona parte degli argomenti su Linux riguardano UNIX in generale, come l'amministrazione di sistema, la gestione di una rete e via dicendo.

Ho rimosso la bibliografia in inglese, questo perché la sentivo poco consona a questo HOWTO; e poi oramai non è cosí difficile trovare un buon testo su UNIX/Linux in Italiano. Se siete alla caccia di una buona bibliografia su UNIX potete dare una occhiatina alla FAQ dell'area di discussione di FidoNet UNIX.ITA nel sito http://www.fidoitalia.net/echo/faq/unix/indice.htm, oppure se cercate riferimenti a tutti i libri su Linux potete dare una occhiata anche a http://www.psci.net/~dbklem/books.html o anche a http://shell.nanospace.com/~markst/linux/.

Difficile fare invece un ``censimento'' di tutto il materiale reperibile in rete. Ho cercato, soprattutto, di mettere le fonti di informazioni più importanti, o quelle più vicine a noi.

Questo elenco non aspira a essere completo, tutt'altro. Come sempre sono felicissimo di ricevere suggerimenti, correzioni e aggiunte.

Libri

- *D. A. Tauber*, **Guida a Linux**, McGraw-Hill Italia, ISBN: 88-386-3406-8. Contiene un CD-ROM.
- *Giuseppe Zanetti*, **LINUX - il sistema operativo free software per personal computer 386/486**, edizioni Libreria Progetto, Padova, 1996. Poco più di 300 pagine, simpatica copertina a colori col logo di Linux creato da Rohde-Hass. Se volete contattare l'autore, scrivete a beppe@freddy.iperv.it. Ne è uscita una nuova versione che ho solo visto di sfuggita, ma che è visibilmente piú grossa della precedente; infatti è stato aggiunto un corposo capitolo sul TCP/IP.
- *Luca Righi*, **LINUX**, Jackson, 1996. Contiene un CD-ROM.
- *Naba Barkakati*, **I segreti di Linux**, Apogeo, 1996, 800 pp., ISBN: 88-7303-191-9. Contiene un CD-ROM con Linux Slackware 3.0 e altre utility.
- *Richard Petersen*, **Guida completa a Linux**, McGraw Hill, 1996, 721 pp. È una guida completa completa! In allegato CD-ROM con distribuzione RedHat e Caldera Network Desktop. ;-) (da Daniele Sanna)
- *Baffa, Calculli*, **Internet: manuale avanzato**. Un buon libro che, a dispetto del titolo, parla quasi solo di Linux, e include un CD-ROM con la distribuzione Slakware. (da Gianni Comoretto)
- *Olaf Kirch*, **Linux guida per l'amministratore di rete**, Jackson Libri. É la traduzione del libro O.

Kirch, Linux Network Administrator's Guide, O'Reilly & Associates. Eccezionale. (da Ermenegildo Colpo)

- *M.Welsh, L.Kaufman*, **Il Manuale di Linux**, Jackson Libri. Traduzione del ``Running Linux'' di Welsh-Kaufman (da S. Crespi Reghizzi)
- *Progetto ILDP*, **Linux HowTo**, Apogeo, 1996, pp 960, ISBN 88-7303-317-2. I piu' importanti Howto di Linux interamente in italiano, tradotti (buona parte) e revisionati (tutti) dal gruppo Pluto.
- *David Pitts*, **RedHat Tutto e Oltre**, Apogeo e SAMS Publishing. Il libro fornisce informazioni su tutto ciò che è il pacchetto Linux e suoi svariati utilizzi, trattando dettagliatamente tutto ciò che riguarda la reti ed internet in particolare. Consigliato a chi si avvicina a Linux per la prima volta, ed anche ai più esperti, per la precisione e la completezza con cui vengono trattati i vari argomenti, a mio giudizio un ottimo manuale di riferimento. Il linguaggio è di facile comprensione, aiutato da esempi e parti di codice ben spiegate (da Osvaldo Tulini).
- *Witherspoon, Hall*, **Linux For Dummies**, Apogeo. Linux For Dummies è al contrario un ottimo testo che spiega, in termini semplici e con un ``piglio'' tipicamente da bigino (chi non ne ha mai usato uno a scuola?), tutto quello che c'è da sapere per far funzionare il sistema operativo in maniera decente. L'utente inesperto, che grazie a quel libro riuscirà a muoversi da solo nel sistema operativo, potrà poi approfondire gli argomenti più delicati (da Paolo Besser).
- **Linux For Dummies - Espresso**, Apogeo. Veloce prontuario per le operazioni piú semplice e frequenti (da Fabio).

Riviste

Sono ormai molte le riviste di informatica che hanno una rubrica fissa su Linux o che distribuiscono dei CD con materiale Linux; molto spesso è il metodo piú semplice per venire in possesso di un CD con Linux.

Non sono abbonato a nessuna di queste riviste, quindi le informazioni che sono qui presenti sono, per forza, approssimative; ricevo volentieri, quindi, integrazioni.

- **Dev**, Infomedia. Ha una rubrica fissa o comunque pubblica in ogni numero articoli su Linux, oltre a CD (da Marco d'Itri).
- **Login**, Infomedia. La rivista esiste da circa due anni e mezzo, e ogni anno pubblicano 3 cd, di cui uno è sempre dedicato ad una distribuzione Linux (da Marco Marongiu).
- **Computer Programming**, Infomedia. Spesso sono presenti articoli che trattano problemi di programmazione sotto Linux/UNIX.
- **MC MicroComputer**, Technimedia. Da sempre ha avuto ``l'AngoLinux'', ora la rubrica è stata ampliata (da Dario Burnelli).
- **Computer Shopper**. È in edicola una nuova rivista (COMPUTER SHOPPER, in italiano) nella quale scrivo ogni mese una rubrica tutta per Linux (dovrebbe partire da zero con l'installazione per trattare via via argomenti piú tecnici). Se qualcuno l'ha avuta tra le mani, sappia che sono sempre interessato a ricevere commenti e soprattutto critiche sui pezzi (da Ubi, ubi@srd.it).
- **Inter.Net**. Pubblica spesso CD con distribuzioni Linux e altre cose che è un po' difficile reperire altrimenti.

- **IoProgrammo**. Idem a sopra, ogni tanto pubblica CD con materiale Linux.
- **PC Inter@ctive**, Mondadori Informatica. Abbiamo iniziato a dedicare uno spazio fisso a Linux sul nostro giornale: un esperimento, vediamo un po' le reazioni dei lettori (da Marco Mazzei).
- **PC Action**, Xenia On Line. Scrivo sulle riviste del gruppo Xenia Edizioni da molti anni ormai e, dopo mesi di trattative e rimandi, sono finalmente riuscito ad ``accaparrarmi'' un po' di spazio su PC ACTION e sul relativo CD da dedicare a Linux. Ogni suggerimento è ovviamente ben accetto, mandatemelo per favore alla casella di posta personale paolone@besser.it (da Paolo Besser).

9.2 Pubblicazioni online

All'inizio c'era il nulla. Poi pian piano, i volenterosi, hanno cominciato a pubblicare delle FAQ (ovvero a raccogliere le domande fatte più spesso in modo che non siano più poste...). Da questo stadio poi è partito un vero e proprio ``progetto editoriale'' su Linux.

Ora, dopo parecchi anni, il ``panorama'' editoriale online su Linux è ricco e ben organizzato; punto di riferimento è **LDP**, ovvero **Linux Documentation Project**, il progetto internazionale di documentazione su Linux che ha realizzato un grande numero di documenti, dai veloci mini-HOWTO, ai seri HOWTO, fino a veri e propri libri come le guide. Tutto il materiale LDP è liberamente reperibile in rete. Il sito madre, come per molto altro materiale Linux, è sunsite. Più precisamente il materiale è disponibile in versione HTML online all'indirizzo http://sunsite.unc.edu/LDP/, e negli altri formati all'url ftp://sunsite.unc.edu/pub/Linux/docs/. Molti siti, anche italiani, fanno il mirror di sunsite; ne trovate alcuni in fondo a questo HOWTO.

Italian Linux Documentation Project

Con molta fantasia si chiama *ILDP*, ovvero *Italian Linux Documentation Project* il progetto, lanciato dal gruppo Pluto, di tradurre gli HOWTO. Partiti da questa idea iniziale, si è poi fatto molto di piú arrivando a tradurre le guide e perfino a creare dal nulla ottima documentazione.

- Sono stati tradotti un buon numero di **HOWTO** e ormai si è nella fase di ``mantenimento'' del materiale tradotto. Il progetto ILDP comunque cerca volontari per tradurre e mantenere aggiornati gli HOWTO. Se vuoi collaborare, contatta il coordinatore Giovanni Bortolozzo (borto@pluto.linux.it).
- Sono state tradotte una quantità consistente di manpage, e anche qui si è ormai nella fase di mantenimento, almeno per quanto riguarda le sezioni ``critiche'' (1, user commands). Anche qui se vuoi collaborare contatta Alessandro Rubini (rubini@pop.systemy.it).
- Sono state tradotte la Guida dell'Utente e si sta lavorando sulle altre guide, che trovate su http://www.pluto.linux.it/ildp/guide.
- È stata creato ex-novo AppuntiLinux.

Per collaborare puoi inoltre iscriverti alle liste di discussione, in particolare ``it'' e ``pluto-ildp''.

Tutto il materiale tradotto è reperibile online all'indirizzo http://www.pluto.linux.it/ildp/ e negli altri formati all'indirizzo ftp://ftp.pluto.linux.it/pub/ildp/, o comunque in uno dei numerosi mirror di questo

sito che stanno nascendo come funghi.

AppuntiLinux

AppuntiLinux è un ottimo esempio di documentazione autoprodotta, e rappresenta un testo da consigliare vivamente a chi si avvicina per la prima volta a Linux, visto che si configura come un ``corso'' che aiuta a muovere i primi passi in Linux.

Potete trovare AppuntiLinux negli altri formati a questo URL.

Riviste/fanzine

Sono riviste elettroniche (fanzine, da fan+magazine, giornaletti autoprodotti ``in casa'') di ottima qualità (mediamente), ed alcune con una consolidata tradizione alle spalle.

- **PLUTO Open Journal** , Pluto User Group. Il **PLUTO Open Journal** è una rivista interamente in italiano e completamente gratuita. La rivista tratta argomenti inerenti Linux ed è completamente fatta da volontari di tutta Italia. È consultabile on-line all'indirizzo URL: http://www.pluto.linux.it/journal e tutti i mirror PLUTO. È reperibile anche in versione .tar.gz per una comoda lettura off-line all'url ftp://ftp.pluto.linux.it/pub/pluto/journal oppure (con una settimana di ritardo ;) in f'req al nodo FidoNet 2:333/1016 (con estensione .tgz invece che .tar.gz per compatibilità con i sistemi DOS.
- **Linux Gazette Edizione Italiana**, LUGBari. LGEI, come tradisce il nome, è una fanzine su Linux in italiano che contiene per la maggior parte (ma non solo) articoli che sono una traduzione di quelli della Linux Gazzette statunitense. È reperibile all'URL http://www.officine.it/lgei/.
- **LinuxFocus**. All' URL http://mercury.chem.pitt.edu/~angel/LinuxFocus potete trovare l'edizione italiana del magazine on-line ``LinuxFocus'' (da Seipse).
- **AscoLinux Journal**, LUG Ascoli. Il LUG Ascoli ha pubblicato sul Web il proprio giornale all URL http://www.rinascita.it/ascolinux. Trattasi di AscoLinux Journal, una raccolta di articoli scritti in maniera molto ``leggera'' (con contenuti elementari, direi) da noi per far avvicinare i concittadini alle iniziative del LUG. L'idea che vogliamo realizzare è quella di fare in modo che la gente, anche se non esperta di informatica, possa capire che cosa è possibile fare con linux, e in che modo si può fare!
- **LinuxValley Journal**, LinuxValley. Altra fanzine, nata dallo scisma all'interno di LGEI; è reperibile all'URL http://www.linuxvalley.com/lvj.

9.3 Distribuzioni in Italiano

Ebbene sí, si può cominciare a parlare di distribuzioni in Italiano!!!

Ma sfogo la mia amarezza dicendo che solo Debian ha un approccio ``serio'' alla localizzazione di una distribuzione... Mentre Debian ha creato un gruppo per l'internazionalizzazione e da questo dei gruppi locali uno per ogni lingua, e sta cercando di fornire una localizzazione completa (programma di installazione, documentazione, cataloghi dei programmi, vocabolario per ispell, ...) le distribuzioni

commerciali spesso considerano ``distribuzione in Italiano'' una distribuzione in inglese, con localizzazione mancante o non funzionante, il programma di installazione con le descrizioni dei pacchetti in italiano e un manuale (sempre di installazione) in italiano.

Contando che tutto il materiale italiano inserito in Debian è PLUTO o GNU, non c'è scusante a questa, per me barbara, politica...

- **Slackware**, Infomagic. pubblica in Italia un cofanetto con dei CD che ha la copertina in Italiano, e mi sembra il (breve) manualetto di installazione all'interno pure. Non so che cosa ci sia in italiano oltre a queste cose, e chiedo conferme.
- **Debian**, Debian. Oltre alla Debian ``pulita'', che già contiene molte cose in Italiano (manpage, gli HOWTO, i PlutoJournal e il vocabolario per ispell), da una collaborazione tra ProSA S.r.L e Athena S.r.L è nata ``Athena Debian 2.0 Linux - powered by ProSA'', una Debian 2 con installazione parzialmente semplificata e in italiano, che verrà commercializzata con una copia su carta degli AppuntiLinux. Per maggiori informazioni fate riferimento ai due URL qui sopra.
- **Caldera**, Caldera Inc.. tra poco dovrebbe uscire una versione di Caldera Open Linux Base con installazione e manuale in italiano, e tra un po' dovrebbe essere prodotto un corso interattivo (tutorial) in italiano.
- **SuSE**, SuSE. Dopo la mitica Debian Pluto in Italiano è uscita la prima distribuzione commerciale in Italiano si tratta della ormai famosa SuSE 5.2 (falso, la prima è la Caldera, ndr). Mi è arrivata ieri e ho prontamente installato, naturalmente il supporto italiano è limitato all'installazione e al sistema di gestione denominato Yast. Ci sono dei curiosi errori di traduzione, ma mai quanti quelli di sistemi piú blasonati quali AIX (hard disk = disco duro :-) (da Gabriele Paciucci). Ha un manuale in Italiano un po' ridotto (spiega l' installazione e poco altro), e poi c'è la guida completa in Inglese che ti spiega tutti i perché e i per come. (da un anonimo S.u.S.E. Linux 5.1 User).

9.4 Altro materiale in Italiano

Fortune in Italiano

Esiste una collezione di ``fortune'' in italiano; le fortune sono frasi che vengono stampate automaticamente nella fase di login dal programma **fortune(1)**.

Maggiori informazioni si possono avere al sito http://www.pluto.linux.it/~zuse/fortune.it o scrivendo all'autore Andrea `Zuse' Balestrero.

Linux FAQ

Con questa guida mi propongo di riunire all'interno di un unico documento in Italiano, informazioni che possano essere utili a coloro che vogliono avvicinarsi a questo ottimo sistema operativo e nello stesso tempo fornire delle notizie che possano essere utilizzate dai system integrator per giustificare la scelta di Linux all'interno di un'azienda.

Una versione piú o meno aggiornata di questa guida è disponibile nei newsgroup it.faq e it.comp.linux e su http://www.pluto.linux.it/ildp/FAQ (da Marco Iannacone).

10. Linux in Italia

Questo spazio vuole essere un tentativo di censimento della realtà Linux in Italia, un tentativo perché questa è una realtà in velocissimo movimento...

10.1 Associazioni, gruppi di utenti

Il mondo Linux in Italia sta subendo una crescita impressionante, cosa che stupisce anche noi che... in queste cose ci crediamo!

Oltre a ILS (che è una associazione vera e propria, con statuto, assemblea e quota associativa) e il PLUTO (che è il gruppo piú noto e attivo, gruppo che coordina le traduzioni e pubblica il PlutoJournal tra le altre cose...) esistono una miriade di LUG locali sparsi per tutta Italia...

PLUTO

Il Pluto, definito recursivamente come ``Pluto Linux/Lumen Utentibus Terrarum Orbis'', è nato nel 1992 per iniziativa di alcuni studenti e professori di Informatica dell'Università di Padova.

Da allora il Pluto è cresciuto in maniera esponenziale estendendosi su tutto il territorio nazionale ed oltre, e costituisce il piú grande ed attivo LUG (Linux User Group) di lingua Italiana.

Lo scopo del Pluto è di diffondere l'uso del Sistema Operativo Linux attraverso l'opera volontaria e non retribuita dei suoi membri/soci. Per conseguire il suo scopo, il Pluto:

- Sostiene e diffonde lo sviluppo del Software Free e della sua filosofia.
- Produce, traduce e diffonde la documentazione di Linux in lingua Italiana.
- Promuove corsi, dimostrazioni e manifestazioni per favorire la conoscenza di Linux.

Ogni utente di Linux, senza restrizioni alcune, che si impegni a partecipare attivamente allo scopo del Pluto, diviene per questo membro del Pluto.

Il Pluto è una organizzazione ``virtuale'' che vive in uno o piú siti connessi ad Internet. La comunicazione tra i membri avviene primariamente attraverso i mezzi elettronici messi a disposizione dalla sede, il newsgroup Usenet it.comp.linux.pluto, le mailing list pluto-*, canali IRC, pagine del World Wide Web, ecc.

Maggiori informazioni possono essere reperite alle pagine ufficiali del PLUTO all'indirizzo http://www.pluto.linux.it/.

TOC

ILS

ILS è un'associazione culturale senza fini di lucro, che ha fra i suoi scopi primari la diffusione della cultura informatica in tutte le sue forme, con una particolare attenzione per il mondo del free software in generale e del software che viene rilasciato sotto la GNU General Public License. ILS è composta da tutti coloro che volontariamente e senza percepire alcun profitto mettono a disposizione il proprio tempo, le proprie capacita e le proprie risorse al fine di realizzare una comunità virtuale basata sulla cultura informatica/telematica. ILS offre ai Soci i servizi di Internet e-mail e Usenet News, che sono compresi nella quota associativa annua di Lire 50.000. Per maggiori informazioni chiamare la BBS della ILS al numero 019-862349, oppure mandare una mail a info@linux.it.

Si può anche fare riferimento per maggiori informazioni a http://www.linux.it/.

Altri LUG locali

Esiste la mappa italiana dei LUG, ovvero delle pagine che contengono riferimenti a tutti i LUG nati qua e la in Italia.

Consiglio a tutti quelli che hanno creato un LUG, o che ne hanno l'intenzione, di chiedere al manutentore (Lindo Nepi, ascoli@ascu.unian.it) di essere aggiunti a questa mappa. Consiglio inoltre di iscriversi alla lista ``nuovi-lug'' per il coordinamento tra LUG e per lo scambio di esperienze, trovate le informazioni nella sezione dedicata alle liste di discussione.

10.2 Aree di discussione pubbliche

Newsgroups

comp.os.linux*

Ovvero **I** newsgroup su Linux. Ultimamente sono diventati supertrafficati, e la qualità non è esaltante (per la maggior parte si cercano rimedi per qualche pezzo di hardware che non funziona). Da seguire assolutamente comp.os.linux.announce, che ``lancia'' tutti i nuovi software per linux.

it.comp.linux

Newsgroup italiano ufficiale su Linux. Valgono i discorsi fatti per la precedente gerarchia, con l'aggiunta della disponibilità tutta italiana (e in italiano) ad aiutare chi si affaccia per la prima volta nel grande mondo Linux.

it.comp.linux.pluto

Gli argomenti trattati dovrebbero essere attinenti a tutto ciò che riguarda il gruppo PLUTO.

it.comp.linux.setup

Discussioni sulla configurazione e installazione di un sistema Linux.

it.comp.linux.annunci

Annunci importanti per la comunità Linux italiana.

Mailing List

TOC

Le mailing list sono dei luoghi virtuali in cui è possibile confrontarsi e discutere, un po' come i newsgroup. Quello che veramente li distingue da questi è il fatto che è possibile avere un controllo piú stringente sull'accesso, e quindi evitare maggiormente il rumore di fondo.

Per iscriversi a queste liste è sufficiente mandare un messaggio all'utente ``majordomo'' dell'indirizzo a cui appartengono le liste e inserire nel corpo del messaggio una riga con:

```
subscribe nomelista email
```

---

dove ``nomelista'' è il nome della lista (quello che c'è a sinistra del simbolo ``@''), e ``email'' il vostro indirizzo email.

pluto-amiga@pluto.linux.it

Linux e m68k, ovvero Linux su Mac, Atari, Amiga...

pluto-arch@pluto.linux.it

coordinamento progetti amiga, i386 e Sparc

pluto-devel@pluto.linux.it

i **Pluto Developers**, ovvero lista in cui si discute sul free software e si sviluppa free software

pluto-fido@pluto.linux.it

Linux e il mondo della FidoTech, ovvero come mettere su una BBS sotto Linux e conquistare il mondo (della FTN)

pluto-humor@pluto.linux.it

per discussioni poco serie

pluto-i386@pluto.linux.it

Linux su macchine PC, per ora discussioni su cosa far fare al proprio PC con linux...

pluto-ildp@pluto.linux.it

maliling list ufficiale del progetto ILDP; occorre iscriversi a questa lista se si ha l'intenzione di tradurre qualche HOWTO o documento simile.

pluto-journal@pluto.linux.it

lista di coordinamento/redazione virtuale del Pluto Open Journal

pluto-meeting@pluto.linux.it

per organizzare il meeting pluto

pluto-ong@pluto.linux.it

Linux on the world, ovvero linux come strumento di networking a basso costo per i PVS

pluto-sparc@pluto.linux.it

Linux su Sparc

pluto-soci@pluto.linux.it

per tutti i soci pluto; questa lista è volutamente diffusa sotto altre forme in diversi net amatoriali e anche altro per permetterne la massima accessibilità.

pluto-scical@pluto.linux.it

Linux e calcolo scientifico

it@li.org

lista per l'Italia del progetto Linux International; occorre iscriversi qui se si ha l'intenzione di tradurre qualche catalogo di messaggi, oppure qualche programma GNU

ispell-it@arci01.bo.cnr.it

lista per la produzione di un vocabolario in italiano per ispell

Per queste altre, che utilizzano il software SmartList, occorre invece mandare un messaggio all'indirizzo della lista con un ``-request'' aggiunto prima del segno ``@'', e inserire nel corpo del messaggio una riga con un semplice:

```
subscribe
```

---

scuola@linux.it

Linux e il mondo della scuola

com@linux.it

Linux e il mondo commerciale. Questa mailing list è per tutte le ditte che vendono prodotti per Linux, consulenza su Linux, sistemi Linux e cose simili.

linux-isp@linux.it

Uso di Linux da parte di provider e simili.

debian-it@linux.it

doppione locale della precedente

nuovi-lug@linux.it

Lista di coordinamento tra i LUG Italiani, e non solo per quelli nuovi (come sembra dire il nome).

debian-italian@lists.debian.org

lista ufficiale degli utenti Debian italiani

linux-italia@lists.cosmos.it

Lo spirito della Mailing List è molto differente dal solito accademico ``RTFM'' (Read The Fucked

Manual): a ogni domanda una risposta. Chi non sa chiede, chi sa mette a disposizione degli alt proprie conoscenze.

Infine per queste occorre mandare un messaggio all'utente ``listserv'' del dominio a cui appartengono le liste e inserire nel corpo del messaggio una riga con:

```
subscribe nomelista Nome Cognome
```

dove ``nomelista'' è il nome della lista (quello che c'è a sinistra del simbolo ``@''), e ``Nome Cognome'' il vostro nome e cognome.

linux-it@nis.garr.it

Mailing list generale su linux.

FidoNet

FidoNet è la mamma di tutte le reti in tecnologia (appunto) FidoNet, ed è la piú grossa rete amatoriale oggi esistente; veicola delle aree di discussione che sono, ai fini pratici, equivalenti ai newsgroup come tecnologia, ma molto piú simili alle mailing list come qualità dei contenuti.

Per maggiori informazioni vi rimando a http://www.fidoitalia.net/ oppure alle aree locali di ogni buona BBS.

UNIX.ITA

L'area UNIX.ITA tratta di tutto quello che riguarda i sistemi operativi *IX, sia come filosofia generale che come applicazione a sistemi particolari, compresi - ma non limitati a - AIX, A/UX, Coherent, ESIX, Interactive, Linux, Minix, NetBSD, SCO-UNIX, SunOS, Ultrix, Xenix. È tollerato anche parlare dei vari pacchetti software che storicamente sono associati all'idea di ``ambiente aperto'', e che sono disponibili in sorgente, come ad esempio i programmi GNU. **In via sperimentale**, è permesso parlare di programmazione di rete Unix (leggasi socket, RPC e simili). L'area UNIX.ITA **non** tratta di TeX (esiste l'area TEX.ITA al proposito) e soprattutto **non tratta** di ``Internet at large'' e affini. Per queste tematiche esiste l'area COMMS_RETI.ITA. Si invitano gli abituali frequentatori di UNIX.ITA a non rispondere in quest'area ai messaggi di questo genere: molto meglio una risposta in matrix comprendente anche l'invito a linkare l'altra area in questione. La parte di Internet che riguarda l'area comprende essenzialmente i server http (non i lettori) e i protocolli http, ftp e simili (non HTML).

Moderatore: **Marco d'Itri**, Fidonet: 2:332/206.10, md@linux.it

TEX.ITA

I temi della conferenza sono il linguaggio per la creazione di documenti TeX (con l'estensione LaTeX) ed il linguaggio per la creazione di font METAFONT, che a TeX è direttamente correlato. SGML, che è, a titolo sperimentale, ammesso come tema di discussione in area (HTML escluso a meno di autorizzazione) e sulla possibilità di discutere di applicazioni correlate a (La)TeX, ad esempio AUC TeX e cosi` via.

Moderatore: **Paolo Cascini**, FidoNet: 2:332/101.12, laurgeo@udini.math.unifi.it

COMMS_RETI.ITA

Qui si tratta di reti geografiche e metropolitane, dei loro contenuti, strumenti protocolli e modalità di accesso. L'argomento può portare facilmente, data la natura della principali di queste reti, internet, a parlare di sistemi UNIX. Si ricorda che per questo argomento esiste un'area nazionale apposita. Anche per la discussione sulle reti locali esiste un'altra area dedicata. Si invitano gli utenti a scrivere qui solo di reti geografiche e metropolitane e a spostare in matrix ogni comunicazione non attinente il soggetto della conferenza (ringraziamenti, saluti, insulti e divagazioni). Qualora un nuovo utente ponga una domanda a cui si è già più volte risposto in area o/e di cui si ritiene nota ai frequentatori la risposta, si prega di soddisfarne la richiesta in matrix e di inviare, sempre per questa via, copia della risposta al moderatore che provvederà a inserirla nelle FAQ dell'area (è un buon proposito).

Moderatore: **Federico Baraldi**, Fidonet: 2:332/414.6, Federico.Baraldi@p6.f414.n332.z2.fidonet.org

PLUTO.ITA

I Plutini in FidoNet

Questa lista vuole essere punto di incontro dei soci PLUTO in FidoNet, ed infatti è posta in gateway con la lista pluto-soci@pluto.linux.it.

UNIX_BBS.ITA

Linux alla conquista delle BBS

In quest'area si dovrebbe discutere dell'uso di UNIX (e quindi Linux) come software di BBS/gate.

LINUX_BBS

Linux alla conquista delle BBS (internazionale)

Idem a sopra, solo che quest'area è internazionale, e sta già producendo FAQ a manetta.

LINUX

International LINUX Conference

Area internazionale ufficiale su Linux. Niente di speciale, a parte il fatto che si leggono i messaggi provenienti da Sud America e Giappone qualche settimana dopo che sono stati spediti...

UNIX

International UNIX Conference

Idem a sopra.

SPUTNIK.500

Area del LUG Siciliano

Area locale all'HUB 500 (Sicilia) su Linux creata appositamente come area di supporto per il

gruppo SPUTNIK.

PNet

Rete ``alternativa'' a FidoNet, piú piccola (è solo nazionale) ma non meno tecnicamente preparata.

Per maggiori informazioni vi rimando a http://www.retepnet.it/.

P_UNIX

Linux, Unix e affini

Discussioni sul sistema operativo Linux (software, hardware, compatibilità, sorgenti, etc) e confronto con altri sistemi operativi Unix-like

P_PLUTO

PLUTO in PNet

gate con la mailing lists pluto-soci@pluto.linux.it

P_PLFTN

PNet, PLUTO e le BBS UNIX

gate con la mailing lists pluto-fido@pluto.linux.it

EuroNet

COMEON_LINUX.ITA

Area di supporto ComeOn Linux!

Per i suoi attivisti e per quanti vogliano collaborare.

PeaceLink

Rete amatoriale ecopacifista, membro APC. Per maggiori informazioni si può vedere http://www.peacelink.it/.

PCK.PUB.PEACELINK_ARC

Supporto tecnico PeaceLink

non è proprio un'area dedicata a Linux, ma è l'area di supporto tecnico tra gli utenti di PeaceLink e Linux è il sistema operativo ufficiale di rete e associazione PeaceLink.

ScoutNet

Rete del movimento scout mondiale, maggiori informazioni si possono trovare al sito http://www.scoutnet.org/.

LINUX.GSI

TOC

Il Gruppo Sperimentazione Informatica e Linux

area del GSI per le discussioni sull'applicazione di Linux nelle attività scout.

Rete packet AX25

Il sistema operativo Linux è molto apprezzato nell'ambiente radioamatoriale in quanto utilizzato per far funzionare molti gateway tcp/ip che collegato la rete ax25 ham con la rete internet. Sulla rete radioamatoriale esistono delle aree ``Linux'' nelle quali confluiscono tutti i messaggi con indirizzo linux@ita linux@eu e linux@ww (da Valter Bruno, ik1qew).

IRC

Se siete interessati a Linux, se lo conoscete abbastanza bene, molto bene o volete trovare un aiuto, se usate IRC, siete invitati a intervenire nel canale #linux-it di IRCNet. Sono bene accetti utenti di qualsiasi livello, sia per dare che per ricevere informazioni in tempo reale (ottimo quando avete un problema urgente). Piú persone ci sono, piú probabilità c'è di trovare chi dà risposte (da Enrico Cherubini).

Ora il canale #linux-it ha anche un URL, ovvero http://members.tripod.com/~DarkServer.

Meeting

Pluto Meeting

Più o meno due volte l'anno i Plutiani si ritrovano a parlare dei loro progetti e delle loro ambizioni. Almeno una di queste due riunioni è aperta a tutti, viene organizzata con tanto di conferenze e dimostrazioni e viene pubblicamente annunciata nei soliti canali.

Ottimo punto di riferimento è http://meeting.pluto.linux.it/

10.3 Siti Linux

Siti internet

- ftp://tsx-11.mit.edu/pub/linux Casa madre di buona parte del software Linux, amichevolmente chiamato semplicemente ``tsx''.
- ftp://sunsite.unc.edu/pub/Linux Idem come sopra, in più sunsite rappresenta il sito dove guardare per primo se si va a caccia di informazioni e documentazione, anche questo chiamato ``sunsite'' e basta (basta la parola ;).
- ftp://ftp.funet.fi/pub/linux Uno dei più grandi siti ftp al mondo, mirror di tante cose tra cui sunsite, tsx e casa madre dei kernel di Linux.
- ftp://ftp.pluto.linux.it/ Casa madre di tutto quello prodotto dal gruppo Pluto, in particolare aggiornamenti della documentazione tradotta in italiano e Pluto Journal.
- ftp://ftp.linux.it/, ftp://svpop.com.dist.unige.it/ Casa madre di ILS, correntemente mirror completo

di Debian e di molte altre cose relative a Linux, soprattutto relative all'uso di Linux in ambient FTN (Ifmail-HOWTO e compagnia).

- http://gnu.linux.it Mirror italiano ufficiale di http://www.gnu.org.
- ftp://cnuce_arch.cnr.it/pub/Linux Mirror ufficiale italiano di sunsite.
- ftp://ftp.lnf.infn.it/packages/Linux/ Il sito dell'INFN che tra l'altro è mirror italiano di SuSE.

BBS

- **OneWay BBS**, +39 2 4491062, v.32bis,v.42bis, Milano, Fidonet 2:331/333, Virnet
- **DOC!**, +39 41 5905472, ZyXEL19.2k,V.32b,V.42b, Mogliano Veneto (TV), Fidonet 2:333/704
- **Sierra BBS**, +39 6 39721568, ZyXEL19.2k,V.32b,V.42b, Roma, Fidonet 2:335/336
- **Aloha BBS**, +39 923 546164/7, V.32bis, V.42bis, Trapani, Fidonet 2:335/519, MultiMediaNet
- **Pappaconda on-line**, +39 51 6233671, Zyxel 19.200, Bologna, Fidonet 2:332/424
- **ILS BBS**, +39 19 862349, V32bis,V42bis, Savona, Internet (E-Mail, News) Praticamente la versione BBS di svpop.
- **XCalBr8 BBS**, +39 535 780191 Fidonet 2:332/521 Sysop Andrea Baldoni. File per sysadm, sorgenti di dosemu60, doom, driver per Gravis. patch del kernel, sorgente completo del kernel.
- **Eko BBS**, +39 6 5215306, V.32bis, V.42bis, Roma, Fidonet 2:335/374 Files: Slackware (Agosto95), Howto, LDP...
- **cmb2000 BBS**, +39 332 820009, v.32bis, v.42bis, Varese Fidonet 2:331/134 Files: Linux Slackware 3.0, IfMail-HowTo, MIT Sources, VuBbs.
- **Universal Mind BBS**, +39 434 561305 , V.34+, Pordenone, Fidonet 2:333/1016, Pnet 91:13/1, PeaceLink 61:3917/1, Sysop Marco Gaiarin. Mirror (?!) di tutto quello prodotto per il PLUTO (soprattutto Pluto Journal), vario altro. Magic f'req ``PLUTO'' per la lista files Linux.
- **Car Systems Service '93 BBS**, +39 85 4492408, Creatix SG 2834, Montesilvano PE, Fidonet 2:335/626, Euronet 230:1027/0, VirNet 9:395/0, Intelec Network 239:102/0, RIME 900:310/0, Eagle's Nest Network 145:1/0, IFDNet 14:5004/103, OS2Net 81:439/68, SysOp Stefano Brandimarte, Sorgenti del kernel costantemente aggiornati, patch, dosemu, XFree86, librerie e applicazioni varie.

Java CGI HOWTO

by David H. Silber
javacgi-document@orbits.com

v0.5, 1 December 1998

This HOWTO document explains how to set up your server to allow CGI programs written in Java and how to use Java to write CGI programs. Although HOWTO documents are targetted towards use with the Linux operating system, this particular one is not dependant on the particular version of unix used.

1. Introduction

- 1.1 Prior Knowledge
- 1.2 This Document
- 1.3 The Package
- 1.4 The Mailing List

2. Setting Up Your Server to Run Java CGI Programs (With Explanations)

- 2.1 System Requirements
- 2.2 Java CGI Add-On Software
- 2.3 Unpacking the Source
- 2.4 Decide On Your Local Path Policies
- 2.5 Testing your installation.

3. Setting Up Your Server to Run Java CGI Programs (The Short Form)

TOC

4. Executing a Java CGI Program

- 4.1 Obstacles to Running Java Programs Under the CGI Model
- 4.2 Overcoming Problems in Running Java CGI Programs

5. Using the Java CGI Classes.

- 5.1 CGI
- 5.2 CGI_Test
- 5.3 Email
- 5.4 Email_Test
- 5.5 HTML
- 5.6 HTML_Test
- 5.7 Text

6. Future Plans

7. Changes

- 7.1 Changes from 0.4 to 0.5
- 7.2 Changes from 0.3 to 0.4
- 7.3 Changes from 0.2 to 0.3
- 7.4 Changes from 0.1 to 0.2

---

1. Introduction

Because of the way that Java is designed the programmer does not have easy access to the system's environment variables. Because of the way that the Java Development Kit (JDK) is set up, it is necessary to use multiple tokens to invoke a program, which does not mesh very well with the standard HTML forms/CGI manner of operations. There are ways around these limitations, and I have implemented one of them. Read further for details.

Since I wrote the previous paragraph in 1996, there have been many changes in the Java technology. It is likely that a better solution to running server-side Java programs is now available -- perhaps you should take a look at servlets.

1.1 Prior Knowledge

I am assuming that you have a general knowledge of HTML and CGI concepts and at least a minimal knowledge of your HTTP server. You should also know how to program in Java, or a lot of this will not make sense.

1.2 This Document

The latest version of this document can be read at http://www.orbits.com/software/Java_CGI.html.

1.3 The Package

The latest version of the package described here can be accessed via anonymous FTP at ftp://ftp.orbits.com/pub/software/java_cgi-0.5.tgz. The package distribution includes SGML source for this document.

The package is distributed under the terms of the GNU Library General Public License. This document can be distributed under the terms of the Linux HOWTO copyright notice.

If you use this software, please make some reference to http://www.orbits.com/software/Java_CGI.html, so that others will be able to find the Java CGI classes.

I have run out of time to maintain and support this package, so this will probably be its final release. If anyone out there is sufficiently enamoured of this software that they wish to take over the maintenace of it, please contact me at `javacgi-document@orbits.com`.

1.4 The Mailing List

TOC

I have created a majordomo list to allow people to help each-other work through their mutual problems in installing and using this software. Send a message to `javacgi-request@orbits.com`, containing the word *subscribe*.

---

2. Setting Up Your Server to Run Java CGI Programs (With Explanations)

This section will lead you through installing my *Java CGI* package with copious explanations so that you know what the effects of your actions will be. If you just want to install the programs and don't care about the whys & wherefores, skip to Setting Up Your Server to Run Java CGI Programs (The Short Form).

2.1 System Requirements

This software should work on any unix-like web server that has the Java Development Kit installed. I am using it on a *Debian Linux* system running *apache* as the HTTP daemon. If you find that it does not run on your server, please contact the mailing list. See The Mailing List for details.

Unfortunatly, the Java run-time interpreter seems to be something of a memory hog -- you may want to throw another few megabytes of RAM onto your server if you will be using Java CGI programs a lot.

2.2 Java CGI Add-On Software

The software that I wrote to aid in this is called *Java CGI*. You can get it from ftp://ftp.orbits.com/pub/software/java_cgi-0.5.tgz. (The version number may have changed.)

2.3 Unpacking the Source

Find a convenient directory to unpack this package into. (If you don't already have a standard place to put packages, I suggest that you use `/usr/local/src`.) Unpack the distribution with this command:

```
gzip -dc java_cgi-0.5.tgz | tar -xvf -
```

This will create a directory called `java_cgi-0.5`. In there you will find the files referenced in the rest of this document. (If the version number has changed, use the instructions from within that distribution from this point on.)

2.4 Decide On Your Local Path Policies

You need to decide where you want your Java CGI programs to live. Generally, you will want to put them in a directory in parallel with your `cgi-bin` directory. My *apache* server came configured to use `/var/www/cgi-bin` as the `cgi-bin` directory, so I use `/var/www/javacgi` as the directory to put Java CGI programs in. You probably do not want to put your Java CGI programs into one of the

TOC

existing `CLASSPATH` directories. Edit the Makefile to reflect your system configuration. Make sure you are logged in as the root user and run `make install`. This will compile the Java programs, modify the `java.cgi` script to fit in with your system and install the programs in the appropriate places. If you want the HTML version of this documentation and an HTML test document in addition, run `make all` instead.

2.5 Testing your installation.

Installed from the distribution are HTML documents called `javacgitest.html`, `javaemailtest.html` and `javahtmltest.html`. If you installed `all` in the previous section, it will be in the directory you specified for WEBDIR in the `Makefile`. If you didn't, you can run `make test` to build them from `javacgitest.html-dist`, `javaemailtest.html-dist` and `javahtmltest.html-dist`.

When you are sure that your installation is working correctly, you may wish to remove `CGI_Test.class`, `Email_Test.class` and `HTML_Test.class` from your JAVACGI directory and `javacgitest.html`, `javaemailtest.html` and `javahtmltest.html` from your WEBDIR directory as they show the user information that is normally only available to the server.

---

3. Setting Up Your Server to Run Java CGI Programs (The Short Form)

- Get the *Java CGI* package from ftp://ftp.orbits.com/pub/software/java_cgi-0.5.tgz. (The version number may have changed.)
- Unpack the distribution with this command:

  ```
  gzip -dc java_cgi-0.5.tgz | tar -xvf -
  ```

  (If the version number has changed, use the instructions from within that distribution from this point on.)
- Edit the `Makefile` you will find in the newly created directory `java_cgi-0.5` as appropriate to your system.
- As root, run `make install`. This will compile the Java programs, apply your system-specific information and install the various files. If you want the HTML version of this documentation and an HTML test document, run `make all` instead.
- You should be ready to go.

4. Executing a Java CGI Program

4.1 Obstacles to Running Java Programs Under the CGI Model

There are two main problems in running a Java program from a web server:

You can't run Java programs like ordinary executables.

You need to run the Java run-time interpreter and provide the initial class (program to run) on the command-line. With an HTML form, there is no provision for sending a command-line to the web server.

Java does not have general access to the environment.

Every environment variable that will be needed by the Java program must be explicitly passed in. There is no method similar to the **C** `getenv()` function.

4.2 Overcoming Problems in Running Java CGI Programs

To deal with these obstacles, I wrote a shell CGI program that provides the information needed by the Java interpreter.

The java.cgi script.

This shell script manages the interaction between the HTTP daemon and the Java CGI program that you wish to use. It extracts the name of the program that you want to run from the server-provided data. It collects all of the environment data into a temporary file. Then, it runs the Java run-time interpreter with the name of the file of environment information and the program name added to the command-line.

The `java.cgi` script was configured and installed in Decide On Your Local Path Policies.

Invoking java.cgi from an HTML form.

My forms that use Java CGI programs specify a form action as follows:

```
<form action="/cgi-bin/java.cgi/CGI_Test" method="POST">
```

Where `/cgi-bin/` is your local CGI binary directory, `java.cgi` is the Java front-end that allows us

to run Java programs over the web and `CGI_Test` is an example of the name of the Java program to run.

5. Using the Java CGI Classes.

There are currently three main classes supported -- CGI, Email and HTML. I am considering adding classes to deal with MIME-formatted input and output -- MIMEin & MIMEout, respectively.

There are also a few support and test classes. CGI_Test, Email_Test and HTML_Test are intended to be used to test your installation. They can also be used as a starting-point for your own Java programs which use this class library. The Text class is the superclass for both the `Email` and the `HTML` classes.

5.1 CGI

Class Syntax

```
public class CGI
```

Class Description

The CGI class holds the ``CGI Information'' -- Environment variables set by the web server and the name/value sent from a form when its **submit** action is selected. All information is stored in a `Properties` class object.

This class is in the ``Orbits.net'' package.

Member Summary

```
        CGI()          //  Constructor.
        getNames()     //  Get the list of names.
        getValue()     //  Get form value by specifying name.
```

See Also

`CGI_Test`.

CGI()

Purpose

Constructs an object which contains the available CGI data.

Syntax

```
public CGI()
```

Description

When a CGI object is constructed, all available CGI information is sucked-up into storage local to the new object.

getNames()

Purpose

List the names which are defined to have corresponding values.

Syntax

```
public Enumeration getKeys ()
```

Description

Provides the full list of names for which coresponding values are defined.

Returns

An `Enumeration` of all the names defined.

getValue()

Purpose

Retrieves the **value** associated with the **name** specified.

Syntax

```
public String getValue ( String name )
```

Description

This method provides the corespondence between the `names` and `values` sent from an HTML form.

Parameter

name

The key by which values are selected.

Returns

A `String` containing the value.

5.2 CGI_Test

This class provides both an example of how to use the `CGI` class and a test program which can be used to confirm that the *Java CGI* package is functioning correctly.

Member Summary

TOC

```
main()      //  Program main().
```

See Also

`CGI`.

main()

Purpose

Provide a `main()` method.

Syntax

```
public static void main( String argv[] )
```

Description

This is the entry point for a CGI program which does nothing but return a list of the available name/value pairs and their current values.

Parameter

argv[]

Arguments passed to the program by the `java.cgi` script. Currently unused.

5.3 Email

Class Syntax

```
public class Email extends Text
```

Class Description

Messages are built up with the `Text` class `add*()` methods and the e-mail-specific methods added by this class. When complete, the message is sent to its destination.

This class is in the ``Orbits.net'' package.

Member Summary

```
Email()      //  Constructor.
send()       //  Send the e-mail message.
```

```
sendTo()     //  Add a destination for message.
subject()    //  Set the Subject: for message.
```

TOC

See Also

`Email_Test, Text`.

Email()

Purpose

Constructs an object which will contain an email message.

Syntax

```
public Email()
```

Description

Sets up an empty message to be completed by the Email methods.

See Also

`Text`.

send()

Purpose

Send the e-mail message.

Syntax

```
public void send ()
```

Description

This formats and sends the message. If no destination address has been set, there is no action taken.

sendTo()

Purpose

Add a destination for this message.

Syntax

```
public String sendTo ( String address )
```

Description

Add `address` to the list of destinations for this method. There is no set limit to the number of destinations an e-mail message may have. I'm sure that if you build up the list large enough, you can exceed the size of the parameter list that the *Mail Transport Agent* can accept or use up your memory.

TOC

Parameter/

address

A destination to send this message to.

subject()

Purpose

Set the subject for this message.

Syntax

```
public void subject ( String subject )
```

Description

This method sets the text for the e-mail's `Subject:` line. If called more than once, the latest subject set is the one that is used.

Parameter

subject

The text of this message's `Subject:` line.

5.4 Email_Test

This class provides both an example of how to use the `Email` class and a test program which can be used to confirm that the *Java CGI* package is functioning correctly.

Member Summary

```
main()      //  Program main().
```

See Also

`Email`.

main()

Purpose

Provide a `main()` method.

Syntax

```
public static void main( String argv[] )
```

Description

TOC

This is the entry point for a CGI program which returns a list of the available name/value pairs a their current values. It will also send this list to the address specified in the `Email` variable.

Parameter

argv[]

Arguments passed to the program by the `java.cgi` script. Currently unused.

5.5 HTML

Class Syntax

```
public class HTML extends Text
```

Class Description

Messages are built up with the `Text` class `add*()` methods and the HTML-specific methods added by this class. When complete, the message is sent to its destination.

Currently, there is no error checking to confirm that the list-building methods are being used in a correct order, so the programmer must take pains not to violate HTML syntax.

This class is in the ``Orbits.net'' package.

Member Summary

```
HTML()                  //  Constructor.
author()                //  Set the name of the document author.
definitionList()        //  Start a definition list.
definitionListTerm()    //  Add a term to a definition list.
endList()               //  End a list.
listItem()              //  Add an entry to a list.
send()                  //  Send the HTML message.
title()                 //  Set the text for the document title.
```

See Also

`HTML_Test, Text.`

HTML()

Purpose

Constructs an object which will contain an HTML message.

TOC

Syntax

```
public HTML()
```

Description

Sets up an empty message to be completed by the HTML methods.

See Also

`Text`.

author()

Purpose

Set the name of the document author.

Syntax

```
public void author ( String author )
```

Description

Set the name of the document author to `author`.

Parameter/

author

The text to use as the author of this message.

See Also

`title()`.

definitionList()

Purpose

Start a definition list.

Syntax

```
public void definitionList ()
```

Description

Start a definition list. A *definition list* is a list specialized so that each entry in the list is a *term* followed by the definition *text* for that term. The start of a definition list should be followed by the creation of (at least) one term/text pair and a call to the `endList()` method. *Note that, currently, lists cannot be nested.*

See Also

`definitionListTerm()`, `endList()`, `listItem()`.

TOC

definitionListTerm()

Purpose

Add a term to a definition list.

Syntax

```
public void definitionListTerm ()
```

Description

Add a term to a definition list. The text for the term part of the current list entry should be appended to the message after this method is called and before a corresponding `listItem` method is called.

See Also

`definitionList()`, `listItem()`.

endList()

Purpose

End a list.

Syntax

```
public void endList ()
```

Description

End a list. This method closes out a list. *Note that, currently, lists cannot be nested.*

See Also

`definitionList()`.

listItem()

Purpose

Add an entry to a list.

Syntax

```
public void listItem ()
```

```
public void listItem ( String item )
```

```
public boolean listItem ( String term, String item )
```

Description

Add an entry to a list. If the first form is used, the text for the current list item should be appended to the message after this method is called and before any other list methods are called. In the second and third forms, the `item` text is specified as a parameter to the method instead of (or in addition to) being appended to the message. The third form is specific to definition lists and provides both the

TOC

term and the definition of the list entry.

Parameters

item

The text of this list entry.

term

The text of this definition list entry's term part.

See Also

`definitionList()`, `definitionListTerm()`, `endList()`.

send()

Purpose

Send the HTML message.

Syntax

```
public void send ()
```

Description

Send the HTML message.

title()

Purpose

Set the text for the document title.

Syntax

```
public void title ( String title )
```

Description

Set the text for the document title.

Parameter

title

The text of this message's title.

See Also

`author()`.

5.6 HTML_Test

This class provides both an example of how to use the `HTML` class and a test program which can be used to confirm that the *Java CGI* package is functioning correctly.

Member Summary

```
main()      //  Program main().
```

See Also

`HTML`.

main()

Purpose

Provide a `main()` method.

Syntax

```
public static void main( String argv[] )
```

Description

This is the entry point for a CGI program which returns a list of the available name/value pairs in an HTML document, with each name/value pair displayed in a definition list element.

Parameter

argv[]

Arguments passed to the program by the `java.cgi` script. Currently unused.

5.7 Text

Class Syntax

```
public abstract class Text
```

Class Description

This class is the superclass of the `Email` and `HTML` classes. Messages are built up with the methods in this class and completed and formatted with the methods in subclasses.

This class is in the ``Orbits.text'' package.

Member Summary

TOC

```
Text()            //  Constructor.
add()             //  Add text to this object.
addLineBreak()    //  Add a line break.
addParagraph()    //  Add a paragraph break.
```

See Also

`Email`, `HTML`.

add()

Purpose

Add text to this item.

Syntax

```
public void add ( char addition )
```

```
public void add ( String addition )
```

```
public void add ( StringBuffer addition )
```

Description

Add `addition` to the contents of this text item.

Parameter

addition

Text to be added to the text item.

See Also

`addLineBreak()`, `addParagraph()`.

addLineBreak()

Purpose

Force a line break at this point in the text.

Syntax

```
public void addLineBreak ()
```

Description

Add a line break to the text at the current point.

TOC

See Also

`add()`, `addParagraph()`.

addParagraph()

Purpose

Start a new paragaph.

Syntax

```
public void add ()
```

Description

Start a new paragraph at this point in the text flow.

See Also

`add()`, `addLineBreak()`.

---

6. Future Plans

- Add to the Email class:
  Email(int capacity)

  Used when we know how much space the message will need to have allocated.
  sendTo(String [] address)

  Add a list of primary destinations to the e-mail message.
  sendCc(String address)

  Add a Carbon-Copy destination to the e-mail message.
  sendCc(String [] address)

  Add a list of Carbon-Copy destinations to the e-mail message.
  sendBcc(String address)

  Add a Blind Carbon-Copy destination to the e-mail message.
  sendBcc(String [] address)

  Add a list of Blind Carbon-Copy destinations to the e-mail message.
- Add to the HTML class:
  HTML(int capacity)

  Used when we know how much space the message will need to have allocated.
  public void unorderedList()

  Start an unordered list.
  public void orderedList()

  Start an ordered list.
  public void directoryList()

  Start a directory list.
  public void menuList()

  Start a menu list.
  void anchor(String anchorName)

  Specify an anchor.
  void link(String url, String text)

  Specify a link.

void applet(String url, String altText)

Specify an applet link.

- Allow HTML lists to be nested.
- Add error checking code to enforce correct ordering of HTML list formatting codes.
- The location of the file of environment data should be configurable from the `Makefile`.
- Get rid of the spurious empty name/value pair that appears in the list when we are dealing with the GET method of data transfer.
- Consider having CGI implement the java.util.Enumeration interface to successively provide variable names.
- Add a `Test` class, which would use every method in this package.
- Document how `CGI_Test`, `Email_Test` and `HTML_Test` build on each other to provide incremental tests for debugging purposes.
- Document how Test uses every feature available in this package.

---

7. Changes

7.1 Changes from 0.4 to 0.5

- Changed documentation and comments to reflect the final nature of this release.

7.2 Changes from 0.3 to 0.4

- Fleshed out the HTML class to provide minimal functionality.
- Wrote the HTML_Test class and javahtmltest.html-dist.
- Added the HTML methods to deal with a definition list.

7.3 Changes from 0.2 to 0.3

- Added the Text and Email classes. HTML was also added, but it is merely a stub at this point.
- Put the various classes into packages. The main classes are in `Orbits.net.*`, the support class `Text` is in `Orbits.text.Text`.
- Changed `CGItest` to `CGI_Test`.
- Added the `Email_Test` class.

7.4 Changes from 0.1 to 0.2

- The environment variables are put into a temportary file instead of being crammed into the Java inperpreter command-line. The `CGI` class and `java.cgi` had to be modified.
- The `javacgitest.html` document is made part of the distribution.
- The text files which are modified by `make` upon installation are provided with names that end with *-dist*.

The Linux keyboard and console HOWTO

Andries Brouwer, aeb@cwi.nl

v2.8, 25 February 1998

This note contains some information about the Linux keyboard and console, and the use of non-ASCII characters. It describes Linux 2.0.

1. Useful programs

2. Keyboard generalities

3. Console generalities

4. Resetting your terminal

- 4.1 Keyboard hardware reset

5. Delete and Backspace

- 5.1 How to tell Unix what character you want to use to delete the last typed character
- 5.2 How to tell Linux what code to generate when a key is pressed
- 5.3 How to tell X to interchange Delete and Backspace
- 5.4 How to tell emacs what to do when it receives a Delete or Backspace
- 5.5 How to tell emacs to interchange Delete and Backspace
- 5.6 How to tell kermit to interchange Delete and Backspace
- 5.7 How to tell xterm about your favourite tty modes
- 5.8 How to tell xmosaic that the Backspace key generates a DEL
- 5.9 A better solution for Motif-using programs, like netscape
- 5.10 What about termcap and terminfo?

TOC

6. The console character sets

7. Console switching

- 7.1 Changing the number of Virtual Consoles

8. Ctrl-Alt-Del and other special key combinations

- 8.1 Ctrl-Alt-Del (Boot)
- 8.2 Other combinations
- 8.3 X Combinations
- 8.4 Dosemu Combinations
- 8.5 Composing symbols
- 8.6 The SysRq key

9. How to get out of raw mode

10. The keyboard LEDs

11. The TERM variable

- 11.1 Terminfo

12. How to make other programs work with non-ASCII chars

13. What precisely does XFree86-2.1 do when it initializes its keymap?

14. Unusual keys and keyboards

15. Examples of use of loadkeys and xmodmap

- 15.1 `I can use only one finger to type with'

16. Changing the video mode

- 16.1 Instructions for the use of resizecons

17. Changing the keyboard repeat rate

18. Scrolling

19. Screensaving

20. Screen dumps

21. Some properties of the VT100 - application key mode

22. Hardware incompatibility

23. Copyright

1. Useful programs

The following packages contain keyboard or console related programs.

`kbd-0.95.tar.gz` contains `loadkeys`, `dumpkeys`, `showkey`, `setmetamode`, `setleds`, `setfont`, `showfont`, `mapscrn`, `kbd_mode`, `loadunimap`, `chvt`, `resizecons`, `deallocvt`, `getkeycodes`, `setkeycodes`.

`util-linux-2.6` contains `setterm`, `kbdrate`. (Yes, the `more` in `util-linux-2.6` dumps core due to a name conflict. Preserve your old copy, or use `util-linux-2.5`, or change `savetty' to `my_savetty' in `more.c`.)

`sh-utils-1.12` contains `stty`.

`open-1.4.tgz` contains `open` (that should be renamed to `openvt`). (See also `dynamic-vc-1.1.tar.gz`.)

`SVGATextMode-1.8.tar.gz` contains `SVGATextMode`, a program that obsoletes `resizecons`.

The X distribution contains `xmodmap`, `xset`, `kbd_mode`. (See also X386keybd(1) for the situation under XFree86 1.3, and Xserver(1) for the XKEYBOARD extension under X11R6.)

`termcap-2.0.8.tar.gz` contains `termcap`, an old terminal capabilities data base. `ncurses-1.9.9e.tar.gz` contains the `termlib` data base which obsoletes `termcap`. (However, there are still many programs using `termcap`.)

See loadkeys(1), setleds(1) and setmetamode(1) for the codes generated by the various keys and the setting of leds when not under X. Under X, see xmodmap(1) and xset(1).

See setfont(8) for loading console fonts. Many people will want to load a font like `iso01.f16` because the default font is the hardware font of the video card, and often is a `Code Page 437' font missing accented characters and other Latin-1 symbols.

See setterm(1) and kbdrate(8) for properties such as foreground and background colors, screen blanking and character repeat rate when not under X. Under X, see xset(1), also for key click and bell volume.

The file `/etc/termcap` defines the escape sequences used by many programs addressing the console (or any other terminal). See termcap(5). A more modern version is found in `/usr/lib/terminfo`. See terminfo(5). Terminfo files are compiled by the terminfo compiler `/usr/lib/terminfo/tic`, see tic(1). Their contents can be examined using the program `infocmp`, see infocmp(1). The Linux console sequences are documented in console_codes(4).

2. Keyboard generalities

You press a key, and the keyboard controller sends scancodes to the kernel keyboard driver. Some keyboards can be programmed, but usually the scancodes corresponding to your keys are fixed. The kernel keyboard driver just transmits whatever it receives to the application program when it is in *scancode mode*, like when `X` is running. Otherwise, it parses the stream of scancodes into keycodes, corresponding to key press or key release events. (A single key press can generate up to 6 scancodes.) These keycodes are transmitted to the application program when it is in *keycode mode* (as used, for example, by `showkey`). Otherwise, these keycodes are looked up in the keymap, and the character or string found there is transmitted to the application, or the action described there is performed. (For example, if one presses and releases the `a` key, then the keyboard produces scancodes 0x1e and 0x9e, this is converted to keycodes 30 and 158, and then transmitted as 0141, the ASCII or latin-1 code for `a'; if one presses and releases `Delete`, then the keyboard produces scancodes 0xe0 0x53 0xe0 0xd3, these are converted to keycodes 111 and 239, and then transmitted as the 4-symbol sequence ESC [3 ~, all assuming a US keyboard and a default keymap. An example of a key combination to which an action is assigned is Ctrl-Alt-Del.)

The translation between unusual scancodes and keycodes can be set using the utility `setkeycodes` - only very few people will need it. The translation between keycodes and characters or strings or actions, that is, the keymap, is set using the utilities `loadkeys` and `setmetamode`. For details, see getkeycodes(8), setkeycodes(8), dumpkeys(1), loadkeys(1), setmetamode(1). The format of the files output by `dumpkeys` and read by `loadkeys` is described in keytables(5).

Where it says `transmitted to the application' in the above description, this really means `transmitted to the terminal driver'. That is, further processing is just like that of text that comes in over a serial line. The details of this processing are set by the program `stty`.

3. Console generalities

Conversely, when you output something to the console, it first undergoes the standard tty processing, and then is fed to the console driver. The console driver emulates a VT100, and parses the input in order to recognize VT100 escape sequences (for cursor movement, clear screen, etc.). The characters that are not part of an escape sequence are first converted into Unicode, using one of four mapping tables if the console was not in UTF-8 mode to start with, then looked up in the table describing the correspondence between Unicode values and font positions, and the obtained 8- or 9-bit font indices are then written to video memory, where they cause the display of character shapes found in the video card's character ROM. One can load one's own fonts into character ROM using `setfont`, load the corresponding Unicode map with `loadunimap`, and load a user mapping table using `mapscrn`. More details will be given below.

There are many consoles (called *Virtual Consoles* or *Virtual Terminals*, abbreviated VCs or VTs) that share the same screen. You can use them as independent devices, either to run indendent login sessions, or just to send some output to, perhaps from `top`, or the tail of the system log or so. See below (`Console switching') on how to set them up and switch between them.

4. Resetting your terminal

There is garbage on the screen, or all your keystrokes are echoed as line drawing characters. What to do?

Many programs will redraw the screen when ^L is typed. This might help when there is some modem noise or broadcast message on your screen. The command `clear` will clear the screen.

The command `reset` will reset the console driver. This helps when the screen is full of funny graphic characters, and also if it is reduced to the bottom line. If you don't have this command, or if it does something else, make your own by putting the following two lines in an executable file `reset` in your PATH:

```
#!/bin/sh
echo -e \\033c
```

that is, you want to send the two characters ESC c to the console.

Why is it that the display sometimes gets confused and gives you a 24-line or 1-line screen, instead of the usual 25 lines? Well, the main culprit is the use of `TERM=vt100` (or some other entry with 24 lines) instead of `TERM=linux` when logged in remotely. If this happens on `/dev/tty2` then typing

```
% cat > /dev/tty2
^[c
^D
```

on some other VT (where 4 symbols are typed to `cat`: ESC, c, ENTER, Ctrl-D) and refreshing the screen on `/dev/tty2` (perhaps using ^L) will fix things. Of course the permanent fix is to use the right termcap or terminfo entry.

Why is it that you sometimes get a lot of line-drawing characters, e.g., after catting a binary to the screen? Well, there are various character set changing escape sequences, and by accident your binary might contain some of these. The ESC c is a general reset, a cure for all, but if you know precisely what went wrong you can repair it without resetting other console attributes. For example, after

```
% cat
^N
^D
```

your shell prompt will be all line-drawing characters. Now do (typing blindly)

```
% cat
^O
^D
```

and all is well again. (Three symbols typed to each `cat`: ^N (or ^O), ENTER, Ctrl-D.) To understand

TOC

what is happening, see `The console character sets' below.

If you loaded some strange font, and want to return to the default,

```
% setfont
```

will do (provided you stored the default font in the default place). If this default font does not contain an embedded Unicode map (and gives the wrong symbols for accented characters), then say

```
% loadunimap
```

For example, if I do

```
% loadkeys de-latin1
```

then I have a German keyboard, and the key left of the Enter key gives me a-umlaut. This works, because the a-umlaut occurs on the CP437 code page and the kernel Unicode map is initialized to CP437, and my video card has a CP437 font built-in. If I now load an ISO 8859-1 font with

```
% setfont iso01.f16
```

then everything still works, because `setfont` invalidates the kernel Unicode map (if there is no Unicode map attached to the font), and without map the kernel goes directly to the font, and that is precisely correct for an ISO 8859-1 system with `iso01.f16` font. But going back to the previous font with

```
% setfont
```

gives capital Sigma's instead of a-umlaut - all accented letters are mixed up because also this font has no embedded Unicode map. After

```
% loadunimap
```

which loads the default Unicode map (which is right for the default font) all works correctly again. Usually `loadunimap` is not invoked directly, but via `setfont`. Thus, the previous two commands may be replaced by

```
% setfont -u def
```

The Ethiopian fonts and the `lat1u*.psf` fonts have embedded Unicode code map. Most of the others don't.

On old terminals output involving tabs may require a delay, and you have to say

```
% stty tab3
```

(see stty(1)).

You can change the video mode using `resizecons` or `SVGATextMode`. This usually settles the output side. On the input side there are many things that might be wrong. If `X` or `DOOM` or some other program using raw mode crashed, your keyboard may still be in raw (or mediumraw) mode, and it is

difficult to give commands. (See "How to get out of raw mode" below.) If you loaded a bad keymap, then

```
% loadkeys -d
```

loads the default map again, but it may well be difficult to type `-'! An alternative is

```
% loadkeys defkeymap
```

Sometimes even the letters are garbled. It is useful to know that there are four main types of keyboards: QWERTY, QWERTZ, AZERTY and DVORAK. The first three are named after the first six letter keys, and roughly represent the English, German and French speaking countries. Compared to QWERTY, the QWERTZ map interchanges Y and Z. Compared to QWERTY, the AZERTY map interchanges Q and A, W and Z, and has its M right of the L, at the semicolon position. DVORAK has an entirely different letter ordering.

4.1 Keyboard hardware reset

Things may be wrong on a lower level than Linux knows about. There are at least two distinct lower levels (keyboard and keyboard controller) where one can give the command "keyboard disable" to the keyboard hardware. Keyboards can often be programmed to use one out of three different sets of scancodes.

However, I do not know of cases where this turned out to be a problem.

Some keyboards have a remapping capability built in. Stormy Henderson (`stormy@Ghost.Net`) writes: `If it's your keyboard accidently being reprogrammed, you can (on a Gateway AnyKey keyboard) press control-alt-suspend_macro to reset the keys to normal.'

---

5. Delete and Backspace

Getting Delete and Backspace to work just right is nontrivial, especially in a mixed environment, where you talk to console, to `X`, to `bash`, to `emacs`, login remotely, etc. You may have to edit several configuration files to tell all of the programs involved precisely what you want. On the one hand, there is the matter of which keys generate which codes (and how these codes are remapped by e.g. `kermit` or `emacs`), and on the other hand the question of what functions are bound to what codes.

People often complain `my backspace key does not work', as if this key had a built-in function `delete previous character'. Unfortunately, all this key, or any key, does is producing a code, and one only can hope that the kernel tty driver and all application programs can be configured such that the backspace key indeed does function as a `delete previous character' key.

Most Unix programs get their tty input via the kernel tty driver in `cooked' mode, and a simple `stty` command determines the erase character. However, programs like `bash` and `emacs` and `X` do their own input handling, and have to be convinced one-by-one to do the right thing.

5.1 How to tell Unix what character you want to use to delete the last typed character

```
% stty erase ^?
```

If the character is erased, but in a funny way, then something is wrong with your tty settings. If `echoprt` is set, then erased characters are enclosed between `\` and `/`. If `echoe` is not set, then the erase char is echoed (which is reasonable when it is a printing character, like #). Most people will want `stty echoe -echoprt`. Saying `stty sane` will do this and more. Saying `stty -a` shows your current settings. How come this is not right by default? It is, if you use the right `getty`.

Note that many programs (like `bash`, `emacs` etc.) have their own keybindings (defined in `~/.inputrc`, `~/.emacs` etc.) and are unaffected by the setting of the erase character.

The standard Unix tty driver does not recognize a cursor, or keys (like the arrow keys) to move the current position, and hence does not have a command `delete current character' either. But for example you can get `bash` on the console to recognize the Delete key by putting

```
set editing-mode emacs
"\e[3~":delete-char
```

into `~/.inputrc`.

`Getty used to do the right thing with DEL and BS but is broken now?'

Earlier, the console driver would do BS Space BS (`\010\040\010`) when it got a DEL (`\177`). Nowadays, DEL's are ignored (as they should be, since the driver emulates a vt100). Get a better getty, i.e., one that does not output DEL.

`Login behaves differently at the first and second login attempts?'

At the first attempt, you are talking to `getty`. At the second attempt, you are talking to `login`, a different program.

5.2 How to tell Linux what code to generate when a key is pressed

TOC

On the console, or, more precisely, when not in (MEDIUM)RAW mode, use

```
% loadkeys mykeys.map
```

and under X use

```
% xmodmap mykeys.xmap
```

Note that (since XFree86-2.1) X reads the Linux settings of the keymaps when initialising the X keymap. Although the two systems are not 100% compatible, this should mean that in many cases the use of `xmodmap` has become superfluous.

For example, suppose that you would like the Backspace key to send a BackSpace (^H, octal 010) and the grey Delete key a DEL (octal 0177). Add the following to `/etc/rc.local` (or wherever you keep your local boot-time stuff):

```
/usr/bin/loadkeys << EOF
keycode 14 = BackSpace
keycode 111 = Delete
EOF
```

Note that this will only change the function of these keys when no modifiers are used. (You need to specify a keymaps line to tell which keymaps should be affected if you want to change bindings on more keymaps.) The Linux kernel default lets Ctrl-Backspace generate BackSpace - this is sometimes useful as emergency escape, when you find you can only generate DELs.

The left Alt key is sometimes called the Meta key, and by default the combinations AltL-X are bound to the symbol MetaX. But what character sequence is MetaX? That is determined (per-tty) by the Meta flag, set by the command `setmetamode`. The two choices are: ESC X or X or-ed with 0200.

`Why doesn't the Backspace key generate BackSpace by default?'

(i) Because the VT100 had a Delete key above the Enter key.

(ii) Because Linus decided so.

5.3 How to tell X to interchange Delete and Backspace

```
% xmodmap -e "keysym BackSpace = Delete" -e "keysym Delete = BackSpace"
```

Or, if you just want the Backspace key to generate a BackSpace:

```
% xmodmap -e "keycode 22 = BackSpace"
```

Or, if you just want the Delete key to generate a Delete:

```
% xmodmap -e "keycode 107 = Delete"
```

(but usually this is the default binding already).

5.4 How to tell emacs what to do when it receives a Delete Backspace

TOC

Put in your `.emacs` file lines like

```
(global-set-key "\?" 'help-command)
(global-set-key "\C-h" 'delete-backward-char)
```

Of course you can bind other commands to other keys in the same way. Note that various major and minor modes redefine keybindings. For example, in incremental search mode one finds the code

```
(define-key map "\177" 'isearch-delete-char)
(define-key map "\C-h" 'isearch-mode-help)
```

This means that it may be a bad idea to use the above two global-set-key commands. There are too many places where there are built-in assumptions about ^H = help and DEL = delete. That doesn't mean that you have to setup keys so that Backspace generates DEL. But if it doesn't then it is easiest to remap them at the lowest possible level in emacs.

5.5 How to tell emacs to interchange Delete and Backspace

Put in your `.emacs` file lines

```
(setq keyboard-translate-table (make-string 128 0))
(let ((i 0))
  (while (< i 128)
    (aset keyboard-translate-table i i)
    (setq i (1+ i))))
(aset keyboard-translate-table ?\b ?\^?)
(aset keyboard-translate-table ?\^? ?\b)
```

Recent versions of emacs have a function `keyboard-translate` and one may simplify the above to

```
(keyboard-translate ?\C-h ?\C-?)
(keyboard-translate ?\C-? ?\C-h)
```

Note that under X emacs can distinguish between Ctrl-h and the Backspace key (regardless of what codes these produce on the console), and by default emacs will view the Backspace key as DEL (and do deletion things, as bound to that character, rather than help things, bound to ^H). One can distinguish Backspace and Delete, e.g. by

```
(global-unset-key [backspace] )
(global-set-key [backspace] 'delete-backward-char)
(global-unset-key [delete] )
(global-set-key [delete] 'delete-char)
```

5.6 How to tell kermit to interchange Delete and Backspace

Put in your `.kermrc` file the lines

```
set key \127 \8
set key \8 \127
```

TOC

5.7 How to tell xterm about your favourite tty modes

Normally xterm will inherit the tty modes from its invoker. Under `xdm`, the default erase and kill characters are # and @, as in good old Unix Version 6. If you don't like that, you might put something like

```
XTerm*ttymodes: erase ^? kill ^U intr ^C quit ^\ eof ^D \
                susp ^Z start ^Q stop ^S eol ^@
```

in `/usr/lib/X11/app-defaults/XTerm` or in `$HOME/.Xresources`, assuming that you have a line

```
xrdb $HOME/.Xresources
```

in your `$HOME/.xinitrc` or `$HOME/.xsession`.

5.8 How to tell xmosaic that the Backspace key generates a DEL

Putting

```
*XmText.translations: #override\n\
   <Key>osfDelete: delete-previous-character()
*XmTextField.translations: #override\n\
   <Key>osfDelete: delete-previous-character()
```

in your `$HOME/.Xresources` helps.

The netscape FAQ, however, says:

```
Why doesn't my Backspace key work in text fields?
By default, Linux and XFree86 come with the Backspace and Delete keys
misconfigured. All Motif programs (including, of course, Netscape
Navigator) will malfunction in the same way.

The Motif spec says that Backspace is supposed to delete the previous
character and Delete is supposed to delete the following character.
Linux and XFree86 come configured with both the Backspace and Delete
keys generating Delete.

You can fix this by using any one of the xmodmap, xkeycaps, or
loadkeys programs to make the key in question generate the BackSpace
keysym instead of Delete.

You can also fix it by having a .motifbind file; see the man page
for VirtualBindings(3).

Note: Don't use the *XmText.translations or *XmTextField.translations
resources to attempt to fix this problem. If you do, you will blow
away Netscape Navigator's other text-field key bindings.
```

5.9 A better solution for Motif-using programs, like netscap

TOC

Ted Kandell (`ted@tcg.net`) suggests the following:

Somewhere in your .profile add the following:

```
stty erase ^H
```

If you are using `bash`, add the following lines to your `.inputrc`:

```
"\C-?": delete-char
"\C-h": backward-delete-char
```

Add the following lines to your .xinitrc file:

```
xmodmap <<-EOF
keycode 22  =  BackSpace osfBackSpace
keycode 107 =  Delete
EOF

# start your window manager here,  for example:
#(fvwm) 2>&1 | tee /dev/tty /dev/console

stty sane
stty erase ^H
loadmap <<-EOF
keycode 14  = BackSpace
keycode 111 = Delete
EOF
```

This will definitely work for a PC 101 or 102 key keyboard with any Linux/XFree86 layout.

The important part to making Motif apps like Netscape work properly is adding osfBackSpace to keycode 22 in addition to BackSpace.

Note that there must be spaces on either side of the = sign.

5.10 What about termcap and terminfo?

When people have problems with backspace, they tend to look at their termcap (or terminfo) entry for the terminal, and indeed, there does exist a kb (or kbs) capability describing the code generated by the Backspace key. However, not many programs use it, so unless you are having problems with one particular program only, probably the fault is elsewhere. Of course it is a good idea anyway to correct your termcap (terminfo) entry. See also below under "The TERM variable".

---

6. The console character sets

The kernel first tries to figure out what symbol is meant by any given user byte, and next where this symbol is located in the current font.

The kernel knows about 5 translations of bytes into console-screen symbols. In Unicode (UTF-8) mode, the UTF-8 code is just converted directly into Unicode. The assumption is that almost all symbols one needs are present in Unicode, and for the cases where this does not hold the codes 0xff** are reserved for direct font access. When not in Unicode mode, one of four translation tables is used. The four tables are: a) Latin1 -> Unicode, b) VT100 graphics -> Unicode, c) PC -> Unicode, d) user-defined.

There are two character sets, called G0 and G1, and one of them is the current character set. (Initially G0.) Typing ^N causes G1 to become current, ^O causes G0 to become current.

These variables G0 and G1 point at a translation table, and can be changed by the user. Initially they point at tables a) and b), respectively. The sequences ESC (B and ESC (0 and ESC (U and ESC (K cause G0 to point at translation table a), b), c) and d), respectively. The sequences ESC) B and ESC) 0 and ESC) U and ESC) K cause G1 to point at translation table a), b), c) and d), respectively.

The sequence ESC c causes a terminal reset, which is what you want if the screen is all garbled. The oft-advised `echo ^V^O` will only make G0 current, but there is no guarantee that G0 points at table a). In some distributions there is a program reset(1) that just does `echo ^[c`. If your termcap entry for the console is correct (and has an entry `:rs=\Ec:`), then also `setterm -reset` will work.

The user-defined mapping table can be set using mapscrn(8). The result of the mapping is that if a symbol c is printed, the symbol `s = map[c]` is sent to the video memory. The bitmap that corresponds to `s` is found in the character ROM, and can be changed using setfont(8).

7. Console switching

By default, console switching is done using Alt-Fn or Ctrl-Alt-Fn. Under `X` (or recent versions of `dosemu`), only Ctrl-Alt-Fn works. Many keymaps will allow cyclic walks through all allocated consoles using Alt-RightArrow and Alt-LeftArrow.

XFree86 1.3 does not know that Alt is down when you switch to the X window. Thus, you cannot switch immediately to some other VT again but have to release Alt first. In the other direction this should work: the kernel always keeps track of the up/down status of all keys. (As far as possible: on some keyboards some keys do not emit a scancode when pressed (e.g.: the PFn keys of a FOCUS 9000) or released (e.g.: the Pause key of many keyboards).)

XFree86 1.3 saves the fonts loaded in the character ROMs when started, and restores it on a console switch. Thus, the result of `setfont` on a VT is wiped out when you go to X and back. Using `setfont` under X will lead to funny results.

One can change VT under program control using the `chvt` command.

7.1 Changing the number of Virtual Consoles

This question still comes up from time to time, but the answer is: you already have enough of them. Since kernel version 1.1.54, there are between 1 and 63 virtual consoles. A new one is created as soon as it is opened. It is removed by the utility `deallocvt` (but it can be removed only when no processes are associated to it anymore, and no text on it has been selected by programs like `selection` or `gpm`).

For older kernels, change the line

```
#define NR_CONSOLES     8
```

in `include/linux/tty.h` (don't increase this number beyond 63), and recompile the kernel.

If they do not exist yet, create the tty devices with `MAKEDEV` or `mknod ttyN c 4` N where N denotes the tty number. For example,

```
for i in 9 10 11 12; do mknod /dev/tty$i c 4 $i; done
```

or, better (since it also takes care of owner and permissions),

```
for i in 9 10 11 12; do /dev/MAKEDEV tty$i; done
```

If you want the new VCs to run `getty`, add lines in `/etc/inittab`. (But it is much better to have only two `getty`'s running, and to create more consoles dynamically as the need arises. That way you'll have more memory when you don't use all these consoles, and also more consoles, in case you really need them. Edit `/etc/inittab` and comment out all `getty`'s except for the first two.)

When the consoles are allocated dynamically, it is usually easiest to have only one or two running `getty`. More are opened by `open -l -s bash`. Unused consoles (without associated processes) are deallocated using `deallocvt` (formerly `disalloc`). But, you say, I am involved in activities when I suddenly need more consoles, and do not have a bash prompt available to give the `open` command. Fortunately it is possible to create a new console upon a single keystroke, regardless of what is happening at the current console.

If you have `spawn_login` from `kbd-0.95.tar.gz` and you put

```
loadkeys << EOF
alt keycode 103 = Spawn_Console
EOF
spawn_login &
```

in `/etc/rc.local`, then typing Alt-UpArrow will create a fresh VC running `login` (and switch to it). With `spawn_console &` instead of `spawn_login &` you'll have bash running there. See also `open-1.4.tgz` and `dynamic-vc-1.1.tar.gz`.

What action should be taken upon this Spawn_Console keypress can also be set in `/etc/inittab` under `kbrequest`, if you have a recent `init`. See inittab(5).

(This action can be something entirely different - I just called the key Spawn_Console because that is what I used it for. When used for other purposes it is less confusing to use its synonym KeyboardSignal. For example, some people like to put the lines

```
kb::kbrequest:/sbin/shutdown -h now
```

in `/etc/inittab`, and

```
control alt keycode 79 = KeyboardSignal
control alt keycode 107 = KeyboardSignal
```

in their keymap. Now Ctrl-Alt-End will do a system shutdown.)

You can only login as "root" on terminals listed in `/etc/securetty`. There exist programs that read terminal settings from files `/etc/ttys` and `/etc/ttytype`. If you have such files, and create additional consoles, then it might be a good idea to also add entries for them in these files.

---

8. Ctrl-Alt-Del and other special key combinations

8.1 Ctrl-Alt-Del (Boot)

If you press Ctrl-Alt-Del (or whatever key was assigned the keysym Boot by loadkeys) then either the machine reboots immediately (without sync), or `init` is sent a SIGINT. The former behaviour is the default. The default can be changed by root, using the system call reboot(), see ctrlaltdel(8). Some `init`'s change the default. What happens when `init` gets SIGINT depends on the version of `init` used - often it will be determined by the pf entry in `/etc/inittab` (which means that you can run an arbitrary program in this case). In the current kernel Ctrl-AltGr-Del is no longer by default assigned to Boot.

8.2 Other combinations

```
Name            Default binding
--------------------------------
Show_Memory     Shift-Scrollock
Show_Registers  AltGr-ScrollLock
Show_State      Ctrl-ScrollLock
Console_n       Alt-Fn and Ctrl-Alt-Fn  (1 <= n <= 12)
Console_{n+12}  AltGr-Fn                (1 <= n <= 12)
Incr_Console    Alt-RightArrow
Decr_Console    Alt-LeftArrow
Last_Console    Alt[Gr]-PrintScreen
Scroll_Backward Shift-PageUp
Scroll_Forward  Shift-PageDown
Caps_On                                 (CapsLock is a toggle; this key sets)
Compose         Ctrl-.
```

8.3 X Combinations

```
Ctrl-Alt-Fn     Switch to VT n
Ctrl-Alt-KP+    Next mode
Ctrl-Alt-KP-    Previous mode
Ctrl-Alt-Backspace      Kill X
```

On some motherboards, Ctrl-Alt-KP- and Ctrl-Alt-KP+ will be equivalent to pressing the Turbo button. That is, both will produce the scancodes 1d 38 4a ca b8 9d and 1d 38 4e ce b8 9d, and both will switch between Turbo (>= 25MHz) and non-Turbo (8 or 12 MHz). (Often these key combinations only function this way when enabled by jumpers on the motherboard.)

Perry F Nguyen (`pfnguyen@netcom22.netcom.com`) writes: AMI BIOS has a feature that locks up the keyboard and flashes the LED's if the Ctrl-Alt-Backspace combination is pressed while a BIOS password is enabled, until the CMOS/BIOS password is typed in.

8.4 Dosemu Combinations

```
Ctrl-Alt-Fn     Switch to VT n (from version 0.50; earlier Alt-Fn)
Ctrl-Alt-PgDn   Kill dosemu (when in RAW keyboard mode)
(and many other combinations - see the dosemu documentation)
```

8.5 Composing symbols

One symbol may be constructed using several keystrokes.

- LeftAlt-press, followed by a decimal number typed on the keypad, followed by LeftAlt-release, yields the symbol with code given by this number. (In Unicode mode this same mechanism, but then with 4 hexadecimal digits, may be used to define a Unicode symbol.)
- A dead diacritic followed by a symbol, yields that symbol adorned with that diacritic. If the combination is undefined, both keys are taken separately. Which keys are dead diacritics is user-settable; none is by default. Five (since 2.0.25 six) dead diacritics can be defined (using loadkeys(1)): dead_grave, dead_acute, dead_circumflex, dead_tilde, dead_diaeresis (and dead_cedilla). Precisely what this adorning means is also user-settable: dead-diacritic, symbol is equivalent to Compose + diacritic + symbol.
- Compose followed by two symbols yields a combination symbol. These combinations are user-settable. Today there are 68 combinations defined by default; you can see them by saying "dumpkeys | grep compose".
- Then there are `Sticky' modifier keys (since 1.3.33). For example, one can type ^C as SControl, C and Ctrl-Alt-BackSpace as SControl, SAlt, BackSpace.

Note that there are at least three such composition mechanisms:

1. The Linux keyboard driver mechanism, used in conjunction with loadkeys.
2. The X mechanism - see X386keybd(1), later XFree86kbd(1). Under X11R6: edit `/usr/X11R6/lib/X11/locale/iso8859-1/Compose`.

   See also Andrew D. Balsa's comments at http://wauug.erols.com/~balsa/linux/deadkeys/index.html.
3. The emacs mechanism obtained by loading "iso-insert.el" or calling `iso-accents-mode'.

For X the order of the two symbols is arbitrary: both Compose-,-c and Compose-c-, yield a c-cedilla; for Linux and emacs only the former sequence works by default. For X the list of compose combinations is fixed. Linux and emacs are flexible. The three default lists are somewhat similar, but the details are different.

8.6 The SysRq key

In case your kernel was compiled with CONFIG_MAGIC_SYSRQ enabled (a feature that is present since Linux 2.1.43) there is a single key (defined in `<linux/keyboard.h>`) to which special system functions are attached, regardless of the current keyboard mode. For the PC architecture this special key is, naturally, the Alt+SysRq key, and any of the two Alt keys will work. (Note that if CONFIG_MAGIC_SYSRQ was not enabled, the default action of this key is to return to the previous console.)

If you press this key, do not release it, and hit another key, a corresponding action is performed. The action is performed whether anybody is logged in or not, is root or not. For the details, see `drivers/char/sysrq.c`. Since this feature is meant only for kernel hackers, that should suffice. Still, let me add a few remarks.

TOC

For the key r the keyboard mode is reset to K_XLATE. For the key k a SAK and console reset is done. For the key b the machine is rebooted immediately. (See, not something you want to have enabled on a production machine.) For the key o the power is turned off (when the machine is capable of that). For the key s an emergency sync is scheduled. For the key u an emergency read-only remount is scheduled. For the keys p,t,m various information is shown (namely the same information also shown for RAlt,RCtrl,RShift+ScrollLock). For the keys e,i,l all processes get a SIG_TERM or SIG_KILL, respectively; for l even the init process is killed. Digits set the log level. Anything else prints a short summary: `SysRq: unRaw saK Boot Off Sync Unmount showPc showTasks showMem loglevel0-8 tErm kIll killalL`.

Note: These are very dangerous actions! And they do not use your keymap - indeed, are meant for emergency cases where the state of your keymap, or even of the entire kernel, is uncertain. If you use a dvorak keyboard - bad luck! Most other people will be able to survive: the dangerous letters A,M,Q,W,Y,Z that are differently placed on English, French and German keyboards, are not used for actions.

---

9. How to get out of raw mode

If some program using K_RAW keyboard mode exits without restoring the keyboard mode to K_XLATE, then it is difficult to do anything - not even Ctrl-Alt-Del works. However, it is sometimes possible to avoid hitting the reset button. (And desirable as well: your users may get angry if you kill their Hack game by rebooting; you might also damage your file system.) Easy solutions involve logging in from another terminal or another machine and doing `kbd_mode -a`. The procedure below assumes that no X is running, that the display is in text mode, and that you are at your bash prompt, that you are using a US keyboard layout, and that your interrupt character is Ctrl-C.

Step 1. Start X. As follows: press 2 (and don't release), press F12 (and don't release) and immediately afterwards press = . This starts X. (Explanation: if a key press produces keycode K, then the key release produces keycode K+128. Probably your shell does not like these high characters, so we avoid generating them by not releasing any key. However, we have to be quick, otherwise key repeat starts. The digit 2 produces a Ctrl-C that discards previous junk, the F12 produces an X and the = a Return.) Probably your screen will be grey now, since no `.xinitrc` was specified. However, Ctrl-Alt-Fn will work and you can go to another VT. (Ctrl-Alt-Backspace also works, but that exits X, and gets you back into the previous state, which is not what you want.)

Step 2. Setup to change the keyboard mode. (For example, by `sleep 5; kbd_mode -a`.)

Step 3. Leave X again. Alt-Fx (often Alt-F7) brings you back to X, and then Ctrl-Alt-Backspace exits X. Within 5 seconds your keyboard will be usable again.

If you want to prepare for the occasion, then make `\215A\301` (3 symbols) an alias for `kbd_mode -a`. Now just hitting = F7 = (3 symbols) will return you to sanity.

10. The keyboard LEDs

1. There are per-tty keyboard flags: each VC has its own NumLock, CapsLock, ScrollLock. By default these keyboard flags are shown in the LEDs. The usual way to change them is by pressing the corresponding key. (Side remark: pressing the NumLock key when in application key mode will not change the NumLock status, but produce an escape sequence. If you want the NumLock key to always change the Numlock status, bind it to Bare_Num_Lock.)

2. Next, there are per-tty default keyboard flags, to initialize the keyboard flags when a reset occurs. Thus if you want NumLock on all the time, that is possible. The usual way to change them is by `` `setleds -D ...'. ``

3. There is the possibility that the leds do not reflect the keyboard flags, but something else.

3A. This something else can be three bits somewhere in the kernel - which can be used if you want to monitor some hardware or software status bit(s). If you want this, edit the kernel source to call `register_leds()` somewhere.

3B. This something else can also be whatever some user program wants to show in the LEDs. Thus, people who like such things can make nice patterns of lights. If you want this, use the KDSETLED ioctl.

This latter use is not per-tty, but the choice between former and latter use is per-tty.

Summarizing: Each tty has a flag `kbd->ledmode`. If this has the value LED_SHOW_FLAGS then the keyboard flags (NumLock etc.) of that tty are shown. If this has the value LED_SHOW_MEM then three selected memory addresses are shown. If this has the value LED_SHOW_IOCTL then the leds show whatever value was last assigned to them using the KDSETLED ioctl.

One may add that X uses ioctl's to set the LEDs, but fails to reset its VT when it exits, so after using X there may be one VT that is not in the default LED_SHOW_FLAGS state. This can be fixed by doing `` `setleds -L' `` on that VT. See setleds(1).

11. The TERM variable

Many programs use the `TERM` variable and the database `/etc/termcap` or `/usr/lib/terminfo/*` to decide which strings to send for clear screen, move cursor, etc., and sometimes also to decide which string is sent by the users backspace key, function keys etc. This value is first set by the kernel (for the console). Usually, this variable is re-set by `getty`, using `/etc/ttytype` or the argument specified in `/etc/inittab`. Sometimes, it is also set in `/etc/profile`.

Older systems use `TERM=console` or `TERM=con80x25`. Newer systems (with ncurses 1.8.6) use the more specific `TERM=linux` or `TERM=linux-80x25`. However, old versions of `setterm` test for `TERM=con*` and hence fail to work with `TERM=linux`.

Since kernel version 1.3.2, the kernel default for the console is `TERM=linux`.

If you have a termcap without entry for linux, add the word linux to the entry for the console:

```
console|con80x25|linux:\
```

and make `/usr/lib/terminfo/l/linux` a copy of or symbolic link to `/usr/lib/terminfo/c/console`.

11.1 Terminfo

The terminfo entry for the linux console from ncurses 1.8.6 misses the entry `kich1=\E[2~`, needed by some programs. Edit the file and `tic` it.

12. How to make other programs work with non-ASCII chars

In the bad old days this used to be quite a hassle. Every separate program had to be convinced individually to leave your bits alone. Not that all is easy now, but recently a lot of gnu utilities have learned to react to `LC_CTYPE=iso_8859_1` or `LC_CTYPE=iso-8859-1`. Try this first, and if it doesn't help look at the hints below. Note that in recent versions of libc the routine setlocale() only works if you have installed the locale files (e.g. in `/usr/lib/locale`).

First of all, the 8-th bit should survive the kernel input processing, so make sure to have `stty cs8 -istrip -parenb` set.

A. For `emacs` the details strongly depend on the version. The information below is for version 19.34. Put lines

```
(set-input-mode nil nil 1)
(standard-display-european t)
(require 'iso-syntax)
```

into your `$HOME/.emacs`. The first line (to be precise: the final 1) tells `emacs` not to discard the 8-th bit from input characters. The second line tells `emacs` not to display non-ASCII characters as octal escapes. The third line specifies the syntactic properties and case conversion table for the Latin-1 character set These last two lines are superfluous if you have something like `LC_CTYPE=ISO-8859-1` in your environment. (The variable may also be `LC_ALL` or even `LANG`. The value may be anything with a substring `88591' or `8859-1' or `8859_1'.)

This is a good start. On a terminal that cannot display non-ASCII ISO 8859-1 symbols, the command

```
(load-library "iso-ascii")
```

will cause accented characters to be displayed comme {,c}a. If your keymap does not make it easy to produce non-ASCII characters, then

```
(load-library "iso-transl")
```

will make the 2-character sequence Ctrl-X 8 a compose character, so that the 4-character sequence Ctrl-X 8 , c produces c-cedilla. Very inconvenient.

The command

```
(iso-accents-mode)
```

will toggle ISO-8859-1 accent mode, in which the six characters ', `, ", ^, ~, / are dead keys modifying the following symbol. Special combinations: ~c gives a c with cedilla, ~d gives an Icelandic eth, ~t gives

an Icelandic thorn, "s gives German sharp s, /a gives a with ring, /e gives an a-e ligature, ~< and ~> gives guillemots, ~! gives an inverted exclamation mark, ~? gives an inverted question mark, and " gives an acute accent. This is the default mapping of accents. The variable `iso-languages` is a list of pairs (language name, accent mapping), and a non-default mapping can be selected using

```
(iso-accents-customize LANGUAGE)
```

Here LANGUAGE can be one of `"portuguese"`, `"irish"`, `"french"`, `"latin-2"`, `"latin-1"`.

Since the Linux default compose character is Ctrl-. it might be convenient to use that everywhere. Try

```
(load-library "iso-insert.el")
(define-key global-map [?\C-.] 8859-1-map)
```

The latter line will not work under `xterm`, if you use `emacs -nw`, but in that case you can put

```
XTerm*VT100.Translations:       #override\n\
      Ctrl <KeyPress> . : string("\0308")
```

in your `.Xresources`.)

B. For `less`, put `LESSCHARSET=latin1` in the environment. This is also what you need if you see `\255` or `<AD>` in `man` output: some versions of `less` will render the soft hyphen (octal 0255, hex 0xAD) this way when not given permission to output Latin-1.

C. For `ls`, give the option `-N`. (Probably you want to make an alias.)

D. For `bash` (version 1.13.*), put

```
set meta-flag on
set convert-meta off
```

and, according to the Danish HOWTO,

```
set output-meta on
```

into your `$HOME/.inputrc`.

E. For `tcsh`, use

```
setenv LANG     US_en
setenv LC_CTYPE iso_8859_1
```

If you have nls on your system, then the corresponding routines are used. Otherwise `tcsh` will assume iso_8859_1, regardless of the values given to LANG and LC_CTYPE. See the section NATIVE LANGUAGE SYSTEM in tcsh(1). (The Danish HOWTO says: `setenv LC_CTYPE ISO-8859-1; stty pass8`)

F. For `flex`, give the option `-8` if the parser it generates must be able to handle 8-bit input. (Of course it must.)

G. For `elm`, set `displaycharset` to `ISO-8859-1`. (Danish HOWTO: `LANG=C` and `LC_CTYPE=ISO-8859-1`)

H. For programs using curses (such as `lynx`) David Sibley reports: The regular curses package uses the high-order bit for reverse video mode (see flag _STANDOUT defined in `/usr/include/curses.h`). However, `ncurses` seems to be 8-bit clean and does display iso-latin-8859-1 correctly.

I. For programs using `groff` (such as `man`), make sure to use `-Tlatin1` instead of `-Tascii`. Old versions of the program `man` also use `col`, and the next point also applies.

J. For `col`, make sure 1) that it is fixed so as to do `setlocale(LC_CTYPE,"");` and 2) put `LC_CTYPE=ISO-8859-1` in the environment.

K. For `rlogin`, use option `-8`.

L. For `joe`, `sunsite.unc.edu:/pub/Linux/apps/editors/joe-1.0.8-linux.tar.gz` is said to work after editing the configuration file. Someone else said: `joe`: Put the `-asis` option in `/isr/lib/joerc` in the first column.

M. For LaTeX: `\documentstyle[isolatin]{article}`. For LaTeX2e: `\documentclass{article}\usepackage{isolatin}` where `isolatin.sty` is available from ftp://ftp.vlsivie.tuwien.ac.at/pub/8bit.

A nice discussion on the topic of ISO-8859-1 and how to manage 8-bit characters is contained in the file `grasp.insa-lyon.fr:/pub/faq/fr/accents` (in French). Another fine discussion (in English) can be found in ftp.vlsivie.tuwien.ac.at:/pub/8bit/FAQ-ISO-8859-1, which is mirrored in rtfm.mit.edu:pub/usenet-by-group/comp.answers/character-sets/iso-8859-1-faq.

If you need to fix a program that behaves badly with 8-bit characters, one thing to keep in mind is that if you have a signed char type then characters may be negative, and using them as an array index will fail. Several programs can be fixed by judiciously adding (unsigned char) casts.

---

13. What precisely does XFree86-2.1 do when it initializes its keymap?

Since version 2.1, XFree86 will initialize its keymap from the Linux keymap, as far as possible. However, Linux had 16 entries per key (one for each combination of the Shift, AltGr, Ctrl, Alt modifiers) and presently has 256 entries per key, while X has 4 entries per key (one for each combination of Shift, Mod), so some information is necessarily lost.

First `X` reads the `Xconfig` file, where definitions of the LeftAlt, RightAlt, RightCtl, ScrollLock keys as Meta, ModeShift, Compose, ModeLock or ScrollLock might be found - see X386keybd(1), later XFree86kbd(1).

For Mod the LeftAlt key is taken, unless RightCtl was defined as ModeShift or ModeLock, in which case RightCtl is taken, or RightAlt was so defined, in which case RightAlt is taken. This determines how the 4 XFree86 meanings of a key are selected from the 16 Linux meanings. Note that Linux today does not distinguish by default between the two Ctrl keys or between the two Shift keys. `X` does distinguish.

Now the kernel keymap is read and the usually obvious corresponding X bindings are made. The bindings for the "action keys" Show_Memory, Show_State, Show_Registers, Last_Console, Console_n, Scroll_Backward, Scroll_Forward, Caps_On and Boot are ignored, as are the dead diacriticals, and the locks (except for ShiftLock), and the "ASCII-x" keys.

Next, the definitions in the `Xconfig` file are used. (Thus, a definition of Compose in `Xconfig` will override its value as found in the Linux keymap.)

What happens to the strings associated with the function keys? Nothing, X does not have such a concept. (But it is possible to define strings for function keys in `xterm` - note however that the window manager gets the keys first.)

I don't know how to convince `xterm` that it should use the X keymap when Alt is pressed; it seems just to look at its resource `eightBitInput`, and depending on whether that is true or false either set the high order bit of the character, or generate an additional Escape character (just like setmetamode(1) does for the console).

14. Unusual keys and keyboards

The two keys PrintScrn/SysRq and Pause/Break are special in that they have two keycodes: the former has keycode 84 when Alt is pressed simultaneously, and keycode 99 otherwise; the latter has keycode 101 when Ctrl is pressed simultaneously, and keycode 119 otherwise. (Thus, it makes no sense to bind functions to Alt keycode 99 or Ctrl keycode 119.)

If you have strange keys, that do not generate any code under Linux (or generate messages like "unrecognized scancode"), and your kernel is 1.1.63 or later, then you can use setkeycodes(1) to tell the kernel about them. They won't work under X, however. Once they have gotten a keycode from `setkeycodes`, they can be assigned a function by `loadkeys`.

15. Examples of use of loadkeys and xmodmap

Switching Caps Lock and Control on the keyboard (assuming you use keymaps 0-15; check with `dumpkeys | head -1`)

```
% loadkeys
keymaps 0-15
keycode 58 = Control
keycode 29 = Caps_Lock
%
```

Switching them under X only:

```
% xmodmap .xmodmaprc
```

where `.xmodmaprc` contains lines

```
remove Lock = Caps_Lock
remove Control = Control_L
keysym Control_L = Caps_Lock
keysym Caps_Lock = Control_L
add Lock = Caps_Lock
add Control = Control_L
```

What is this about the key numbering? Backspace is 14 under Linux, 22 under X? Well, the numbering can best be regarded as arbitrary; the Linux number of a key can be found using showkey(1), and the X number using xev(1). Often the X number will be 8 more than the Linux number.

Something else people like to change are the bindings of the function keys. Suppose that you want to make F12 produce the string "emacs ". Then

```
% loadkeys
keycode 88 = F12
string F12 = "emacs "
%
```

will do this. More explicitly, the procedure is like this: (i) find the keycodes of the keys to be remapped, using showkey(1). (ii) save the current keymap, make a copy and edit that:

```
% dumpkeys > my_keymap
% cp my_keymap trial_keymap
% emacs trial_keymap
% loadkeys trial_keymap
%
```

TOC

The format of the table can be guessed by looking at the output of `dumpkeys`, and is documented in keytables(5). When the new keymap functions as desired, you can put an invocation

```
loadkeys my_new_keymap
```

in `/etc/rc.local` or so, to execute it automatically at boot-up. Note that changing modifier keys is tricky, and a newbie can easily get into a situation only an expert can get out of.

The default directory for keymaps is `/usr/lib/kbd/keytables`. The default extension for keymaps is `.map`. Thus, `loadkeys uk` would probably load `/usr/lib/kbd/keytables/uk.map`.

(On my machine) `/dev/console` is a symbolic link to `/dev/tty0`, and the kernel regards `/dev/tty0` as a synonym for the current VT. XFree86 1.3 changes the owner of `/dev/tty0`, but does not reset this after finishing. Thus, `loadkeys` or `dumpkeys` might fail because someone else owns `/dev/tty0`; in such a case you might run X first. Note that you cannot change keyboard mappings when not at the console (and not superuser).

15.1 `I can use only one finger to type with'

"Can the Shift, Ctrl and Alt keys be made to behave as toggles?"

Yes, after saying

```
% loadkeys
keymaps 0-15
keycode 29 = Control_Lock
keycode 42 = Shift_Lock
keycode 56 = Alt_Lock
%
```

the left Control, Shift and Alt keys will act as toggles. The numbers involved are revealed by showkey (and usually are 29, 97, 42, 54, 56, 100 for left and right control, shift and alt, respectively), and the functions are Control_Lock, Shift_Lock, Alt_Lock, ALtGr_Lock.

"What about `sticky' modifier keys?"

Since version 1.3.33, the kernel knows about `sticky' modifier keys. These act on the next key pressed. So, where one earlier needed the 3-symbol sequence Shift_Lock a Shift_Lock to type `A', one can now use the 2-symbol sequence SShift_Lock a. Versions of the kbd package older than 0.93 do not yet include code for these sticky modifiers, and have to invoke them using their hexadecimal codes. For example,

```
% loadkeys
keymaps 0-15
keycode 54 = 0x0c00
keycode 97 = 0x0c02
```

```
keycode 100 = 0x0c03
%
```

will make the right Shift, Ctrl, Alt sticky versions of the left ones. >From 0.93 on you can say

```
% loadkeys
keymaps 0-15
keycode 54 = SShift
keycode 97 = SCtrl
keycode 100 = SAlt
%
```

to obtain the same result. This will allow you to type Ctrl-Alt-Del in three keystrokes with one hand.

The keymaps line in these examples should cover all keymaps you have in use. You find what keymaps you have in use by

```
% dumpkeys | head -1
```

16. Changing the video mode

As far as I know there are 6 ways to change resolution:

1. At compile time: change the line

```
SVGA_MODE=      -DSVGA_MODE=NORMAL_VGA
```

in `/usr/src/linux/Makefile`.

1A. After compilation: use `rdev -v` - a terrible hack, but it exists.

2. At boot time: put `vga=ask` in the lilo config file, and lilo will ask you what video mode you want. Once you know, put `vga=`mypreference.

3. At run time: A. Use the `resizecons` command. (This is a very primitive wrapper around the VT_RESIZE ioctl.) B. Use the `SVGATextMode` command. (This is a less primitive wrapper around the VT_RESIZE ioctl.)

4. Not "on the console": Under `dosemu`, or with svgalib etc. you can change the hardware video mode without the console driver being aware of it. Sometimes this is useful in getting `resizecons` or `SVGATextMode` set up: use `dosemu` and some DOS program to get into the desired videomode, dump (say from another VT) the contents of all video hardware registers, and use that in the initialization that `resizecons` and `SVGATextMode` require. In some cases where the video mode has gotten into some unusable state, starting `dosemu`, relying on the BIOS to set up the video mode, and then killing `dosemu` (with `kill -9`), is the easiest way to get into shape again.

16.1 Instructions for the use of resizecons

Get svgalib and compile the program `restoretextmode`. Boot up your machine in all possible video modes (using `vga=ask` in the lilo config file), and write the video hardware register contents to files CxR (C=cols, R=rows), e.g., 80x25, 132x44, etc. Put these files in `/usr/lib/kbd/videomodes`. Now `resizecons 132x44` will change videomode for you (and send SIGWINCH to all processes that need to know about this, and load another font if necessary).

At present, `resizecons` only succeeds when there is memory enough for both the old and the new consoles at the same time.

17. Changing the keyboard repeat rate

At startup, the Linux kernel sets the repeat rate to its maximal value. For most keyboards this is reasonable, but for some it means that you can hardly touch a key without getting three copies of the corresponding symbol. Use the program kbdrate(8) to change the repeat rate, or, if that doesn't help, edit or remove the section

```
     ! set the keyboard repeat rate to the max

         mov     ax,#0x0305
         xor     bx,bx           ! clear bx
         int     0x16
```

of `/usr/src/linux/[arch/i386/]boot/setup.S`.

18. Scrolling

There are two ways to get a screen to scroll. The first, called `hard scrolling', is to leave the text in video memory as it is, but change the viewing origin. This is very fast. The second, called `soft scrolling', involves moving all screen text up or down. This is much slower. The kernel console driver will write text starting at the top of the video memory, continuing to the bottom, then copy the bottom part to the top again, and continue, all the time using hard scrolling to show the right part on the screen. You can scroll back until the top op the video memory by using Shift-PageUp (the grey PageUp) and scroll down again using Shift-PageDown (the grey PageDown), assuming a default keymap. The amount of scrollback is thus limited to the amount of video memory you happen to have and you cannot increase this amount. If you need more scrollback, use some program that buffers the text, like `less` or `screen` - by using a buffer on disk you can go back to what you did last week. (One can set the amount of scrollback for `xterm` by adding a line like `XTerm*saveLines: 2500` in `.Xresources`.)

Upon changing virtual consoles, the screen content of the old VT is copied to kernel memory, and the screen content of the new VT is copied from kernel memory to video memory. Only the visible screen is copied, not all of video memory, so switching consoles means losing the scrollback information.

Sometimes, hard scrolling is undesirable, for example when the hardware does not have the possibility to change viewing origin. The first example was a Braille machine that would render the top of video memory in Braille. There is a kernel boot-time option `no-scroll` to tell the console driver not to use hard scrolling. See bootparam(7).

19. Screensaving

`setterm -blank` *nn* will tell the console driver to blank the screen after *nn* minutes of inactivity. (With *nn* = 0, screensaving is turned off. In some old kernels this first took effect after the next keyboard interrupt.)

The `s` option of xset(1) will set the X screensaving parameters: `xset s off` turns off the screensaver, `xset s 10` blanks the screen after 10 minutes.

The video hardware powersaving modes can be enabled/disabled using the `setvesablank` program given in the starting comment of `/usr/src/linux/drivers/char/vesa_blank.c`.

20. Screen dumps

`setterm -dump` *N* will dump the contents of the screen of `/dev/tty`*N* to a file `screen.dump` in the current directory. See setterm(1).

The current contents of the screen of `/dev/tty`*N* can be accessed using the device `/dev/vcs`*N* (where `vcs' stands for `virtual console screen'). For example, you could have a clock program that displays the current time in the upper right hand corner of the console screen (see the program `vcstime` in `kbd-0.95.tar.gz`). Just dumping the contents goes with `cat /dev/vcs`*N*. These device files `/dev/vcs`*N* do not contain newlines, and do not contain attributes, like colors. From a program it is usually better to use `/dev/vcsa`*N* (`virtual console screen with attributes') instead - it starts with a header giving the number of rows and columns and the location of the cursor. See vcs(4).

21. Some properties of the VT100 - application key mode

: Sometimes my cursor keys or keypad keys produce strange codes?

When the terminal is in application cursor key mode the cursor keys produce Esc O x and otherwise Esc [x where x is one of A,B,C,D. Certain programs put the terminal in application cursor key mode; if you kill them with `kill -9`, or if they crash, then the mode will not be reset.

```
% echo -e '\033c'
```

resets all properties of the current VC. Just changing the cursor application key mode is done by

```
% echo -e '\033[?1h'
```

(set) and

```
% echo -e '\033[?1l'
```

(clear).

When the terminal is in application keypad key mode the keypad keys produce Esc O y and otherwise Esc [z ~ for certain y and z. Setting application keypad key mode is done by

```
% echo -e '\033='
```

and

```
% echo -e '\033>'
```

clears it again.

22. Hardware incompatibility

Several people have noticed that they lose typed characters when a floppy disk is active. It seems that this might be a problem with Uni-486WB motherboards. (Please mail me (`aeb@cwi.nl`) to confirm [yes, I have the same problem], deny [no, nothing wrong with my Uni-486WB], modify [My Xyzzy machine has the same problem].)

Tjalling Tjalkens (`tjalling@ei.ele.tue.nl`) reports very similar problems with "a no-brand GMB-486 UNP Vesa motherboard with AMD 486DX2-66 CPU" - during floppy activity some keystrokes are lost, during floppy tape streamer (Conner C 250 MQ) activity many keystrokes are lost.

Some people experience sporadic lockups - sometimes associated to hard disk activity or other I/O.

Ulf Tietz (`ulf@rio70.bln.sni.de`) wrote: `I have had the same problems, when I had my motherboard tuned too fast. So I reset all the timings (CLK, wait statements etc) to more conventional values, and the problems are gone.'

Bill Hogan (`bhogan@crl.com`) wrote: `If you have an AMI BIOS, you might try setting the Gate A20 emulation parameter to "chipset" (if you have that option). Whenever I have had that parameter set to any of the other options on my machine ("fast", "both", "disabled") I have had frequent keyboard lockups.'

23. Copyright

Copyright (c) 1993-1998 by Andries Brouwer. This document may be distributed under the terms set forth in the LDP license at http://sunsite.unc.edu/LDP/COPYRIGHT.html or ftp://www.win.tue.nl/pub/linux/LDP/COPYRIGHT.txt.

Additions and corrections are welcome. Andries Brouwer - `aeb@cwi.nl`

RedHat Linux KickStart HOWTO

Martin Hamilton <martinh@gnu.org>

v0.2, 11 January 1999

This HOWTO briefly describes how to use the RedHat Linux KickStart system to rapidly install large numbers of identical Linux boxes. For advanced users, we describe how to modify the KickStart installation procedure to do your own thing, and give a quick guide to building RPM packages of your own.

1. Copyright

2. Homepage

3. Introduction

4. Prerequisites

5. Setting up a boot floppy

6. The KickStart config file

- 6.1 System info
- 6.2 Packages to install
- 6.3 Post-installation shell commands

7. Installation itself

8. Mounting the boot/supp disks

9. Modifying the RedHat installer

10. FAQs/Wish list

11. Credits

12. Appendix A - Configuring BOOTP/DHCP and NFS

13. Appendix B - Making your own RPMs

14. Appendix C - Munging your own RPMs into the distribution

1. Copyright

Copyright (c) 1998 Martin Hamilton, All rights reserved. This is free documentware; you can redistribute it and/or modify it under the terms of version 2 or later of the GNU General Public License.

2. Homepage

If you got this document from a Linux HOWTO mirror site or a CD-ROM, you might want to check back to the KickStart HOWTO home page to see if there's a newer version around.

3. Introduction

RedHat Linux version 5 comes with a a little-known (and until now, not hugely documented) feature called *KickStart*. This lets you automate most/all of a RedHat Linux installation, including:

- Language selection
- Network configuration and distribution source selection
- Keyboard selection
- Boot loader installation (e.g. lilo)
- Disk partitioning and filesystem creation
- Mouse selection
- X Window system server configuration
- Timezone selection
- Selection of an (initial) root password
- Which packages to install

Eagle eyed RedHat users will probably have realised by now that these are essentially the main steps involved in the manual installation of a RedHat Linux system. KickStart makes it possible for you to script the regular installation process, by putting the information you would normally type at the keyboard into a configuration file.

But wait - there's more :-)

Having finished the normal installation process, KickStart also lets you specify a list of shell level commands which you would like to be executed. This means that you can automatically install extra local software not distributed as part of RedHat Linux (yes, there are even more free software programs than the ones you get with the RedHat distribution. Some can't be distributed by RedHat on legal grounds, e.g. the `ssh` and `PGP` encryption systems) and carry out any tidying up you may need to do in order to get a fully operational system.

4. Prerequisites

There are two approaches to a KickStart install - one is to simply copy your KickStart configuration file to a RedHat boot floppy. The other is to use a regular boot floppy and get your KickStart config file off the network.

In both cases, you'll need:

1. Intel (i386) class machines - KickStart appears to only work on these at the time of writing.
2. KickStart config file - we'll talk about this in the next section.
3. RedHat boot disk - preferably from the *updates* directory, to take advantage of any fixes/driver updates.
4. DNS entries for the IP addresses you'll be using - optional, but will stop the installation from prompting you for your machine's domain name.

If you want to fetch your config file over the network, you'll need to export it via NFS - this is the only access method supported at the moment. The config file lets you specify a different NFS server to fetch the RedHat distribution itself from.

You can configure a static IP address for your machine - e.g. a special one reserved for KickStart installations. Alternatively, if you don't want to hard code an IP address in the config file you can tell KickStart to use a BOOTP/DHCP server to fetch this. Some servers will allocate new addresses in a given range automatically, e.g. the CMU BOOTP server with dynamic addressing extensions.

More information on NFS and BOOTP/DHCP is in Appendix A.

5. Setting up a boot floppy

Essentially, all you have to do is copy your KickStart config file to */ks.cfg* on the RedHat boot floppy, e.g.

```
mcopy ks.cfg a:
```

However - the RedHat boot floppy is pretty full, and you may find that you have to delete some of the other files to make room for the KickStart config file. I was able to make enough room for mine by deleting the various message files normally displayed by the `SYSLINUX` boot loader, e.g.

```
mdel a:\*.msg
```

Another approach would be to throw away the drivers for some of the hardware you don't have - see the section on modifying the boot floppy below.

You may also want to edit *syslinux.cfg*, the `SYSLINUX` config file. This also lives in the top level directory of the RedHat boot floppy. For instance, the following *syslinux.cfg* will cause KickStart mode to be entered into automatically as the machine boots up, without the normal delay:

```
default ks
prompt 0
label ks
  kernel vmlinuz
  append ks=floppy initrd=initrd.img
```

Note that you almost probably want to base your boot and supplementary floppies on the most recent disk images available in the *updates/i386* on your local RedHat mirror site. Older images may be buggy or have driver support for less hardware.

6. The KickStart config file

There are three main sections to the config file:

1. System info, e.g. disk partitioning and network config
2. RedHat packages to install
3. Post-installation shell commands to run

There are some other possibilities which we won't talk about here, but **might** work. For more information check out the sample KickStart config in *misc/src/install/ks.samp* and *doc/README.ks* under the top level *i386* RedHat distribution directory on your CD-ROM or local RedHat mirror site.

6.1 System info

The available directives which I've been using are:

lang

Language configuration, e.g. for English

```
lang en
```

network

Network configuration, e.g. to use BOOTP/DHCP

```
network --bootp
```

nfs

NFS server and directory to install from, e.g.

```
nfs --server chicken.swedish-chef.org /mnt/cdrom
```

to use the NFS server *chicken.swedish-chef.org* and try to mount the RedHat distribution from the directory */mnt/cdrom*.

keyboard

Select keyboard type, e.g. for UK keyboards

```
keyboard uk
```

zerombr

Clear the Master Boot Record - removes any existing operating system boot loader from your disk

clearpart

Clear existing partitions - e.g. to remove all existing disk partitions prior to installation

```
clearpart --all
```

part

TOC

Partition the disk, e.g. to make a root filesystem of 500MB

```
part / --size 500
```

install

Make a fresh installation of RedHat Linux.

mouse

Set the mouse being used, e.g. for a PS/2 or compatible "bus mouse"

```
mouse ps/2
```

timezone

Set the timezone, e.g. for local time in the UK

```
timezone --utc Europe/London
```

rootpw

Set the initial root password, based on a previously derived encrypted password

```
rootpw --iscrypted XaacoeGPmf/A.
```

lilo

Install the LILO boot loader, e.g. in the Master Boot Record

```
lilo --location mbr
```

%packages

Packages to install - see below.

%post

Post-installation shell commands - see below.

Note that the directory where KickStart is looking for the RedHat distribution should have a subdirectory *RedHat*, which contains the RedHat distribution tree for the platform in question. In the above example, we should see something like the following files and directories:

```
/mnt/cdrom/RedHat
/mnt/cdrom/RedHat/base
/mnt/cdrom/RedHat/contents
/mnt/cdrom/RedHat/i386
/mnt/cdrom/RedHat/instimage
/mnt/cdrom/RedHat/RPMS
/mnt/cdrom/RPM-PGP-KEY
```

If you're installing off a CD-ROM rather than off the network, the contents should look something like this:

```
RedHat
RedHat/base
RedHat/contents
RedHat/i386
```

```
RedHat/instimage
RedHat/RPMS
RPM-PGP-KEY
```

If you have the RedHat distribution for multiple architectures (e.g. on an NFS server - they're too big to fit more than one architecture's version onto a single CD-ROM), you'll notice that each distribution has the same files and directories under a subdirectory, e.g.

```
alpha/RPM-PGP-KEY
i386/RPM-PGP-KEY
sparc/RPM-PGP-KEY
```

There should be a file `architecture`/Redhat/`architecture`, e.g. *i386/Redhat/i386*.

If you want to create your own encrypted passwords, it's very easy using Perl, e.g.

```
% perl -e 'print crypt("schmurrdegurr", "Xa") . "\n";'p
```

Other options (or mooted options), which I've not tried:

cdrom

Install off CD-ROM rather than network.

device

Explicitly declare device details, e.g.

```
device ethernet 3c509 --opts "io=0x330, irq=7"
```

Alternative values of `device` include `scsi` for SCSI controllers and `cdrom` for proprietary CD-ROM drives.

upgrade

Upgrade an existing installation rather than make a fresh installation.

xconfig

Configure X Window server, graphics card and monitor. e.g.

```
xconfig --server "Mach64" --monitor "tatung cm14uhe"
```

I've not delved too deeply into this last one, because I'm not ever planning to run X on the console of any of my KickStarted machines. I'm told that running `xconfig` within KickStart itself is a bit flaky, but the same functionality is also available from the command line via `Xconfigurator` - so you might be best off leaving this to the post-installation script.

Here's how this first part of a KickStart config file looks when we put all the bits together:

```
lang en
network --static --ip 198.168.254.253 --netmask 255.255.255.0
  --gateway 198.168.254.1 --nameserver 198.168.254.2
nfs --server chicken.swedish-chef.org /mnt/cdrom
keyboard uk
zerombr yes
clearpart --all
part / --size 500
```

TOC

```
part swap --size 120
install
mouse ps/2
timezone --utc Europe/London
rootpw --iscrypted XaacoeGPmf/A.
lilo --location mbr
```

Note that some of the RedHat documentation refers to an invocation of the `network` directive which doesn't actually work in practice: `network --option`. The correct invocation is to put `network` followed by `--static`, `--bootp` or `--dhcp`. Be aware that the BOOTP and DHCP options are different - to the extent that they even use different code.

You can add the `--grow` parameter to a `part` directive to indicate that it's OK to grow the partition beyond the size you specify. It probably only makes sense to have one partition tagged with `--grow`.

6.2 Packages to install

The start of the packages section of the KickStart config file is indicated by the presence of a `%packages` directive on a line of its own. This is followed by one or both of two types of package specifier - individual packages may be installed by giving the name of their RPM (excluding the version and platform information), and groups of packages may be installed by giving their group name.

Here's a sample `packages` section for a KickStart config file:

```
%packages
@ Base
netkit-base
bind-utils
ncftp
rdate
tcp_wrappers
traceroute
cmu-snmp
```

So, what are these groups ? Well, there are a number of groups defined by default in a file called *base/comps* under the RedHat distribution's top level directory. Here are the ones which were current at the time of writing:

- Base
- Printer Support
- X Window System
- Mail/WWW/News Tools
- DOS/Windows Connectivity
- File Managers
- Graphics Manipulation
- X Games
- Console Games
- X multimedia support
- Console Multimedia
- Print Server
- Networked Workstation

TOC

- Dialup Workstation
- News Server
- NFS Server
- SMB (Samba) Connectivity
- IPX/Netware(tm) Connectivity
- Anonymous FTP/Gopher Server
- Web Server
- DNS Name Server
- Postgres (SQL) Server
- Network Management Workstation
- TeX Document Formatting
- Emacs
- Emacs with X windows
- C Development
- Development Libraries
- C++ Development
- X Development
- Extra Documentation

You'll notice that they correspond to the various configurations which you're prompted for during a manual installation. Note that some of the packages in a given package group are duplicated in other groups, and that you can install multiple groups of packages without this causing problems. Each group's entry in the *comps* listing looks similar to this:

```
0 Extra Documentation
sag
lpg
howto
faq
man-pages
end
```

It seems that groups with a *1* next to their name (the first line above) are selected for installation by default. You can customise the Linux installation process even further by creating your own groups or redefine existing ones by editing this file.

6.3 Post-installation shell commands

This is probably the best feature of all, and something which there is no direct equivalent to in the manual installation process. What we can do here is specify a sequence of shell level commands which should be executed after the main installation (disk partitioning, package installation, and so on) is complete.

The beginning of this section is signified by the `%post` directive in the KickStart config file. In what follows you can take advantage of all of the utilities which have been installed on your newly built Linux system, e.g.

```
%post
ln -s /etc/rc.d/init.d /etc/init.d
```

```
ln -s /etc/rc.d/rc.local /etc/rc.local
ln -s /usr/bin/md5sum /usr/bin/md5
ln -s /usr/bin/perl /usr/local/bin/perl
chmod ug-s /bin/linuxconf
mkdir /var/tmp/tmp
perl -spi -e 's!image=/boot/vmlinuz-.*!image=/boot/vmlinuz!' /etc/lilo.conf
rm /etc/rc.d/rc*.d/*sendmail
```

You can also use I/O redirection and here documents:

```
cat <<EOF >>/etc/passwd
squid:*:102:3500:Squid Proxy:/usr/squid:/bin/bash
EOF

cat <<EOF >>/etc/group
cache:x:3500:
EOF
```

Modify the run-time startup scripts:

```
cat <<EOF >>/etc/rc.local
echo 8192 > /proc/sys/kernel/file-max
echo 32768 > /proc/sys/kernel/inode-max

[ -x /usr/sbin/sshd ] && /usr/sbin/sshd
[ -x /usr/sbin/cfd ] && /usr/sbin/cfd

EOF
```

Set up *crontab* entries:

```
cat <<EOF >/tmp/crontab.root
# Keep the time up to date
0,15,30,45 * * * * /usr/sbin/ntpdate -s eggtimer 2>&1 >/dev/null
# Recycle Exim log files
1 0 * * * /usr/exim/bin/exicyclog
# Flush the Exim queue
0,15,30,45 * * * * /usr/exim/bin/exim -q
EOF

crontab /tmp/crontab.root
rm /tmp/crontab.root
```

And even install other RPMs which you made yourself:

```
rpm -i ftp://chicken.swedish-chef.org/rpms/squid.rpm
rpm -i ftp://chicken.swedish-chef.org/rpms/ssh.rpm
rpm -i ftp://chicken.swedish-chef.org/rpms/exim.rpm
rpm -i ftp://chicken.swedish-chef.org/rpms/cfengine.rpm
rpm -i ftp://chicken.swedish-chef.org/rpms/linux.rpm

ssh-keygen -b 1024 -f /etc/ssh_host_key -N ""
```

```
depmod -a
```

Note that you can achieve the same effect by making your own RPMs containing the commands you want executed - see below for more information. Give them a carefully chosen name and you can force them to be installed first (e.g. name starts with 'aaa') or last (e.g. name starts with 'zzz').

Be aware that a less painful way of doing root crontab entries is to create them as files in one or more of the directories */etc/cron.hourly*, */etc/cron.daily*, */etc/cron.weekly* and */etc/cron.monthly*.

More information about making your own RPMs is available in Appendix B.

---

7. Installation itself

Boot the to-be-installed machine off your RedHat boot floppy as usual, but instead of pressing `RETURN` at the `SYSLINUX` prompt, type `linux ks`.

If you're lucky, this will be all you have to type!

If you customised your RedHat boot floppy as outlined above, you won't even need to do this bit :-)

Since we're really just automating the normal steps involved in a RedHat installation, the normal dialogs may appear if/when KickStart gets confused about what to do next. The most likely case is that your network interface won't be detected automatically, and you'll be prompted for its IRQ and I/O address space. KickStart tends to need help for ISA bus cards, but detects PCI bus cards automatically.

You can keep an eye on what KickStart is doing by by switching virtual consoles as usual:

- Alt-F1 - installation dialog
- Alt-F2 - shell prompt
- Alt-F3 - install log (messages from install program)
- Alt-F4 - system log (messages from kernel, etc.)
- Alt-F5 - other messages

8. Mounting the boot/supp disks

The RedHat boot disk *boot.img* is in MS-DOS format, using the `SYSLINUX` program to boot up. The supplementary disk *supp.img* is a Linux `ext2` filesystem. If you have support for the loopback filesystem in your Linux kernel, you can mount both of these files in your filesystem and hack at them:

```
# mkdir -p /mnt/boot /mnt/supp
# mount -o loop -t msdos boot.img /mnt/boot
# mount -o loop supp.img /mnt/supp
```

Now you should be able to see and manipulate the files on the boot and supplementary disk under */mnt/boot* and */mnt/supp* respectively. Phew! Note that older versions of `mount` may not be able to handle the `-o loop` option. In these cases you'll need to explicitly use `losetup` to configure the loopback device for each file, e.g.

```
# losetup /dev/loop0 boot.img
# mount -t msdos /dev/loop0 /mnt/boot
```

You might also need to explicitly use the `-t ext2` option when mounting an `ext2` filesystem like the one on the supplementary disk. But, it looks like people with modern Linux distributions shouldn't have to worry about this.

Of course, if you don't want to mess around too much, you can cut a corner and manipulate actual floppy disks rather than these floppy disk images. If time is important, you'll probably prefer to use the loopback devices, since you can hack around with the disk images without incurring the latency associated with a genuine floppy disk read/write.

9. Modifying the RedHat installer

If you want to mess around with the installation procedure itself, the source code can be found on the RedHat CD-ROM or your local RedHat mirror site. It's in *misc/src/install* under the *i386* distribution top level directory.

If you examine the RedHat boot disk you'll see that, in addition to the Linux kernel *vmlinuz*, there's a large file *initrd.img*:

```
- -rwxr-xr-x   1 root     root          559 May 11 15:48 boot.msg
- -rwxr-xr-x   1 root     root          668 May 11 15:48 expert.msg
- -rwxr-xr-x   1 root     root          986 May 11 15:48 general.msg
- -rwxr-xr-x   1 root     root       968842 May 11 15:48 initrd.img
- -rwxr-xr-x   1 root     root         1120 May 11 15:48 kickit.msg
- -r-xr-xr-x   1 root     root         5352 May 11 15:48 ldlinux.sys
- -rwxr-xr-x   1 root     root          875 May 11 15:48 param.msg
- -rwxr-xr-x   1 root     root         1239 May 11 15:48 rescue.msg
- -rwxr-xr-x   1 root     root          402 May 11 15:48 syslinux.cfg
- -rwxr-xr-x   1 root     root       444602 May 11 15:48 vmlinuz
```

You guessed it, this is another `ext2` filesystem saved as a file - - but with a twist. It's actually compressed as well. You can uncompress it and then mount the result, e.g.

```
# gzip -dc /mnt/boot/initrd.img >/tmp/initrd.ext2
# mkdir /mnt/initrd
# mount -o loop /tmp/initrd.ext2 /mnt/initrd
```

Probably the most important part of this filesystem is the collection of loadable kernel modules which are included with the boot disk. If you need to merge in a new version of a driver, you'll need to either replace *vmlinuz* with a new kernel which has this statically linked, or replace it in the modules collection. What's more, you may need to throw other modules away to make room.

The modules collection is the file *modules/modules.cgz*. Wondering what that might be ? It's actually a compressed `cpio` archive, believe it or not! And you thought nobody used `cpio` any more... Actually RPM itself uses `cpio` internally, too. Here's how to hack around with it:

```
# gzip -dc /mnt/initrd/modules/modules.cgz >/tmp/modules.cpio
# cpio -itv <modules.cpio >modules.listing
# mkdir modules
# cpio -idumv <../modules.cpio
```

I don't believe that there is currently a way under Linux (at least in mainstream distributions) to transparently access compressed filesystems. Let me know if you know better!

If you change anything, remember to:

TOC

1. Use `cpio` to recreate the archive. How to do this is left as an exercise for the reader...
2. Use `gzip` to compress the resulting archive.
3. Copy it to */mnt/initrd*, or wherever you put the uncompressed *initrd.img* archive.
4. Unmount */mnt/initrd* (or whatever you called it).
5. Compress the new *initrd.img* using `gzip` again.
6. Copy the resulting archive onto the boot disk image - */mnt/boot/initrd.img* in our example.
7. Unmount the boot disk image, e.g. */mnt/boot*.

Finally, you can now create new boot floppies using this modified boot disk setup, e.g.

```
# cat boot.img >/dev/fd0
```

---

10. FAQs/Wish list

Q: After KickStart installation, my machine won't boot up. The BIOS complains with a message like `Missing operating system`.

A: Sounds like the partition with the root filesystem on isn't bootable. Use `fdisk` to toggle its bootable status.

Q: After the floppy boots, I get the message: `Error opening files for kickstart copy: File exists`.

A: Use a more recent version of *boot.img* and *supp.img* - look in the *updates* directory of your local RedHat mirror site. There was a bug in some older versions of these for RedHat 5.1.

Q: Can you have all outstanding patches (update RPMs) applied automatically too ? How ?

A1: Copy the RPMs you want installing to the RPMS directory from which the installation is going to take place, get rid of the older RPMs, and update the file *RedHat/base/hdlist* with the new RPM details. See Appendix C for a script from Eric Doutreleau to do this for you. If you do this yourself, remember to run *genhdlist* afterwards!

A2: Try this Perl script: patchup. This compares the RPMs your system has installed with those in a nominated directory and reports on the ones it thinks need updating. It can even install them for you if you trust it to.

A3: rpm2hml has a much more powerful (12MB of C code vs. a page of Perl!) version of A2.

Q: A single config file on the install server for all of the clients, perhaps as a fallback after trying *IPADDR-kickstart* ?

A1: Use the BOOTP/DHCP 'boot file' parameter *bf* to set the filename.

A2: Add a a record `bf=/kickstart/ks.cfg` to the relevant entry in */etc/bootptab*.

Q: More flexibility when things go wrong - e.g. prompt for alternate locations if distribution not found on CD-ROM.

A: ?

Q: Explicit exclusion of packages - e.g. everything apart from *sendmail*.

A: Rebuild the **BASE** package without sendmail.

Q: Choose which services are started automatically on boot-up by the run-level scripts under */etc/rc.d/*.

A: The *chkconfig* utility lets you configure which services are run automatically on boot-up. You can run this in your post-installation script section, e.g. to run *ypbind* in run levels 3, 4 and 5:

TOC

```
chkconfig --level 345 ypbind on
```

and it will start the ypbind level on the 345 level.

Q: When executing the shell commands in the `%post` section, bring any output up in another virtual console rather than overwriting the main screen. *Could be done in the shell commands section using `open`?*.

A: No problem - do something like this:

```
exec >/dev/tty5
```

Q: Does the filesystem creation code check for bad blocks ?

A: If you switch to the virtual console where the filesystem creation output is being displayed, you won't see any mention of the 'read-only' test being performed. Looks like the answer is no.

Q: Can I arrange things so some of my machines are configured differently from others ?

A: You could move the host dependent stuff into the scripted section of the KickStart config - e.g. only install a given RPM if on a given machine. It would be useful to have a conditional installation feature in the packages section of the config file, e.g. switching on architecture, or hostname/domain name/IP address.

Q: Are there any changes between RedHat 5.1 and 5.2 ?

A1: Lots of changes in the installer, but mostly bug fixes or cosmetic improvements. No impact on KickStart as far as I can tell - from a *diff -rcs* of the two *misc/src/install* directories.

A2: RH5.2 now apparently includes the automatic IP allocation/DHCP patches to `bootpd`, but they have left out the documentation which tells you how to use it.

Q: (How) can you clear a specific partition or partitions ? e.g. to leave */home* but zap */*.

A: You can't - yet!

Q: Can you arrange to have your partitions created across multiple drives ? e.g. */* on `sda` and */home* on `sdb`.

A: Don't think so - looks like you only get access to the first drive from the partitioning tool.

Q: Is it possible to specify existing partitions to be included in the mount table, or is it only possible to specify the creation of new partitions that will then be included?

A: ?

Q: After running `mkkickstart`, where is the file it creates?

A: It doesn't create a file - it dumps the KickStart config to `stdout`.

Q: In virtual console 4 (Alt-F4) I get `Unable to load NLS charset cp437(nls_cp437)`.

What does this mean ? Should I be worried ?

A: Sounds like you're trying to mount a CD-ROM burned with the Joliet (Unicode extensions to ISO 9660. In theory the filenames on the CD-ROM might get munched and not make it through to Linux correctly. In practice it doesn't seem to cause any problems - could be a spurious dependency ?

Q: Why am i getting the X Window System installed ? I didn't put it in my list of packages.

A: The `XFree86-VGA16` RPM is a 'base' component, and as such always gets installed - unless you change the definition of the base class.

Q: In my post-installation script, can I use the packages which have been installed by now to do funky things not possible with the limited tools on the floppies ?

A: Yep - e.g. if you chose to install Perl when you put your KickStart config together, almost anything is possible in about five lines :-)

---

11. Credits

Thanks to Eric Doutreleau for the info about *chkconfig*, the `SYSLINUX` config file hack, and the Perl script for updating your distribution server's RPMs. Thanks to Robert Kaminsky for extensive investigations. Thanks to Piete Brooks, Flavia Regina Munhoz, Tom Toffoli, Bob Robbins, Charlie Brady and Ragen Herrington, for their comments and questions.

12. Appendix A - Configuring BOOTP/DHCP and NFS

If you're wondering what on earth this BOOTP and DHCP stuff is, more information is available at the DHCP WWW site. NFS is documented separately in detail in the NFS HOWTO, and there's a DHCP mini-HOWTO too. I've tried to provide enough details here to help you get started, whilst not treating the topics in depth - let me know if you think this is overkill.

In the BOOTP/DHCP + NFS configuration we're discussing, the KickStart config file should be NFS mountable by the machine being installed from */kickstart/IPADDR-kickstart* on the BOOTP/DHCP server, where *IPADDR* is the IP address of the new machine, e.g. */kickstart/198.168.254.254-kickstart* for the machine *198.168.254.254*.

You should be able to override this location by returning the `bf` parameter (boot file) in your BOOTP/DHCP response. It may even be possible to have this NFS mounted off another machine entirely.

To NFS export some directories from an existing Linux box, create the file */etc/exports* with contents something like:

```
/kickstart *.swedish-chef.org(ro,no_root_squash)
/mnt/cdrom *.swedish-chef.org(ro,no_root_squash)
```

Note that if you didn't register the IP addresses you're going to be using in the DNS, you may be told to get lost by the NFS server and/or the RPC portmapper. In this you can probably get away with putting IP address/netmask pairs in the config files, e.g.

```
/kickstart 198.168.254.0/255.255.255.0(ro,no_root_squash)
```

and in */etc/hosts.allow*:

```
ALL: 194.82.103.0/255.255.255.0: ALLOW
```

This is because most Linux distributions use TCP wrappers to do access control for some or all of the NFS related daemons. Be aware that the */etc/exports* syntax tends to be different on other Unix variants - the NFS servers bundled with Linux distributions tend to offer a much wider range of options than the ones shipped with other versions of Unix.

Be aware that if you include a root password in your KickStart config file, or NFS export directories containing sensitive information, you should take care to expose this information to as few people as possible. This can be done by making the NFS export permissions as fine grained as you can, e.g. by specifying a particular host or subnet to export to rather than a whole domain. If you keep a special IP address free for KickStart installations, everything's nice and simple, but you'll have to change it later - or

reconfigure the machine to get its IP address via BOOTP/DHCP.

Most NFS servers require you to tell `mountd` and `nfsd` (on some versions of Unix they're prefixed with a `rpc.`) that the */etc/exports* file has changed - usually by sending a `SIGHUP`. There's often a program or script called `exportfs`, which will do this for you, e.g.

```
# exportfs -a
```

If you didn't have NFS up and running when this machine was booted, the directories may not be exported automatically. Try rebooting, or running the following programs as root:

```
# portmap
# rpc.nfsd
# rpc.mountd
```

As noted, on some systems the `rpc.` prefix isn't used. In most modern Unix distributions, these programs can be found in the */usr/sbin* or */usr/libexec* directories. These might not be in your path already, e.g. if you used `su` to become *root*. The `portmap` program is also sometimes called `rpcbind`, e.g. on Solaris, some versions of `nfsd` require a command line argument specifying the number of instances of the server to run, and you may find you also need to run another daemon called `biod`. The above should suffice on most (all?) Linux systems.

If you're using the CMU BOOTP server with DHCP and dynamic addressing extensions referred to earlier, a sample */etc/bootptab* entry (*/etc/bootptab* is the normal location of the BOOTP/DHCP configuration file) would look something like this:

```
.dynamic-1:ip=198.168.254.128:T254=0x30:T250="ds=198.168.254.2:
dn=swedish-chef.org:sm=255.255.255.0:gw=198.168.254.1:
dl=0xFFFFFFFF":
```

(wrapped for clarity)

This says to allocate IP addresses dynamically on encountering new machines, starting at *198.168.254.128* and continuing for the next 48 (the hexadecimal value *30*) addresses. Each client will be passed back the value of *T250*. In this case that sets:

- the DNS server `ds` to *198.168.254.2*
- the domain name `dn` to *swedish-chef.org*
- the subnet mask `sm` to *255.255.255.0*
- the default gateway `gw` to *198.168.254.1*
- the lease length `dl` (how long the address is valid for) to "forever"

There seem to be a number of other versions of this server kicking around which do not support dynamic addressing. For these, you would have to list the hardware (typically Ethernet MAC) address of each to-be-installed machine in */etc/bootptab*, and the entries would look something like this:

```
bork.swedish-chef.org:ip=198.168.254.128:ha=0000E8188E56:
```

```
ds=198.168.254.2:dn=swedish-chef.org:sm=255.255.255.0:
gw=198.168.254.1:dl=0xFFFFFFFF":
```

(wrapped for clarity)

Note that the parameter `ha` corresponds to the hardware address of the machine being installed.

---

13. Appendix B - Making your own RPMs

The RPM package format is already very well documented, particularly in the book *Maximum RPM* by Ed Bailey, which you can download from the RPM WWW site - also available from all good book stores! This is just a couple of quick hints for people in a hurry.

RPM packages are built from a *spec* file. This consists (in a similar fashion to the KickStart config file) of a recipe of steps that need to be taken in order to build the package - it's expected that you'll have to build it from source, potentially for multiple platforms, and may need to apply patches before compiling. Once built and installed, a binary RPM will be created from the files and directories you specify as being associated with the package. It's important to note that RPM has no idea of which files and directories are related to a given package - you have to tell it.

Here's a sample specification for a custom RPM of the Squid WWW cache server:

```
Summary: Squid Web Cache server
Name: squid
Version: 1.NOVM.22
Release: 1
Copyright: GPL/Harvest
Group: Networking/Daemons
Source: squid-1.NOVM.22-src.tar.gz
Patch: retry-1.NOVM.20.patch
%description
This is just a first attempt to package up the Squid Web Cache for easy
installation on our RedHat Linux servers

%prep
%setup
%build
configure --prefix=/usr/squid
perl -spi -e 's!#( -DALLOW_HOSTNAME_UNDERSCORES)!$1!' src/Makefile
make

%install
make install

%files
/usr/squid
```

Here's how to build this RPM:

```
% mkdir -p SOURCES BUILD SRPMS RPMS/i386
% cp ~/squid-1.NOVM.22-src.tar.gz SOURCES
% cp ~/retry-1.NOVM.20.patch SOURCES
% rpm -ba squid-1.NOVM.22+retry-1.spec
```

This will automatically create a subdirectory under the *BUILD* directory, into which it'll unpack the source c and then apply the patch (there are a number of options available for patching - check the book for details). Now, RPM will automatically build the package by running `configure` and then `make`, install it using `make install`, and take a snapshot of the files under */usr/squid*. It's the latter which will form the binary RPM of the Squid software.

Note that we can insert arbitrary shell commands into the unpacking, building and installing processes, e.g. the call to `perl` which tweaks one of Squid's compile-time parameters.

The final binary RPM will be left under the *RPMS* directory in the platform specific subdirectory *i386*. In this case it will be called *squid-1.NOVM.22-1.i386.rpm*. Note that the filename is created by concatenating the values of the following parameters from the spec file: `Name`, `Version` and `Release` - plus the hardware platform in question, *i386* in this case. Try to bear this in mind when creating your own RPMs, to avoid giving them overly long or painful names!

It's also worth bearing in mind that you can build RPMs without having to rebuild the whole software package, e.g.

```
Summary: Linux 2.0.36 kernel + filehandle patch + serial console patch
Name: linux
Version: 2.0.36+filehandle+serial_console
Release: 1
Copyright: GPL
Group: Base/Kernel
Source: linux-2.0.36+filehandle+serial_console.tar.gz
%description
This is just a first attempt to package up the Linux kernel with patches
for installation on our RedHat Linux servers

%prep
echo

%setup
echo

%build
echo

%install
echo

%post
/sbin/lilo

%files
/lib/modules/2.0.36
/boot/vmlinuz
```

In this case we simply create an RPM based on the */boot/vmlinuz* file and the contents of the directory */lib/modules/2.0.36*, and execute */sbin/lilo* after the package has been installed on a target machine. Let me know

if you know much neater way of writing the spec file than this.

14. Appendix C - Munging your own RPMs into the distribution

Here is Eric's script for munging updated RPMs into the RedHat distribution area:

```
#!/usr/bin/perl
#
$redhatdir="/cdrom/i386";
$rpmdir="/cdrom/i386/RedHat/RPMS/";
$updatedir="/cdrom/updates/";
@OTHERDIR=($updatedir);
foreach $dir (@OTHERDIR)
        {
        print "update for $dir\n";
        system(" find $dir -name \"*.rpm\" -exec cp {} $rpmdir \\; ");
        }
chdir($contribdir) || die "peux pas aller dans $contribdir $!\n";
system("chmod -R 755 $redhatdir");
chdir($rpmdir) || die "problem to go in $rpmdir $!\n";
#
# remove the old file
#
opendir(DIR,'.');
@package=grep(/\.rpm$/,readdir(DIR));
foreach $file (@package)
        {
        $file =~ /(.*)\-([\d+|\.]+\w*)\-(\d+)\.[i386|noarch].*/;
        $nom=$1;
        $version=$2;
        $buildvers=$3;
        if ($NOM{$nom})
                {
                $version2=$VERSION{$nom};
                $buildver2=$BUILDVERS{$nom};
                $file2=$FILE{$nom};
                $nom2=$NOM{$nom};
                if ( $version2 gt $version )
                        {
                        print "$file2 is newer than $file\n";
                        unlink($file);
                        }
                else
                        {
                        if ( $version2 lt $version )
                                {
                                print "$file is newer than $file2\n";
                                unlink($file2);
                                $VERSION{$nom}=$version;
                                $BUILDVERS{$nom}=$buildvers;
                                $FILE{$nom}=$file;
                                $NOM{$nom}=$nom;
                                }
                        else
```

```
                                        {
                                        if ( $buildver2 > $buildvers )
                                                        {
                                                   print "$file2 : $buildver2 est mieux que
$file : $buildvers\n";
                                                   unlink($file);
                                                        }
                                        else
                                                        {
                                                print "$file2 : $buildver2 is older than
$file : $buildvers\n";
                                                    unlink($file2);
                                                    $VERSION{$nom}=$version;
                                                    $BUILDVERS{$nom}=$buildvers;
                                                    $FILE{$nom}=$file;
                                                    $NOM{$nom}=$nom;
                                                        }
                                        }
                                }
                        }
                else
                        {
                        $VERSION{$nom}=$version;
                        $BUILDVERS{$nom}=$buildvers;
                        $FILE{$nom}=$file;
                        $NOM{$nom}=$nom;
                        }
                }

# we do the hard thing here
#
system("$redhatdir/misc/src/install/genhdlist $redhatdir");
```

LinuxDoc+Emacs+Ispell-HOWTO

Author: Philippe MARTIN (feloy@wanadoo.fr)
Translator: Sébastien Blondeel (Sebastien.Blondeel@lifl.fr)

v0.4, 27 February 1998

This document is aimed at writers and translators of Linux HOWTOs or any other paper for the Linux Documentation Project. It gives them hints at using tools including Emacs and Ispell.

1. Preamble

- 1.1 Copyright
- 1.2 Credits
- 1.3 Comments
- 1.4 Versions

2. Introduction

- 2.1 SGML
- 2.2 The `LinuxDoc` Type Definition
- 2.3 `SGML-Tools`

3. Your first document.

- 3.1 From a text document

4. Configuring `Emacs`

- 4.1 Accented Characters
- 4.2 SGML mode

- 4.3 PSGML mode
- 4.4 Miscellaneous

TOC

5. Ispell

- 5.1 Choosing your default dictionaries
- 5.2 Selecting special dictionaries for certain files
- 5.3 Spell-checking your document
- 5.4 Personal dictionary versus local file dictionary
- 5.5 Typing spell-checking

6. Dirty Tricks

- 6.1 Inserting a header automatically

Appendix

7. An `insert-sgml-header` function

---

1. Preamble

1.1 Copyright

Copyright Philippe Martin 1998
You may redistribute and/or modify this document as long as you comply with the terms of the GNU General Public Licence, version 2 or later.

1.2 Credits

Special thanks go to Sébastien Blondeel, who is a nasty bugger and asked me so much about Emacs setup. His clever questions have allowed me to understand it better and pass the knowledge to you through this document.

1.3 Comments

Do not hesitate to tell me any thing you think will help make this document better. I will examine your critics thoroughly.
Do not hesitate as well to ask me any questions related to topics discussed here. I will be more than happy to answer them, as they may help me further improve this document.

> Translator note: If the English is ugly, well then that goes to me!

1.4 Versions

This paper is about the following versions:

- Sgml-tools version 0.99,
- Emacs version 19.34,
- Ispell version 3.1,
- All Emacs libraries referred to in this document are distributed with the above Emacs version, apart from `iso-sgml`, which is distributed with XEmacs, and `psgml`, which is a stand-alone library.

2. Introduction

2.1 SGML

Standard Generalised Mark-up Language, or **SGML**, is a language to define document types.

For instance, one may define the document type *recipe*, with a first part presenting the ingredients, a second part introducing the accessories, a third part giving step by step instructions for baking the cake, and a nice final picture to show the outcome of it all.

This is called a *Document Type Definition*. It does not define what the final product will look like, it only defines what it may contain.

To use the same example again, I'm sure that upon reading my idea of a recipe, you recognised yours, or your favourite cook's. Nevertheless, they actually look different: mine have a picture in the upper left corner of the bathroom cupboard, and the ingredients list can be found in the back garden, between the swimming pool and the barbecue. Yours?

Thanks to this standard definition, one can write a document, without taking into account what it will look like in the end to the reader.

2.2 The `LinuxDoc` Type Definition

This type is used to write, as you might have guessed, documents related to Linux.

Such documents are generally built as follows: they start with a title followed by the name of the author, and the version number and date. Then comes the abstract (so you don't have to browse through it before realizing it isn't what you were looking for after all), then the contents which show the structure so that those in a rush can go directly to the part they want to read.

Then comes a list of chapters, sections, paragraphs. Among these, one can insert bits of programs, change the font to emphasise a word or a sentence, insert lists, refer to another part of the document, etc.

To write such a document, you just need to specify at the right time the title, the author, the date, and the document version, the chapters and sections, say when a list is to be inserted, what its elements are etc.

2.3 `SGML-Tools`

SGML-Tools will turn the specification of a document into the final result in the form you prefer. If you want it in your personal library, you will choose *PostScript*. If you want to share it with the world through the Web, it will be *HTML*. If you can't help it and must read it under Windows, you can turn it into *RTF* to be able to read it with any word processor. Or maybe use all three formats to accommodate

your changing moods.

SGML-Tools are available via anonymous FTP at ftp://ftp.lip6.fr/pub/sgml-tools/

---

3. Your first document.

3.1 From a text document

If you want to turn a text document into SGML to port it to other formats, this is the way to go:

1. Add the following lines at the very beginning:

```
<!doctype linuxdoc system>
 <article>
  <title>Title Goes Here</title>
  <author>
   name of author, author's e-mail, etc.
  </author>
  <date>
   version and date
  </date>
```

2. If you describe briefly the contents of the document in the beginning, surround that paragraph with the `<abstract>` and `</abstract>` tags.
3. Then insert the `<toc>` tag, which stands for *Table Of Contents*.
4. At the beginning of each new chapter, replace the line giving the number and title of the chapter with:

```
<sect>The Title Of The Chapter
```

   and add the `</sect>` tag at the end of the chapter.

   Note : You don't have to put the chapter number, this is done automatically.
5. Proceed in the same way for sections. You need to delete their numbers and tag their titles with `<sect1>` and they end with `</sect1>`.
6. You can also define as many as 4 levels of nesting in the sections, using `<sectn>` and `</sectn>` where `n=` 2, 3, or 4 in a similar way.
7. In the beginning of each paragraph, insert the `<p>` tag.
8. If you need to emphasise some parts, tag them with `<it>` and `</it>` (*italics*), `<bf>` and `</bf>` (**bold face**), or `<tt>` and `</tt>` (`typewriter style`).
9. To insert a list like the following one:

```
This is a four lines list:
```

```
- first line goes here
- second line comes next
- yet another one
- that's it.
```

you must replace it with:

```
 This is a four lines list:
<itemize>
<item>first line goes here
<item>second line come next
<item>yet another one
<item>that's it.
</itemize>
```

10. When a whole block is a part of a program, or something else that needs to stick out:

```
<verb>
 10 REM Oh my God what's this?
 20 REM I thought this had long disappeared!
 30 PRINT "I am back to";
 40 PRINT "save the world."
 50 INPUT "From whom, do you reckon? ",M$
 60 IF M$="Bill" THEN PRINT "Thou art wise.":GOTO PARADISE
 70 ELSE PRINT "You ain't got a clue...":GOTO RICHMOND
 </verb>
```

11. Thus far, your SGML formating skills are fairly decent. If you want to refine your document, you may have a look at the user's guide for **SGML-Tools**, which gives more details about the **LinuxDoc** document type.

4. Configuring Emacs

4.1 Accented Characters

If you want to write documents in French or in any other western European language, you will need 8-bit characters. This is how to set Emacs up to tell it to accept such characters.

The displaying of 8-bit characters

To let Emacs display 8-bit characters, you will need the following lines in your `.emacs` file:

```
(standard-display-european 1)
(load-library "iso-syntax")
```

If you are using Emacs on a terminal which has no 8-bit support, you can use the `iso-ascii` library (`(load-library "iso-ascii")`), which tells Emacs to display such characters to its best approximation.

The typing of 8-bit characters

If your keyboard allows you to enter accented characters, no problem. If not here are some remedies:

The `iso-acc` library

The Emacs `iso-acc` library will let you type 8-bit characters from a 7-bit keyboard.

To use it, insert the following in your `.emacs` file:

```
(load-library "iso-acc")
```

Then, upon running Emacs and opening the file you need to edit, type `Meta-x iso-accents-mode`.

You can then enter the **é** of the French word *café* typing `'` then `e`. More generally, you will type an accented character typing the accent first, then the letter to accent (upper or lower case). The following are the accents you may use:

- `'` : Acute
- `` ` `` : Grave
- `^` : Circumflex
- `"` : Dieresis

- ~ : Tilde, cedilla, and other particular cases (cf iso-acc.el).
- / : To bar a letter, etc.

If you need one of these characters and not an accented letter, type a space next to it. For instance, to type *l'éléphant*, type `l ' spc ' e l ' e ...`

You will find all the possible combinations in the `iso-acc.el` file.

The Meta key

Some terminals will let you type 8-bit characters with the Meta (or Alt) key. For example, pressing **Meta-i** will get you the **é** character.

But Emacs reserved the Meta key for other uses, and I know of no library which lets you use it for accented characters.

This is a solution:

```
(global-set-key "\ei" '(lambda () (interactive) (insert ?\351)))
```

Such a line, if inserted in your `.emacs` file, will let you type **é** using the **Meta-i** keystroke. You can redefine in such a way the combinations you need if you replace **i** with the right key and **351** with the right code (the code being taken from the ISO-8859-1 character set).

Warning! Some local modes may redefine such key combinations.

The displaying of 8-bit SGML characters

Under SGML, you can type accented characters with macros. For example, the **é** key is **é**. Generally, the applications that need to read SGML can read 8-bit characters and there is no need to use these macros. But some may not be able to do so. Given that there is a way to solve this problem, it would be a waste to let these crash.

The `iso-sgml` library will let you type accented characters under Emacs, like always, but upon saving your file to the disk, it will turn these 8-bit characters into their SGML equivalent.

It is therefore easy, thanks to this library, to type and reread your document under Emacs, and you can be sure a non 8-bit clean application will accept you document.

To use this library, you just need to add the following lines to your `.emacs` file:

```
(setq sgml-mode-hook
'(lambda () "Defaults for SGML mode."
 (load-library "iso-sgml")))
```

TOC

4.2 SGML mode

Upon loading a file with the **.sgml** extension, Emacs enters the **sgml mode** automatically. If it doesn't, you can tell it to do so manually by typing `Meta-x sgml-mode`, or automatically by adding the following lines to your `.emacs` file:

```
(setq auto-mode-alist
(append '(("\.sgml$"  . sgml-mode))
          auto-mode-alist))
```

This mode will let you choose how to insert 8-bit characters for example. With `Meta-x sgml-name-8bit-mode` (or the menu item *SGML/Toggle 8-bit insertion*), you can choose to type 8-bit characters as is, or in SGML form, i.e. in the form **&...;**.

It will as well let you hide or show SGML tags, with `Meta-x sgml-tags-invisible` (or the menu item *SGML/Toggle Tag Visibility*).

4.3 PSGML mode

PSGML mode helps a lot to edit SGML documents with Emacs.

The psgml-linuxdoc documentation explains how to install this mode and use it with *LinuxDoc*.

4.4 Miscellaneous

auto-fill mode

In the normal mode, when you type a paragraph and get to the end of the line, you must use the Return key yourself to get to the next line, or else your line goes on through the whole paragraph. When you use Return to get to the next line, you get a paragraph with ragged right margins.

If you let some lines go beyond a reasonable width, you won't be able to see them with some editors.

The **auto-fill** mode automates this boring task: when you go further than a certain column (the 70th by default), you are automatically taken to the next line.

This is how to use this mode, and set the width of your lines to 80:

```
(setq sgml-mode-hook
      '(lambda () "Defaults for SGML mode."
          (auto-fill-mode)
          (setq fill-column 80)))
```

---

5. Ispell

If you want to spell-check your document from within Emacs, you may use the **Ispell** package and its Emacs mode.

5.1 Choosing your default dictionaries

You can set up Emacs so that upon loading a file, it chooses automatically which dictionaries to use (you can use several). The first one, certainly the most important, is the main dictionary, distributed with Ispell. You can choose among several languages. The second one is your personal dictionary, where Ispell will insert words it couldn't find in the main dictionary but you told it to remember.

If you wish to use as a default dictionary the French dictionary that comes with Ispell, and if you wish to use the file `.ispell-dico-perso` in your home directory as a personal dictionary, insert the following lines in your `.emacs` file:

```
(setq sgml-mode-hook
'(lambda () "Defauts for SGML mode."
(setq ispell-personal-dictionary "~/.ispell-dico-perso")
(ispell-change-dictionary "francais")
))
```

5.2 Selecting special dictionaries for certain files

You may have a little problem if you do not spell-check documents in the same language at all times. If you translate documents, it is very likely that you swap languages (and dictionaries) very often.

I don't know of any Lisp way of selecting, either automatically, or with a single mouse click, the main and personal dictionaries associated to the language currently being used. (If you do, please tell me!)

However, it is possible to indicate, at the end of the file, which dictionaries you want to use for the current file (and only this one). It suffices to add them as commentaries, so that Ispell can read them upon launching a spell-check:

```
<!-- Local IspellDict: english -->
<!-- Local IspellPersDict: ~/emacs/.ispell-english -->
```

If you have previously defined, in your `.emacs` file, that your default dictionaries are the French dictionaries, then you can add these lines in the end of any file written in English.

TOC

5.3 Spell-checking your document

To spell-check the whole of your document, use, from anywhere in the document the `Meta-x ispell-buffer` command. You may as well only run the checking on a region in your document:

- Mark the beginning of the region with `Ctrl-Spc` (mark-set-command),
- Go to the end of the region to check,
- type `Meta-x ispell-region`.

Emacs then runs Ispell. Upon meeting an unknown word, this one shows you said word (usually highlighted) and prompts you for a key:

- **spc** accepts the word, this time only,
- **i** accepts the word and inserts it in your personal dictionary,
- **a** accepts the word for this session,
- **A** accepts the word for this file, and inserts it in the local file dictionary
- **r** allows you to correct the word by hand
- **R** allows you to correct all the occurrences of the misspelled word,
- **x** stops the checking, and puts the cursor back in place,
- **X** stops the checking and leaves the cursor where it is, letting you correct your file; you will be able to continue the spell-checking later if you type `Meta-x ispell-continue`,
- **?** gives you online help.

If ispell finds one or several words close to the unknown one, it will show them in a little window, each one of them preceded by a digit. Just type this digit to replace the misspelled word with the corresponding word.

5.4 Personal dictionary versus local file dictionary

The **i** key will let you insert a word in your personal dictionary, whereas **A** will let you insert a word in the local file dictionary.

The local file dictionary is a sequence of words inserted at the end of the file, as comments, reread by Ispell each time it is run on the file. This way, you can accept some words, acceptable in this file, but not necessarily acceptable in other files.

As far as I am concerned, I think it is better that the personal dictionary be reserved for words the main dictionary doesn't know but which belong to the language (like hyphenated words), plus some common words like proper nouns or others (like *Linux*), if they don't look too much like a real word of the main dictionary; adding too many words in the personal dictionary, such as first names, may be dangerous, because they may look like a word of the language (one can imagine Ispell being mystified on the following: `*When the going gets tof, the tof get going*

Tof is a French abbreviation for the first name *Christophe*.

'!).

5.5 Typing spell-checking

TOC

Ispell can spell-check your file while you're typing. You need to use **ispell-minor-mode** for this. To start it or stop it, type `Meta-x ispell-minor-mode`. Ispell will *beep* you each time you type a word it doesn't know.

If those *beeps* hassle you (or your roommate is taking a nap), you can replace those annoying *beeps* with a flash on the screen, with the command `Meta-x set-variable RET visible-bell RET t RET`. You can add the following line in your `.emacs` and silence Emacs forever:

```
(setq visible-bell t)
```

---

6. Dirty Tricks

6.1 Inserting a header automatically

Emacs allows you to *hook* some actions to any event (opening of a file, saving, running a new mode, etc).

The **autoinsert** library uses this feature: when you open a new file under Emacs, this library inserts, according to the type of the file, a *standard* header.

In our case, this *standard* header could well be the part declaring the document type (LinuxDoc), the title, the author, and the date.

I will describe here two ways to insert such a header. You could insert a template file containing the information to insert, or you could run an **elisp** routine.

by inserting a file

You must first tell Emacs to run the `auto-insert` when opening a file, then to read the **autoinsert** library which declares the `auto-insert-alist` list which we need to change. This list defines the header to insert for each file type. By default, the file to insert must be in the `~/insert/` directory, but it is possible to redefine the `auto-insert-directory` variable if you want to put it somewhere else.

Add the following lines to your `.emacs` file to insert the `~/emacs/sgml-insert.sgml` file each time you open a new SGML file:

```
(add-hook 'find-file-hooks 'auto-insert)
(load-library "autoinsert")
(setq auto-insert-directory "~/emacs/")
(setq auto-insert-alist
      (append '((sgml-mode .  "sgml-insert.sgml"))
              auto-insert-alist))
```

You can then write in the `~/emacs/sgml-insert.sgml` file your customised header, then re-run Emacs and open some `foobar.sgml` file. Emacs should ask you to confirm the automatic insertion, and if you answer yes, insert your header.

by running a routine

This works like before, but instead of setting the `auto-insert-alist` to a file to insert, you need to set it to a function to execute. This is how to proceed, taking for granted you want to write this function in a file named `~/emacs/sgml-header.el`. (there's no need to burden your `.emacs` file with such a function, as it may turn out to be quite long):

```
(add-hook 'find-file-hooks 'auto-insert)
(load-library "autoinsert")
(add-to-list 'load-path "~/emacs")
(load-library "sgml-header")
(setq auto-insert-alist
      (append '(((sgml-mode .  "SGML Mode") . insert-sgml-header))
              auto-insert-alist))
```

You will find in appendix an example of `insert-sgml-header` function.

---

7. An `insert-sgml-header` function

This function will let the user insert a customised header for a Linux Documentation Project document in a file. It can be called automatically when one opens a new file, or explicitly, by the user.

This function prompts the user, through the *mini-buffer*, for some pieces of information, some of which are necessary, some of which are not.

First comes the title. If none is given, the function returns immediately, and inserts nothing. Then comes the date, the author, his e-mail and home page (these last two are optional).

Then comes a request for the name of the translator. If there is none, just type *Return*, and no further prompting about a hypothetical translator will be done. If there is one, you are asked for his e-mail and home page (optional as well).

This function then prints your request to the current buffer, including of course all the information you typed in a set up form, and including as well the tags which will serve for the abstract and the first chapter. It finally puts the cursor in the place where the abstract needs to be typed.

```
(defun insert-sgml-header ()
  "Inserts the header for a LinuxDoc document"
  (interactive)
  (let (title author email home translator email-translator home-translator date
              starting-point)
    (setq title (read-from-minibuffer "Title: "))
    (if (> (length title) 0)
        (progn
          (setq date (read-from-minibuffer "Date: ")
                author (read-from-minibuffer "Author: ")
                email (read-from-minibuffer "Author e-mail: ")
                home (read-from-minibuffer "Author home page: http://")
                translator (read-from-minibuffer "Translator: "))
          (insert "<!doctype linuxdoc system>\n<article>\n<title>")
          (insert title)
          (insert "</title>\n<author>\nAuthor: ") (insert author) (insert
"<newline>\n")
          (if (> (length email) 0)
              (progn
                (insert "<htmlurl url=\"mailto:")
                (insert email) (insert "\" name=\"") (insert email)
                (insert "\"><newline>\n")))
          (if (> (length home) 0)
              (progn
                (insert "<htmlurl url=\"http://")
                (insert home) (insert "\" name=\"") (insert home)
                (insert "\">\n<newline>")))
          (if (> (length translator) 0)
              (progn
                (setq email-translator (read-from-minibuffer "Translator e-mail: ")
                      home-translator (read-from-minibuffer "Translator home page:
http://"))
                (insert "Translator : ")
                (insert translator)
```

```
            (insert "<newline>\n")
            (if (> (length email-translator) 0)
                (progn
                  (insert "<htmlurl url=\"mailto:")
                  (insert email-translator) (insert "\" name=\"")
                  (insert email-translator)
                  (insert "\"><newline>\n")))
            (if (> (length home-translator) 0)
                (progn
                  (insert "<htmlurl url=\"http://")
                  (insert home-translator) (insert "\" name=\"")
                  (insert home-translator)
                  (insert "\"><newline>\n")))))
      (insert "</author>\n<date>\n")
      (insert date)
      (insert "\n</date>\n\n<abstract>\n")
      (setq point-beginning (point))
      (insert "\n</abstract>\n<toc>\n\n<sect>\n<p>\n\n\n</sect>\n\n</article>\n")
      (goto-char point-beginning)
      ))))
```

Linux Meta-FAQ

Michael K. Johnson <johnsonm@redhat.com>

v4.7, 25 October 1997

This is the Meta-FAQ for Linux. It is mainly a list of valuable sources of information. Check these sources out if you want to learn more about Linux, or have problems and need help.

1. Introduction

2. Getting Linux

- 2.1 Linux FTP sites
- 2.2 Linux on physical media
- 2.3 AFS
- 2.4 Commercial networks
- 2.5 Mailservers and such

3. Linux distributions

4. Linux mailing-lists

5. Documentation for various programs

6. More Documentation

7. Keeping track of current releases

TOC

8. This Document

9. Legalese

---

1. Introduction

What is Linux?

Linux is an independent implementation of the POSIX operating system specification, with SYSV and BSD extensions, that has been written entirely from scratch (this means it looks and acts just like Unix). It has no proprietary code in it. Linux is freely distributable under the GNU General Public License.

Linux works on IBM PC compatibles with an ISA or EISA bus (including local bus variants VLB and PCI) and a 386 or higher processor. Some Amiga and Atari computers with MMU's are also supported. This means 68020 with an external MMU, 68030, 68040, or 68060. Support for the Digital Alpha is now stable. Red Hat and Craftworks have Alpha distributions of Linux. Support for Sparc is stable, and Red Hat Linux is available for Sparc. Support for PowerPC is in development for multiple platforms, including Nubus and PCI Macintosh, Motorola Powerstack, IBM 830 and 850, and other platforms. Support for ARM, StrongARM, and MIPS is in various stages of completion, but don't hold your breath. Read comp.os.linux.announce instead.

See the Linux INFO-SHEET for more technical information on these ports, and the Hardware Compatibility HOWTO for more exact hardware requirements.

The Linux kernel is written by Linus Torvalds <torvalds@transmeta.com> and other volunteers. Most of the programs running under Linux are generic Unix freeware, many of them from the GNU project.

The Linux INFO-SHEET

More specific technical information on Linux. Includes pointers to information on the various ports, a feature list, information about how to get Linux, and more.

The Linux HOWTO's

These are somewhat like FAQ's, but instead of answering common questions, they explain how to do common tasks, like ordering a release of Linux, setting up print services under Linux, setting up a basic UUCP feed, etc. See http://sunsite.unc.edu/LDP/HOWTO/HOWTO-INDEX.html or ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/ for the definitive versions of all the HOWTO's. Other sites with up-to-date copies of the HOWTOs are ftp.cc.gatech.edu and tsx-11.mit.edu.

In addition, there are many short, free-form documents called "mini-HOWTOs". These documents cover very specific subjects, such as BogoMIPS or Color-ls. These are available at ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/mini/ and at http://sunsite.unc.edu/LDP/HOWTO/HOWTO-INDEX.html.

Linux newsgroups

There are several Usenet newsgroups for Linux. It is a good idea to follow at least

TOC

comp.os.linux.announce if you use Linux. comp.os.linux.announce is moderated by Lars Wirzenius. To make submissions to the newsgroup, send mail to linux-announce@news.ornl.g You may direct questions about comp.os.linux.announce to Lars Wirzenius <wirzeniu@iki.fi>

The newsgroup comp.os.linux.announce is a moderated newsgroup for announcements about Linux (new programs, bug fixes, etc).

The newsgroup comp.os.linux.answers is a moderated newsgroup to which the Linux FAQ, HOWTO documents, and other documentation postings are made.

The newsgroup comp.os.linux.setup is an unmoderated newsgroup for discussion of issues and problems involved in setting up Linux systems.

The newsgroup comp.os.linux.admin is an unmoderated newsgroup for discussion of administration of Linux systems.

The newsgroup comp.os.linux.development.system is an unmoderated newsgroup specifically for discussion of Linux **kernel** development. The only application development questions that should be discussed here are those that are intimately associated with the kernel. All other development questions are probably generic Unix development questions and should be directed to a comp.unix group instead, unless they are very Linux-specific applications questions, in which case they should be directed at comp.os.linux.development.apps.

The newsgroup comp.os.linux.development.apps is an unmoderated newsgroup specifically for discussion of Linux-related applications development. It is not for discussion of where to get applications for Linux, nor a discussion forum for those who would like to see applications for Linux.

The newsgroup comp.os.linux.hardware is for Linux-specific hardware questions.

The newsgroup comp.os.linux.networking is for Linux-specific networking development and setup questions.

The newsgroup comp.os.linux.x is for Linux-specific X Windows questions.

The newsgroup comp.os.linux.misc is an unmoderated newsgroup for any Linux discussion that doesn't belong anywhere else.

In general, **do not** crosspost between the Linux newsgroups. The **only** crossposting that is appropriate is an occasional posting between one unmoderated group and comp.os.linux.announce. The whole point of splitting the old comp.os.linux group into many groups was to reduce traffic in each. Those that do not follow this rule will be flamed without mercy...

Other newsgroups

Do not assume that all your questions are appropriate for a Linux newsgroup just because you are running Linux. Is your question really about shell programming under any unix or unix clone? Then ask in comp.unix.shell. Is it about GNU Emacs? Then try asking in gnu.emacs.help. Also, if you don't know another group to ask in, but think there might be, politely ask in your post if there is another group that would be more appropriate for your question. At least the groups comp.unix.{questions,shell,programming,bsd,admin} and comp.windows.x.i386unix should be

useful for a Linux user.

The World-Wide Web

Greg Hankins <gregh@cc.gatech.edu> maintains the home WWW page for the Linux project. The URL is http://sunsite.unc.edu/LDP/

Linux Journal

A magazine called *Linux Journal* was launched several years ago. It includes articles intended for almost all skill levels, and is intended to be helpful to all Linux users. Subscriptions are $22 in the U.S., $27 in Canada and Mexico, and $32 elsewhere around the world, all payable in U.S. funds. Subscription inquiries can be sent via email to subs@ssc.com or faxed to (U.S.) 1-206-782-7191 or mailed to Linux Journal, PO Box 85867, Seattle, WA 98145-1867 USA. SSC has a PGP public key if you wish to send your credit card number via encrypted email: finger info@ssc.com

The Linux Software Map

Information on free software available for Linux can be found in the Linux Software Map, which can be found at http://www.execpc.com/~lsm/

---

2. Getting Linux

2.1 Linux FTP sites

A more complete list of Linux FTP sites is in the Linux INFO-SHEET, which can always be found at http://sunsite.unc.edu/LDP/HOWTO/INFO-SHEET.html The most important sites are listed here; please see the INFO-SHEET for a site nearer to you (there are many mirrors).

```
textual name             numeric addr    Linux directory
=======================  ==============  ===============
tsx-11.mit.edu           18.86.0.44      /pub/linux
sunsite.unc.edu          152.2.22.81     /pub/Linux
ftp.kernel.org           206.184.214.34  /pub/linux
```

These sites are the main ``home'' sites for Linux where most uploads take place. There are many mirror sites; please use the closest (network-wise) site to you.

2.2 Linux on physical media

Linux is distributed on physical media, mainly CD-ROM, by several commercial vendors. Please read the distribution HOWTO, posted regularily to comp.os.linux.announce, and available at http://sunsite.unc.edu/LDP/HOWTO/Distribution-HOWTO.html

2.3 AFS

Linux is available over AFS by mounting the volume project.linux from sipb.mit.edu

2.4 Commercial networks

Compu$erve has some Linux archives.

2.5 Mailservers and such

Sunsite offers ftp-mail service --- mail <ftpmail@sunsite.unc.edu>.

3. Linux distributions

Linux is distributed by its author only as a kernel. Other people have put together ``distributions'' that pair the Linux kernel with utilities and application software to make a complete working package.

There are several distributions of Linux, which are available at various sites. Sunsite mirrors many of the distributions at ftp://sunsite.unc.edu/pub/Linux/distributions/. The most commonly-recommended freely-available distributions are Red Hat http://www.redhat.com and Debian http://www.debian.org. These are available for free over the internet, and are also sold on CD-ROM.

There are other distributions of Linux as well. Most commercial distributors of Linux advertise in *Linux Journal*.

4. Linux mailing-lists

Used mostly for discussion between developers of new features and testers of pre-release versions. See addresses in the FAQ. Send mail to majordomo@vger.rutgers.edu with the single word `help` in the body of the message , and you will get mail explaining how to subscribe to the many Linux mailing lists there. Save this mail, as it tells you how to unsubscribe from the lists, and if you post annoying messages to the list complaining about not being able to get off the list (because you didn't follow instructions and save the mail telling you how to unsubscribe), you will likely be flamed for wasting international bandwidth and money.

5. Documentation for various programs

Many programs come with some sort of documentation, often in a file called README or something similar. It is a VERY good idea to read them with care. It is boring to see (and answer) questions that are answered in the documentation. Most programs also have ``man pages''; use the command `man programname` to get documentation on a program named `programname`. To get help using the man program, use `man man`.

Most distributions put other documentation about programs in the directory /usr/doc/; your distribution should include documentation on how to access that documentation.

6. More Documentation

The Linux Documentation Project is working on a lot of documentation. Already, over 3000 pages of book-style documentation has been released to the general public, and another 2000 or so printed pages of man pages have also been released, with more to follow. Check http://sunsite.unc.edu/LDP/ for documents written by the LDP.

7. Keeping track of current releases

Important new releases, programs, and ports are usually announced in comp.os.linux.announce.

8. This Document

The latest version of this document should always be available from
http://sunsite.unc.edu/LDP/HOWTO/META-FAQ.html

9. Legalese

Trademarks are owned by their owners. Satisfaction not guaranteed. No warranties about this document. Void where prohibited.

The content of this document is placed in the public domain, but if you quote it, please be polite and attribute your source.

Lars Wirzenius <wirzeniu@iki.fi> wrote the first version of this document; it is now maintained by Michael K. Johnson <johnsonm@redhat.com>. Mail me if you have any questions about this document.

The MGR Window System HOWTO

Vincent Broman

Draft, 30 May 1996

1. This HOWTO

- 1.1 Archiving
- 1.2 Authentication
- 1.3 Credit for the HOWTO

2. What is the MGR window system?

- 2.1 Function
- 2.2 Requirements
- 2.3 How do MGR, X11, and 8.5 compare?

3. Installing MGR

4. Running MGR

- 4.1 Applications not aware of MGR
- 4.2 MGR Applications (clients) distributed with the server
- 4.3 MGR-aware clients distributed separately, see "SUPPORT" file

5. Programming for MGR

6. More documentation

7. Credit for MGR

1. This HOWTO

Copyright Vincent Broman 1995.
Permission granted to make and distribute copies of this HOWTO
under the conditions of the GNU General Public License.

1.1 Archiving

This HOWTO is archived in `ftp://archimedes.nosc.mil/pub/Mgr/MGR-HOWTO.sgml`, and also distributed from `ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/MGR-HOWTO`. In nearby directories the same document may appear in alternate formats like `MGR-HOWTO.txt`.

1.2 Authentication

Copies of the MGR distribution due to Broman should be accompanied by PGP signature files, signed by "Vincent Broman <broman@nosc.mil>".

1.3 Credit for the HOWTO

While Vincent Broman first put together this HOWTO, much of the information and text was obtained from FAQs, READMEs, etc. written by Stephen Uhler, Michael Haardt, and other public-spirited net-persons. Email corrections and suggested changes to `broman@nosc.mil`.

Uhler was the main architect of **MGR** -- see the Credit section below.

2. What is the MGR window system?

2.1 Function

MGR (ManaGeR) is a graphical window system. The **MGR** server provides a builtin window manager and windowed graphics terminal emulation on color and monochrome bitmap displays. **MGR** is controlled by mousing pop-up menus, by keyboard interaction, and by escape sequences written on pseudo-terminals by client software.

MGR provides each client window with: termcap-style terminal control functions, graphics primitives such as line and circle drawing; facilities for manipulating bitmaps, fonts, icons, and pop-up menus; commands to reshape and position windows; and a message passing facility enabling client programs to rendezvous and exchange messages. Client programs may ask to be informed when a change in the window system occurs, such as a reshaped window, a pushed mouse button, or a message sent from another client program. These changes are called events. **MGR** notifies a client program of an event by sending it an ASCII character string in a format specified by the client program. Existing applications can be integrated into the windowing environment without modification by having **MGR** imitate keystrokes in response to user defined menu selections or other events.

2.2 Requirements

MGR currently runs on Linux, FreeBSD, Sun 3/4 workstations with SunOS, and Coherent. Various older versions of **MGR** run on the Macintosh, Atari ST MiNT, Xenix, 386-Minix, DEC 3100, and the 3b1 Unix-pc. Many small, industrial, real-time systems under OS9 or Lynx in Europe use (another variant of) Mgr for their user interface. The programming interface is implemented in C and in ELisp, although supporting clients written in other languages is quite easy.

Running **MGR** requires much less in resources than X, or even gcc. It does not have the user-base, software repertory, or high-level libraries of X or MS-Windows, say, but it is quite elegant and approachable.

It has been said that **MGR** is to X as Unix was to Multics.

2.3 How do MGR, X11, and 8.5 compare?

MGR consists of a server with builtin window manager and terminal emulator, and clients which run in this terminal emulator and use it to communicate with the server. No resource multiplexing is done.

X11 consists of a server and clients, which usually connect to the server using a socket. All user visible things like terminal emulators, window managers etc are done using clients. No resource multiplexing is done.

8.5, the Plan 9 window system, is a resource multiplexer, as each process running in a window can a `/dev/bitblt`, `/dev/mouse` and `/dev/kbd` in its own namespace. These are multiplexed to th `/dev/bitblit`, `/dev/mouse` and `/dev/kbd` in the namespace of 8.5. This approach allows one to run 8.5 in an 8.5 window, a very clean design. 8.5 further has an integrated window manager and terminal emulator.

3. Installing MGR

The latest source distribution can be FTPed from the directory `ftp://archimedes.nosc.mil/pub/Mgr/69` or Mosaiced from `http://archimedes.nosc.mil/Mgr/69`. The same should be found at `ftp://sunsite.unc.edu/pub/Linux/apps/MGR` and its mirrors. Older versions of this distribution from Haardt can be found on `tsx-11.mit.edu` and perhaps elsewhere. Pre-Linux versions of **MGR** from Uhler and others have been found at `ftp://bellcore.com/pub/mgr`, but I think they are gone now. I have saved a copy of everything about **MGR** seen on the Internet, but I am not aware of anything weighty that is missing from this Linux/Sun distribution. **MGR** has been through a lot of versions and releases, but the current *Linux* version number is 0.69. This version number could jump to 1.0 when stable 256-color VGA code for Linux appears (for more than one video card type). RCS version numbers have increased from Bellcore's 4.3 up to our 4.13 now.

Required tools to build this distribution of **MGR** are m4 (GNU, or perhaps another supporting the -D option), make (GNU, or perhaps another supporting include) and *roff for the docs. Also sh, awk, and POSIX install. Binary distributions are not assembled often so you need an ANSI C compiler environment, e.g. gcc.

A Linux installation requires Linux 0.99.10 or better (1.2.13 is what I actually test on now), an HGC, EGA, VGA, or SVGA graphics card, and a mouse. Mouses supported are: serial Microsoft mouse, serial MouseSystems 3 and 5 byte mouse, serial MMSeries mouse, serial Logitech mouse, PS/2 mouse, or a bus mouse. With Buckey (Meta) hot keys enabled, even a mouseless system could do a certain amount of useful work under **MGR**. The VGA 640x480 monochrome graphics mode is supported out of the box, as is 640x350 and 640x200. To run 800x600, or other modes that your BIOS can initialize and which do not require bank-switching, you need to run a small program (supplied as `src/vgamisc/regs.exe`) under DOS or an emulator to read the VGA registers and write a header file which you place in the directory `src/libbitblit/linux`, so that it can be `#include`'d by the `vga.c` file there. Samples of these files are supplied, but please create your own. Some VGA cards can use 128k windows, and these might run higher monochrome resolutions.

The Linux-colorport code also runs in the standard 320x200x256 color VGA mode without difficulty, because no bank switching is required. If you think of how few 64000 pixels is, you would realize this color mode is quite limited. Non-fast, but simple, bank-switching code has been added in version 0.65, and it works with a Tseng ET4000 card in 640x480x256 and 800x600x256 modes. The S3 code does not work in super VGA resolutions, yet. Supporting new super VGA cards requires writing one function to switch banks and then making sure that the desired screen mode can be initialized from a register dump, possibly with hand-tweaking. The Linux color servers generally mangle the screen fonts, necessitating use of restorefont as in runx. If someone were to extract the VGA initialization code out of X, this might make MGR work on a lot more color systems.

Suns with SunOS 4.1.2+ and `bwtwo`, `cgthree`, or `cgsix` frame buffers are supported. Their speed handling color is good. Coherent installations should refer to the `Versions/README.Coh` file in the

TOC

source distribution. Porting the latest-and-greatest **MGR** to another POSIX-like system which provi `select()` and pty's and direct access to a bitmapped frame-buffer ought to be straightforward, just implementing the `libbitblit` library based on the `sunmono` or `colorport` code, say.

If you want to install everything, you need 7 MB disk space for binaries, fonts, manual pages etc. The sources are about 4.5 MB, plus object files during compilation.

Normally, `/usr/mgr` should be either the directory or a link to the directory where you install **MGR** stuff for runtime use. Typing

```
cd /usr/mgr; tar xvfz whereveryouputit/mgrusr-0.69.tgz
```

and optionally

```
cd /usr/mgr; tar xvfz wherever/morefonts-0.69.tgz
```

will unpack these. The source can be put anywhere, e.g. typing

```
cd /usr/src/local/mgr; tar xvfz wherever/mgrsrc-0.69.tgz
```

to unpack the sources from `archimedes.nosc.mil`.

The source tree can be compiled from one top-level Makefile which invokes lower-level Makefiles, all of which "include" a `"Configfile"` at the top level. The `Configfile` is created by an interactive sh script named `Configure`, which asks you questions, then runs m4 on a `Configfile.m4`. So you type something like this:

```
chdir /usr/src/local/mgr
sh ./Configure
make first
make depend
make install
make clean
```

It might be wise, before running make, to eyeball the `Configfile` generated by the `Configure` script, checking that it looks reasonable. (At least one m4 poops out (Sun `/usr/bin/m4`), creating a very short `Configfile`. If this happens, try hand editing a copy of `Configfile.sun` or `Configfile.lx`) One can also `make all` in any directory with a Makefile as soon as the libraries have been compiled and installed. The server, libraries, and some clients have been linted, but several clients are K&R C code that generates many compiler warnings.

Several flags in MGRFLAGS can be added/omitted in the Configfile to change some optional features in the server, viz:

-DWHO

muck utmp file so "who" works

-DVI

code for clicking the mouse in vi moving the cursor

TOC

-DDEBUG

enable debugging output selectable with -d options.

-DFASTMOUSE

XOR the mouse track

-DBUCKEY

for hot-key server commands without mousing

-DPRIORITY

for priority window scheduling instead of round-robin; the active window gets higher priority

-DCUT

for cut/paste between windows and a global snarf buffer

-DMGR_ALIGN

forces window alignment for fast scrolling (monochrome)

-DKILL

kills windows upon tty i/o errors

-DSHRINK

use only some of the screen ($MGRSIZE in environment)

-DNOSTACK

don't permit event stacking

-DBELL

audibly ring the bell

-DKBD

read `mgr` input from the sun kbd, instead of stdin. This permits redirection of console msgs to a window.

-DFRACCHAR

fractional character movement for proportional fonts

-DXMENU

extended menu stuff (experimental)

-DMOVIE

movie making extension which logs all operations to a file for later replay -- not quite working under Linux

-DEMUMIDMSBUT

Emulate a missing middle mouse button by chording

TOC

Not all combinations of these options have been tested on all systems.

The BITBLITFLAGS macro should contain `-DBANKED` if you're trying out the super VGA color.

C code for the static variables in the server containing icons and fonts is generated by a translator from icon and font files.

Not all the clients are compiled and installed by the Makefiles. Clients found under `src/clients` having capitalized names or not compiled by the supplied Makefiles may have problems compiling and/or running, but they may be interesting to hack on. Most of the screen drivers found under the `libbitblit` directory are of mainly archeological interest. Grave robbing can be profitable.

At some point check that your `/etc/termcap` and/or `terminfo` file contain entries for **MGR** terminals such as found in the `misc` directory. If all your software checks $TERMCAP in the environment, this is not needed, as long as you run `eval `set_termcap` ` in each window.

MGR works better if run setuid root, because it wants to chown ptys and write in the utmp file. This helps the ify iconifier client work better and the event passing mechanism be more secure. On Linux, root permissions are *required* in order to do in/out on the screen device. Otherwise, you decide whether to trust it.

In versions around 0.62 there are troubles on the Sun with using the csh as the default shell. Programs seem to run in a different process group than the foreground process group of the window's pty, in contradiction to man pages and posix specs. There is no trouble with bash, sh, or rc. Ideas why?

---

4. Running MGR

The only file *required* in an **MGR** installation is the server itself. That would give you terminal emulator windows with shells running in them and cutting and pasting with the mouse, but no nice clocks, extra fonts, fancy graphics, etc. Depending on options, a monochrome server needs about 200K of RAM plus dynamic space for windows, bitmaps, etc.

If `/usr/mgr/bin` is in your PATH, then just type "`mgr`" to start up. After enjoying the animated startup screen, press any key. When the hatched background and mouse pointer appear, hold down the left mouse button, highlight the "new window" menu item, and release the button. Then drag the mouse from corner to corner where you want a window to appear. The window will have your default shell running in it. Hold down the left mouse button over an existing window to see another menu for doing things to that window. Left-clicking on an obscured window raises it to the top. The menu you saw that pops-up over the empty background includes the quit command. For people with a two button mouse: press both buttons together to emulate the missing middle button used by some clients.

The quit submenu includes the "really quit" option, a suspend option which should only be used if you run a job-control shell, and a screen saver and locker option, which waits for you to type your login password when you come back to your machine.

When trying to run **MGR**, if you get:

can't find the screen

> make sure you have a `/dev` entry for your display device, e.g. on a Sun `/dev/bwtwo0`. If not, as root cd to `/dev`, and type "MAKEDEV bwtwo0". Otherwise, you might need the `-S/dev/bwtwo0` or (on Linux) the `-S640x480` command line option when starting `mgr`. On Linux, you might also make sure that `/usr/mgr/bin/mgr` was installed setuid root.

can't find the mouse

> make sure `/dev/mouse` exists, usually as a symbolic link to the real device name for your mouse. If you haven't permission to write in `/dev`, then something like a `-m/dev/cua0` option can be given when starting `mgr`. Also, make sure you've supplied the right mouse protocol choice when you configured `mgr`. The mouse may speak Microsoft, even if that is not the brand name.

can't get a pty

> make sure all of `/dev/[tp]ty[pq]?` are owned by root, mode 666, and all programs referenced with the "shell" option in your `.mgrc` startup file (if any) exist and are executable.

none but the default font

> make sure **MGR** is looking in the right place for its fonts. Check the `Configfile` in the source or see whether a `-f/usr/mgr/font` option to `mgr` fixes the problem.

completely hung (not even the mouse track moves)

TOC

login to your machine from another terminal (or rlogin) and kill the `mgr` process. A buckey-Q can quit **MGR** if the keyboard still works.

4.1 Applications not aware of MGR

Any tty-oriented application can be run in an **MGR** window without further ado. Screen-oriented applications using termcap or curses can get the correct number of lines and columns in the window by your using `shape(1)` to reshape the window or using `set_termcap(1)` to obtain the correct termcap entry.

4.2 MGR Applications (clients) distributed with the server

bdftomgr

converts some BDF fonts to MGR fonts

browse

an icon browser

bury

bury this window

c_menu

vi menus from C compiler errors

clock

digital display of time of day

clock2

analog display of time of day

close

close this window, iconify

color

set the foreground and background color for text in this window

colormap

read or write in the color lookup table

cursor

change appearance of the character cursor

cut

TOC

cut text from this window into the cut buffer

cycle

display a sequence of icons

dmgr

crude ditroff previewer

fade

fade a home movie script from one scene to another

font

change to a new font in this window

gropbm

a groff to PBM driver using Hershey fonts

hpmgr

hp 2621 terminal emulator

ico

animate an icosahedron or other polyhedron

iconmail

notification of mail arrival

iconmsgs

message arrival notification

ify

iconify and deiconify windows

loadfont

load a font from the file system

maze

a maze game

mclock

micky mouse clock

menu

create or select a pop-up menu

mgr

bellcore window system server and window manager

mgrbd

TOC

boulder-dash game

mgrbiff

watch mailbox for mail and notify

mgrload

graph of system load average

mgrlock

lock the console

mgrlogin

graphical login controller

mgrmag

magnify a part of the screen, optionally dump to file

mgrmail

notification of mail arrival

mgrmode

set or clear window modes

mgrmsgs

message arrival notification

mgrplot

Unix "plot" graphics filter

mgrsclock

sandclock

mgrshowfont

browse through mgr fonts

mgrsketch

a sketching/drawing program

mgrview

view mgr bitmap images

mless

start up less/more in separate window, menu added for less

mnew

startup up any program in a separate, independent window

mphoon

TOC

display the current phase of the moon

mvi

start up vi in a separate window, with mouse pointing

oclose

(old) close a window

omgrmail

(old) notification of mail arrival

pbmrawtomgr, pgmrawtomgr, ppmrawtomgr

convert raw PBM/PGM/PPM image files to mgr bitmap format

pbmstream

split out a stream of bitmaps

pbmtoprt

printer output from PBM

pgs

ghostscript patch and front end, a PS viewer

pilot

a bitmap browser, or image viewer

resetwin

cleanup window state after client crashes messily

rotate

rotate a bitmap 90 degrees.

screendump

write graphics screen dump to a bitmap file

set_console

redirect console messages to this window

set_termcap

output an appropriate TERM and TERMCAP setting

setname

name a window, for messages and iconifying

shape

reshape this window

square

TOC

square this window

squeeze

compress mgr bitmap using run-length encoding

startup

produce a skeleton startup file for current window layout

texmgr

TeX dvi file previewer

text2font, font2text

convert between mgr font format and text dump

unsqueeze

uncompress mgr bitmap using run length encoding

vgafont2mgr, mgrfont2vga

convert between mgr font format and VGA

window_print

print an image of a window

zoom

an icon editor

bounce, grav, grid, hilbert, mgreyes, stringart, walk

graphics demos

4.3 MGR-aware clients distributed separately, see "SUPPORT" file

calctool

on-screen calculator

chess

frontend to `/usr/games/chess`

gnu emacs

editor with `lisp/term/mgr.el` mouse & menu support

gnuplot

universal scientific data plotting

metafont

font design and creation

origami

folding editor

pbmplus

portable bitmap format conversions, manipulations

plplot

slick scientific data plotting

The Emacs support in `misc/mgr.el` and `misc/mailcap` includes very usable MIME support, via Rmail and metamail.

A general image viewer could be cobbled together from `pilot` and the netPBM filters, but I have not taken the time to do it.

---

5. Programming for MGR

The **MGR** programmers manual, the C language applications interface, is found in the doc directory in troff/nroff form. It covers general concepts, the function/macro calls controlling the server, a sample application, with an index and glossary.

Porting client code used with older versions of **MGR** sometimes requires the substitution of

```
#include <mgr/mgr.h>
```

for

```
#include <term.h>
#include <dump.h>
```

and clients using old-style B_XOR, B_CLEAR, et al instead of BIT_XOR, BIT_CLR, et al can be accommodated by writing

```
#define OLDMGRBITOPS
#include <mgr/mgr.h>
```

Compiling client code generally requires compiler options like the following.

```
-I/usr/mgr/include   -L/usr/mgr/lib -lmgr
```

One can get some interactive feel for the **MGR** server functions by reading and experimenting with the `mgr.el` terminal driver for GNU Emacs which implements the **MGR** interface library in ELisp.

The usual method of inquiring state from the server has the potential of stumbling on a race condition if the client also expects a large volume of event notifications. The problem arises if an (asynchronous) event notification arrives when a (synchronous) inquiry response was expected. If this arises in practice (unusual) then the **MGR** state inquiry functions would have to be integrated with your event handling loop.

The only major drawing function missing from the **MGR** protocol, it seems, is an area fill for areas other than upright rectangles. There is new code for manipulating the global colormap, as well as (advisory) allocation and freeing of color indices owned by windows.

If you are thinking of hacking on the server, you can find the mouse driver in `mouse.*` and `mouse_get.*`, the grotty parts of the keyboard interface in `kbd.c`, and the interface to the display in the `src/libbitblit/*` directories. The main procedure, much initialization, and the top level input loop are in `mgr.c`, and the interpretation of escape sequences is in `put_window.c`.

6. More documentation

The programmer's manual is essential for concepts.

Nearly all the clients supplied come with a man page which is installed into `/usr/mgr/man/man1` or `man6`. Other useful man pages are `bitblit.3`, `font.5`, and `bitmap.5`. There is some ambiguity in the docs in distinguishing the internal bitmap format found in your frame-buffer and the external bitmap format found in files, e.g. icons.

The `mgr.1` man page covers command line options, commands in the `~/.mgrc` startup file, mouse and menu interaction with the server, and hot-key shortcuts available on systems with such hot-keys.

Many of the fonts in `/usr/mgr/font/*` are described to some extent in `/usr/mgr/font/*.txt`, e.g. `/usr/mgr/font/FONTDIR.txt` gives X-style font descriptions for the fonts obtained in .bdf format. Font names end in `WxH`, where `W` and `H` are the decimal width and height in pixels of each character box.

7. Credit for MGR

Stephen Uhler, with others working at Bellcore, was the original designer and implementer of **MGR**, so Bellcore has copyrighted much of the code and documentation for **MGR** under the following conditions.

```
* Permission is granted to copy or use this program, EXCEPT that it
* may not be sold for profit, the copyright notice must be reproduced
* on copies, and credit should be given to Bellcore where it is due.
```

One required showing of the copyright notice is the startup title screen.

Other credits to:

- Stephen Hawley for his wonderful icons.
- Tommy Frandsen for the VGA linux library.
- Tom Heller for his Gasblit library.
- Andrew Haylett for the Mouse driver code.
- Dan McCrackin for his gasblit->linux patches.
- Dave Gymer, dgymer@gdcarc.co.uk, for the Startrek effect fix.
- Alex Liu for first releasing a working Linux version of **MGR**.
- Lars Aronsson (aronsson@lysator.liu.se) for text2font and an ISO8859-1 8-bit font.
- Harry Pulley (hcpiv@grumpy.cis.uoguelph.ca, hcpiv@snowhite.cis.uoguelph.ca) for the Coherent port.
- Vance Petree & Grant Edwards & Udo Munk for their work on Hercules.
- Udo Munk for his work on serial mouse initialization & select.
- Norman Bartek & Hal Snyder at Mark Williams Co. for their help with some bugs & with Coherent device drivers.
- Extra thanks to Zeyd Ben Halim for lots of helpful patches, especially the adaptation of selection.
- Bradley Bosch, brad@lachman.com, for lots of patches from his 3b1 port, which fix bugs and implement new and desirable features.
- Andrew Morton, applix@runxtsa.runx.oz.au, who first wrote the cut-word code.
- Kapil Paranjape, kapil@motive.math.tifr.res.in, for the EGA support.
- Michael Haardt for MOVIE support fixes, bug fixes, separation of the libbitblit code into output drivers, expansion of the libmgr, and origami folding of the code.
- Yossi Gil for many fonts.
- Carsten Emde, carsten@thlmak.pr.net.ch, for mphoon.
- Vincent Broman for middle mouse-button emulation, linting, Sun cgsix support, VGA colormap acess, integration of the sunport code into Haardt's layering scheme, font gathering, the screen saver, and continued maintenance.
- Kenneth Almquist, ka@socrates.hr.att.com, for helpful bug reports.

- Tim Pierce, twpierce@midway.uchicago.edu, for the port to FreeBSD 2.0R with Trident VGA.

All bitmap fonts from any source are strictly public domain in the USA. The 583 fixed-width fonts supplied with **MGR** were obtained from Uhler, the X distribution, Yossi Gil, and elsewhere. The Hershey vector fonts and the code for rendering them are probably freely redistributable.

Alpha Miniloader Howto

David A. Rusling, david.rusling@reo.mts.dec.com

v0.84, 6 December 1996

This document describes the Miniloader, a program for Alpha based systems that can be used to initialize the machine and load Linux. The Alpha Linux Miniloader (to give it it's full name) is also known as MILO.

1. Introduction

- 1.1 Copyright
- 1.2 New Versions of this Document

2. What is MILO?

3. Pre-Built Standard MILO Images.

4. How To Build MILO

5. How To Load MILO

- 5.1 Loading MILO from the Windows NT ARC firmware
- 5.2 Loading MILO from the Evaluation Board Debug Monitor
- 5.3 Loading MILO from a Failsafe Boot Block Floppy
- 5.4 Loading MILO from Flash
- 5.5 Loading MILO from the SRM Console
- 5.6 System Specific Information

TOC

6. MILO's User Interface

- 6.1 The "help" Command
- 6.2 Booting Linux
- 6.3 Rebooting Linux
- 6.4 The "bootopt" command

7. Running the Flash Management Utility

- 7.1 The "help" command
- 7.2 The "list" command
- 7.3 The "program" command
- 7.4 The "environment" command
- 7.5 The "bootopt" command
- 7.6 The "quit" command

8. Restrictions.

9. Problem Solving.

10. Acknowledgements.

---

1. Introduction

This document describes the Miniloader for Linux on Alpha AXP (MILO). This firmware is used to initialize Alpha AXP based systems, load and start Linux and, finally, provide PALcode for Linux.

1.1 Copyright

The Alpha Miniloader (MILO) HOWTO is copyright (C) 1995, 1996 David A Rusling.

Copyright. Like all Linux HOWTO documents, it may be reproduced and distributed in whole or in part, in any medium, physical or electronic, so long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however the author would *like* to be notified of such distributions. You may translate this HOWTO into any language whatsover provided that you leave this copyright statement and disclaimer intact, and that you append a notice stating who translated the document.

Disclaimer. While I have tried to include the most correct and up to date information available to me, I cannot guarantee that usage of information in this document does not result in loss of data or equipment. I provide NO WARRENTY about the information in the HOWTO and I cannot be made liable for any consequences resulting from using the information in this HOWTO.

1.2 New Versions of this Document

The latest version of this document can be found in ftp://gatekeeper.dec.com/pub/Digital/Linux-Alpha/Miniloader/docs and David Mosberger-Tang is kind enough to include the html form of it in his excellent Linux Alpha FAQ site http://www.azstarnet.com/~axplinux.

2. What is MILO?

On Intel based PC systems, the BIOS firmware sets up the system and then loads the image to be run from the boot block of a DOS file system. This is more or less what MILO does on an Alpha based system, however there are several interesting differences between BIOS firmware and MILO, not least of which is that MILO includes and uses standard Linux device drivers unmodified. MILO is firmware, unlike LILO, which relies on the BIOS firmware to get itself loaded. The main functional parts of MILO are:

1. PALcode,
2. Memory set up code (builds page tables and turns on virtual addressing),
3. Video code (BIOS emulation code and TGA (21030)),
4. Linux kernel code. This includes real Linux kernel code (for example, the interrupt handling) and ersatz or mock Linux kernel,
5. Linux block device drivers (for example, the floppy driver),
6. File system support (ext2, MS-DOS and ISO9660),
7. User inteface code (MILO),
8. Kernel interface code (sets up the HWRPB and memory map for linux),
9. NVRAM code for managing environment variables.

The following paragraphs describe these functional parts in more detail.

PALcode can be thought of as a tiny software layer that tailors the chip to a particular operating system. It runs in a special mode (PALmode) which has certain restrictions but it uses the standard Alpha instruction set with just five extra instructions. In this way, the Alpha chip can run such diverse operating systems as Windows NT, OpenVMS, Digital Unix and, of course, Linux. The PALcode that MILO uses (and therefore Linux itself) is, like the rest of MILO, freeware. It is derived from Digital's Evaluation Board software example Digital Unix PALcode.. The differences between the different PALcodes are because of differences in address mapping and interrupt handling that exist between the Alpha chips (21066 based systems have a different I/O map to 21064+2107x systems) and different Alpha based systems.

For MILO to operate properly it needs to know what memory is available, where Linux will eventually be running from and it must be able to allocate temporary memory for the Linux device drivers. The code maintains a memory map that has entries for permanent and temporary allocated pages. As it boots, MILO uncompresses itself into the correct place in physical memory. When it passes control to the Linux kernel, it reserves memory for the compressed version of itself, the PALcode (which the kernel needs) and some data structures. This leaves `most` of the memory in the system for Linux itself.

The final act of the memory code is to set up and turn on virtual addressing so that the data structures that Linux expects to see are at the correct place in virtual memory.

MILO contains video code that initialises and uses the video device for the system. It will detect and a VGA device if there is one, otherwise it will try to use a TGA (21030) video device. Failing that, it assume that there is no graphics device. The BIOS emulation that the standard, pre-built, images include is Digital's own BIOS emulation which supports most, if not all, of the standard graphics devices available.

Linux device drivers live within the kernel and expect certain services from the kernel. Some of these services are provided directly by Linux kernel code, for example the interrupt handling and some is provided by kernel look-alike routines.

MILO's most powerful feature is that you can embed unaltered Linux device drivers into it. This gives it the potential to support every device that Linux does. MILO includes all of the block devices that are configured into the Linux kernel that it is built against as well as a lot of the block device code (for example, ll_rw_blk()).

MILO loads the Linux kernel from real file systems rather than from boot blocks and other strange places. It understands MSDOS, EXT2 and ISO9660 filesystems. Gzip'd files are supported and these are recommended, particularly if you are loading from floppy which is rather slow. MILO recognises these by their *.gz* suffix.

Built into MILO is a simple keyboard driver which, together with an equally simple video driver allows it to have a simple user interface. That interface allows you to list file systems on configured devices, boot Linux or run flash update utilities and set environment variables that control the system's booting. Like LILO, you can pass arguments to the Kernel.

MILO must tell the Linux kernel what sort of system this is, how much memory there is and which of that memory is free. It does this using the HWRPB (Hardware Restart Parameter Block) data structure and associated memory cluster descriptions. These are placed at the appropriate place in virtual memory just before control is passed to the Linux kernel.

---

3. Pre-Built Standard MILO Images.

If you are planning to run Linux on a standard Alpha based system, then there are pre-built "standard" MILO images that you might use. These (along with the sources and other interesting stuff) can be found in ftp://gatekeeper.dec.com/pub/Digital/Linux-Alpha/Miniloader.

The `images` subdirectory contains a directory per standard system (eg AlphaPC64) with MILO images having the following naming convention:

1. `MILO` - Miniloader executable image, this image can be loaded in a variety of ways,
2. `fmu.gz` - Flash management utility,
3. `MILO.dd` - Boot block floppy disk image. These should be written using rawrite.exe or `dd` on Linux.

The `test-images`, like the `images` subdirectory contains a directory per standard system. These images are somewhat experimental but tend to contain all the latest features.

4. How To Build MILO

You build MILO seperately from the Kernel. As MILO requires parts of the kernel to function (for example interrupt handling) you must first configure and build the kernel that matches with MILO that you want to build. Mostly this means building the kernel with the same version number. So, MILO-2.0.25.tar.gz will build against linux-2.0.25.tar.gz. MILO `may` build against a higher version of the kernel, but there again it may not. Also, now that ELF shared libraries are fully supported, there are two versions of the MILO sources. To build under an ELF system you must first unpack the standard MILO sources and then patch those sources with the same version numbered ELF patch. In the remainder of this discussion, I assume that your kernel sources and object files are stored in the subtree at `/usr/src/linux` and that the linux kernel has been fully built with the command `make boot`

To build MILO, change your working directory to the MILO source directory and invoke `make` with:

```
$    make KSRC=/usr/src/linux config
```

Just like the Linux kernel, you will be asked a series of questions

```
        Echo output to the serial port (MINI_SERIAL_ECHO) [y]
```

It's a good idea to echo kernel printk to `/dev/ttyS0` if you can. If you can (and want to), then type "y", otherwise "n". All of the standard, pre-built, MILO images include serial port I/O using COM1.

```
        Use Digital's BIOS emulation code (not free) (MINI_DIGITAL_BIOS_EMU) [y]
```

This code is included as a library which is freely distributable so long as it is used on an Alpha based system. The sources are not available. If you answer `n` then the freeware alternative BIOS emulation will be built. It's sources are included with MILO. Note that you cannot right now build choose Digital's BIOS emulation code in an ELF system (the library is not yet ready) and so you must answer no to this question.

```
        Build PALcode from sources (Warning this is dangerous)
    (MINI_BUILD_PALCODE_FROM_SOURCES) [n]
```

You should only do this if you have changed the PALcode sources, otherwise use the standard, pre-built PALcode included with MILO.

You are now all set to build the MILO image itself:

```
$    make KSRC=/usr/src/linux
```

When the build has successfully completed, the MILO image is in the file called `milo`. There are a lot of images called `milo.*`, these should be ignored.

5. How To Load MILO

The most commonly supported method of loading MILO is from the Windows NT ARC firmware as most shipping systems support this. However, there are a wide variety of loading MILO. It may be loaded from:

- a failsafe boot block floppy,
- the Windows NT ARC firmware,
- Digital's SRM console,
- an Alpha Evaluation Board Debug Monitor,
- flash/ROM.

5.1 Loading MILO from the Windows NT ARC firmware

Most, if not all, Alpha AXP based systems include the Windows NT ARC firmware and this is the prefered method of booting MILO and thus Linux. Once the Windows NT firmware is running and you have the correct MILO image for your system, this method is completely generic.

The Windows NT ARC firmware is an environment in which programs can run and make callbacks into the firmware to perform actions. The Windows NT OSLoader is a program that does exactly this. Linload.exe is a much simpler program which does just enough to load and execute MILO. It loads the appropriate image file into memory at 0x00000000 and then makes a swap-PAL PALcall to it. MILO, like Linux, uses a different PALcode to Windows NT which is why the swap has to happen. MILO relocates itself to 0x200000 and continues on through the PALcode reset entry point as before.

Before you add a Linux boot option, you will need to copy linload.exe and the appropriate MILO that you wish to load to someplace that the Windows NT ARC firmware can read from. In the following example, I assume that you are booting from a DOS format floppy disk:

1. At the boot menu, select `"Supplementary menu..."`
2. At the `"Supplementary menu"`, select `"Set up the system..."`
3. At the `"Setup menu"`, select `"Manage boot selection menu..."`
4. In the `"Boot selections menu"`, choose `"Add a boot selection"`
5. Choose `"Floppy Disk 0"`
6. Enter `"linload.exe"` as the osloader directory and name
7. Say "yes" to the operating system being on the same partition as the osloader
8. Enter "\" as the operating system root directory
9. I usually enter `"Linux"` as the name for this boot selection

10. Say "No" you do not want to initialise the debugger at boot time
11. You should now be back in the `"Boot selections menu"`, choose the `"Change a boot selection option"` and pick the selection you just created as the one to edit
12. Use the down arrow to get `"OSLOADFILENAME"` up and then type in the name of the MILO image that you wish to use, for example `"noname.arc"` followed by return.
13. Press ESC to get back to the `"Boot Selections menu"`
14. Choose `"Setup Menu"` (or hit ESC again) and choose `"Supplementary menu, and save changes"` option
15. ESC will get you back to the `"Boot menu"` and you can attempt to boot MILO. If you do not want Linux as the first boot option, then you can alter the order of the boot options in the `"Boot selections menu"`.

At the end of all this, you should have a boot selection that looks something like:

```
LOADIDENTIFIER=Linux
SYSTEMPARTITION=multi(0)disk(0)fdisk(0)
OSLOADER=multi(0)disk(0)fdisk(0)\linload.exe
OSLOADPARTITION=multi(0)disk(0)fdisk(0)
OSLOADFILENAME=\noname.arc
OSLOADOPTIONS=
```

You can now boot MILO (and then Linux). You can load linload.exe and MILO directly from a file system that Windows NT understands such as NTFS or DOS on a hard disk.

The contents `OSLOADOPTIONS` are passed to MILO which interprets it as a command. So, in order to boot Linux directly from Windows NT without pausing in MILO, you could pass the following in `OSLOADOPTIONS`:

```
boot sda2:vmlinux.gz root=/dev/sda2
```

See MILO's User Interface for more information on the commands available.

Another (rather sneaky) way of loading of loading MILO via the WNT ARC firmware is to put MILO onto an MS-DOS floppy and call it `fwupdate.exe` and then choose the "Upgrade Firmware" option.

5.2 Loading MILO from the Evaluation Board Debug Monitor

Evaluation boards (and often designs cloned from them) include support for the Alpha Evaluation Board Debug Monitor. Consult your system document before considering this method of booting MILO. The following systems are *known* to include Debug Monitor support:

- AlphaPC64 (Section AlphaPC64)
- EB64+ (Section EB64+)

- EB66+ (Section EB66+)
- EB164 (Section EB164)
- PC164 (Section PC164)

Before you consider this method, you should note that the early versions of the Evaluation Board Debug Monitor did not include video or keyboard drivers and so you must be prepared to connect another system via the serial port so that you can use the Debug Monitor. Its interface is very simple and typing help shows a whole heap of commands. The ones that are most interesting include the word `boot` or `load` in them.

The Evaluation Board Debug Monitor can load an image either via the network (netboot) or via a floppy (flboot). In either case, set the boot address to 0x200000 (`> bootadr 200000`) before booting the image.

If the image is on floppy (and note that only DOS formatted floppies are supported), then you will need to type the following command:

```
AlphaPC64> flboot <MILO-image-name>
```

5.3 Loading MILO from a Failsafe Boot Block Floppy

Only the AxpPCI33 is *known* to include failsafe boot block floppy support (Section < id="noname-section" name="Noname">).

If you do not have a standard pre-built MILO .dd image, then you may need to build an SRM boot block floppy. Once you have built MILO, you need to do the following on Digital Unix box:

```
fddisk -fmt /dev/rfd0a
cat mboot bootm > /dev/rfd0a
disklabel -rw rfd0a 'rx23' mboot bootm
```

Or on a Linux box:

```
cat mboot bootm > /dev/fd0
```

If you have a standard MILO image available (say `MILO.dd`) then you would build a boot block floppy using the following command:

```
dd if=MILO.dd of=/dev/fd0
```

TOC

5.4 Loading MILO from Flash

There are a number of systems where MILO can be blown into flash and booted directly (instead of via the Windows NT ARC firmware):

- AlphaPC64 (Section AlphaPC64)
- Noname (Section Noname)
- EB66+ (Section EB66+)
- EB164 (Section EB164)
- PC164 (Section PC164)

5.5 Loading MILO from the SRM Console

The SRM (short for System Reference Manual) Console knows nothing about filesystems or disk-partitions, it simply expects that the secondary bootstrap loader occupies a consecutive range of physical disk sectors starting from a given offset. The information describing the secondary bootstrap loader (its size and offset) is given in the first 512 byte block. To load MILO via the SRM you must generate that structure on a device which the SRM can access (such as a floppy disk). This is what `mboot` and `bootm`, `mboot` is the first block (or boot description) and `mboot` is the `MILO` image rounded up to a 512 byte boundary.

To load MILO from a boot block device, either build `mboot` and `bootm` and push them onto the boot device using the following command:

```
$ cat mboot bootm > /dev/fd0
```

Or, grab the appropriate `MILO.dd` from a web site and write it onto the boot device using either `RAWRITE.EXE` or `dd`.

Once you have done that you can boot the SRM console and use one of its many commands to boot MILO. For example, to boot MILO from a boot block floppy you would use the following command:

```
>>>boot dva0
(boot dva0.0.0.0.1 -flags 0)
block 0 of dva0.0.0.0.1 is a valid boot block
reading 621 blocks from dva0.0.0.0.1
bootstrap code read in
base = 112000, image_start = 0, image_bytes = 4da00
initializing HWRPB at 2000
initializing page table at 104000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
MILO Stub: V1.1
```

```
Unzipping MILO into position
Allocating memory for unzip
####...
```

The following systems are *known* to have SRM Console support:

- Noname (Section Noname)
- AlphaPC64 (Section AlphaPC64)
- EB164 (Section EB164)
- PC164 (Section PC164)

5.6 System Specific Information

AxpPCI33 (Noname)

The Noname board can load MILO from the Windows NT ARC firmware (Section booting from Windows NT ARC firmware), from the SRM Console (Section Loading MILO from the SRM Console). and from a failsafe boot block floppy (Section Loading from a Failsafe Boot Block Floppy). A flash management utility, runnable from MILO is available so that once MILO is running, it can be blown into flash (Section running the flash management utility). However, be warned that once you have done this you will lose the previous image held there as there is only room for one image.

The way that Noname boots is controlled by a set of jumpers on the board, J29 and J28. These look like:

```
                4
J29     2 x x x 6
        1 x x x 5

J28     2 x x x 6
        1 x x x 5
                3
```

The two options that we're interested in are J28, pins 1-3 which boots the console/loader from flash and J29, pins 1-3 which boots the console/loader from a boot block floppy. The second option is the one that you need to first boot MILO on the Noname board.

Once you've selected the boot from floppy option via the jumpers, put the SRM boot block floppy containing MILO into the floppy and reboot. In a few seconds (after the floppy light goes out) you should see the screen blank to white and MILO telling you what's going on.

If you are really interested in technical stuff, the Noname loads images off of the floppy into physical address 0x104000 and images from flash into 0x100000. For this reason, MILO is built with it's PALcode starting at 0x200000. When it is first loaded, it moves itself to the correct location (see relocate.S).

AlphaPC64 (Cabriolet)

TOC

The AlphaPC64 includes the Windows NT ARC firmware (Section booting from Windows NT ARC firmware), the SRM Console (Section Loading MILO from the SRM Console) and the Evaluation Debug Monitor (Section Loading from the Debug Monitor). These images are in flash and there is room to add MILO so that you can boot MILO directly from flash. A flash management utility, runnable from MILO is available so that once MILO is running, it can be blown into flash (Section running the flash management utility). This system supports MILO environment variables.

You select between the boot options (and MILO when it is been put into flash) using a combination of jumpers and a boot option which is saved in the NVRAM of the TOY clock.

The jumper is J2, SP bits 6 and 7 have the following meanings:

- SP bit 6 should always be out. If this jumper is set then the SROM mini-debugger gets booted,
- SP bit 7 in is boot image selected by the boot option byte in the TOY clock,
- SP bit 7 out is boot first image in flash.

So, with bit 7 out, the Debug Monitor will be booted as it is `always` the first image in flash. With bit 7 in, the image selected by the boot option in the TOY clock will be selected. The Debug Monitor, the Windows NT ARC firmware and MILO all support setting this boot option byte but you must be very careful using it. In particular, you cannot set the boot option so that next time the system boots MILO when you are running the Windows NT ARC firmware, it only allows you to set Debug Monitor or Windows NT ARC as boot options.

To get MILO into flash via the Evaluation Board Debug Monitor, you will need a flashable image. The build proceedures make MILO.rom, but you can also make a rom image using the makerom tool in the Debug Monitor software that comes with the board:

```
> makerom -v -i7 -l200000 MILO -o mini.flash
```

(type makerom to find out what the arguments mean, but 7 is a flash image id used by the srom and -l200000 gives the load address for the image as 0x200000).

Load that image into memory (via the Debug Monitor commands flload, netload, and so on) at 0x200000 and then blow the image into flash:

```
AlphaPC64> flash 200000 8
```

(200000 is where the image to be blown is in memory and 8 is the segment number where you put the image. There are 16 1024*64 byte segments in the flash and the Debug Monitor is at seg 0 and the Windows NT ARC firmware is at seg 4).

Set up the image that the srom will boot by writing the number of the image into the TOY clock.

```
AlphaPC64> bootopt 131
```

(131 means boot the 3rd image, 129 = 1st, 130 = 2nd and so on).

TOC

Power off, put jumper 7 on and power on and you should see the MILO burst into life. If you don't th take jumper 7 back off and reboot the Debug Monitor.

EB66+

The EB66+, like all of the Alpha Evaluation Boards built by Digital contains the Evaluation Board Debug Monitor and so this is available to load MILO (Section Loading from the Debug Monitor). Quite often (although not always) boards whose design is derived from these include the Debug Monitor also. Usually, these boards include the Windows NT ARC firmware (Section booting from Windows NT ARC firmware). A flash management utility, runnable from MILO is available so that once MILO is running, it can be blown into flash (Section running the flash management utility). This system supports MILO environment variables.

These systems have several boot images in flash controlled by jumpers. The two jumper banks are J18 and J16 and are located at the bottom of the board in the middle (if the Alpha chip is at the top). You select between the boot options (and MILO when it is been put into flash) using a combination of jumpers and a boot option which is saved in the NVRAM of the TOY clock.

Jumper 7-8 of J18 in means boot the image described by the boot option. Jumper 7-8 of J18 out means boot the Evaluation Board Debug Monitor.

Blowing an image into flash via the Evaluation Board Debug Monitor is exactly the same proceedure as for the AlphaPC64 (Section AlphaPC64).

EB64+/Aspen Alpine

This system is quite like the AlphaPC64 except that it does not contain flash which MILO can be loaded from. The EB64+ has two ROMs, one of which contains the Windows NT ARC firmware (Section booting from Windows NT ARC firmware). and the other contains the Evaluation Board Debug Monitor (Section Loading from the Debug Monitor).

The Aspen Alpine is a little different in that it only has one ROM; this contains the Windows NT ARC firmware.

Universal Desktop Box (Multia)

This is a very compact pre-packaged 21066 based system that includes a TGA (21030) graphics device. Although you can *just* fit a half height PCI graphics card in the box you are better off waiting for full TGA support in XFree86. It includes the Windows NT ARC firmware and so booting from that is the prefered method (Section Loading from Windows NT).

TOC

EB164

The EB164, like all of the Alpha Evaluation Boards built by Digital contains the Evaluation Board Debug Monitor and so this is available to load MILO (Section Loading from the Debug Monitor). Quite often (although not always) boards whose design is derived from these include the Debug Monitor also. Usually, these boards include the Windows NT ARC firmware (Section booting from Windows NT ARC firmware). The SRM console is also available (Section Loading MILO from the SRM Console). A flash management utility, runnable from MILO is available so that once MILO is running, it can be blown into flash (Section running the flash management utility). This system supports MILO environment variables.

These systems have several boot images in flash controlled by jumpers. The two jumper bank is J1 and is located at the bottom of the board on the left (if the Alpha chip is at the top). You select between the boot options (and MILO when it is been put into flash) using a combination of jumpers and a boot option which is saved in the NVRAM of the TOY clock.

Jumper SP-11 of J1 in means boot the image described by the boot option. Jumper SP-11 of J1 out means boot the Evaluation Board Debug Monitor.

Blowing an image into flash via the Evaluation Board Debug Monitor is exactly the same proceedure as for the AlphaPC64 (Section AlphaPC64).

PC164

The PC164, like all of the Alpha Evaluation Boards built by Digital contains the Evaluation Board Debug Monitor and so this is available to load MILO (Section Loading from the Debug Monitor). Quite often (although not always) boards whose design is derived from these include the Debug Monitor also. Usually, these boards include the Windows NT ARC firmware (Section booting from Windows NT ARC firmware). The SRM console is also available (Section Loading MILO from the SRM Console). A flash management utility, runnable from MILO is available so that once MILO is running, it can be blown into flash (Section running the flash management utility). This system supports MILO environment variables.

These systems have several boot images in flash controlled by jumpers. The main jumper block, J30, contains the system configuration jumpers and jumper CF6 in means that the system will boot the Debug Monitor, the default is out.

Blowing an image into flash via the Evaluation Board Debug Monitor is exactly the same proceedure as for the AlphaPC64 (Section AlphaPC64).

XL266

The XL266 is one of a family of systems that are known as Avanti. It has a riser card containing the Alpha chip and cache which plugs into the main board at right angles. This board can replace the equivalent Pentium board.

Some of these systems ship with the SRM console but others, notably the XL266 ship with only the Windows NT ARC firmware (Section booting from Windows NT ARC firmware).

TOC

Here is my list of compatible systems:

- AlphaStation 400 (Avanti),
- AlphaStation 250,
- AlphaStation 200 (Mustang),
- XL. There are two flavours, XL266 and XL233 with the only difference being in processor speed and cache size.

Note The system that I use to develop and test MILO is an XL266 and so this is the only one that I can guarentee will work. However, technically, all of the above systems are equivalent; they have the same support chipsets and the same interrupt handling mechanisms.

Platform2000

This is a 233Mhz 21066 based system.

---

6. MILO's User Interface

Once you have correctly installed/loaded/run MILO you will see the MILO (for MIniLOader) prompt displayed on your screen. There is a very simple interface that you must use in order to boot a particular Linux kernel image. Typing "help" is a good idea as it gives a useful summary of the commands.

6.1 The "help" Command

Probably the most useful command that MILO has:

```
MILO> help
MILO command summary:

ls [-t fs] [dev:[dir]]
                    - List files in directory on device
boot [-t fs] [dev:file] [boot string]
                    - Boot Linux from the specified device and file
run [-t fs] dev:file
                    - Run the standalone program dev:file
show                - Display all known devices and file systems
set VAR VALUE       - Set the variable VAR to the specified VALUE
unset VAR           - Delete the specified variable
reset               - Delete all variables
print               - Display current variable settings
help [var]          - Print this help text

Devices are specified as: fd0, hda1, hda2, sda1...
Use the '-t filesystem-name' option if you want to use
  anything but the default filesystem  ('ext2').
Use the 'show' command to show known devices and filesystems.
Type 'help var' for a list of variables.
```

Note that the `bootopt` command only appears on AlphaPC64 (and similar) systems. Refer to the board's dcoumentation to find out just what it means.

Devices. Until you use a command that needs to make use of a device, no device inititalisation will take place. The first `show`, `ls`, `boot` or `run` commands all cause the devices within MILO to be initialised. Devices are named in the same way (exactly) that Linux itself will name them. So, the first IDE disk will be called 'hda' and it's first partition will be 'hda1'. Use the `show` command to show what devices are available.

File Systems. MILO supports three file systems, MSDOS, EXT2 and ISO9660. So long as a device is available to it, MILO can `listboot` or `run` an image stored on one of these file systems. MILO's default file system is `EXT2` and so you have tell MILO that the file system is something other than that. All of the commands that use filenames allow you to pass the file system using the `-t [filesystem]`option. So, if you wanted to list the contents of a SCSI CD ROM, you might type the following:

```
MILO> ls -t iso9660 scd0:
```

Variables. MILO contains some settable variables that help the boot process. If you are loading via the Windows NT ARC firmware, then MILO makes use of the boot option environment variables set up by that firmware. For some systems, MILO (for example, the AlphaPC64) maintains its own set of environment variables that do not change from boot to boot. These variables are:

TOC

```
MILO> help var
Variables that MILO cares about:
  MEMORY_SIZE      - System memory size in megabytes
  BOOT_DEV         - Specifies the default boot device
  BOOT_FILE        - Specifies the default boot file
  BOOT_STRING      - Specifies the boot string to pass to the kernel
  SCSIn_HOSTID     - Specifies the host id of the n-th SCSI controller.
  AUTOBOOT         - If set, MILO attempts to boot on powerup
                     and enters command loop only on failure.
  AUTOBOOT_TIMEOUT - Seconds to wait before auto-booting on powerup.
```

6.2 Booting Linux

The `boot` command boots a linux kernel from a device. You will need to have a linux kernel image on an EXT2 formated disk (SCSI, IDE or floppy) or an ISO9660 formatted CD available to MILO. The image can be gzip'd and in this case MILO will recognise that it is gzip'd by the .gz suffix.

You should note that the version of MILO does not usually have to match the version of the Linux kernel that you are loading. You boot Linux using the following command syntax:

```
MILO> boot [-t file-system] device-name:file-name [[boot-option] [boot-option] ...]
```

Where `device-name` is the name of the device that you wish to use and `file-name` is the name of the file containing the Linux kernel. All arguments supplied after the file name are passed directly to the Linux kernel.

If you are installing Red Hat, then you will need to specify a root device and so on. So you would use:

```
MILO> boot fd0:vmlinux.gz root=/dev/fd0 load_ramdisk=1
```

MILO will automatically contain the block devices that you configure into your vmlinux. I have tested the floppy driver, the IDE driver and a number of SCSI drivers (for example, the NCR 810), and these work fine. Also, it is important to set the host id of the SCSI controller to a reasonable value. By default, MILO will initialize it to the highest possible value (7) which should normally work just fine. However, if you wish, you can explicitly set the host id of the *n*-th SCSI controller in the system by setting environment variable `SCSI`*n*`_HOSTID` to the appropriate value. For example, to set the hostid of the first SCSI controller to 7, you can issue the following command at the MILO prompt:

```
setenv SCSI0_HOSTID 7
```

6.3 Rebooting Linux

You may want to reboot a running Linux system using the `shutdown -r now` command. In this case, the Linux kernel returns control to MILO (via the HALT CallPAL entrypoint). MILO leaves a compressed copy of itself in memory for just this reason and detects that the system is being rebooted from information held in the HWRPB (Hardware Restart Parameter Block). In this case it starts to reboot using exactly the same command that was used to boot the Linux kernel the last time. There is a 30 second timeout that allows you to interrupt this process and boot whatever kernel you wish in whatever way you wish.

6.4 The "bootopt" command

For flash based systems such as the AlphaPC64, EB164 and the EB66+, there are a number of possible boot options and these are changed using the `bootopt` command. This has one argument, a decimal number which is the type of the image to be booted the next time the system is power cycled or reset:

0 Boot the Evaluation Board Debug Monitor,

1 Boot the Windows NT ARC firmware.

In order to tell the boot code to boot the MILO firmware from flash then you need a boot option that means boot the N'th image. For this, you need to 128 plus N, so if MILO is the third image, you would use the command:

```
MILO> bootopt 131
```

`Note:` Be very careful with this command. A good rule is never to set bootopt to 0 (the Evaluation Board Debug Monitor), but instead use the system's jumpers to achieve the same thing.

---

7. Running the Flash Management Utility

The `run` command is used to run the flash management utility. Before you start you will need a device available to MILO that contains the updateflash program. This (like vmlinux) can be gzip'd. You need to run the flash management utility program from the MILO using the (`run`) command:

```
MILO> run fd0:fmu.gz
```

Once it has loaded and initialised, the flash management utility will tell you some information about the flash device and give you a command prompt. Again the `help` command is most useful.

```
Linux MILO Flash Management Utility V1.0

Flash device is an Intel 28f008SA
  16 segments, each of 0x10000 (65536) bytes
Scanning Flash blocks for usage
Block 12 contains the environment variables
FMU>
```

`Note` that on systems where environment variables may be stored and where there is more than one flash block (for example, the AlphaPC64) the flash management utility will look for a block to hold MILO's environment variables. If such a block already exists, the flash management utility will tell you where it is. Otherwise, you must use the `environment` command to set a block and initialise it. In the above example, flash block 12 contains MILO's environment variables.

7.1 The "help" command

```
FMU> help
FMU command summary:

list                - List the contents of flash
program             - program an image into flash
quit                - Quit
environment         - Set which block should contain the environment variables
bootopt num         - Select firmware type to use on next power up
help                - Print this help text
FMU>
```

Note that the `environment` and `bootopt` commands are only available on the EB66+, the AlphaPC64, EB164 and PC164 systems (and their clones).

7.2 The "list" command

The "list" command shows the current usage of the flash memory. Where there is more than one flash block, the usage of each flash block is shown. In the example below you can see that Windows NT ARC is using blocks 4:7 and block 15.

```
FMU> list
Flash blocks:  0:DBM  1:DBM  2:DBM  3:WNT  4:WNT  5:WNT  6:WNT  7:WNT  8:MILO
        9:MILO 10:MILO 11:MILO 12:MILO 13:U 14:U 15:WNT
```

TOC

```
Listing flash Images
  Flash image starting at block 0:
    Firmware Id:  0 (Alpha Evaluation Board Debug Monitor)
    Image size is 191248 bytes (3 blocks)
    Executing at 0x300000
  Flash image starting at block 3:
    Firmware Id:  1 (Windows NT ARC)
    Image size is 277664 bytes (5 blocks)
    Executing at 0x300000
  Flash image starting at block 8:
    Firmware Id:  7 (MILO/Linux)
    Image size is 217896 bytes (4 blocks)
    Executing at 0x200000
FMU>
```

7.3 The "program" command

The flash management utility contains a compressed copy of a flash image of MILO. The "program" command allows you to blow this image into flash. The command allows you to back out, but before you run it you should use the "list" command to see where to put MILO. If MILO is already in flash, then the flash management utility will offer to overwrite it.

```
FMU> program
Image is:
    Firmware Id:  7 (MILO/Linux)
    Image size is 217896 bytes (4 blocks)
    Executing at 0x200000
Found existing image at block 8
Overwrite existing image? (N/y)? y
Do you really want to do this (y/N)? y
Deleting blocks ready to program: 8 9 10 11
Programming image into flash
Scanning Flash blocks for usage
FMU>
```

Wait until it has completed before powering off your system.

`Note:` I cannot emphasise just how careful you must be here not to overwrite an existing flash image that you might need or render your system useless. A very good rule is never to overwrite the Debug Monitor.

7.4 The "environment" command

This selects a flash block to contain MILO's environment variables.

7.5 The "bootopt" command

This is just the same as MILO's "bootopt" command, see (Section The "bootopt" command).

7.6 The "quit" command

This is really pretty meaningless. The only way back to MILO (or anything else) once the flash management utility has run is to reboot the system.

---

8. Restrictions.

Unfortunately this is not a perfect world and there, as always, some restrictions that you should be aware of.

MILO is not meant to load operating systems other than Linux, although it can load images linked to run at the same place in memory as Linux (which is 0xFFFFFC0000310000). This is how the flash management utilities can be run.

The PALcode sources included in `miniboot/palcode/`*blah* are correct, however there are problems when they are built using the latest `gas`. They *do* build if you use the ancient a.out gas that's supplied in the Alpha Evaluation Board toolset (and that's how they were built). I'm trying to get someone to fix the new gas. Meanwhile, as a workaround, I have provided pre-built PALcode for the supported boards and David Mosberger-Tang has a fixed gas on his ftp site.

9. Problem Solving.

Here are some common problems that people have seen, together with the solutions.

Reading MS-DOS floppies from the Evaluation Board Debug Monitor.

Some of the older versions of the Evaluation Board Debug Monitor (pre-version 2.0) have a problem with DOS format flopies generated from Linux. Usually, the Debug Monitor can load the first few sectors all right, but then goes into an endless loop complaining about "bad sectors." Apparently, there is an incompatibility between the DOS file system as expected by the Debug Monitor and the Linux implementation of DOSFS. To make the long story short: if you run into this problem, try using DOS to write the floppy disk. For example, if loading the file `MILO.cab` doesn't work, use a DOS machine, insert the floppy and then do:

```
copy a:MILO.cab c:
copy c:MILO.cab a:
del c:MILO.cab
```

Then try booting from that floppy again. This normally solves the problem.

MILO displays a long sequence of `0>` and does not accept input.

This usually happens when MILO was built to use COM1 as a secondary console device. In such a case, MILO echo output to COM1 and accepts input from there also. This is great for debugging but not so great if you have a device other than a terminal connected. If this happens, disconnect the device or power it down until the Linux kernel has booted. Once Linux is up and running, everything will work as expected.

MILO complains that the kernel image has the wrong magic number

Older versions of MILO did not support the ELF object file format and so could not recognise an ELF image and this might be your problem. If this is reported, upgrade to the latest MILO that you can find. All 2.0.20 and beyond MILOs support ELF. On the other hand it could be that the image is indeed damaged. You should also note that MILO does not yet automatically distinquish between GZIP'd and non-GZIP'd images; you need to add the ".gz" suffix to the file name.

MILO prints "...turning on virtual addressing and jumping to the Linux Kernel" and nothing else happens

One obvious problem is that the kernel image is wrongly built or is built for another Alpha system altogether. Another is that the video board is a TGA (Zlxp) device and the kernel has been built for a VGA device (or vice versa). It is worth building the kernel to echo to COM1 and then connecting a terminal to that serial port or retrying the kernel that came with the Linux distribution that you installed.

MILO does not recognise the SCSI device

TOC

The standard MILO images include as many device drivers as are known to be stable for Alpha (as of now that includes the NCR 810, QLOGIC ISP, Buslogic and Adaptec 2940s and 3940 cards). If your card is not included, it may be that the driver is not stable enough on an Alpha system yet. Again, the latest MILO images are worth trying. You can tell which SCSI devices a MILO image has built into it by using the "show" command.

---

10. Acknowledgements.

I would like to thank:

- Eric Rasmussen and Eilleen Samberg the authors of the PALcode,
- Jim Paradis for the keyboard driver and the original MILO interface,
- Jay Estabrook for his help and bugfixes,
- David Mosberger-Tang for the freeware BIOS emulation and his support and encouragement,
- Last (and `not` least) Linus Torvalds for the timer code and his kernel.

There are a number of things that still need doing to MILO, if you want to add something yourself, then do let me know david.rusling@reo.mts.dec.com so that we do not duplicate our efforts.

Finally, a big thank you to Digital for producing such a wonderful chip (and paying me to do this).

Linux/MIPS HOWTO

Ralf Bächle, `ralf@gnu.org`

v0.1, 31 March 1999

This FAQ describes the MIPS port of the Linux operating system, common problems and their solutions, availability and more. It also tries to be a little helpful to other people who might read this FAQ in an attempt to find information that actually should be covered elsewhere.

1. What is Linux/MIPS?

2. What hardware does Linux/MIPS support?

- 2.1 Hardware platforms
- 2.2 Processor types

3. Linux distributions.

- 3.1 RedHat
- 3.2 Debian

4. Linux/MIPS net resources.

- 4.1 Anonymous FTP servers.
- 4.2 Anonymous CVS servers.
- 4.3 Web servers.
- 4.4 Mailing lists.

5. Installation of Linux/MIPS and common problems.

- 5.1 NFS booting fails.

- 5.2 Self compiled kernels crash when booting.
- 5.3 Booting the kernel on the Indy fails with PROM error messages
- 5.4 Where can I get the little endian firmware for my SNI?
- 5.5 ld dies with signal 6

6. Milo

- 6.1 Building Milo
- 6.2 Pandora

7. Loadable Modules

8. How do I setup a crosscompiler?

- 8.1 Diskspace requirements
- 8.2 Byte order
- 8.3 Configuration names
- 8.4 Installation of GNU Binutils.
- 8.5 Assert.h
- 8.6 First installation of egcs
- 8.7 float.h
- 8.8 Installing the kernel sources
- 8.9 Installing GNU libc
- 8.10 Building egcs again
- 8.11 Should I build the C++, Objective C or F77 compilers?
- 8.12 GDB

9. Related Literature

- 9.1 See MIPS Run
- 9.2 The MIPS Programmer's Handbook
- 9.3 Computer Architecture - A Quantitative Approach

10. Linux/MIPS news

1. What is Linux/MIPS?

Linux/MIPS is a port of the widespread UNIX clone Linux to the MIPS architecture. Linux/MIPS is running on a large number of technically very different systems ranging from little embedded systems and servers to large desktop machines and servers that, at least at the time when they were introduced into the market, were the best of their class.

Linux/MIPS advantages over other operating systems at this time are

- The entire Linux system consists only of Free Software.
- Excellent Price/Performance ratio.
- Availability of large amounts of software of which a large part again is Free Software.
- Binary compatibility across a growing number of platforms.
- Small footprint making Linux/MIPS suitable for many embedded systems.

In short, Linux has been designed and ships with Fahrvergnügen. However as usual your mileage may vary and you should examine Linux's suitability for your purpose which purpose this document tries to serve.

2. What hardware does Linux/MIPS support?

2.1 Hardware platforms

Many machines are available with a number of different CPU options of which not all are currently supported. Please check section Processor Types to make sure your CPU type is supported. This is a listing of machines that are running Linux/MIPS, systems to which Linux/MIPS could be ported or systems that people have an interest in running Linux/MIPS.

Acer PICA

The *Acer PICA* is derived from the *Mips Magnum 4000* design. It has a R4400PC CPU running at 133Mhz or optionally 150Mhz plus a 512kb (optionally 2mb) second level cache; the Magnum's G364 gfx card was replaced with a S3 968 based one. The system is supported with the exception of the X server.

Baget/MIPS series

The Baget series includes several boxes which have R3000 processors: Baget 23, Baget 63, and Baget 83. Baget 23 and 63 have BT23-201 or BT23-202 motherboards with R3500A (which is basically a R3000A chip) at 25 MHz and R3081E at 50 MHz respectively. The BT23-201 board has VME bus and VIC068, VAC068 chips as system controllers. The BT23-202 board has PCI as internal bus and VME as internal. Support for BT23-201 board has been done by Gleb Raiko (rajko@mech.math.msu.su) and Vladimir Roganov (vroganov@msiu.ru) with a bit help from Serguei Zimin (zimin@msiu.ru). Support for BT23-202 is under development along with Baget 23B which consists of 3 BT23-201 boards with shared VME bus.

Baget 83 is mentioned here for completeness only. It has only 2mb RAM and it's too small to run Linux. The Baget/MIPS code has been merged with the DECstation port; source for both is available at http://decstation.unix-ag.org/.

Cobalt Qube and Raq

The Cobalt Qube product series are low cost headless server systems based on a IDT R5230. Cobalt has developed its own Linux/MIPS variant to fit the special requirements of the Qube as well as possible. Basically the Qube kernel has been derived from Linux/MIPS 2.1.56, then backported to 2.0.30 for stability's sake, then optimized. Cobalt kernels are available from Cobalt's ftp site http://www.cobaltnet.com. The Cobalt Qube support has never been integrated into the official Linux/MIPS 2.1.x kernels.

Netpower 100

The *Netpower 100* is apparently an *Acer PICA* in disguise. It should therefore be supported but this is untested. If there is a problem then it is probably the machine detection.

TOC

Nintendo 64

The *Nintendo 64* is R4300 based game console with 4mb RAM. Its graphics chips were developed by Silicon Graphics for Nintendo. Right now this port has pipe dream status and will continue to be in that state until Nintendo decides to publish the necessary technical information. The question remains as to whether this is a good idea.

Silicon Graphics Indy

The Indy is currently the only (mostly) supported Silicon Graphics machine. The only supported graphics card is the Newport card aka “XL” graphics. The Indy is available with a large number of CPU options at various clock rates all of which are supported. There is currently no X server available for the Indy; Alan Cox (alan@lxorguk.ukuu.org.uk) is working on one.

Strange numbers of available memory

On bootup the kernel on the Indy will report available memory with a message like

```
Memory: 27976k/163372k available (1220k kernel code, 2324k data)
```

The large difference between the first pair of numbers is caused by a 128mb area in the Indy's memory address space which mirrors up to the first 128mb of memory. The difference between the two numbers will always be about 128mb and does not indicate a problem of any kind.

Indy PROM related problems

Several people have reported these problems with their machines after upgrading them typically from surplus parts. There are several PROM versions for the Indy available. Machines with old PROM versions which have been upgraded to newer CPU variants like a R4600SC or R5000SC module can crash during the self test with an error message like

```
Exception: <vector=Normal>
Status register: 0x30004803<CU1,CU0,IM7,IM4,IPL=???,MODE=KERNEL,EXL,IE>
Cause register: 0x4000<CE=0,IP7,EXC=INT>
Exception PC: 0xbfc0b598
Interrupt exception
CPU Parity Error Interrupt
Local I/O interrupt register 1: 0x80 <VR/GIO2>
CPU parity error register: 0x80b<B0,B1,B3,SYSAD_PAR>
CPU parity error: address: 0x1fc0b598
NESTED EXCEPTION #1 at EPC: 9fc3df00; first exception at PC: bfc0b598
```

In that case you'll have to upgrade your machine's PROMs to a newer version or go back to an older CPU version. Usually R4000SC or R4400SC modules should work in that case. Just to be clear, this is a problem which is unrelated to Linux. It's only mentioned here because several Linux users have asked about it.

TOC

ELF support in old PROM versions

Old PROM versions don't know about the ELF binary format which the Linux kernel uses, that is can't boot Linux directly. The preferable solution for this is of course a PROM upgrade. Alternatively you can use Sash of IRIX 5 or newer to boot the kernel. Sash knows how to load ELF binaries and doesn't care if it's an IRIX or Linux kernel. Simply type “Sash” to the prom monitor. You should get another shell prompt, this time from Sash. Now launch Linux as usual.

Sash can read EFS or XFS filesystems or read the kernel from bootp / tftp. That means if you intend to use Sash for booting the kernel from local disk you'll still have to have a minimal IRIX installation on your system.

Why is so much memory reserved on my Indy?

On bootup the `Memory: ...' message on an Indy says that there is 128mb of RAM reserved. That is ok; just like the PC architecture has a gap in its memory address space between 640kb and 1024kb, the Indy has a 128mb-sized area in its memory map where the first 128mb of its memory is mirrored. Linux knows about it and just ignores that memory, thus this message.

Silicon Graphics Challenge S

This machine is very similar to the Indy; the difference is that it doesn't have a keyboard and a GFX card but has an additional SCSI WD33C95 based adapter. This WD33C95 hostadapter is currently not supported.

Silicon Graphics Indigo

This machine is only being mentioned here because occasionally people have confused it with Indys. The Indigo series is a different architecture however and therefore yet unsupported. Andrew R. Baker (andrewb@uab.edu) announced a university project to port Linux to the Indigo on January 2, 1999.

Serial console on SGI machines

Make sure the kernel you're using includes the appropriate driver for a serial interface and serial console. Set the *console* ARC environment variable to either the value *d1* or *d2* for Indy and Challenge S depending on which serial interface you're going to use as console.

If you have the problem that all kernel messages appear on the serial console on bootup but everything is missing from the point when init starts, then you probably have the wrong setup for your /dev/console. You can find more information about this in the Linux kernel source documentation; it's in /usr/src/linux/Documentation/serial-console.txt if you have the kernel source installed.

Motorola 68k based machines like the Iris 3000

These are *very* old machines, probably more than ten years old by now. As these machines are not based on MIPS processors this document is the wrong place to search for information. However, in order to make things easy, these machines are currently not supported.

TOC

SGI VisPC

This is actually an x86 based system, therefore not covered by this FAQ. But to make your search for answers simple, here it is. Ken Klingman (kck@mailbox.esd.sgi.com) posted on January 17, 1999 to SGI's Linux mailing list:

```
We are working on it.  We're actually close to getting
the base level system support into the 2.2 release.
Software-only X and OpenGL should follow relatively
shortly, but hardware-accelerated OpenGL is still
some time off.  See www.precisioninsight.com for
news about hardware-accelerated OpenGL.
```

For more information see the Documentation/ of Linux kernel versions from 2.2.0 and newer. There is additional information available on the web on http://www.linux.sgi.com/intel/. Note that the SGI/MIPS and SGI/Intel people are working independently of each other, therefore the sources in the anonymous CVS on linus.linux.sgi.com may or may not work for Intel machines; we don't test this.

Other Silicon Graphics machines

At this time no other Silicon Graphics machine is supported. This also applies to the *very* old Motorola 68k based systems.

Sony Playstation

The Sony Playstation is based on an R3000 derivative and uses a set of graphics chips developed by Sony themselves. While the machine in theory would be capable of running Linux, a port is difficult, since Sony so far hasn't provided the necessary technical information. This still leaves the question of whether the port would be worthwhile. So in short, nothing has happend yet even though many people have shown their interest in trying Linux on a Playstation so far.

SNI RM200C

In contrast to the RM200 (see below) this machine has EISA and PCI slots. The RM200 is supported with the exception of the availability of the onboard NCR53c810A SCSI controller.

SNI RM200

If your machine has both EISA and PCI slots, then it is an RM200C; please see above. Due to the slight architectural differences of the RM200 and the RM200C this machine isn't currently supported in the official sources. Michael Engel (engel@numerik.math.uni-siegen.de) has managed to get his RM200 working partially but the patches haven't yet been included in the official Linux/MIPS sources.

SNI RM300C

The RM300 is technically very similar to the RM200C. It should be supported by the current Linux kernel, but we haven't yet received any reports.

SNI RM400

The RM400 isn't supported.

Algorithmics P4032

The Algorithmics P4032 port is at the time of this writing still running Linux 2.1.36.

Algorithmics P5064

The P5064 is basically an R5000-based 64bit variant of the P4032. It's not yet supported but a Linux port will be quite easy.

DECstation series

During the late 80's and the early 90's Digital (now Compaq) built MIPS based Workstations named DECstation resp. DECsystem. Other x86 and Alpha based machines were sold under the name DECstation, but these are obviously not subject of this FAQ. Support for DECstations is still under development, started by Paul M. Antoine. These days most of the work is done by Harald Koerfgen (Harald.Koerfgen@home.ivm.de) and others. On the Internet, DECstation-related information can be found at http://decstation.unix-ag.org/.

The DECstation family ranges from the DECstation 2100 with an R2000/R2010 chipset at 12 Mhz to the DECstation 5000/260 with a 60 MHz R4400SC.

The following DECstation models are actively supported:

- 2100, codename PMAX
- 5000/xx (Personal DECstation), codename MAXine
- 5000/1xx, codename 3MIN
- 5000/2x0, codename 3MAX+
- 5900/2x0 (identical to the 3MAX+).

These DECstation models are orphaned because nobody is working on them, but support for these should be relatively easy to achieve.

- 3100, identical to the 2100 except the R2000A/R2010A @ 16 MHz
- 5100, codename MIPSMATE, almost identical to the 2100 but with an R3000/R3010 chipset.
- 5000/200, codename 3MAX

The other members of the DECstation family, besides the x86 based ones, should be considered as VAXen with the CPU replaced by a MIPS CPU. There is absolutely no information available about these machines and support for them is unlikely to happen ever unless the VAXLinux port comes to a new life. These are:

- 5400, codename MIPSFAIR

- 5500, codename MIPSFAIR2
- 5800, codename ISIS

The R2000/R3000 support in the Linux/MIPS kernel is a merge of the DECstation and Baget/MIPS code and isn't yet integrated into the official Linux/MIPS source tree.

Mips Magnum 4000 / Olivetti M700-10

These two machines are almost completely identical. Back during the ACE initiative Olivetti licensed the Jazz design and marketed the machine with Windows NT as OS. MIPS Computer Systems, Inc. itself bought the Jazz design and marketed it as the MIPS Magnum 4000 series of machines. Magnum 4000 systems were marketed with Windows NT and RISC/os as operating systems.

The firmware on the machine depended on the operating system which was installed. Linux/MIPS supports only the little endian firmware on these two types of machines. Since the M700-10 was only marketed as an NT machine all M700-10 machines have this firmware installed. The MIPS Magnum case is somewhat more complex. If your machine has been configured big endian for RISC/os then you need to reload the little endian firmware. This firmware was originally included on a floppy with the delivery of every Magnum. If you don't have the floppy anymore you can download it via anonymous ftp from ftp://ftp.fnet.fr.

It is possible to reconfigure the M700 for headless operation by setting the firmware environment variables ConsoleIn and ConsoleOut to multi()serial(0)term(). Also try the command *listdev* which will show the available ARC devices.

In some cases, like where the G364 graphics card is missing but the console is still configured to use normal graphics it will be necessary to set the configuration jumper JP2 on the board. After the next reset the machine will reboot with the console on COM2.

MIPS Magnum 4000SC

The Mips Magnum 4000SC is the same as a Magnum 4000 (see above) with the exception that it uses an R4000SC CPU.

VaxStation

As the name already implies this machine is a member of Digital Equipment's VAX family. It's mentioned here because people often confuse it with Digital's MIPS based DECstation family due to the similar type numbers. These two families of architectures share little technical similarities. Unfortunately the VaxStation, like the entire VAX family, is currently unsupported.

2.2 Processor types

R2000, R3000 family

The R2000 is the original MIPS processor. It's a 32 bit processor which was clocked at 8MHz back in '85 when the first MIPS processors came to the market. Later versions were clocked faster: for instance, the R3000 is a 100% compatible redesign of the R2000, just clocked faster. Because of their high compatibility, where this document mentions the R3000, in most cases the same facts also apply to the R2000.

The R3000A is basically an R2000 plus an R3010 FPU and 64k cache running at up to 40Mhz and integrated into the same chip. Support for the R3000 processor is currently in the works by various people. Harald Koerfgen (Harald.Koerfgen@home.ivm.de) and Gleb O. Raiko (raiko@niisi.msk.ru) have both independently worked on patches which haven't yet been integrated into the official Linux/MIPS sources.

R6000

Sometimes people confuse the R6000, a MIPS processor, with RS6000, a series of workstations made by IBM. So if you're reading this in hope of finding out more about Linux on IBM machines you're reading the wrong document.

The R6000 is currently not supported. It is a 32-bit MIPS ISA 2 processor and a pretty interesting and weird piece of silicon. It was developed and produced by a company named *BIT Technology*. Later NEC took over the semiconductor production. It was built in ECL technology, the same technology that was and still is being used to build extremely fast chips like those used in some Cray computers. The processor had its TLB implemented as part of the last couple of lines of the external primary cache, a technology called *TLB slice*. That means its MMU is substantially different from those of the R3000 or R4000 series, which is also one of the reasons why the processor isn't supported.

R4000 and R5000 family

Linux supports many of the members of the R4000 family. Currently these are R4000PC, R4400PC, R4300, R4600, R4700, R5000, R5230, R5260. Many others are probably working as well.

Not supported are R4000MC and R4400MC CPUs (that is multiprocessor systems) as well as R5000 systems with a CPU controlled second level cache. This means where the cache is controlled by the R5000 itself in contrast to some external external cache controller. The difference is important because, unlike other systems, especially PCs, on MIPS the cache is architecturally visible and needs to be controlled by software.

Special credit goes to Ulf Carlsson (grim@zigzegv.ml.org) who provided the CPU module for debugging the R4000SC / R4400SC support.

R8000

The R8000 is currently unsupported partly because this processor is relatively rare and has only been used in a few SGI machines, partly because the Linux/MIPS developers don't have such a machine.

The R8000 is a pretty interesting piece of silicon. Unlike the other members of the MIPS family it is a set of seven chips. Its cache and TLB architecture is pretty different from the other members of the MIPS family. It was born as a hack to get the floating point crown back to Silicon Graphics before the R10000 is finished.

R10000

The R10000 is currently unsupported because the Linux/MIPS developers don't have an R10000 machine.

3. Linux distributions.

3.1 RedHat

For MIPSeb, there's Rough Cuts Linux, previously known as Hard Hat Linux, which is most of Red Hat Linux 5.1 ported for MIPSeb. You can get this at ftp://ftp.linux.sgi.com/pub/hardhat.

It is also bundled along with M68k, UltraSparc and PowerPC in a package called "Rough Cuts" pressed by Red Hat, and available wherever Red Hat products are sold. This is a very convenient way to get it without having to download 280MB. You can order Rough Cuts directly from Red Hat at http://www.redhat.com/product.phtml/RC1000.

As well, there's a distribution based on Red Hat 5.2 that's targetting the Cobalt Qubes; those binaries will work perfectly on other MIPSel architectures available at ftp://intel.cleveland.lug.net/pub/Mipsel.

3.2 Debian

A Debian port is underway. Current efforts are being bootstrapped using SGI/Linux as a base, and dpkg compiles natively with few changes. In addition to the SGI version, some interest has been shown in little endian platforms. Keep an eye on the Debian-MIPS Port page, http://www.debian.org/ports/mips/ for developments.

4. Linux/MIPS net resources.

4.1 Anonymous FTP servers.

The two primary anonymous FTP servers for Linux/MIPS are

ftp.linux.sgi.com

This server should satisfy almost all your Linux/MIPS related ftp desires. Really.

ftp.fnet.fr

This server is currently pretty outdated; it's included here mostly for completeness and for people with interest in prehistoric software.

On all these ftp servers there is a list of mirror sites you may want to use for faster access.

4.2 Anonymous CVS servers.

For those who always want to stay on the bleeding edge and want to avoid having to download patch files or full tarballs we also have an anonymous CVS server. Using CVS you can checkout the Linux/MIPS source tree with the following commands:

```
cvs -d :pserver:cvs@linus.linux.sgi.com:/cvs login
(Only needed the first time you use anonymous CVS, the password is "cvs")
cvs -d :pserver:cvs@linus.linux.sgi.com:/cvs co <repository>
```

where you insert linux, libc, or gdb for <repository>.

The other important CVS archive of the Linux community is vger.rutgers.edu where a lot of code is being collected before being sent to Linus for distribution. Although vger itself no longer offers anonymous access, there are mirror sites which do provide anonymous access. For details how to access them see http://cvs.on.openprojects.net/. The modules which are of interest are “linux”, “modutils”, “pciutils”, “netutils”.

4.3 Web servers.

The two primary anonymous web servers for Linux/MIPS are

www.linux.sgi.com

This server covers most of Linux/MIPS; it's somewhat SGI centric but since Linux/MIPS tries to be the same on every platform most of its information is of interest to all users.

lena.fnet.fr

This server is currently pretty outdated; it's included here mostly for completeness.

All these servers have mirrors scattered all over the world; you may want to use one for best performance.

4.4 Mailing lists.

TOC

There are three Linux/MIPS oriented mailing lists:

linux-mips@fnet.fr

This mailing list is used for most non-SGI related communication of all kinds. Subscription is handled by a human; send your subscription requests to linux-mips-mips@fnet.fr. You can unsubscribe from this mailing list by sending *unsubscribe* <*your-email-address*> to the same address.

linux@engr.sgi.com

This mailing list currently has the most traffic. It's somewhat SGI-centric but is nevertheless of interest especially to developers as a good number of SGI engineers are subscribed to this list. Subscription to this list is handled via Majordomo (majordomo@engr.sgi.com); just send an email with the words *subscribe linux-mips*. In order to unsubscribe send *unsubscribe linux-mips*. Note that you have to be subscribed if you want to post; the growth of spam forced us into that policy.

linux-mips@vger.rutgers.edu

This mailing list has only very low traffic as most people tend to use one of the above mailing lists. Subscription is handled via Majordomo (majordomo@vger.rutgers.edu); just send an email with the words *subscribe linux-mips*. In order to unsubscribe send *unsubscribe linux-mips*.

---

5. Installation of Linux/MIPS and common problems.

5.1 NFS booting fails.

Usually the reason for this is that people have unpacked the tar archive under IRIX, not Linux. Since the representation of device files over NFS is not standardized between various Unices, this fails. The symptom is that the system dies with the error message ``Warning: unable to open an initial console.'' right after mounting the NFS filesystem.

For now the workaround is to use a Linux system (doesn't need to be MIPS) to unpack the installation archive onto the NFS server. The NFS server itself may be any type of UNIX.

5.2 Self compiled kernels crash when booting.

When I build my own kernel, it crashes. On an Indy the crash message looks like the following; the same problem hits other machines as well but may look completely different.

```
Exception: <vector=UTLB Miss>
Status register: 0x300004803<CU1,CU0,IM4,IPL=???,MODE=KERNEL,EXL,IE>
Cause register: 0x8008<CE=0,IP8,EXC=RMISS>
Exception PC: 0x881385cc, Exception RA: 0x88002614
exception, bad address: 0x47c4
Local I/O interrupt register 1: 0x80 <VR/GIO2>
Saved user regs in hex (&gpda 0xa8740e48, &_regs 0xa8741048):
  arg: 7 8bfff938 8bfffc4d 880025dc
  tmp: 8818c14c 8818c14c 10 881510c4 14 8bfad9e0 0 48
  sve: 8bfdf3e8 8bfffc40 8bfb2720 8bfff938 a8747420 9fc56394 0 9fc56394
  t8 48 t9 8bfffee66 at 1 v0 0 v1 8bfff890 k1 bad11bad
  gp 881dfd90 fp 9fc4be88 sp 8bfff8b8 ra 88002614

PANIC: Unexpected exception
```

This problem is caused by a still unfixed bug in Binutils newer than version 2.7. As a workaround, change the following line in arch/mips/Makefile from:

```
LINKFLAGS       = -static -N
```

to:

```
LINKFLAGS       = -static
```

5.3 Booting the kernel on the Indy fails with PROM error messages

```
>> boot bootp()/vmlinux
73264+592+11520+331680+27848d+3628+5792 entry: 0x8df9a960
Setting $netaddres to 192.168.1.5 (from server deadmoon)
Obtaining /vmlinux from server deadmoon

Cannot load bootp()/vmlinux
Illegal f_magic number 0x7f45, expected MIPSELMAGIC or MIPSEBMAGIC.
```

This problem only happens for Indys with very old PROM versions which cannot handle the ELF binary format which Linux uses. A solution for this problem is in the works.

5.4 Where can I get the little endian firmware for my SNI?

SNI's system can be operated in both big and little endian modes. At this time Linux/MIPS only supports the little endian firmware. This is somewhat unlucky since SNI hasn't shipped that firmware for quite some time, since they dropped NT.

When running in big endian mode the firmware looks similar to an SGI Indy which is already supported, therefore fixing the SNI support will be relativly easy. Interested hackers should contact Ralf Bächle (ralf@gnu.org).

5.5 ld dies with signal 6

```
collect2: ld terminated with signal 6 [Aborted]
```

This is a known bug in older binutils versions. You will have to upgrade to binutils 2.8.1 plus very current patches.

6. Milo

Milo is the boot loader used to boot the little endian MIPS systems with ARC firmware, currently the Jazz family and the SNI RM 200. While Milo uses the same name and has a similar purpose to the Alpha version of Milo, these two Milos have nothing else in common. They were developed by different people, don't share any code, and work on different hardware platforms. The fact that both have the same name is just a kind of historic ``accident''.

Plans are to remove the need for Milo in the near future.

6.1 Building Milo

The building procedure of Milo is described in detail in the README files in the Milo package. Since Milo has some dependencies to kernel header files which have changed over time Milo often cannot be built easily; however the Milo distribution includes binaries for both Milo and Pandora.

6.2 Pandora

Pandora is a simple debugger. It has been primarily developed in order to analyze undocumented systems. Pandora includes a dissassembler, memory dump functions and more. If you only want to use Linux there is no need to install Pandora. It's small though.

7. Loadable Modules

Using modules on Linux/MIPS is quite easy; it should work as expected for people who have used it on other Linux systems. If you want to run a module-based system then you should have at least kernel version 980919 and modutils newer than version 2.1.121 installed. Older versions won't work.

8. How do I setup a crosscompiler?

First of all go and download the following source packages:

- binutils-2.8.1.tar.gz
- egcs-1.0.2.tar.gz
- glibc-2.0.6.tar.gz
- glibc-crypt-2.0.6.tar.gz
- glibc-localedata-2.0.6.tar.gz
- glibc-linuxthreads-2.0.6.tar.gz

These are the currently recommended versions. Older versions may or may not be working. If you're trying to use older versions please don't send bug reports; we don't care. When installing please install things in the order binutils, egcs, then glibc. Unless you have older versions already installed, changing the order *will* fail. The installation description below mentions a number of patches which you can get from the respective SRPM packages on ftp.linux.sgi.com. However since these SRPM packages are intended to be compiled natively it's not possible to just rebuild them.

8.1 Diskspace requirements

For the installation you'll have to choose a directory for installation. I'll refer to that directory below with <prefix>. To avoid a certain problem it's best to use the same value for <prefix> as your native gcc. For example if your gcc is installed in /usr/bin/gcc then choose /usr for <prefix>. You must use the same <prefix> value for all the packages that you're going to install.

During compilation you'll need about 31mb diskspace for binutils; for installation you'll need 7mb diskspace for on <prefix>'s partition. Building egcs requires 71mb and installation 14mb. GNU libc requires 149mb diskspace during compilation and 33mb for installation. Note these numbers are just a guideline and may differ significantly for different processor and operating system architectures.

8.2 Byte order

One of the special features of the MIPS architecture is that all processors except the R8000 can be configured to run either in big or in little endian mode. Byte order means the way the processor stores multibyte numbers in memory. Big endian machines store the byte with the highest value digits at the lowest address while little endian machines store it at the highest address. Think of it as writing multi-digit numbers from left to right or vice versa.

In order to setup your crosscompiler correctly you have to know the byte order of the crosscompiler target. If you don't already know, check the section Hardware Platforms for your machine's byteorder.

TOC

8.3 Configuration names

Many of the packages based on autoconf support many different architectures and operating systems. In order to differentiate between these many configurations, names are constructed with <cpu>-<company>-<os> or even <cpu>-<company>-<kernel>-<os>. Expressed this way the configuration names of Linux/MIPS are mips-unknown-linux-gnu for big endian targets or mipsel-unknown-linux-gnu for little endian targets. These names are a bit long and are allowed to be abbreviated to mips-linux or mipsel-linux. You *must* use the same configuration name for all packages that comprise your crosscompilation environment. Also, while other names like mips-sni-linux or mipsel-sni-linux are legal configuration names, use mips-linux or mipsel-linux instead; these are the configuration names known to other packages like the Linux kernel sources and they'd otherwise have to be changed for crosscompilation.

I'll refer to the target configuration name below with <target>.

8.4 Installation of GNU Binutils.

This is the first and simplest part - at least as long as you're trying to install on any halfway-sane UNIX flavour. Just cd into a directory with enough free space and do the following:

```
gzip -cd binutils-<version>.tar.gz | tar xf -
cd binutils-<version>
patch -p1 < ../binutils-<version>-mips.patch
./configure --prefix=<prefix> --target=<target>
make CFLAGS=-O2
make install
```

This usually works very easily. On certain machines using GCC 2.7.x as compiler is known to dump core. This is a known bug in GCC and can be fixed by upgrading to GCC 2.8.1 or egcs.

8.5 Assert.h

Some people have an old assert.h headerfile installed, probably a leftover from an old crosscompiler installation. This file may cause autoconf scripts to fail silently; it was never necessary and was only installed because of a bug in older GCC versions. Check to see if the file <prefix>/<target>/include/assert.h exists in your installation. If so, just delete the it: it should never have been installed.

8.6 First installation of egcs

Now the not-so-funny part begins: there is a so-called bootstrap problem. In our case that means the installation process of egcs needs an already- installed glibc, but we cannot compile glibc because we don't have a working crosscompiler yet. Luckily you'll only have to go through this once when you install a crosscompiler for the first time. Later when you already have glibc installed things will be much smoother. So now do:

```
gzip -cd egcs-<version>.tar.gz | tar xf -
cd egcs-<version>
for i in egcs-1.0.2-libio.patch egcs-1.0.2-hjl.patch \
```

```
        egcs-1.0.2-rth1.patch egcs-1.0.2-rth2.patch egcs-1.0.2-rth3.patch
        egcs-1.0.2-rth4.patch egcs-1.0.2-hjl2.patch egcs-1.0.2-jim.patch
        egcs-1.0.2-haifa.patch egcs-1.0.1-objcbackend.patch \
        egcs-1.0.2-mips.patch; do patch -p1 -d < ../$i; done
./configure --prefix=<prefix> --with-newlib --target=<target>
cd gcc
make LANGUAGES="c"
```

Note that we deliberately don't build gcov, protoize, unprotoize and the libraries. Gcov doesn't make sense in a crosscompiler environment and protoize and unprotoize might even overwrite your native programs - this is a bug in the gcc makefiles. Finally we cannot build the libraries because we don't have glibc installed yet. If everything went successfully, install with:

```
make LANGUAGES="c" install
```

8.7 float.h

Another bootstrap problem is that building GCC requires running programs on the machine for which GCC will generate code, but since a crosscompiler is running on a different type of machine this cannot work. When building GCC this happens for the header file float.h. Luckily there is a simple solution: download the header file from one of the Linux/MIPS ftp servers or rip it from one of the native Linux/MIPS binary packages. Later when recompiling or upgrading egcs usually the already-installed float.h file will do because float.h changes rarely. Install it with:

```
cp float.h <prefix>/<target/<version>/include/float.h
```

where <version> is the internal version number of the egcs version you're using. For egcs 1.0.2 for example you would use egcs-2.90.27 for <version>. If not sure - ls is your friend.

8.8 Installing the kernel sources

Installing the kernel sources is simple. Just place them into some directory of your choice and configure them such that some files which are generated by the procedure will be installed. This works the same as you're used to when configuring the kernel sources for native compilation. The only problem you may run into is that you may need to install some required GNU programs like bash or have to override the manufacturer-provided versions of programs by placing the GNU versions earlier in the PATH variable. When configuring you should answer the question ``Are you using a crosscompiler", that is the option CONFIG_CROSSCOMPILE, with ``yes". When you're done with configuring type make clean; make depend; make. The last make command will generate the header file <linux/version.h> which compiling some programs depends on. This file is generated right at the beginning of the make command, so if you're not interested in actually building a kernel you may interrupt the compilation after this file has been built. It may be a good idea, however, to compile the kernel as a test for your newly-built crosscompiler.

If you only want the crosscompiler for building the kernel, you're done. Crosscompiling libc is only required to be able to compile user applications.

8.9 Installing GNU libc

TOC

Do:

```
gzip -cd glibc-2.0.6.tar.gz | tar xf -
cd glibc-2.0.6
gzip -cd glibc-crypt-2.0.6.tar.gz | tar xf -
gzip -cd glibc-localedata-2.0.6.tar.gz | tar xf -
gzip -cd glibc-linuxthreads-2.0.6.tar.gz | tar xf -
patch -p1 < ../glibc-2.0.6-mips.patch
mkdir build
cd build
CC=<target>-gcc BUILD_CC=gcc AR=<target>-ar RANLIB=<target>-ranlib \
      ../configure --prefix=/usr --host=<target> \
      --enable-add-ons=crypt,linuxthreads,localedata --enable-profile
make
```

You now have a compiled GNU libc which still needs to be installed. Do *not* just type make install. That would overwrite your host system's files with Linux/MIPS-specific files with disastrous effects. Instead install GNU libc into some other arbitrary directory <somedir> from which we'll move the parts we need for crosscompilation into the actual target directory:

```
make install_root=<somedir> install
```

Now cd into <somedir> and finally install GNU libc manually:

```
cd usr/include
find . -print | cpio -pumd <prefix>/<target>/include
cd ../../lib
find . -print | cpio -pumd <prefix>/<target>/lib
cd ../usr/lib
find . -print | cpio -pumd <prefix>/<target>/lib
```

GNU libc also contains extensive online documentation. Your systems might already have a version of this documentation installed, so if you don't want to install the info pages, which will save you a less than a megabyte, or already have them installed, skip the next step:

```
cd ../info
gzip -9 *.info*
find . -name \*.info\* -print | cpio -pumd <prefix>/info
```

If you're not bootstrapping your installation is now finished.

8.10 Building egcs again

The first attempt of building egcs was stopped by lack of a GNU libc. Since we now have libc installed we can rebuild egcs but this time as complete as a crosscompiler installation can be:

```
gzip -cd egcs-<version>.tar.gz | tar xf -
cd egcs-<version>
for i in egcs-1.0.2-libio.patch egcs-1.0.2-hjl.patch \
      egcs-1.0.2-rth1.patch egcs-1.0.2-rth2.patch egcs-1.0.2-rth3.patch \
      egcs-1.0.2-rth4.patch egcs-1.0.2-hjl2.patch egcs-1.0.2-jim.patch \
      egcs-1.0.2-haifa.patch egcs-1.0.1-objcbackend.patch \
      egcs-1.0.2-mips.patch; do patch -p1 < ../$i; done
./configure --prefix=<prefix> --target=<target>
make LANGUAGES="c c++ objective-c f77"
```

As you can see the procedure is the same as the first time with the exception that we dropped the --with-newlib option. This option was necessary to avoid the libgcc build breaking due to the lack of libc. Now install with:

```
make LANGUAGES="c c++ objective-c f77" install
```

You're almost finished. All you have left to do now is to reinstall float.h, which has been overwritten by the last make install command. You'll have to do this every time you reinstall egcs as a crosscompiler. If you think you don't need the Objective C or F77 compilers you can omit them from above commands; each will save you about 3mb. Do not build gcov, protoize or unprotoize.

8.11 Should I build the C++, Objective C or F77 compilers?

The answer to this question largely depends on your use of your crosscompiler environment. If you only intend to rebuild the Linux kernel then you have no need for the full blown setup and can safely omit the Objective C and F77 compilers. You must, however, build the C++ compiler, because building the libraries included with the egcs distribution requires C++.

8.12 GDB

Building GDB as crossdebugger is only of interest to kernel developers; for them GDB may be a life saver. Such a remote debugging setup always consists of two parts: the remote debugger GDB running on one machine and the target machine running the Linux/MIPS kernel being debugged. The machines are typically interconnected with a serial line. The target machine's kernel needs to be equipped with a ``debugging stub'' which communicates with the GDB host machine using the remote serial protocol.

Depending on the target's architecture you may have to implement the debugging stub yourself. In general you'll only have to write very simple routines for serial. The task is further simplified by the fact that most machines are using similar serial hardware typically based on the 8250, 16450 or derivatives.

---

9. Related Literature

9.1 See MIPS Run

author Dominic Sweetman, published Morgan Kaufmann, ISBN 1-55860-410-3.

This is intended as a pretty comprehensive guide to programming MIPS, wherever it's different from programming any other 32-bit CPU. It's the first time anyone tried to write a readable and comprehensive explanation and account of the wide range of MIPS CPUs available, and should be very helpful for anyone programming MIPS who isn't insulated by someone else's operating system. And the author is a free-unix enthusiast who subscribes to the Linux/MIPS mailing list!

John Hennessey, father of the MIPS architecture, was kind enough to write in the foreword: “ ... this book is the best combination of completeness and readability of any book on the MIPS architecture ...”

It includes some context about RISC CPUs, a description of the architecture and instruction set including the "co-processor 0" instructions used for CPU control; sections on caches, exceptions, memory management and floating point. There's a detailed assembly language guide, some stuff about porting, and some fairly heavy-duty software examples.

Available from:

- http://www.algor.co.uk/algor/info/seemipsrun.html (europe)
- http://www.mkp.com/books_catalog/1-55860-410-3.asp (US)

and from good bookshops anywhere. It's 512 pages and costs around $50 in the US, £39.95 in the UK.

I'd be inclined to list two other books too, both from Morgan Kaufmann and available from www.mkp.com or any good bookshop:

9.2 The MIPS Programmer's Handbook

authors Farquhar and Bunce, published by Morgan Kaufmann, ISBN 1-55860-297-6.

A readable introduction to the practice of programming MIPS at the low level, by the author of PMON. Strengths: lots of examples; weakness: leaves out some big pieces of the architecture (such as memory management, floating point and advanced caches) because they didn't feature in the LSI “embedded” products this book was meant to partner.

9.3 Computer Architecture - A Quantitative Approach

TOC

authors Hennessy & Patterson, published Morgan Kaufmann, ISBN 1-58860-329-8.

The bible of modern computer architecture and a must-read if you want to understand what makes programs run slow or fast. Is it about MIPS? Well, it's mostly about something very *like* MIPS... Its sole defect is its size and weight - but unlike most big books it's worth every page.

---

10. Linux/MIPS news

Some of this chapter is pretty historic ...

04-Dec-98

Ariel Faigon announces that SGI has joined Linux International.

13-Oct-98

Ralf Bächle fixes the support for R4000SC / R4400SC CPUs.

12-Oct-98

Vladimir Roganov reports that his R3000 system is now stable enough to compile GDB.

03-Oct-98

Harald Körfgen reports that his DECstation 5000/133 is now running single user. Congratulations!

29-Sep-98

Ralf starts rewriting this FAQ to fit with reality.

10-Jun-98

ftp.linux.sgi.com now offers anonymous CVS access.

01-Feb-98

First commercial Linux/MIPS based product accounced.

26-Jan-98

One more timewarp in this list because the maintainer is lazy^H^H^H^H busy coding. The driver for the NCR53c8xx has been modified and has been successfully tested with several machines, most notably the SNI RM200. Even better, the initial version seems to be reliable.

Already some time ago Thomas Bogendörfer implemented the necessary changes to the NCR53C9x driver aka ESP driver, so there is now SCSI support for the builtin hostadapters in the Mips Magnum 4000, Olivetti M700-10 and Acer PICA.

28-Nov-97

First public release of X11 client binaries.

30-Aug-97

Duh, time warp in this page once again. A lot has happend in the meantime and the maintainer of this pages is a lazy person that rather prefers to code and hack than write docs...

SGI now has its own Linux/MIPS server reachable as http://www.linux.sgi.com, with lots of SGI specific information and many links. The server is also reachable under ftp.linux.sgi.com. In addition to binaries, sources and docs specific to Silicon Graphic machines this server also has all

TOC

the other Linux/MIPS stuff in stock. Only available on this server is the developers' cvs archiv download. Sorry, no anonymous CVS yet.

Silicon Graphics has supported some of the Linux key developers' work on Linux/MIPS with hardware. As a result the work is now advancing more quickly and Ralf is no longer the lone workhorse ...

Already available for some time the Indy port is now in the standard kernel source tree.

Long missing, but finally there: Thomas Bogendoerfer contributed patches to the NCR53C9x driver for Mips Magnum 4000, Olivetti M700 and Acer PICA.

Many more packages of a RedHat port to MIPS are now available for ftp download. Installing is still more a thing for experts ... but we're working on it!

Eeecmacs lovers will be pleased to hear that this FAQ has been edited by Emacs running on a Linux/MIPS machine.

6-May-97

David Monro releases version 1.01 of bfsd. bfsd is a daemon that can be used to boot the machines built by Mips Computersystems, Inc. over a network.

10-Jun-96

Release of Linux/MIPS kernel 2.0.4. This release features a partially rewritten signal handler that should match POSIX.1.

3-Jun-96

First release of shared libraries for Linux/MIPS based on GNU libc snapshot 960619.

Release of Linux/MIPS kernel 2.0.1.

25-May-96

David S. Miller starts working on SGI support at Silicon Graphics.

20-May-96

Release 1.3.98 of the kernel adds support for the SNI RM200 PCI.

27-Mar-96

Linux/MIPS works as NFS server.

The IDE CD driver now also supports Linux/MIPS.

24-Mar-96

Added reference to literature available online form SGI to the FAQ.

23-Mar-96

New chapter in the FAQ about the ARC standard.

27-Jan-96

TOC

Release of Milo 0.26 and a kernel patch to use it. This release passes parameters to the kernel completly different way that makes porting Linux/MIPS to another architecture a lot easier.

24-Jan-96

Release of crosscompiler binaries based on the FSF's Binutils version 2.6. This release brings lots of new features and many bugfixes.

21-Jan-96

Warner Losh started working on a port of Linux/MIPS to Deskstation rPC44.

20-Jan-96

Linux/MIPS kernel updated to version 1.3.58.

Patch gcc-2.7.2-1.diffs.gz has been released.

Patch binutils-2.6-1.diffs.gz has been released. This patch contains lots of bugfixes. The Linux kernel Makefiles will automatically detect whether Binutils 2.6 or an older version is installed and use the new features resulting in a much smaller kernel executable which is especially useful for bootdisks.

15-Jan-96

Release of a complete root and /usr filesystem that can be NFS mounted to use a Linux/MIPS system as a diskless client. A native development kit based on GCC 2.7.2, Binutils 2.6 and GNU libc snapshot 951218 is included as well as many of the standard utilities.

25-Dec-95

Linux/MIPS boots off an NFS filesystem as a diskless client. This also means that the rest of Linux/MIPS networking is operational now.

7-Jan-95

Soft-N-Hard GMBH and SNI sign a contract. SNI will loan an RM200 to Soft-N-Hard for porting Linux/MIPS to it.

22-Sep-95

The Linux/MIPS FTP archive and mailing list have been moved to fnet.fr. (There is much more news I currently have no time to document)

18-Jul-95

New crossdevelopment tools released. GCC-2.6.3-2 and Binutils-2.5.2-2 for Linux/i386 need kernels with ELF support and libc-5.0.9 installed. The new crossdev tools are required for Linux/MIPS kernels above 1.2.9. A.out versions of the crossdev tools will follow soon.

14-Jul-95

We have a working shell!

12-Jul-95

Patches 2.6.3-2 for Linux/MIPS GCC released. This compiler better complies with the MIPS standard of symbol names.

TOC

10-Jul-95

Linux/MIPS kernel 1.2.9 released.

9-Jun-95

Milo 0.24 released. This version features improved machine type detection and many cleanups and bugfixes.

24-May-95

Linux/MIPS kernel 1.2.8 released. This version features many bugfixes and has the Magnum 4000 specific changes from Linux-1.2.7 integrated.

Milo 0.23 released. This version features built-in support for Olivetti M700 machines. Milo is now split into two binaries: A simple bootloader and a standalone debugger/monitor with boot capability.

23-5-95

Linux/MIPS kernel 1.2.7 on Olivetti M700 mounts root file system.

22-May-95

Linux/MIPS kernel 1.2.7 on Mips Magnum 4000 mounts root file system.

Added NEC RiscStation and RiscServer to target list.

Milo 0.22 successfully tested on NEC RiscStation and RiscServer.

18-May-95

Linux/MIPS kernel 1.2.7 released. This release features initial Magnum 4000 support and tons of bugfixes.

12-May-95

Milo 0.22 released. This version contains some cleanups and several bugfixes.

5-May-95

The Linux/MIPS archive is now also available from ftp://ftp.mcc.ac.uk/pub/linux/MIPS.

3-May-95

Milo 0.21 released. This version features more built-in debugger/monitor commands and contains some important bug fixes.

30-Apr-95

Milo 0.20 released. This version features a built-in debugger/monitor and a lot of new library functions.

Port to Olivetti M700 started.

26-Apr-95

Linux/MIPS kernel 1.2.6 released.

13-Apr-95

Milo 0.19 released. This version includes some minor fixes plus initial support for kernels in ELF format.

13-Apr-95

Milo 0.18b released. This version includes support for Mips Magnum 4000. Port to Mips Magnum 4000 started.

27-Mar-95

Linux/MIPS kernel 1.2.2 released. Kernel now mounts its root file system.

22-Mar-95

Milo 0.18 released. This version includes support for Deskstation rPC44 systems.

Port to DeskStation rPC44 started.

The Linux Electronic Mail Administrator HOWTO

Guylhem Aznar <guylhem at oeil.qc.ca>

v3.1, July 1999

*This document describes the setup, care and feeding of Electronic Mail (e-mail) under Linux. It is primarily intended for administrators, rather than users. (See the Mail-User's-HOWTO for information on user issues and user agents.) You need to read this if you plan to communicate locally or to remote sites via electronic mail. You probably do *not* need to read this document if don't exchange electronic mail with other users on your system or with other sites.*

1. Introduction, copyright and standard disclaimer

- 1.1 Email and spamming
- 1.2 Goals
- 1.3 New versions
- 1.4 Feedback
- 1.5 Copyright
- 1.6 Limited warranty

2. Other sources of information

- 2.1 Mail User's HOWTO
- 2.2 USENET
- 2.3 Mailing Lists
- 2.4 Other documents from LDP
- 2.5 Books

TOC

3. How Electronic Mail Works

- 3.1 Mail between full-time Internet machines
- 3.2 Notifiers
- 3.3 Mail to part-time Internet machines
- 3.4 Remote mail and remote-mail protocols
- 3.5 Mailbox formats

4. Requirements

- 4.1 Hardware

5. Choosing a Mail Transport Agent

- 5.1 sendmail
- 5.2 smail v3.2
- 5.3 qmail
- 5.4 exim

6. Intalling Transport Software

- 6.1 Qmail v1.03
- 6.2 Smail v3.1
- 6.3 "fidopaths" file
- 6.4 Restarting inetd
- 6.5 Smail with smtp
- 6.6 OUTDATED SECTION: Sendmail+IDA
- 6.7 Sendmail 8.x
- 6.8 Sendmail v8 tidbits
- 6.9 Local Delivery Agents

7. User Agent Administration

- 7.1 Mutt
- 7.2 Elm

- 7.3 Mailx

8. Handling remote mail

- 8.1 History
- 8.2 Getting mail
- 8.3 Sending mail
- 8.4 Reading mail
- 8.5 Testing
- 8.6 Using

9. Acknowledgements

---

1. Introduction, copyright and standard disclaimer

1.1 Email and spamming

To send mail to anyone mentioned in this document, convert "at" in email addresses to "@".

This conversion is simple for humans, but not spammers' address harvesters; therefore it's useful to protect generous contributors from being spammed!

1.2 Goals

The intent of this document is to answer some of the questions and comments that appear to meet the definition of "frequently asked questions" about e-mail software under Linux in general and the version in the Linux Debian and RedHat distributions in particular.

1.3 New versions

New versions of this document will be periodically posted to comp.os.linux.announce, comp.answers and mail.answers. They will also be added to the various anonymous ftp sites who archive such information including *sunsite.unc.edu:/pub/Linux/docs/HOWTO*.

In addition, you should be generally able to find this document on the Linux WorldWideWeb home page at *http://sunsite.unc.edu/mdw/linux.html*.

1.4 Feedback

I am interested in any feedback, positive or negative, regarding the content of this document via e-mail. Definitely contact me if you find errors or obvious omissions.

I read, but do not necessarily respond to, all e-mail I receive. Requests for enhancements will be considered and acted upon based on that day's combination of available time, merit of the request and daily blood pressure :-)

Flames will quietly go to /dev/null so don't bother.

Feedback concerning the actual format of the document should go to the HOWTO coordinator : Greg Hankins (`gregh at sunsite.unc.edu`).

TOC

1.5 Copyright

The Mail-Administrator HOWTO is copyrighted (c) 1998 Guylhem Aznar. Distributed under LDP copyright license. If you have questions, please contact the Linux HOWTO coordinator, at `linux-howto at sunsite.unc.edu`.

1.6 Limited warranty

Of course, I disavow any potential liability for the contents of this document. Use of the concepts, examples, and/or other content of this document is entirely at your own risk.

---

2. Other sources of information

2.1 Mail User's HOWTO

There is a Mail User's HOWTO, which focuses on user issues.

It is currently maintained by Eric S. Raymond; you should be able to view it on the World Wide Web at http://metalab.unc.edu/LDP/HOWTO/Mail-User-HOWTO.html.

2.2 USENET

There is nothing special about configuring and running mail software under Linux (any more). Accordingly, you almost certainly do *NOT* want to be posting generic mail-related questions to the comp.os.linux.* newsgroups.

Don't post in comp.os.linux hierarchy unless it's really linux specific, for example : "Which options was Debian 1.2 sendmail compiled with ?" or "RedHat 5.0 smail crashes when I run it".

Let me repeat that.

There is virtually no reason to post anything mail-related in the comp.os.linux hierarchy any more. There are existing newsgroups in the comp.mail.* hierarchy to handle *ALL* your questions.

IF YOU POST TO COMP.OS.LINUX. FOR NON-LINUX-SPECIFIC QUESTIONS, YOU ARE LOOKING IN THE WRONG PLACE FOR HELP. THE MAIL EXPERTS HANG OUT IN THE PLACES INDICATED ABOVE AND GENERALLY DO NOT RUN LINUX.*

POSTING TO THE LINUX HIERARCHY FOR NON-LINUX-SPECIFIC QUESTIONS WASTES YOUR TIME AND EVERYONE ELSE'S AND IT FREQUENTLY DELAYS YOUR GETTING THE ANSWER TO YOUR QUESTION.

GOOD PLACES are :

```
comp.mail.elm           the ELM mail system.
comp.mail.mh            The Rand Message Handling system.
comp.mail.mime          Multipurpose Internet Mail Extensions.
comp.mail.misc          General discussions about computer mail.
comp.mail.multi-media   Multimedia Mail.
comp.mail.mush          The Mail User's Shell (MUSH).
comp.mail.sendmail      the BSD sendmail agent.
comp.mail.smail         the smail mail agent.
comp.mail.uucp          Mail in the uucp environment.
```

TOC

2.3 Mailing Lists

There are many sendmail, smail and qmail mailing lists.

You can find addresses in /usr/doc/the_one_you_have_chosen.

2.4 Other documents from LDP

There is plenty of excellent material provided in the other Linux HOWTO documents and from the Linux DOC project.

In particular, you might want to take a look at the following:

- on your own computer in /usr/doc/ :-)
- the Linux Networking Administrators' Guide
- the Mail Users HOWTO
- the Serial Communications HOWTO
- the Ethernet HOWTO
- the UUCP HOWTO if you're fed via UUCP

2.5 Books

The following is a non-inclusive set of books that will help:

- "`Managing UUCP and USENET`" from O'Reilly and Associates is in my opinion the best book out there for figuring out the programs and protocols involved in being a USENET site.
- "`Unix Communications`" from The Waite Group contains a nice description of all the pieces (and more) and how they fit together.
- "`Sendmail`" from O'Reilly and Associates looks to be the definitive reference on sendmail-v8 and sendmail+IDA. It's a "must have" for anybody hoping to make sense out of sendmail without bleeding in the process.
- "`The Internet Complete Reference`" from Osborne is a fine reference book that explains the various services available on Internet and is a great source for information on news, mail and various other Internet resources.
- "`The Linux Networking Administrators' Guide`" from Olaf Kirch of the Linux Documentation Project is available on the net and is also published by (at least) O'Reilly and SSC. It makes a fine one-stop shop to learn about everything you ever imagined you'd need to know about Unix networking.

---

3. How Electronic Mail Works

Now we'll explain the flow of information that typically takes place when two people to communicate by email. Let us suppose that Alice, on her machine `wonderland.com`, wants to send mail to Bob, on his machine `dobbs.com`. Both machines are connected to the Internet.

It helps to know that an Internet mail message consists of two parts; mail headers and a mail body, separated by a blank line. The mail headers contain the source and destination of the mail, a user-supplied subject line, the date it was sent, and various other kinds of useful information. The body is the actual content of the message. Here's an example:

```
From: "Alice" <alice@wonderland.com>
Message-Id: <199711131704.MAA18447@wonderland.com>
Subject: Have you seen my white rabbit?
To: bob@dobbs.org (Bob)
Date: Thu, 13 Nov 1997 12:04:05 -0500 (EST)
Content-Type: text

I'm most concerned.  I fear he may have fallen down a hole.
--
                                                    >>alice>>
```

The arrangement and meaning of Internet mail headers are defined by an Internet standard called RFC822.

3.1 Mail between full-time Internet machines

Here's a diagram of the whole process -- I'll explain all the stages and terminology below.

```
                   +---------+          +-------+
+-------+  types   | sending |  calls   |sending|
| Alice |--------->|   MUA   |--------->|  MTA  |::::>::::
+-------+          |         |          |       |       ::   on the
                   +---------+          +-------+       ::   sending
                                                        ::   machine
.......................................................................
                              SMTP                      ::
 ::::::::::::::::::::::::::::<::::::::::::::::::::::::::
 ::
 ::   +---------+          +-----+                +-------+
 ::   |receiving|  calls   |     |  delivers to   | Bob's |
 ::::>|   MTA   |--------->| LDA |===============>|mailbox|  on the
      |         |          |     |                |       |  receiving
      +---------+          +-----+                +-------+  machine
```

```
                                                |   |
                                                |   |
                 +----------------<-------------+   |
                 |                                  |
            +---------+         +-------+           |
            |  Bob's  |         | Bob's |<----------+
            | notifier|         |  MUA  |
            +---------+         +-------+
                 |                  |
                 |      +-----+     |
                 +----->| Bob |<----+
                        +-----+
```

To send mail, Alice will invoke a program called a mail user agent (or MUA for short). The MUA is what users think of as `the mailer'; it helps her compose the message, usually by calling out to a text editor of her choice. When she hits the MUA `send' button, her part of the process is done. Later in this HOWTO we will survey popular MUAs.

The MUA she uses immediately hands her message to a program called a mail transport agent (or MTA). Usually this program will be `sendmail`, though some alternative MTAs are gaining popularity and may appear in future Linux distributions. Later in this HOWTO we will also survey MTAs.

The MTA's job is to pass the mail to an MTA on Bob's machine. It determines Bob's machine by analyzing the To header and seeing the `dobbs.com` on the right-hand side of Bob's address. It uses that address to open an Internet connection to Bob's machine. The mechanics of making that connection are a whole other topic; for this explanation, it's enough to know that that connection is a way for Alice's MTA to send text commands to Bob's machine and recieve replies to those commands.

The MTA's commands don't go to a shell. Instead they go to a service port on Alice's machine. A service port is a sort of rendezvous, a known place where Internet service programs listen for incoming requests. Service ports are numbered, and Alice's MTA knows that it needs to talk to port 25 on Bob's machine to pass mail.

On port 25, Bob's machine has its own MTA listening for commands (probably another copy of sendmail). Alice's MTA will go through a dialogue with Bob's using Simple Mail Transfer Protocol (or SMTP). Here is what an SMTP dialogue looks like. Lines sent by Alice's machine are shown with S:, responses from Bob's machine are shown with R:.

```
S: MAIL FROM:<alice@wonderland.com>
R: 250 OK
S: RCPT TO:<bob@dobbs.com>
R: 250 OK
S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: From: "Alice" <alice@wonderland.com>
S: Message-Id: <199711131704.MAA18447@wonderland.com>
S: Subject: Have you seen my white rabbit?
S: To: bob@dobbs.org (Bob)
S: Date: Thu, 13 Nov 1997 12:04:05 -0500 (EST)
S: Content-Type: text
```

```
S:
S: I'm most concerned.  I fear he may have fallen down a hole.
S: --
S:                                                  >>alice>>
S: .
R: 250 OK
```

Usually an SMTP command is a single text line and so is its response. The DATA command is an exception; after seeing that, the SMTP listener accepts message lines until it sees a period on a line by itself. (SMTP is defined by the Internet standard RFC821.)

Now Bob's MTA has Alice's message. It will add a header to the message that looks something like this:

```
Received: (from alice@wonderland.com)
        by mail.dobbs.com (8.8.5/8.8.5) id MAA18447
        for alice; Thu, 13 Nov 1997 12:04:05 -0500
```

This is for tracking purposes in case of mail errors (sometimes a message has to be relayed through more than one machine and will have several of these). Bob's MTA will pass the modified message to a local delivery agent or LDA. On Linux systems the LDA is usually a program called `procmail`, though others exist.

The LDA's job is to append the message to Bob's mailbox. It's separate from the MTA so that both programs can be simpler, and so the MTA can concentrate on doing Internet things without worrying about local details like where the user mailboxes live.

Bob's mailbox will normally be a file called /usr/spool/mail/bob or /var/mail/bob. When he reads mail, he runs his own MUA (mail user agent) to look at and edit that file.

3.2 Notifiers

There's yet another kind of program that is important in the mail chain, though it does not itself read or transmit mail. It's a *mail notifier*, a program that watches your email in-box for activity and signals you when new mail is present.

The original notifier was a pair of Unix programs called biff(1) and comsat(8). The biff program is a front end that enables you to turn on the comsat service. When this service is on, the header of news mail will be dumped to your terminal as it arrived. This facility was designed for people using line-oriented programs on CRTs; it's not really a good idea in today's environment.

Most Unix shells have built-in mailcheck facilities that allow them to function as notifiers in a rather less intrusive way (by emitting a message just before the prompt when new mail is detected). Typically you can enable this by setting environment variables documented on the shell's manual page. For shells in the sh/ksh/bash family, see the MAIL and MAILPATH variables

Systems supporting X come with one of several little desktop gadgets that check for new mail periodically and give you both visible and audible indication of new mail. The oldest and most widely used of these is called *xbiff*; if your Linux has a preconfigured X desktop setup, xbiff is probably on it. See the xbiff(1) manual page for details.

TOC

3.3 Mail to part-time Internet machines

If you were reading carefully, you may have noticed that the information flow we described above depends on Alice's machine being able to talk to Bob's machine immediately. What happens if Bob's machine is down, or is up but not connected to the Internet?

If Alice's MTA can't reach Bob's immediately, it will stash Alice's message in a mail queue on `wonderland.com`. It will then retry sending the mail at intervals until an expiration time is reached, at which point a bounce message notifying Alice of the failure will be sent back to her. In the default configuration of the most popular MTA (sendmail), the retry interval is 15 minutes and the expiration time is 4 days.

3.4 Remote mail and remote-mail protocols

Many Linux users nowadays are connected to the Internet via ISPs (Internet Service Providers) and don't have their own Internet domains. Instead they have accounts on an ISP machine. Their mail gets delivered to a mailbox on that ISP machine. But typically these users want to read and reply to their mail using their own machines, which connect to the ISP intermittently using SLIP or PPP. Linux supports remote mail protocols to support this.

Note how this is different from the scenario we discussed in the last section. Mail sitting in a queue awaiting retransmission is not the same as mail dispatched to a server mailbox; mail in a queue is not considered to have been delivered and is subject to expiration, but mail delivered to an ISP server mailbox *is* considered `delivered' and can sit there indefinitely.

A remote-mail protocol allows mail on a server to be pulled across a network link by a client program (this is the opposite of normal delivery in which an MTA pushes mail to a receiving MTA). There are two remote-mail protocols in common use; POP3 (defined by the Internet standard RFC1939) and IMAP (defined by the Internet standard RFC2060). Effectively all ISPs support POP3; a growing number support IMAP (which is more powerful).

Here is what an example POP3 session looks like:

```
S: <client connects to service port 110>
R:    +OK POP3 server ready <1896.697170952@mailgate.dobbs.org>
S:    USER bob
R:    +OK bob
S:    PASS redqueen
R:    +OK bob's maildrop has 2 messages (320 octets)
S:    STAT
R:    +OK 2 320
S:    LIST
R:    +OK 2 messages (320 octets)
R:    1 120
R:    2 200
R:    .
S:    RETR 1
R:    +OK 120 octets
R:    <the POP3 server sends message 1>
```

TOC

```
R:    .
S:    DELE 1
R:    +OK message 1 deleted
S:    RETR 2
R:    +OK 200 octets
R:    <the POP3 server sends message 2>
R:    .
S:    DELE 2
R:    +OK message 2 deleted
S:    QUIT
R:    +OK dewey POP3 server signing off (maildrop empty)
S:  <client hangs up>
```

An IMAP session uses different commands and responses, but is logically very similar.

To take advantage of POP3 or IMAP, you need a remote mail client program to pull your mail. Some mail user agents have client capabilities built in (which one supports which is noted below), and the Netscape browser's mail facility supports both POP and IMAP natively.

The main drawback of POP client facilities built into MUAs is that you have to explicitly tell your mailer to poll the server; you don't get notified by xbiff(1) or equivalent, as you would for mail that is either local or delivered by a conventional SMTP `push' connection. Also, of course, not all MUAs can do POP/IMAP, so you may find yourself compromising on other features.

Your Linux probably comes with a program called fetchmail that is designed specifically to talk to remote-mail servers, fetch mail, and feed it into your normal mail delivery path by speaking SMTP to your listener.

Unless you need to keep your mail on the server (for example, because you move around between client machines a lot) fetchmail is probably a better solution than whatever POP/IMAP features your user agent has. Fetchmail can be told to run in background and poll your server periodically, so your xbiff(1) or other mail-notifier program will work as it would for SMTP mail. Also, fetchmail is rather more tolerant of various idiosyncracies and nonstandard server quirks than the clients in MUAs, and has better error recovery.

Here's a diagram showing how both cases (with and without fetchmail) work:

```
                    +---------+          +-------+
+-------+  types    | sending |  calls   |sending|
| Alice |--------->|   MUA   |--------->|  MTA  |::::>::::
+-------+          |         |          |       |       ::
                    +---------+          +-------+       ::   on the
                                                         ::   sending
                              SMTP                       ::   machine
 ::::::::::::::::::::::::::::<::::::::::::::::::::::::::::
 ::
.::.......................................................................
 ::
 ::   +---------+          +-----+             +-------+
 ::   |receiving|  calls   |     |  delivers   | Bob's |
 ::::>|   MTA   |--------->| LDA |============>|server |::::>::::
```

```
     |         |          |     |    to       |mailbox|       ::  on t
     +---------+          +-----+             +-------+       ::   mai
                                                              ::  server
                          POP or IMAP                         ::
 ::::::::::::::::::::::::::::<::::::::::::::::::::::::::::::::::
 ::
.::.............................................................
 ::
 ::                  +-----------+
 ::                  |           |
 :::::::>:::::::::::| fetchmail |::::::::                on the
 ::                  |           |      ::              receiving
 ::                  +-----------+      ::               machine,
 ::                                     ::             with fetchmail
 ::   ::::::::::::::::<::::::::::::::::::
 ::   ::
 ::   ::   +---------+          +-----+                +-------+
 ::   ::   |receiving|  calls   |     |  delivers to   | Bob's |
 ::   ::::>|   MTA   |--------->| LDA |===============>|mailbox|
 ::        |         |          |     |                |       |
 ::        +---------+          +-----+                +-------+
 ::                                                       |   |
 ::                                                       |   |
 ::                      +----------------<-------------+   |
 ::                      |                                  |
 ::                  +---------+         +-------+          |
 ::                  |  Bob's  |         | Bob's |<----------+
 ::                  | notifier|         |  MUA  |
 ::                  +---------+         +-------+
 ::                      |                  |
.::.............................................................
 ::                   .  |                 |
 ::     without       .  |                 |
 ::    fetchmail      .  |                 |
 ::                   .  |      +-----+    |
 ::   +----------+    .  +----->|     |<----+
 ::   |  Bob's   |    .          | Bob |
 :::::| POP/IMAP |----.--------->|     |
      |   MUA    |    .          +-----+
      +----------+    .
```

3.5 Mailbox formats

When incoming mail gets appended to a mailbox, it's up to the MTA to provide some kind of delimiters that tell where one message stops and the next begins.

Under Unix, the convention almost all mailers use is that each line beginning with ``From '' (the space is

significant) begins a new message. If ``From " occurs at the beginning of a line in text, a Unix MTA will generally prefix it with a less-than sign, so it looks like ``>From ". RFC822 headers follow this From line (which usually continues with the sender name and receipt date).

This convention originated with Unix Version 7, so this kind of mailbox is referred to as a ``V7 mailbox"; it is also sometimes called ``mbox format". Unless otherwise noted, all programs mentioned in this HOWTO expect this format. It is not, however, quite universal, and tools expecting and generating different formats can confuse each other badly.

The four other formats to know about (and beware of!) are BABYL, MMDF, MH, and qmail maildir. Of these, MMDF is the simplest; it uses a delimiter line consisting four control-As (ASCII 001) characters followed by CR-LF. MMDF was an early and rather crude Internet mail transport; a descendant is still in use on SCO systems.

BABYL is another survival, from an early mail system at MIT. It is still used by Emacs's mail-reader mode.

MH and qmail maildir are `mailbox' formats which actually burst each mailbox into a directory of files, one per message. Running grep on such a `mailbox' will get you nowhere, since all grep will see are the directory bits.

Microsoft Outlook Express .mbx mailboxes can be converted to RFC822 format with mbx2mbox app.

---

4. Requirements

4.1 Hardware

There are no specific hardware requirements for mail under Linux.

You'll need some sort of 'transport' software to connect to remote systems, which means either TCP/IP or uucp.

This could mean that you need a modem or ethernet card, depending on your setup. In most cases, you'll want the fastest modem you can afford, i.e. V90 57 600 bps currently. In general, you want to have a 16550 UART on your serial board or built into your modem to handle speeds of above 9600 baud.

If you don't know what that last sentence means, please consult the `comp.dcom.modems` group or the various fine modem and serial communications FAQs and periodic postings on USENET.

5. Choosing a Mail Transport Agent

Mail transport agents are the software that transfers mail from your local system to remote systems. It is very seldom necessary to mess with or replace your MTA on a modern Linux, and you're better off not fixing what isn't broken. Nevertheless, here's a survey to get you started on understanding what the tradeoffs are if you decide you need more security or performance than your system's default can offer.

(There are other Unix MTAs besides these, but you are quite unlikely to encounter them on a Linux box.)

Each has its own unique features, but the best compromise is qmail. It features high security (even if vmail is more secure), high speed (even if smail is faster for local uses) and ease of configuration. Of course, feel free to choose any mail software. The information provided here is intended to help you choose well.

Sendmail can be nice for many sites with complicated options, but I think its configuration is too hard for beginners while it is not very secure or very fast, so there is only a **really** outdated sendmail section in this HOWTO.

If you know what you're doing, choose sendmail (and you shouldn't be reading this HOWTO!); otherwise I generally recommend qmail.

Detailed descriptions of these programs follow.

5.1 sendmail

BSD sendmail is the grandaddy of Internet MTAs. It has outlasted a few would-be successors. Most Linux distributions now use it and have it preinstalled.

Sendmail has a long-standing reputation for being an administrator's nightmare -- hard to understand, tricky to configure, rife with security holes. As Internet technology and standards have stabilized, however, many of the sendmail options and configurable rules that gave rise to this reputation have ceased to require per-site tweaking (the effective demise of non-TCP/IP network layers like UUCP has helped a lot). Also, recent sendmail versions have an improved configuration system that insulates you from the legendary hideousness of the sendmail.cf configuration file. Most importantly, sendmail now normally comes preconfigured, and you should never need to touch it unless you have unusual requirements (such as needing to route mail over a non-TCP/IP network).

There is a sendmail home page at http://www.sendmail.org. It includes references to extensive documentation of sendmail, should you actually need to wrestle with custom-configuring it.

Other MTAs, if called as `sendmail', may mimic the semantics of sendmail's command-line options. This is convenient for mail user agents, which often assume they are talking to sendmail.

5.2 smail v3.2

Smail was the first serious attempt to replace sendmail. It has a simpler and much more comprehensible configuration system than sendmail's, and it's fairtrly secure. Some Linux distributions preinstall it rather than sendmail.

At one time smail's excellent support for mixed TCP/IP and UUCP sites was a major selling point for it, but as UUCP has declined, so has smail. Also, smail is less efficient than sendmail on high-volume connections.

As with sendmail, it is unlikely that you will need to tweak a preinstalled smail configuration.

(Very occasionally you might run across references to an `smail 2.5'. This program has been obsolete for a long time. Don't bother with it.)

5.3 qmail

The qmail program is a sendmail-compatible MTA designed specifically for high security. The author has a standing reward of $500 for publication of the first verifiable security hole; this reward has gone unclaimed since March 1997.

The qmail home page is at http://pobox.com/~djb/qmail.html.

5.4 exim

The exim program is similar to smail3, but with more features. It advertises particular strengths in spam-blocking and support of several virtual hosts (virtual DNS domains) on the same host.

The exim home page is at http://www.exim.org/.

I tryed it on my own computer, it looks like a nice merge between smail configuration system and qmail security, moreover it has the advantage of being GPL.

A section explaining how to replace your current MTA by exim will be added soon.

---

6. Intalling Transport Software

6.1 Qmail v1.03

Secured, fast and easy to use, this is my preferred MTA (mail transport agent).

Currently, no distribution comes with qmail preinstalled. We will focus on compiling and installing qmail, since this is the only tricky part: configuration is really straightforward.

Getting qmail

Go to www.qmail.org to download the latest version.

Uncompressing sources

Then decompress it by running:

```
mv qmail.tar.gz /usr/local/src
cd /usr/local/src ; tar -zxvf qmail.tar.gz
```

If you find a bz2 version (new and better compression format), just replace tar with:

```
bunzip2 qmail.tar.bz2
tar -xvf qmail.tar
```

Preparing for compilation

Now enter the qmail directory to examine the configuration defaults:

```
cd qmail; more conf-*
```

You shouldn't need to change any defaults, but you could (for example) specify an alternate installation directory or better compilation flags.

Now run:

```
mkdir /var/qmail
```

to create target dir.

If you haven't installed a Debian distribution, you'll need to add several user IDs for qmail's use: qmail's high security depends on that.

The fact that qmail is divided into modules running each under their own UID makes it much harder for an intruder to break your whole mail system or gain root access by abusing it.

So run:

```
# groupadd nofiles
# useradd -g nofiles -d /var/qmail/alias alias
# useradd -g nofiles -d /var/qmail qmaild
```

```
# useradd -g nofiles -d /var/qmail qmaill
# useradd -g nofiles -d /var/qmail qmailp
# groupadd qmail
# useradd -g qmail -d /var/qmail qmailq
# useradd -g qmail -d /var/qmail qmailr
# useradd -g qmail -d /var/qmail qmails
```

or hand-edit `/etc/passwd` and `/etc/group` to add these users by yourself.

Evan E. reported he had to use "-g groupid" parameter for a vanilla groupadd (Caldera 1.2), else groupadd reported this error : "A group with that name already exists."

For example you can respectively add:

```
qmail:*:2107:
nofiles:*:2108:
```

&

```
alias:*:7790:2108::/var/qmail/alias:/bin/true
qmaild:*:7791:2108::/var/qmail:/bin/true
qmaill:*:7792:2108::/var/qmail:/bin/true
qmailp:*:7793:2108::/var/qmail:/bin/true
qmailq:*:7794:2107::/var/qmail:/bin/true
qmailr:*:7795:2107::/var/qmail:/bin/true
qmails:*:7796:2107::/var/qmail:/bin/true
```

Now you can run

```
make setup check
```

to check your configuration, then :

```
./config
```

to configure qmail.

Attention, your server has to be resolvable by DNS or ./config will get confused.

Now you must install some aliases, since /etc/alias is not used by qmail unless you compile and install an optional package.

Here's my setup :

```
File : ".qmail-MAILER-DAEMON"
&postmaster
File : ".qmail-bin"
&root
File : ".qmail-daemon"
&root
File : ".qmail-decode"
&root
File : ".qmail-dumper"
&root
File : ".qmail-games"
&root
File : ".qmail-ingres"
&root
```

```
File : ".qmail-mailer-daemon"
&postmaster
File : ".qmail-manager"
&root
File : ".qmail-news"
&root
File : ".qmail-nobody"
&root
File : ".qmail-operator"
&root
File : ".qmail-postmaster"
&root
File : ".qmail-root"
&guylhem
File : ".qmail-system"
&root
File : ".qmail-toor"
&root
File : ".qmail-uucp"
&root
File : ".qmail-uucp-default"
|preline -dr /usr/bin/uux - -r -gC -a"${SENDER:-MAILER-DAEMON}" lm!rmail
"($DEFAULT@$HOST)"
```

You need to create each of these file in `~alias`, replacing `&guylhem` in `.qmail-root` by your own login to get root mail.

ATTENTION UUCP USERS !

DO NOT TRUST THE QMAIL FAQ FOR UUCP, USE MY .qmail-uucp-default INSTEAD! ELSE YOU WILL NOT BE ABLE TO SEND ANY MAIL BY YOUR UUCP CONNEXION!

Now you'll need to decide in which format your users will get their mail.

Here's my suggestion :

- For NFS mounted home dirs, use MAILDIR format with a patch for local mail readers (patchs are available on www.qmail.org)
- If no patch is available, prefer MAILFILE format : any mail reader can read a file containing mail, people will only need to create an alias (for bash) or a setenv (for csh) for their mail reader
- Avoid /var/spool/mail/$USER format, too unsecure

To fix the default format, read each file in `/var/qmail/boot` then copy the one you best like to `/var/qmail/rc`.

`home` or `proc` are safe choices, but prefer `home` for security reasons.

Configuring qmail

In /var/qmail/control, edit:

defaultdomain, me, plusdomain

- me is you local FQDN (full qualified domain name), for example on my machine it is `barberouge.linux.lmm.com`
- defaultdomain will be added to any host name without dots, including defaulthost, for example you can set it to `localnetwork` so any mail sent to joe@hisbox will be completed to be sent to joe@hisbox.localnetwork instead

TOC

- plusdomain is the exception: is is added to any host name that ends with a plus sign, including defaulthost (set i if it ends with a plus sign.

These 3 examples show you the power and ease of configuration of qmail!

locals, rcpthosts

If you want to support virtual domain names, just put additional names in these files. Any mail you receive for these names will be handled locally.

The difference between `locals` and `rcpthosts` is the latter isn't considered as a local alias, which is useful if you receive mail from some free email address like yahoo.com or lemel.fr while you also send mail to other users of these non local services, i.e. you don't want to handle locally mail send to someone@yahoo.com!

virtualdomains

There can you specify defaut outgoing mode, for example :

```
#:alias-uucp
```

if you don't want to send outgoing mail by uucp but by smtp (default) or

```
:alias-ucp
```

if you send your outgoing mail by uucp.

Testing qmail

Now it is configured, try:

```
sh -cf '/var/qmail/rc &'
```

to launch qmail (it won't interfere with your local MTA), then:

```
echo to: mylogin | /var/qmail/bin/qmail-inject
```

You should receive this mail in the format you've chosen in `/var/qmail/boot/`.

Removing your other MTA

If this test was successful, just kill your previous MTA:

killall -STOP daemon_name ; if any children are running, you should killall -CONT their_name, wait, killall -STOP again, and repeat ad nauseam.

If there aren't any children, killall -TERM and then killall -CONT.

Remove it (how you can do this depends on the distribution you installed, for example rpm -e --nodeps on RedHat, Caldera and Suse, or dpkg -r --force-depends on Debian) then run:

```
# ln -s /var/qmail/bin/sendmail /usr/lib/sendmail
# ln -s /var/qmail/bin/sendmail /usr/sbin/sendmail
```

Now set up qmail-smtpd in /etc/inetd.conf (all on one line):

```
smtp stream tcp nowait qmaild /var/qmail/bin/tcp-env tcp-env
```

```
/var/qmail/bin/qmail-smtpd
```

If you are using a old non-SYSV-init distribution like redhat, just add to your boot scripts:

```
sh -cf '/var/qmail/rc &'
```

Usually this should be /etc/rc.local but your mileage may vary.

For actual SYSV-init compliant distributions (RedHat, Caldera, Suse, Debian), add this script to /etc/init.d/:

DEBIAN version:

```
#!/bin/sh

test -x /var/qmail/rc || exit 0

case "$1" in
  start)
     echo -n "Starting mta: "
     sh -cf '/var/qmail/rc &'
     echo "qmail."
     ;;
  stop)
     echo -n "Stopping mta: "
     killall qmail-lspawn
     echo "qmail."
     ;;
  restart)
     echo -n "Restarting mta: "
     killall -HUP qmail-lspawn
     killall -ALRM qmail-lspawn
     echo "qmail."
     ;;
  *)
     echo "Usage: /etc/init.d/qmail {start|stop|restart}"
     exit 1
esac

exit 0
```

REDHAT version:

```
#!/bin/sh
#
# qmail      This shell script takes care of starting and stopping qmail.
#
# description: qmail is a Mail Transport Agent, which is the program \
#              that moves mail from one machine to another.
# processname: qmail
# config: /var/qmail/control/

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
```

```
. /etc/sysconfig/network

export PATH=$PATH:/var/qmail/bin

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

[ -f /usr/sbin/sendmail ] || exit 0

# See how we were called.
case "$1" in
  start)
        # Start daemons.
        echo -n "Starting qmail: "
        qmail-start '|preline procmail' splogger qmail &
        touch /var/lock/subsys/qmail
        echo
        ;;
  stop)
        # Stop daemons.
        echo -n "Shutting down qmail: "
        killproc qmail-lspawn
        echo
        rm -f /var/lock/subsys/qmail
        ;;
  restart)
        $0 stop
        $0 start
        ;;
  status)
        status qmail
        ;;
  *)
        echo "Usage: qmail {start|stop|restart|status}"
        exit 1
esac

exit 0
```

And make symlinks to each /etc/rcN.d/, for example:

```
ln -sf /etc/init.d/qmail /etc/rc1.d/K19qmail
```

If the first letter is K, you will kill qmail on this runlevel (1 for single mode or 6 for boot), but if the first letter is S, you will start qmail on this runlevel (each others runlevel).

- How to decide whether you should put a K or a S? Do what the majority of dæmons in this runlevel do!
- What number should you put after K or S? The number next to your network daemon.

RedHat, Caldera and Suze will use /etc/rc.d/ instead of plain /etc/ for Debian distribution, i.e. /etc/rc.d/rc1.d or /etc/rc.d/init.d for example.

That's all, folks!

No need to reboot (remember, you're using linux, not some other cheap OS!) for the modifications to take effect, just run:

```
killall inetd
init 1
```

To go to single user mode, then:

```
init 2
```

to go back to your default runlevel (indicated in /etc/inittab with initdefault label).

You could also hand-start qmail script but "init" method will show you if qmail script is well positioned, i.e. launched after network scripts but before any program depend on email to warn you (like inn).

6.2 Smail v3.1

Smail3.1 seems to be a de-facto standard transport agent for uucp-only sites and for some smtp sites. It's easy to configure, it compiles without patching from the sources and it's fairly secure.

Configuring smail

Install the smail binary from your distribution (I recommand you choose this) or get smail sources and build smail. If you're building smail from sources, you need to have the following in your os/linux file so that 'sed' gives you shell scripts that work properly.

```
CASE_NO_NEWLINES=true
```

Once it's installed, config. files will certainly go in /etc/smail (but your mileage may vary if you use old distributions); let's start editing them!

"config" file

```
# From
smart_path=polux
smart_transport=uux

# To
hostname=barberouge
domains=linux.lmm.com

visible_name=barberouge.linux.lmm.com
uucp_name=barberouge.linux.lmm.com

# max_message_size=512k
# auth_domains=foo.bar
# more_hostnames=barberouge.polux.freenix.fr
```

Well, first, who is feeding you? I'm fed by "polux" via uucp (i.e. uux transport); naturally you need to change this file according to your own situation. For example, you could by fed by "bargw.bar.foobar.com" via "smtp", in that case you don't need a transport file and can define "-transport_file " to indicate you don't need one.

You can also use "postmaster_address = yourname", hide the network topology in outgoing addresses (if you're a gateway)

using "visible_name", set which aliases address can also be used for the email you receive, using "more_hostnames".

See smail documentation for more details or the examples in /usr/doc/smail/examples to see if any match your situation.

"directors" file

```
# aliasinclude - expand ":include:filename" addresses produced by alias files
# This entry and the next one are pretty much boiler-plate.  Reasons
# for making significant changes are few.  The sole purpose of these
# is to match and expand addresses of the form:
#       :include:pathname
# which may occur in alias files or mailing-list/forward files
# (produced by any director with a driver of forwardfile).
aliasinclude:
        driver = aliasinclude,          # use this special-case driver
        nobody;                         # associate nobody user with addresses
                                        #  when mild permission violations
                                        #  are encountered
        copysecure,                     # get permissions from alias director
        copyowners,                     # get owners from alias director



# forwardinclude - expand ":include:filename" addrs produced by forward files
forwardinclude:
        driver = forwardinclude,        # use this special-case driver
        nobody;
        copysecure,                     # get perms from forwarding director
        copyowners,                     # get owners from forwarding director


# aliases - search for alias expansions stored in a database
# This is the standard aliases file.  It is used for generic things,
# like mapping root, postmaster, MAILER-DAEMON and uucp to site
# admins, creating some small system alias expansions, and such.  In
# this site configuration, the aliases file is used mostly for
# machine-specific aliasing/forwarding information.  Global forwarding
# information should be put in the "forward" database.
aliases:
        driver=aliasfile,               # general-purpose aliasing director
        -nobody,                        # all addresses are associated
                                        # with nobody by default, so setting
                                        # this is not useful.
        sender_okay,                    # don't remove sender from expansions
        owner=owner-$user;              # problems go to an owner address
        file=/etc/aliases,
        modemask=002,                   # should not be globally writable
        optional,                       # ignore if file does not exist
        proto=lsearch,                  # unsorted ASCII file


# forward - search for expansions stored in a forwarding database
# This is the subdomain-wide user forwarding database.  Entries are
```

```
# maintained here for current or past users, to forward their mail to
# their preferred mail-reading machine.  The forward database is
# shipped around the TCP/IP network as changes are made, to keep the
# network consistent.
#forward:
#       driver = aliasfile,             # general-purpose aliasing director
#       -nobody,                        # all addresses are associated
#                                       # with nobody by default, so setting
#                                       # this is not useful.
#       owner = real-$user;             # problems go to an owner address
#
#       file = /etc/forward,
#       modemask = 002,
#       proto = dbm,                    # use dbm(3X) library for access


# dotforward - expand .forward files in user home directories
# For users that have an entry in the "forward" database, a ".forward"
# file is only used if it is on the "home" machine, as identified in
# the forward database.  If used, it is treated as a list of addresses
# to which mail should be delivered, rather than (or in addition to)
# the user identified in the local address.
dotforward:
        driver = forwardfile,           # general-purpose forwarding director
        owner = postmaster, nobody, sender_okay;

        file = ~/.forward,              # .forward file in home directories
        checkowner,                     # the user can own this file
        owners = root,                  # or root can own the file
        modemask = 002,                 # it should not be globally writable
        caution = daemon:root,          # don't run things as root or daemon
        # be extra careful of remotely accessible home directories
        unsecure = "~uucp:/tmp:/usr/tmp:/var/tmp"


# forwardto - expand a "Forward to " in user mailbox files
# This emulates the V6/V7/System-V forwarding mechanism which uses a
# line of forward addresses stored at the beginning of user mailbox files
# prefixed with the string "Forward to "
forwardto:
        driver = forwardfile,
        owner = postmaster, nobody, sender_okay;

        file = /var/spool/mail/${lc:user},      # point at user mailbox files
        forwardto,                      # enable "Forward to " function
        checkowner,                     # the user can own this file
        owners = root,                  # or root can own the file
        modemask = 0002,                # under System V, group mail can write
        caution = daemon:root           # don't run things as root or daemon


# user - match users on the local host with delivery to their mailboxes
user:   driver = user;                  # driver to match usernames
        transport = local               # local transport goes to mailboxes
```

```
# real_user - match usernames when prefixed with the string "real-"
# This is useful for allowing an address which explicitly delivers to a
# user's mailbox file.  For example, errors in a .forward file expansion
# could be delivered here, or forwarding loops between multiple machines
# can be resolved by using a real-username address.  Also, users that
# wish to use mail as a means of transferring data to a machine that
# is not their "home" machine can mail to real-login-name@remote.host.
real_user:
        driver = user;
        transport = local,
        prefix = "real-"                # for example, match real-root


# lists - expand mailing lists stored in a list directory
# mailing lists can be created simply by creating a file in the
# /etc/smail/lists directory.
lists:  driver = forwardfile,
        caution,                        # flag all addresses with caution
        nobody,                         # and then associate the nobody user
        owner = owner-$user;            # system V sites may wish to use
                                        # o-$user, as owner-$user may be
                                        # too long for a 14-char filename.
        file = lists/${lc:user}         # lists is under $smail_lib_dir


# owners - expand mailing lists stored in a list owner directory
# mailing lists owner lists can be created simply by creating a file
# in the /etc/smail/lists/owner directory.  Mailing list owners
# are sent locally generated errors dealing with a mailing list of the
# same name.  To create an owner list for a mailing list, create a
# file with the name of the list in /etc/smail/lists/owner.  This
# will create a list address of owner-listname, as is used by the
# "lists" director above.
owners: driver = forwardfile,
        caution,                        # flag all addresses with caution
        nobody,                         # and then associate the nobody user
        owner = postmaster;             # system V sites may wish to use
                                        # o-$user, as owner-$user may be
                                        # too long for a 14-char filename.
        prefix = "owner-",
        file = lists/owner/${lc:user}   # lists is under $smail_lib_dir


# request - expand mailing lists stored in a list request directory
# mailing lists request lists can be created simply by creating a file
# in the /etc/smail/lists/request directory.  Request addresses
# are typically used as a standard address for queries about a mailing
# list.  For example, requests for additions or deletions to a list
# will generally be sent to "list-request", which should be set up to
# forward to the appropriate person or persons.
request: driver = forwardfile,
        caution,                        # flag all addresses with caution
```

```
        nobody,                         # and then associate the nobody use
        owner = postmaster;             # system V sites may wish to use
                                        # o-$user, as owner-$user may be
                                        # too long for a 14-char filename.
        suffix = "-request",
        file = lists/request/${lc:user} # lists is under $smail_lib_dir
```

You shouldn't need to change anything here, only mailing list options if you intend to run some using smail, or forwards options if, for example, you want to disable forwarding.

6.3 "fidopaths" file

```
.f105.n324.z2.fidonet.org       f105.n324.z2.fidonet.org!%s
.n324.z2.fidonet.org            f105.n324.z2.fidonet.org!%s
.z2.fidonet.org                 f105.n324.z2.fidonet.org!%s
.fidonet.org                    f105.n324.z2.fidonet.org!%s
```

Create such a file only if you're using ifmail and FIDO.

"routers" file

```
# forces - force certain paths
# This database exists as a means of hardcoding the paths to various
# machines or domains.  It is for use in creating temporary tweaks to
# the other routing databases.  To change the database, edit the file
# maps/force.path and type "make" in the maps/ subdirectory.
forces:
        driver = pathalias,             # router to search paths file
        method = /etc/smail/maps/table; # transports are in this file
        file = forcepaths,              # file containing force path info
        proto = lsearch,                # use the sorted path file
        optional,
        reopen                          # close when not being used


uucp_neighbors:
        driver=uuname,                  # use a program which returns neighbors
        transport=uux;
        cmd="/usr/bin/uuname -a",       # specifically, use the uuname program
#        domain=uucp                    # strip ending ".uucp"


# smart_host - a partially specified smarthost director
# If the config file attribute smart_path is defined as a path from the
# local host to a remote host, then hostnames not matched otherwise will
# be sent off to the stated remote host.  The config file attribute
# smart_transport can be used to specify a different transport.
# If the smart_path attribute is not defined, this router is ignored.
smart_host:
        driver = smarthost,             # special-case driver
        transport = uux                 # by default deliver over UUCP
#       path=phreak
```

```
# ifmail - to send mails to fidonet and vice versa
ifmail:
        driver=pathalias,
        transport=ifmail;
        file=fidopaths,
        proto=lsearch
```

You should only include ifmail chapter if you use ifmail for FIDO mails. Note you can also change transport mode from "uux" (ie UUCP) to, for example, "smtp" or even 'hardcode the paths to various machines or domains' in "/etc/smail/maps/table".

This is useful if you want outgoing mail for your local network to be delivered immediately, since there's no need for it to be routed to your uucp connexion of your internet access.

"transports" file

```
# local - deliver mail to local users
# Tell smail to append directly to user mailbox files in the /var/spool/mail
# directory.
#local: driver = appendfile,            # append message to a file
#       -return_path,                   # include a Return-Path: field
#       local,                          # use local forms for delivery
#       from,                           # supply a From_ envelope line
#       unix_from_hack;                 # insert > before From in body
#
#       file = /var/spool/mail/${lc:user},      # use this location for Linux
#                                               # Note, mail spool must be 1777
#       file = ~/mailfile,              # use this location for better security
#       group = mail,                   # group to own file for System V
#       mode = 0660,                    # under System V, group mail can access
#       suffix = "\n",                  # append an extra newline
#       append_as_user,


# This allows each user to have a ~/.procmailrc file to control filtering
# of mail and saving mail from mail lists in separate mailboxes if they wish.
local:  +inet,
        -uucp,
        driver = pipe,                  # append message to a file
        return_path,                    # include a Return-Path: field
        local,                          # use local forms for delivery
        from,                           # supply a From_ envelope line
        unix_from_hack;                 # insert > before From in body

        cmd = "/usr/bin/procmail",      # use procmail for local delivery
        parent_env,                     # environment info from parent addr
        pipe_as_user,                   # use user-id associated with address
        umask = 0022,                   # umask for child process
#       -ignore_status,                 # exit status should be believed
#       -ignore_write_errors,           # retry on broken pipes
```

```
# pipe - deliver mail to shell commands
# This is used implicitly when smail encounters addresses which begin with
# a vertical bar character, such as "|/usr/lib/news/recnews talk.bizarre".
# The vertical bar is removed from the address before being given to the
# transport.
#pipe:  driver = pipe,                  # pipe message to another program
#       return_path, local, from, unix_from_hack;
#
#       cmd = "/bin/sh -c $user",       # send address to the Bourne Shell
#       parent_env,                     # environment info from parent addr
#       pipe_as_user,                   # use user-id associated with address
#       umask = 0022,                   # umask for child process
#       -log_output,                    # do not log stdout/stderr
#       ignore_status,                  # exit status may be bogus, ignore it
#       ignore_write_errors,            # ignore broken pipes


# file - deliver mail to files
# This is used implicitly when smail encounters addresses which begin with
# a slash or squiggle character, such as "/usr/info/list_messages" or
# perhaps "~/Mail/inbox".
#file:  driver = appendfile,
#       return_path, local, from, unix_from_hack;
#
#       file = $user,                   # file is taken from address
#       append_as_user,                 # use user-id associated with address
#       expand_user,                    # expand ~ and $ within address
#       check_path,
#       suffix = "\n",
#       mode = 0644


# uux - deliver to the rmail program on a remote UUCP site
#
# As many as five recipient addresses will be delivered to the remote
# host in one UUCP transaction.
uux:    driver = pipe,
        -uucp,
        inet,
#       uucp,                           # use UUCP-style addressing forms
        from,                           # supply a From_ envelope line
        max_addrs = 5,                  # at most 5 addresses per invocation
        max_chars = 200;                # at most 200 chars of addresses
# the -r flag prevents immediate delivery, parentheses around the
# $user variable prevent special interpretation by uux.
        cmd = "/usr/bin/uux - -r -g$grade $host!rmail $((${strip:user})$)",
#        cmd="/usr/bin/uux - $host!rmail $(($user)$)",
        ignore_write_errors,            # ignore broken pipes
        umask = 0022,
#       pipe_as_sender,


# uux_one_addr - deliver mail over UUCP to a remote host that can take
#                one address at a time.
```

```
# This is often necessary when delivering to a site running an unmodified
# version of 4.1BSD.
uux_one_addr:
        driver = pipe,
        uucp,                           # use UUCP-style addressing forms
        from;                           # supply a From_ envelope line
# the -r flag prevents immediate delivery
        cmd = "/usr/bin/uux - -r -g$grade $host!rmail (${strip:user})",
        umask = 0022,
        pipe_as_sender


queueonly:
        driver = pipe;                  # send the message to a pipe
        cmd = "/usr/lib/sendmail -Q -f $sender -bm $user",
                                        # use getmail for local delivery
        user=root,                      # execute getmail as "root"
        group=mail,                     # execute getmail as "mail"
        parent_env,                     # environment info from parent addr
        -pipe_as_user,                  # use user-id associated with address
        umask = 0007,                   # umask for child process

# to deliver the message.  The smtp transport is included only if BSD
# networking exists.
# The uucp attribute can be specified for transfers within the UUCP
# zone.  The inet attribute must be specified for transfers within the
# Internet.
# NOTE: This is hardly optimal, a backend should exist which can handle
#       multiple messages per connection.
# ALSO: It may be necessary to restrict max_addrs to 100, as this is the
#       lower limit SMTP requires an implementation to handle for one
#       message.
smtp:   driver=tcpsmtp,
        inet,                           # if UUCP_ZONE is not defined
#       uucp,                           # if UUCP_ZONE is defined
        -max_addrs, -max_chars;         # no limit on number of addresses

        short_timeout=5m,               # timeout for short operations
        long_timeout=2h,                # timeout for longer SMTP operations
        service=smtp,                   # connect to this service port
# For internet use: uncomment the below 4 lines
       use_bind,                       # resolve MX and multiple A records
       defnames,                       # use standard domain searching
       defer_no_connect,               # try again if the nameserver is down
       local_mx_okay,                  # fail an MX to the local host


ifmail:
        from,received,max_addrs=5,max_chars=200,
        driver=pipe;
        pipe_as_sender,
        cmd="/usr/local/bin/ifmail -x9 -r$host $((${strip:user})$)"
```

You should include an ifmail chapter only if you use ifmail for FIDO mail. Apart from that, you shouldn't need to edit

anything in this file which defines transport agents (like uux, smtp ...) you can use as parameters in other config. files.

Note I commented out some parts, like "pipes" or "file", to enhance security.

"maps/" directory

It contains `map` and `table` files:

First, `map` file

```
#N      foo.bar foo2.bar2
#S      AT 486/RedHat Linux 1.2.13
#O      organization
#C      contact
#E      administration (email)
#T      phone
#P      address
#R
#U      hosts connected via uucp
#W      created/edited by
#
hname polux

hname linux.eu.org

hname = polux
hname = polux.linux.eu.org
```

Once again, edit this file to match you situation (I'm fed by polux.linux.eu.org).

Now `table` file

```
*       uux
```

You can define different transports to different paths, for exemple "smtp" for the machines in your local network, "uux" (i.e. uucp) for the rest of the world or vice-versa (I'm using uucp for any outgoing mail, therefore I use "*"!).

Other good examples

The previous files are the one I currently use for my site, you shouldn't encounter any problem using them as samples/basis for your own files.

The following files are provided only as good exemples to configure smail a different way.

```
#ident "@(#) transports,v 1.2 1990/10/24 05:20:46 tron Exp"

# See smail(5) for a complete description of the contents of this file.

# local - deliver mail to local users
#
# Tell smail to append directly to user mailbox files in the /usr/mail
# directory.
local:  driver = appendfile,            # append message to a file
        return_path,                    # include a Return-Path: field
```

```
        local,                          # use local forms for delivery
        from,                           # supply a From_ envelope line
        unix_from_hack;                 # insert > before From in body

        file = /usr/mail/${lc:user},    # use this location for System V
        group = mail,                   # group to own file for System V
        mode = 0660,                    # under System V, group mail can access
        suffix = "\n",                  # append an extra newline
        append_as_user,

# pipe - deliver mail to shell commands
#
# This is used implicitly when smail encounters addresses which begin with
# a vertical bar character, such as "|/usr/lib/news/recnews talk.bizarre".
# The vertical bar is removed from the address before being given to the
# transport.
pipe:   driver = pipe,                  # pipe message to another program
        return_path, local, from, unix_from_hack;

        cmd = "/bin/sh -c $user",       # send address to the Bourne Shell
        parent_env,                     # environment info from parent addr
        pipe_as_user,                   # use user-id associated with address
        umask = 0022,                   # umask for child process
        -log_output,                    # do not log stdout/stderr
        ignore_status,                  # exit status may be bogus, ignore it
        ignore_write_errors,            # ignore broken pipes

# file - deliver mail to files
#
# This is used implicitly when smail encounters addresses which begin with a
# slash or squiggle character, such as "/usr/info/list_messages" or perhaps
# "~/Mail/inbox".
file:   driver = appendfile,
        return_path, local, from, unix_from_hack;

        file = $user,                   # file is taken from address
        append_as_user,                 # use user-id associated with address
        expand_user,                    # expand ~ and $ within address
        suffix = "\n",
        mode = 0644

# uux - deliver to the rmail program on a remote UUCP site
#
# As many as five recipient addresses will be delivered to the remote host in
# one UUCP transaction.
uux:    driver = pipe,
        uucp,                           # use UUCP-style addressing forms
        from,                           # supply a From_ envelope line
        max_addrs = 5,                  # at most 5 addresses per invocation
        max_chars = 200;                # at most 200 chars of addresses

        # the -r flag prevents immediate delivery, parentheses around the
        # $user variable prevent special interpretation by uux.
        cmd = "/usr/bin/uux - -r -g$grade $host!rmail $((${strip:user})$)",
```

```
        umask = 0022,
        pipe_as_sender

# uux_one_addr - deliver mail over UUCP to a remote host that can take one
# address at a time.
#
# This is often necessary when delivering to a site running an unmodified
# version of 4.1BSD.
uux_one_addr:
        driver = pipe,
        uucp,                           # use UUCP-style addressing forms
        from;                           # supply a From_ envelope line

        # the -r flag prevents immediate delivery
        cmd = "/usr/bin/uux - -r -g$grade $host!rmail (${strip:user})",
        umask = 0022, pipe_as_sender

# demand - deliver to a remote rmail program, polling on demand
demand: driver = pipe,
        uucp, from, max_addrs = 5, max_chars = 200;

        # with no -r flag, try to contact remote site immediately
        cmd = "/usr/bin/uux - -g$grade $host!rmail $(($user)$)",
        umask = 0022, pipe_as_sender

# uusmtp - deliver to the rsmtp program on a remote UUCP site
#
# Deliver using a simple Batched SMTP protocol to the remote machine.
# This allows much more arbitrary addresses to be used.  It also
# removes the limit on recipient addresses per invocation of uux.
uusmtp: driver = pipe,
        bsmtp,                          # send batched SMTP commands
        -max_addrs,                     # there is no limit on the number or
        -max_chars;                     #   total size of recipient addresses.

        # supply -r to prevent immediate delivery, the recipient addresses
        # are stored in the data sent to the standard input of rsmtp.
        cmd = "/usr/bin/uux - -r -g$grade $host!rsmtp",
        umask = 0022, pipe_as_sender

# demand_uusmtp - deliver to a remote rsmtp program, polling on demand
demand_uusmtp:
        driver = pipe,
        bsmtp, -max_addrs, -max_chars;

        # with no -r flag, try to contact remote site immediately
        cmd = "/usr/bin/uux - -g$grade $host!rsmtp",
        umask = 0022, pipe_as_sender

# smtp - deliver using SMTP over TCP/IP
#
# Connect to a remote host using TCP/IP and initiate an SMTP conversation to
# deliver the message.  The smtp transport is included only if BSD networking
# exists.
```

```
# NOTE: It may be necessary to restrict max_addrs to 100, as this is the
#       lower limit SMTP requires an implementation to handle for one
#       message.
smtp:   driver = smtp,
        -max_addrs,
        -max_chars

#ident "@(#) table,v 1.2 1990/10/24 05:20:31 tron Exp"

# This file names the transports that are to be used in delivering
# to specific hosts from bargw.

#host           transport
#--------       ---------
curdsgw         demand_uusmtp   # deliver using batched SMTP
oldbsd          uux_one_addr    # 4.1BSD sites cannot take more than one addr
sun             demand          # call sun when their is mail to send
*               uux             # for all others, poll at intervals
```

6.4 Restarting inetd

To run smail as a smtp daemon, add one of the following to /etc/inetd.conf:

```
        smtp stream tcp nowait  root  /usr/bin/smtpd smtpd
```

or:

```
        smtp stream tcp nowait  root  /usr/sbin/tcpd  /usr/sbin/in.smtpd
```

Outgoing mail gets sent automatically, when using elm.

6.5 Smail with smtp

Generally, ISPs use smtp, therefore you shouldn't have any problems sending your mail. If your internet link is down when you send mail, then the mail sits in "`/var/spool/smail/input`". When the link next comes up, "`runq`" is run which causes the mail to be sent. However, receiving mail is **the** problem since your provider has many clients to look after, not only you!

Usually, you can retreive your mail via the POP protocol, see POP section below.

6.6 OUTDATED SECTION: Sendmail+IDA

For big sites, sendmail is worth choosing, due to the "incredible ease of use", (very relative feeling when you know qmail) but you must decide which you want between sendmail+IDA and sendmail 8.x:

- If you use an old kernel (1.0): sendmail+IDA
- If you use a not so old kernel (1.2): sendmail+IDA and source code editing
- Recent kernel (2.0) will choose sendmail 8.x

Remember, linux newbies or people concerned by security / ease of configuration should rather try using smail or qmail, which are easier to use and safer.

Source installation

If your distribution doesn't provide you with a ready-to-install sendmail package (.rpm for RedHat, Caldera and Suse, .deb for Debian) just download the sources and run:

- cd / ; tar -zxvf sendmail5.67b+IDA1.5.tpz
- cd to /usr/local/lib/mail/CF and copy the sample.m4 local.m4 file to "yourhostname.m4".

Edit out the distributed hostname, aliases, smarthost and put in the correct one for your site. The default file is for a uucp-only site who has domainized headers and who talks to a smart host. Then "make yourhostname.cf" and move the resulting file to /etc/sendmail.cf

- if you are uucp-only, you do *NOT* need to create any of the tables mentioned in the README.linux file.

You'll just have to touch the files so that the Makefile works. Just edit the .m4 file, make sendmail.cf and start testing it.

- if you're uucp-only and you talk to sites in addition to your "smart-host", you'll need to add uucpxtable entries for each (or mail to them will also go through the smart host) and run dbm against the revised uucpxtable.
- If you run Rich Braun's original binary distribution of 5.67a, you'll need to freeze the configuration if you change your .cf file with "/usr/lib/sendmail -bz" to make the changes take effect.

You should also update your version to at least 5.67b since there is a nasty security hole in 5.67a and earlier. Another nice thing is that if you have mail.debug set and you run syslogd, your incoming and outgoing mail messages will get logged. See the "/etc/syslog.conf" file for details.

The sources for sendmail+IDA can be found at vixen.cso.uiuc.edu ; they require no patching to run under Linux if you're running something like a kernel of 1.00.

If you're running a kernel > 1.1.50, you get the fun of reversing most of the Linux-specific patches that are now in the vanilla sources. (I *did* told you this sendmail was only for old kernels:-)

It's extremely obvious where this needs to be done: just type "make" and when it blows up, go to that line in the sources and comment out the Linux-specific code that's in there.

If you're going to run sendmail+IDA, I strongly recommend you go to the sendmail5.67b+IDA1.5 version since all required Linux-specific patches are now in the vanilla sources and several security holes have been plugged that WERE (!!!) in the older version you may have grabbed or built before about December 1st, 1993.

Now linux kernel is 2.0, you should use sendmail 8.x insted of sendmail+IDA, but I told you'd better choose sendmail 8.x:-)

The sendmail.m4 file

Sendmail+IDA requires you to set up a `sendmail.m4` file rather than editing the `sendmail.cf`file directly. The nice thing about this is that it is simple to set up mail configurations that are extremely difficult (if not totally impossible for most people to set up correctly) in smail or traditional sendmail.

The sendmail.m4 file that corresponds to the above smail example looks like the following:

```
dnl #------------------ SAMPLE SENDMAIL.M4 FILE ------------------
dnl #
dnl # (the string 'dnl' is the m4 equivalent of commenting out a line)
dnl #
dnl # you generally don't want to override LIBDIR from the compiled in paths
dnl #define(LIBDIR,/usr/local/lib/mail)dnl    # where all support files go
define(LOCAL_MAILER_DEF, mailers.linux)dnl    # mailer for local delivery
define(POSTMASTERBOUNCE)dnl                   # postmaster gets bounces
define(PSEUDODOMAINS, BITNET UUCP)dnl         # don't try DNS on these
dnl #
```

TOC

```
dnl #-------------------------------------------------------------
dnl #
dnl # names we're known by
define(PSEUDONYMS, myhostname.subdomain.domain myhostname.UUCP)
dnl #
dnl # our primary name
define(HOSTNAME, myhostname.subdomain.domain)
dnl #
dnl # our uucp name
define(UUCPNAME, myhostname)dnl
dnl #
dnl #-------------------------------------------------------------
dnl #
define(UUCPNODES, |uuname|sort|uniq)dnl       # our uucp neighbors
define(BANGIMPLIESUUCP)dnl                    # make certain that uucp
define(BANGONLYUUCP)dnl                       #  mail is treated correctly
define(RELAY_HOST, my_uucp_neighbor)dnl       # our smart relay host
define(RELAY_MAILER, UUCP-A)dnl               # we reach moria via uucp
dnl #
dnl #-----------------------------------------------------------------
dnl #
dnl # the various dbm lookup tables
dnl #
define(ALIASES, LIBDIR/aliases)dnl            # system aliases
define(DOMAINTABLE, LIBDIR/domaintable)dnl    # domainize hosts
define(PATHTABLE, LIBDIR/pathtable)dnl        # paths database
define(GENERICFROM, LIBDIR/generics)dnl       # generic from addresses
define(MAILERTABLE, LIBDIR/mailertable)dnl    # mailers per host or domain
define(UUCPXTABLE, LIBDIR/uucpxtable)dnl      # paths to hosts we feed
define(UUCPRELAYS, LIBDIR/uucprelays)dnl      # short-circuit paths
dnl #
dnl #-----------------------------------------------------------------
dnl #
dnl # include the 'real' code that makes it all work
dnl # (provided with the source code)
dnl #
include(Sendmail.mc)dnl                         # REQUIRED ENTRY!!!
dnl #
dnl #------------ END OF SAMPLE SENDMAIL.M4 FILE -------
```

Defining a local mailer

Unlike most Unix distributions, Linux does not come with a local mail delivery agent by default. Now, deliver or procmail is generally installed, so no complexity will be added to this already very complex setup. I recommend using the commonly available `deliver` or `procmail` programs, which can be optional packages in a some Linux distributions.

In order to do so, you need to define a `LOCAL_MAILER_DEF` in the `sendmail.m4` file that points to a file that looks like:

```
# -- /usr/local/lib/mail/mailers.linux --
#     (local mailers for use on Linux )
Mlocal, P=/usr/bin/deliver, F=SlsmFDMP, S=10, R=25/10, A=deliver $u
Mprog,  P=/bin/sh,       F=lsDFMeuP,   S=10, R=10, A=sh -c $u
```

TOC

There is a also built-in default for `deliver` in the `Sendmail.mc` file that gets included into the `sendmail.cf` fil specify it, you would not use the `mailers.linux` file but would instead define the following in your sendmail.m4 f

```
dnl --- (in sendmail.m4) ---
define(LOCAL_MAILER_DEF, DELIVER)dnl       # mailer for local delivery
```

Unfortunately, Sendmail.mc assumes deliver is installed in /bin, which is not the case with Slackware1.1.1 (which installs it in /usr/bin). In that case you'd need to either fake it with a link or rebuild deliver from sources so that it resides in /bin. Please note procmail is generally better than deliver, for example for mail filtering.

The sendmail+IDA dbm tables

Setting up special behavior for sites or domains is done through a number of optional `dbm` tables rather than editing the `sendmail.cf` file directly.

Refer to the July-1994 issue of `Linux Journal` (if you can still find it:-), to the docs in the sources, or to the sendmail chapter in the newest version of the Linux Documentation Project `Networking Administration Guide` which will be available real-soon-now for more details.

- mailertable - defines special behavior for remote hosts or domains.
- uucpxtable - forces UUCP delivery of mail to hosts that are in DNS format.
- pathtable - defines UUCP bang-paths to remote hosts or domains.
- uucprelays - short-circuits the pathalias path to well-known remote hosts.
- genericfrom - converts internal addresses into generic ones visible to the outside world.
- xaliases - converts generic addresses to/from valid internal ones.
- decnetxtable - converts RFC-822 addresses to DECnet-style addresses.

So which entries are really required?

When not using any of the optional dbm tables, sendmail delivers mail via the `RELAY_HOST` and `RELAY_MAILER)` defined in the sendmail.m4 file used to generate sendmail.cf. It is easily possible to override this behavior through entries in the domaintable or uucpxtable.

A generic site that is on Internet and speaks Domain Name Service, or one that is UUCP-only and forwards all mail via UUCP through a smart `RELAY_HOST`, probably does not need any specific table entries at all.

Virtually all systems should set the `DEFAULT_HOST` and `PSEUDONYMS` macros, which define the canonical site name and aliases it is known by.

If all you have is a relay host and relay mailer, you don't need to set these defaults since it works automagically. UUCP hosts will probably also need to set `UUCPNAME` to their official UUCP name.

They will also probably set `RELAY_MAILER` and `RELAY_HOST` which enable smart-host routing through a mail relay.

The mail transport to be used is defined in `RELAY_MAILER` and should usually be UUCP-A for UUCP sites. If your site is SMTP-only and talks `Domain Name Service', you would change the `RELAY_MAILER`.

If you're a SLIP site, you might want to take the easy way out and just forward all outgoing mail to your service provider to do the right thing with. To do so, you'd want to define ISOLATED_DOMAINS and VALIDATION_DOMAINS to be your domain, you'd also want to define RELAY_HOST to be your service provider and RELAY_MAILER to be TCP. Of course, you want to ask permission before you set any system up as your general purpose relay.

6.7 Sendmail 8.x

Sendmail 8.7.x from Berkeley was the latest major revision after sendmail5. It had wonderful built-in support for building under Linux : just "make linux" and all was set.

You'll probably be best served by grabbing one of the various binary distributions off of the usual Linux archive sites rather than fighting things like Berkeley dbm yourself.

There's a nice distribution of sendmail 8.6.12 from Jason Haar - `j.haar at lazerjem.demon.co.uk` on sunsite.unc.edu in /pub/Linux/system/Mail/delivery/sendmail-8.6.12-bin.tgz that has the source documentation and a very nice quickie description of how to run sendmail v8 for common configurations.

The bottom line with sendmail v8 is that you want to configure the bare minimum necessary to get the job done ; the following is an example that should get you close at least.

A sample 8.7.x mc file

Much like sendmail+IDA, sendmail v8 uses m4 to process a config file into a full sendmail.cf that sendmail uses. The following is my current mc file for my site (ppp to Internet for outgoing mail, uucp for incoming mail).

```
dnl divert(-1)
#-----------------------------------------------------------------------
#
# this is the .mc file for a linux host that's set up as follows:
#
#       - connected to Internet for outbound mail (ppp here)
#       - connected via UUCP for incoming mail
#       - domainized headers
#       - no local mailer (use 'deliver' instead)
#       - no DNS running so don't canonicalize outgoing via DNS
#       - all non-local outbound mail goes to the RELAY_HOST over smtp
#           (we run ppp and let our service provider do the work)
#
#                                       vds 3/31/95
#
#-----------------------------------------------------------------------
include(`../m4/cf.m4')
VERSIONID(`linux nodns relays to slip service provider smarthost')dnl
Cwmyhostname.myprimary.domain myhostname.UUCP localhost
OSTYPE(linux)
FEATURE(nodns)dnl
FEATURE(always_add_domain)dnl
FEATURE(redirect)
FEATURE(nocanonify)
dnl MAILER(local)dnl
MAILER(smtp)dnl
MAILER(uucp)dnl
define(`RELAY_HOST', smtp:my.relay.host.domain)
define(`SMART_HOST', smtp:my.relay.host.domain)
define(`UUCP_RELAY', smtp:my.relay.host.domain)
define(`LOCAL_MAILER_PATH', `/bin/deliver')
define(`LOCAL_MAILER_ARGS', `deliver $u')
```

6.8 Sendmail v8 tidbits

There are a few differences I suppose to the 'IDA bigots' among us. So far, I've found the following:

Instead of 'runq', you type 'sendmail -q' to run the queue!

6.9 Local Delivery Agents

Unlike most operating systems, Linux did not have mail "built-in": you needed a program to deliver the local mail, like "lmail", "procmail" or "deliver".

However, every recent distribution includes a local mailer now!

Documentation for how to use either for local delivery is in the sendmail5.67b+IDA1.5 binary release (on sunsite) mentioned above.

---

7. User Agent Administration

7.1 Mutt

You should have no problem compiling, installing, or running mutt. Users of qmail can either get the patch or run it with -f flag to read their local mail folder.

If mutt bothers you with an "unknown terminal error" after a distribution upgrading, just recompile it.

7.2 Elm

Elm compiles, installs and runs flawlessly under Linux. For more information, see the elm sources and installation instructions. Elm and filter need to be mode 2755 (group mail) with /var/spool/mail mode 775 and group mail.

Qmail users can get a patch to use nifty qmail features, or will run elm with the -f flag to point to their local mail folder.

One thing you want to be aware of is that if you have Elm compiled to be MIME-able, you need metamail installed and in the standard path or Elm will not be able to read MIME mail you've received. Metamail is available on `thumper.bellcore.com` and of course via "archie".

If you use a binary distribution, you'll need to create a "/usr/local/lib/elm/elm.rc" file to override the compiled-in hostname and domain information:

- replace "subdomain.domain" with your domain name replace
- "myhostname" with you un-domainized hostname replace

```
#---------- /usr/local/lib/elm/elm.rc ------------------
#
# this is the unqualified hostname
hostname = myhostname
#
# this is the local domain
hostdomain = subdomain.domain
#
# this is the fully qualified hostname
hostfullname = myhostname.subdomain.domain
#
#-------------------------------------------------------
```

One thing you want to be aware of is that if you have Elm compiled to be MIME enabled, you need metamail installed and in your path or Elm will not be able to read MIME mail you've received.

Metamail is available on `thumper.bellcore.com` and of course via "archie".

In the "too cool to be true" category, there is a distribution of Elm-2.4.24 that is "PGP-aware". To try it, grab the file `ftp://ftp.viewlogic.com/pub/elm-2.4pl24pgp3.tar.gz`, which is elm2.4.24 with PGP hooks added. You configure and build it the same way you do normal Elm, which means you probably need to add the patches mentioned above. For what it's worth, I run it here and like it a lot. Of course, there must be more recent versions available, including elm-ME+.

While this item is not Linux-specific, it's perceived (wrongly) to be a nagging Elm bug nevertheless. We've heard that Elm sometimes fails with a message that it's unable to malloc() some massive number of bytes. The identified workaround is to remove the post-processed global mail aliases (aliases.dir and aliases.pag).

`THIS IS NOT A BUG IN ELM`, it's an error in configuration of Elm by whomever you got your binary distribution of Elm from.

Elm has an enhanced and non-compatible, format for aliases ; you need to ensure that the path Elm uses for aliases is different from the path sendmail/smail uses. From the volume of reports of this problem, it's apparent that at least one major distribution 'on the street' has in the past been misconfigured. (from `scot at catzen.gun.de (Scot W. Stevenson) )`

The current metamail package requires csh for some of its scripts. Failure to have csh (or tcsh) will cause most interesting errors...

7.3 Mailx

If you don't have a local `mailx` program, save yourself the pain -- just go and grab the mailx kit from Slackware 2.1.0 or later, which has a nice implementation of mailx5.5. If you're into building from sources, mailx v5.5 compiles without patching under Linux if you have `"pmake"` installed.

If anybody is still using it, I strongly recommend removing the old "edmail" stuff from SLS1.00 and replacing it with mailx.

---

8. Handling remote mail

This section describes using POP or IMAP to handle remote mail.

Other options include nfs-mounting the spool partition on client machines (Danger Will Robinson! Is everyone using the same lock method?) or using a mail-to-web gateway (quite popular now).

8.1 History

On a workstation network, mail has always been a problem:

- Either you use "user@computer.foo.com" with problems when "computer" is down, making your network known to the people outside, having different addresses for a same person switching to another computer, ...
- Either you take a mail hub, "mailhost.foo.com" with rules for rewriting, so every user seems to post from the same address, even if they are on different computers.

But in that case, how can user read their mail?

Using a rsh with elm? :-)

It would overload our mail hub! One method was forwarding or UUCP, smtp, etc. but it's too complicated.

Then came POP/IMAP, both with security problems at the beginning, (now fixed using ssh on new versions): a mail program has sometimes to be set locally (like qmail, smail or vmail if, for example, you use elm, but mozilla will avoid that!) however, getting and sending Email is simpler.

8.2 Getting mail

Here come POP's main drawbacks:

- the password is sent as a clear text on the network,
- you must choose a POP-aware mailer; many do now (like Pine, Emacs, Mozilla, Netscape, Mutt, IE, Pegasus, Eudora, Claris...),
- when a user may roam (read mail from different machines) having e-mail popped on the computer used yesterday can be a nuisance,
- some POP servers (e.g. qpopper, ipop3d) on high-use servers can load the machine significantly. Consider controlling options (such as not leaving mail on the server) and/or changing the pop server (e.g. cucipop), as well as avoiding running it from inetd.

The password problem can be solved creating a crypted "channel" to have POP on it or using APOP or RPOP extensions. The mail reader problem can be solved either by changing mail reader (don't underestimate the effort required to re-educate users!) or by using a POP "mail sucker" with a local mail program.

IMAP can be preferable to POP in various situations like remote (and especially roaming) access, while you restrict POP to a LAN where snooping of passwords isn't so much of a concern. Mark Aitchison reported a solution here is to use hosts.deny and hosts.allow files (please see Net-3 HOWTO ; this assumes you are starting pop from inet).

The policy of leaving mail on the server or not has implications for server disk space and easier backup/security of the mail, as well as allowing roaming, so the best solution depends on the type of organisation. Of course, this will not ensure your mail can't be read, but nobody will be able to delete it ; if all your mail is pgp encrypted this is a better solution.

Here are some pop programs worth trying:

- gwpop (a Good Way to POP) is very protected since it creates a crypted "channel" and puts mail directly in the "spool" ; however, it depends on Perl.

- popclient, simple to use: For example if your login is john and your password PrettySecret, you will run:

```
$ popclient -3 -v mail.acme.net -u john -p "PrettySecret" -k -o JOHN-INET-MAIL
```

- fetchmail, which is actively supported and incredibly simple to use: it is configured in `~/.fetchmailrc`, so you only need to run `fetchmail` when you want to retrive your mail. Here's my .fetchmailrc:

```
poll mail.server protocol pop3:
        forcecr
        password PrettySecret;
```

  Please note that the forcecr option is needed to use fetchmail with qmail, which strictly respects RFCs.

8.3 Sending mail

For this, you must use smtp-aware mail software, like qmail, smail, vmail or mozilla (this one does everything: mail reader, POP receive, smtp send!)

Go to one of the previous sections to install and configure the one you like best. Then, when you will reach "Testing", try to send some mail to a local account on the mail hub.

8.4 Reading mail

If your program doesn't do everything itself, you can install elm, pgp, mush, pine ... many good programs are freely available for linux platforms!

8.5 Testing

To check whether your mail server has pop, try:

```
$ telnet mailhost 110
```

If it works, you will get something like "OK Pop server (...) starting": type "quit"!

To install a ssh crypted "channel", first test your mail server typing:

```
$ ssh mailhost date
```

If you get the date, you should be OK. Please note ssh will not ask for a password, therefore you must create a ".shosts" file on the mail server, containing client's name. To test ssh port redirection (which gwpop uses), type:

```
$ ssh -n -f -L 12314:localhost:110 mailhost sleep 30

then

$ telnet localhost 12314
```

Then will you hopefully see mail hub's pop banner. If you don't use ssh, don't forget to comment out $ssh on gwpop script. To check whether procmail is running, try "procmail -v"

8.6 Using

Now you can edit gwpop Perl script to check everything is ok, then run gwpop:

```
$ gwpop -v your-username
```

```
POP password on mailhost: yoursecretpasword
```

If gwpop "error messages" are normal, the mail from mail hub will be downloaded to your local machine wherever you to d gwpop to put it. (please test with some mail!).

You can also use gwpop as a daemon:

```
$ gwpop -d $HOME/tmp your-username
```

gwpop messages are then sent to syslog and gwpop will run endlessly ; a "HUP" signal will force gwpop to get your mail.

You can get POP software here used on:

```
ftp://ftp.unina.it/pub/Unix/pkgs/network/mail/gwpop
ftp://ftp.informatik.rwth-aachen.de/pub/packages/procmail
http://www.cs.hut.fi/ssh/
```

9. Acknowledgements

The following people have helped in the assembly of the information and experience that helped make this document possible:

Steve Robbins, Ian Kluft, Rich Braun, Ian Jackson, Syd Weinstein, Ralf Sauther, Martin White, Matt Welsh, Ralph Sims, Phil Hughes, Scot Stevenson, Neil Parker, Stephane Bortzmayer and especially many thanks to Vince Skahan for his huge contribution.

Eric S. Raymond edited this document, correcting various mistakes and transplanting the section on ``How Electronic Mail Works'' from his Mail User's HOWTO.

If I forgot anybody, my apologies: just email me!

The Linux Mail User HOWTO

Eric S. Raymond, <esr@thyrsus.com>

v2.2, 07 May 1999

This document is an introduction to the world of electronic mail (email) under Linux. It focuses on user-level issues and typical configurations for Linux home and small-business machines connected to the net via an ISP. You need to read this if you plan to communicate locally or to remote sites via electronic mail. You probably do not need to read this document if don't exchange electronic mail with other users on your system or with other sites. For information on configuring and administering mail, see the Mail Administrator HOWTO.

1. Introduction

- 1.1 New versions of this document
- 1.2 Hardware requirements for email programs
- 1.3 Software sources for email programs

2. Mail User Agents

- 2.1 Setting your mail editor
- 2.2 mutt
- 2.3 elm
- 2.4 pine
- 2.5 Netscape
- 2.6 Emacs rmail/smail and vm.
- 2.7 BSD mail
- 2.8 Other user agents

TOC

3. Advanced topics

- 3.1 Aliases
- 3.2 Forwarding
- 3.3 Auto-answering
- 3.4 Mailing lists
- 3.5 Mail filters
- 3.6 Coping with spam

4. Other sources of information

- 4.1 USENET
- 4.2 Books
- 4.3 Periodic USENET Postings
- 4.4 Where *NOT* to look for help

5. Administrivia

- 5.1 Feedback
- 5.2 Copyright Information
- 5.3 Standard Disclaimer
- 5.4 Acknowledgements

---

1. Introduction

The intent of this document is to explain how email works, and answer some of the questions that appear to meet the definition of `frequently asked questions' about e-mail software under Linux.

Modern Linux distributions give you a usable, preconfigured setup for electronic mail out of the box, usually featuring a late version of sendmail-v8. This HOWTO will assume that you have such a setup and a working Internet connection.

(For information on how to set up a PPP or SLIP link to an ISP, see the ISP Hookup HOWTO.)

Accordingly, unlike Vince Skahan's 1.x versions, this HOWTO focuses on user issues and architecture; most technical hair about UUCP, IDA sendmail and other formerly important topics has been dropped.

1.1 New versions of this document

This document will be posted monthly to the newsgroup comp.os.linux.answers You should also be able to view the latest version of this HOWTO on the World Wide Web at http://metalab.unc.edu/LDP/HOWTO/Mail-User-HOWTO.html.

1.2 Hardware requirements for email programs

There are no specific hardware requirements for mail under Linux. If you have the hardware necessary to connect to the Internet, it can support email over that link.

1.3 Software sources for email programs

The software you will need for email support is probably preinstalled in your Linux distribution. You will find updates on the Metalab Linux Archive, especially in the mail subdirectory.

2. Mail User Agents

This section contains information related to user agents, which means the software the user sees and uses. This software relies on the transport agents described in the Mail Administrator's HOWTO (which also include user=agent configuration and troubleshooting tips for administrators).

2.1 Setting your mail editor

Mail user agents call out to some editor to assist composition of mail. Which editor is the default varies. Most of them respect a convention going back to Unix's early days; the contents of the environment variable VISUAL, if it exists, is taken as the name of your preferred editor. If VISUAL is not set, the variable EDITOR is checked.

Popular values for EDITOR include `vi' and `emacs'. But if you are, like me, the sort who always has an Emacs running, the most useful way to set EDITOR is to the value `emacsclient'. Use this with the following lines in your .emacs file:

```
(autoload 'server-edit "server" nil t)
(server-edit)
```

The emacsclient program, when it runs, tries to establish communication with an Emacs instance you already have running and hand the mail message temporary file to that Emacs to be edited. The effect of this will be that when your mailer calls out for an editor, a mail composition window pops open inside your Emacs.

When you are ready to hand the file back to the mailer for sending, type C-x #. The mail buffer will leave your display and the emacsclient instance your mailer called will return, handing control back to the mailer.

It is possible to have more than one emacsclient instance open at once without confusing Emacs. However, calling up another Emacs while an emacsclient session is running can confuse emacsclient enough that it won't be able to find either instance afterwards. If this happens, shut down all your Emacs instances and restart just one.

2.2 mutt

This is what I use and recommend. It is descended from elm and has similar commands by default, but is much more powerful and configurable. It can be a POP3 or IMAP client, and includes excellent support for MIME and PGP. There is a Mutt home page at http://www.mutt.org.

Mutt respects the EDITOR/VISUAL convention.

2.3 elm

Elm was the first modern, screen-oriented Unix mailer, but has been stagnant for years now and is being displaced by Mutt. Some versions of elm have POP3 support built in. For more information, see the elm sources and installation instructions in the Metalab mail user agents directory. Here are a few points that occasionally trip people up:

No, stock elm is not PGP-aware. There are PGP support patches, but Mutt's PGP support is superior. If you want to use PGP, I recommend Mutt.

Elm respects the EDITOR/VISUAL convention.

2.4 pine

Pine is a user agent designed for novices; it includes news-reading capability and built-in support for the IMAP remote-mail protocol. A lot of people swear by it for new users. I find its impoverished command set, limited configurability and native editor hard to take. It has excellent built-in IMAP support, however. If you want to check it out, the distribution is available at http://www.washington.edu/pine.

Pine respects the EDITOR/VISUAL convention.

2.5 Netscape

The Netscape browser has POP3 and IMAP remote-mail capability built into it, so it can be used as a mail user agent. I don't recommend this; it doesn't specialize in being an MUA, and therefore does not offer many of the services that real MUAs do (such as aliases and PGP handling).

Neetscape supplies its own mini-editor, the same one used throughout the browser (e.g. for text fields in forms).

2.6 Emacs rmail/smail and vm.

Emacs has a mode called smail that can send mail, and another called rmail that can read mail. The smail mode can be quite useful, as you get to compose mail inside a full Emacs environment (but see also the discussion of emacsclient elsewhere in this document).

The rmail mode, on the other hand, is not recommended. Every time you run it, it converts your inbox to BABYL format; ordinary mail tools will choke on that. (If this happens to you, do `M-x unrmail` from the Emacs command line.)

There is a mailreader for emacs called `vm' that writes and reads standard V7 mailboxes. It is not distributed with GNU Emacs, but you can find its home page at http://www.wonderworks.com/vm/.

TOC

Emacs smail/rmail/vm do not respect the EDITOR/VISUAL convention. Instead, you use the Emacs they're embedded in.

2.7 BSD mail

If you simply type `mail' to the shell on a Linux or any other modern Unix, you will invoke some variant of the BSD Mail program. It has a line-oriented interface originally designed for use on TTYs. It is, at this point, only of historical interest.

BSD Mail invented the EDITOR/VISUAL convention.

2.8 Other user agents

The following also are known to run under Linux. Consult `archie' to find them...

- mush - mail user's shell, very powerful for filtering and batch processing
- mh - mail handler, yet another mail user agent

I don't know enough about mh or mush to describe them in detail. They both have rather complex interfaces and are designed for sophisticated mail users.

---

3. Advanced topics

3.1 Aliases

An `alias' is a way to set up a pseudo-address that simply directs mail to another (single) address. There are two kinds of aliases: MUA aliases and MTA aliases.

An MUA alias is one you set up in your MUA as a kind of personal shorthand. Other people will not be able to see or use this alias. For example, you could write

```
alias esr       Eric S. Raymond <esr@thyrsus.com>
```

in your mutt configuration file. This would tell mutt that when it sees `esr' in an address line, it should behave as through you had typed `esr@thyrsus.com', Or you can type `mutt esr' and the expanded address will be automatically filled in on the `to' line.

An MTA alias is one your MTA expands; it will be usable by everyone, both on your machine and remotely. To create MTA aliases you must modify a system file, usually but not always /etc/aliases (the location depends on your MTA). It may be instructive for you to look at the /etc/aliases on your system; it should contain a number of standard aliases such as `postmaster'.

Your MTA may also allow the target of an alias to be a filename, which will be treated as a mailbox the mail is to be appended to (this is useful for archiving mail). It may also allow the target of an alias to be a program, in which case mail to that alias will be passed to an instance of the program on its standard input.

3.2 Forwarding

MTA aliases usually require administrator privileges to set up. But it is desirable for mail users to be able to set up forwarding of their own mail without administrator intervention.

To support this, most MTAs follow sendmail's lead and look for a file called `.forward` in your home directory. The contents of this file is interpreted like the target of an alias which should receive all your mail. The most common use for this facility is to redirect your mail to an account on another machine.

3.3 Auto-answering

Another common use for the `.forward` facility is to pass your mail to a `vacation' program. A vacation program reads incoming mail and automatically generates a canned reply to it; they are so called because the most common form of canned reply is to inform the sender that you are on vacation and will not be

reachable until a given date.

There is no one standard vacation program that is in universal use. There are two good reasons for this: one, that such a program is very easy to write as a shellscript of filter rule (see below); and two, that vacation programs interact badly with mailing lists.

You should temporarily unsubscribe from all mailing lists you are on before setting up auto-answering; otherwise, all members of the mailing lists mail find they are being flooded with canned messages by your vacation program. This is considered very rude behavior and will guarantee you quite a frosty reception on your return.

3.4 Mailing lists

A mailing list is a pseudo-address that sends mail to more than one user.

In its simplest form, mailing list is just an MTA alias with more than one recipient. Some small mailing lists are maintained this way. Sendmail assists by supporting a syntax in /etc/aliases that includes the contents of a given mailing list file in the target side of an alias. It looks like this:

```
admin-list:     ":include:/usr/home/admin/admin-list"
```

---

with the advantage that the admin-list file can live in unprivileged-user space somewhere (root is only needed to set up the original inclusion). Some other MTAs have similar features.

These simple lists are commonly called `mail reflectors'. There are a couple of problems with mail reflectors. One is that bounce messages from failed attempts to broadcast goes to all users. Another is that all subscriptions and unsubscriptions have to be done manually by the mailing list administrator.

A kind of software called a mailing list manager has evolved to address these problems and other related ones. Its most important function is to permit mailing list users to subscribe and unscubscribe without going through the list maintainer.

A mailing-list manager keeps its own user-list information and hooks up to the MTA through a program alias in /etc/aliases. For example, if the admin-list above were going through the mailing list manager called SmartList on a sendmail system, a portion of /etc/aliases might look like this:

```
admin-list: "|/usr/home/smartlist/bin/flist admin-list"
admin-list-request: "|/usr/home/smartlist/bin/flist admin-list-request"
```

Note that this is a pair of aliases. It is conventional for real mailing lists to have a request address to be used for user subscription and unsubscription requests. It is considered rude and ignorant to send subscription/unsubscription requests to the main address of such a list -- don't do it.

The robot sitting behind the request address may offer other features besides just

subscription/unsubscription. It may respond to help requests, allow you to query who is on the list, or you automated access to list archives. It may also allow list administrators to restrict posting to know members, set the list to auto-subscribe nonmembers when they first post, or set various security policy options. Mailing-list managers differ primarily in the design and range of these secondary features.

Unfortunately, the format for sending commands to mailing-list request robots is not standard. Some expect commands in the subject line, some ignore the subject line and expect commands in the message body. You need to pay attention to the response mail you get when you first subscribe; it's a good idea to save such mail to a subscriptions mailbox for later reference.

The most important mailing-list managers to know about are majordomo, listserv, listproc, and smartlist; majordomo is the most popular by a considerable margin. There is a rather comprehensive list of such packages on the Web.

For more about mailing list managers, consult the resources at the List-Managers Mailing List, including the FAQ (note: this list is *not* appropriate for how-to questions).

3.5 Mail filters

A mail filter is a program that sits between your local delivery agent and you, automatically dispatching or rejecting mail before you see it.

Mail filters have a number of uses. The most important are spam filtering, dispatching to multiple mailboxes by topic or sender, and auto-answering mail.

Typically, you set up mail filtering by putting a program alias for the filter program in your .forward file, and writing a file of filtering rules. The format and location of the filter rules file varies between filter programs.

There are good feature summaries of the three major mail filters (procmail, mailagent, and deliver) in part 3 of Chris Lewis's Email Software Survey. The most popular of these is (despite its rather nasty rule syntax) procmail, which is universally present on Linux systems (and, indeed, is generally used as the system's local delivery agent).

3.6 Coping with spam

Spam is sometimes known as `UCE' (Unsolicited Commercial Email) or `UBE' (Unsolicited Bulk Email). As these names imply, it is an obnoxious form of advertising that stuffs your mailbox with form letters. (The term `spam' comes from a Monty Python's Flying Circus skit in which a choir of Vikings endlessly repeats the chant "Spam spam spam spam...").

Most spam seems to consist of solicitations for pyramid schemes, ads for pornography, or (annoyingly) attempts to sell spam-sending programs. A few individual spams (like MAKE MONEY FAST or the Craig Shergold postcard hoax) have been so persistent as to become legendary. Spam tends to be both verbose and illiterate. It's a waste of time and a huge waste of network bandwidth.

The spam epidemic seems to have peaked in mid-1997 and been slowly in decline since, but it can still be

a serious annoyance. If you're being deluged with spam, get educated. Browse the Fight Spam on the Internet! page. The Death To Spam! page is particularly effective on methods for stopping or backtracking spam.

4. Other sources of information

4.1 USENET

There are a number of Usenet groups devoted to electronic-mail technical issues:

- comp.mail.elm the ELM mail system.
- comp.mail.mh The Rand Message Handling system.
- comp.mail.mime Multipurpose Internet Mail Extensions.
- comp.mail.misc General discussions about computer mail.
- comp.mail.multi-media Multimedia Mail.
- comp.mail.mush The Mail User's Shell (MUSH).
- comp.mail.sendmail the BSD sendmail agent.
- comp.mail.smail the smail mail agent.
- comp.mail.uucp Mail in the uucp environment.

4.2 Books

The following is a non-inclusive set of books that will help...

- ``*Sendmail*'' from O'Reilly and Associates is the definitive reference on sendmail-v8 and sendmail+IDA. It's a ``must have'' for anybody hoping to make sense out of sendmail without bleeding in the process.
- "*The Internet Complete Reference*" from Osborne is a fine reference book that explains the various services available on Internet and is a great source for information on news, mail, and various other Internet resources.
- "*The Linux Networking Administrators' Guide*" from Olaf Kirch of the LDP is available on the net and is also published by (at least) O'Reilly and SSC. It makes a fine one-stop shopping guide to learn about everything you ever imagined you'd need to know about Unix networking.

4.3 Periodic USENET Postings

Also worth mentioning, is Chris Lewis' periodic posting on unix e-mail software, which is available on ftp://rtfm.mit.edu/pub/usenet/comp.mail.misc as the files named ``UNIX_Email_Software_Survey_*''. An HTMLized version is at http://www.faqs.org/faqs/mail/setup/unix/. At time writing in 1999 this posting has not been seriously updated since 1996, however.

4.4 Where *NOT* to look for help

TOC

There is no longer anything special about configuring and running mail under Linux, relative to other Unixes. Accordingly, you almost certainly do *NOT* want to be posting generic mail-related questions to the comp.os.linux.* newsgroups.

Unless your posting is truly Linux-specific (ie, ``please tell me what routers are already compiled into the SLS1.03 version of smail3.1.28") you should be asking your questions in one of the newsgroups or mailing lists referenced above.

Let me repeat that.

There is virtually no reason to post anything mail-related in the comp.os.linux hierarchy any more. There are existing newsgroups in the comp.mail.* hierarchy to handle *ALL* your questions.

If you post to comp.os.linux. for non-Linux-specific questions, you are looking in the wrong place for help. The electronic mail experts hang out in the places indicated above and generally not in the Linux groups.*

Posting to the Linux hierarchy for non-linux-specific questions wastes your time and everybody else's...and it frequently delays you from getting the answer to your question.

---

5. Administrivia

5.1 Feedback

(Vince wrote this section, but my policy is the same.)

I am interested in any feedback, positive or negative, regarding the content of this document via e-mail. Definitely contact me if you find errors or obvious omissions.

I read, but do not necessarily respond to, all e-mail I receive. Requests for enhancements will be considered and acted upon based on that day's combination of available time, merit of the request, and daily blood pressure :-)

Flames will quietly go to /dev/null so don't bother.

In particular, the Linux filesystem standard for pathnames is an evolving thing. What's in this document is there for illustration only based on the current standard at the time that part of the document was written and in the paths used in the distributions or `kits' I've personally seen. Please consult your particular Linux distribution(s) for the paths they use.

Feedback concerning the actual format of the document should go to the HOWTO coordinator - mail to linux-howto@metalab.unc.edu).

5.2 Copyright Information

The Mail-HOWTO is copyrighted (c)1999 Eric S. Raymond. Copyright is retained for the purpose of enforcing the Linux Documentation Project license terms.

A verbatim copy may be reproduced or distributed in any medium physical or electronic without permission of the author. Translations are similarly permitted without express permission if it includes a notice on who translated it.

Short quotes may be used without prior consent by the author. Derivative work and partial distributions of the Mail-HOWTO must be accompanied with either a verbatim copy of this file or a pointer to the verbatim copy.

Commercial redistribution is allowed and encouraged; however, the maintainer would appreciate being notified of any such distributions (as a courtesy).

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents.

We further want that *all* information provided in the HOWTOS is disseminated. If you have questions, please contact the Linux HOWTO coordinator, at `linux-howto@metalab.unc.edu`.

TOC

5.3 Standard Disclaimer

Of course, we disavow any potential liability for the contents of this document. Use of the concepts, examples, and/or other content of this document is entirely at your own risk.

5.4 Acknowledgements

This was originally authored by Vince Skahan. I have rewritten it for the modern ISP-centric world in which UUCP is little more than a memory.

In May 1999, the name was changed from "The Linux Electronic Mail HOWTO" to avoid a collision with Guylhem Aznar's Mail HOWTO, which will become the Mail Administrator HOWTO.

The Linux Modem-HOWTO

David S.Lawyer mailto:bf347@lafn.org
Contains much info from Serial-HOWTO by Greg Hankins

v0.04, August 1999

Help with selecting, connecting, configuring, trouble-shooting, and understanding modems for a PC. See Serial-HOWTO for multiport serial boards.

1. Introduction

- 1.1 Copyright, Disclaimer, Trademarks, & Credits
- 1.2 Future Plans, You Can Help
- 1.3 New Versions of this HOWTO
- 1.4 What is a Modem ?
- 1.5 Quick Install

2. Modem & Serial Port Basics

- 2.1 Modem Converts Digital to Analog (and conversely)
- 2.2 What is a Serial Port ?
- 2.3 IO Address & IRQ
- 2.4 Names: ttyS0, ttyS1, etc.
- 2.5 Interrupts
- 2.6 Data Compression (by the Modem)
- 2.7 Error Correction
- 2.8 Data Flow (Speeds)
- 2.9 Flow Control
- 2.10 Data Flow Path; Buffers

- 2.11 Modem Commands
- 2.12 Serial Software: Device Driver Module

3. Configuring Modems (including the serial port)

- 3.1 Configuring Overview

4. Configuring the Serial Port

- 4.1 Configuring Overview
- 4.2 Common mistakes made re low-level configuring
- 4.3 I/O Address & IRQ: Boot-time messages
- 4.4 What is the current IO address and IRQ of my Serial Port ?
- 4.5 Choosing Serial IRQs
- 4.6 Choosing Addresses --Video card conflict with ttyS3
- 4.7 Set IO Address & IRQ in the hardware (mostly for PnP)
- 4.8 Giving the IRQ and IO Address to Setserial
- 4.9 Other Configuring

5. Modem Configuration (excluding serial port)

- 5.1 AT Commands
- 5.2 Init Strings: Saving and Recalling
- 5.3 Other Modem Commands

6. Modems for a Linux PC

- 6.1 External vs. Internal
- 6.2 External Modems
- 6.3 Internal Modems
- 6.4 Internal Modems You Must Avoid (winmodems, etc.)
- 6.5 Which Internal Modems are Best to Avoid ?

TOC

7. Serial Port Devices /dev/ttyS2, etc.

- 7.1 Serial Port Device Names & Numbers
- 7.2 Link ttySN to /dev/modem ?
- 7.3 The cua Device

8. Interesting Programs You Should Know About

- 8.1 What is setserial ?
- 8.2 What is isapnp ?
- 8.3 What is wvdialconf ?
- 8.4 What is stty ?

9. Trying Out Your Modem (Dialing Out)

- 9.1 Are You Ready to Dial Out ?
- 9.2 Dialing Out with Minicom
- 9.3 Dialing Out with Kermit

10. Dial-In

- 10.1 Overview
- 10.2 Getty
- 10.3 What Happens when Someone Dials In ?
- 10.4 Why Manual Answer is Best
- 10.5 Callback
- 10.6 Voice Mail

11. Uugetty for Dial-In (from the old Serial-HOWTO)

- 11.1 Installing getty_ps
- 11.2 Setting up uugetty
- 11.3 Customizing uugetty

TOC

12. What Speed Should I Use with My Modem?

- 12.1 Speed and Data Compression
- 12.2 Where do I Set Speed ?
- 12.3 Can't Set a High Enough Speed
- 12.4 Speed Table

13. Communications Programs And Utilities

- 13.1 Minicom vs. Kermit
- 13.2 Lists of Programs
- 13.3 SLiRP and term

14. What Are UARTs? How Do They Affect Performance?

- 14.1 Introduction to UARTS
- 14.2 Two Types of UARTs
- 14.3 FIFOs
- 14.4 UART Model Numbers

15. Troubleshooting

- 15.1 I can't get 56K on my 56K modem
- 15.2 Uploading (downloading) files is broken/slow
- 15.3 For Dial-in I Keep Getting "line NNN of inittab invalid"
- 15.4 When I Try To Dial Out, It Says "/dev/ttySN: Device or resource busy"
- 15.5 I Keep Getting: ``Id "S3" respawning too fast: disabled for 5 minutes"
- 15.6 My Modem is Hosed after Someone Hangs Up, or uugetty doesn't respawn
- 15.7 uugetty Still Doesn't Work
- 15.8 My Modem is Physically There but Can't be Found
- 15.9 The following subsections are in both the Serial and Modem HOWTOs:
- 15.10 My Serial Port is Physically There but Can't be Found
- 15.11 Slow: Text appears on the screen slowly after long delays

- 15.12 The Startup Screen Show Wrong IRQs for the Serial Ports.
- 15.13 "Cannot open /dev/ttyS?: Permission denied"
- 15.14 "Operation not supported by device" (error message) for ttySx
- 15.15 "Cannot create lockfile. Sorry" (error message)
- 15.16 Software which may help

16. Flash Upgrades

17. Other Sources of Information

- 17.1 Misc
- 17.2 Books
- 17.3 HOWTOs
- 17.4 Usenet newsgroups
- 17.5 Web Sites

18. Appendix A: How Modems Work (technical) (unfinished)

- 18.1 Modulation Details

19. Appendix B: "baud" vs. "bps"

- 19.1 A simple example
- 19.2 Real examples

20. Appendix C: Terminal Server Connection

21. Appendix D: Other Types of Modems

- 21.1 Digital-to-Digital "Modems"
- 21.2 ISDN "Modems"
- 21.3 Digital Subscriber Line (DSL)

1. Introduction

This covers conventional modems for PC's, mainly modems on the ISA bus (although much of this should also apply to the PCI bus). For modems on the PCMCIA bus see the PCMCIA-HOWTO: PCMCIA serial and modem devices. This HOWTO doesn't cover PPP (used to connect to the Internet via a modem) or communication programs. Except it does show how to use communication programs to test that your modem works OK and can make phone calls. If you want to use a modem to connect to the Internet then you need to set up PPP. There's a lot of documentation for PPP (including a PPP-HOWTO) but it may be obsolete or not apply to your situation. Some of it might be found in /usr/doc/ppp or the like.

1.1 Copyright, Disclaimer, Trademarks, & Credits

Copyright

Copyright (c) 1998-9 by David S. Lawyer Please freely copy and distribute (sell or give away) this document. For corrections and minor changes contact the maintainer. Otherwise you may create derivative works and distribute them provided you:

1. Discuss it with the maintainer (if there is one). 2. Put the derivative work at an appropriate Internet site for free downloading and supply the url to both LDP and the HOWTO maintainer. 3. License the work in the spirit of this license or use GPL. 4. Give due credit to previous authors and major contributors.

Disclaimer

While I haven't intentionally tried to mislead you, there are likely a number of errors in this document. Please let me know about them. Since this is free documentation, it should be obvious that I cannot be held legally responsible for any errors.

Trademarks

If certain words are trademarks, the context should make it clear to whom they belong. For example "MS Windows" (or just "Windows") implies that "Windows" belongs to Microsoft (MS). "Hayes" is a trademark of Microcomputer Products Inc. I use "winmodem" to mean any modem which requires MS-Windows and not in the trademark sense.

Credits

The following is only a rough approximation of how version 0.0 of this document was created: About 1/3 of the material here was lifted directly from Serial-HOWTO v. 1.11 by Greg Hankins. mailto:gregh@cc.gatech.edu (with his permission). About another 1/3 was taken from that

Serial-HOWTO and revised. The remaining 1/3 is newly created by the author: David S. Lawyer mailto:bf347@lafn.org.

TOC

1.2 Future Plans, You Can Help

An explanation of how to set up modems for dial-in is lacking in this version but should be included in the next. Please let me know of any errors in facts, opinions, logic, spelling, grammar, clarity, links, etc. But first, if the date is over a months old, check to see that you have the latest version. Please send me any other info that you think belongs in this document. The French "Modems-HOWTO" needs to be somehow merged with this document (but I don't know French).

1.3 New Versions of this HOWTO

New versions of this Modem-HOWTO come out every month or so since modem situation is rapidly changing (and since I'm still learning). Your problem might be solved in the latest version. It will be available to browse and/or download at LDP mirror sites. For a list of such sites see: http://metalab.unc.edu/LDP/mirrors.html If you only want to quickly check the date of the latest version you may not want to use a mirror site so check out: http://metalab.unc.edu/LDP/HOWTO/Modem-HOWTO.html.

1.4 What is a Modem ?

A modem is a device that lets one send digital signals over ordinary telephone lines not designed for digital signals. If telephone lines were all digital then you wouldn't need a modem. It permits your computer to connect to and communicate with the rest of the world. When you use a modem, you normally use a communication program or web browser (which includes such a program) to utilize the modem and dial-out on a telephone line. Advanced modem users can set things up so that others may phone in to them and use their computer. This is called "dial-in".

There are two basic types of modems for a PC: external and internal. The external sets on your desk outside the PC while the internal is not visible since it's inside the PC. The external modem plugs into a connector on the back of the PC known as a "serial port". The internal modem is a card that is inserted inside the computer and has an (invisible) serial port built into it. For a more detailed comparison see External vs. Internal. Thus when you get an internal modem, you also get a dedicated serial port (which can only be used with the modem and not with anything else such as another modem or a printer). In Linux, the serial ports are named ttyS0, ttyS1, etc. (usually corresponding respectively to COM1, COM2, etc. in Dos/Windows).

The serial port is not to be confused with the "Universal Serial Bus" (USB) which uses a special modular connector and may be used with modems in the future. See Modem & Serial Port Basics for more details on modems and serial ports.

Modems often include the ability to send Faxes (Fax Modems). See Fax for a list of fax software. "Voice" modems can work like an automatic answering machine and handle voicemail. See Voicemail.

1.5 Quick Install

External Modem Install

With a straight-thru or modem cable, connect the modem to an unused serial port on the PC. Make sure you know the name of the serial port: in most cases COM1 is ttyS0, COM2 is ttyS1, etc. You may need to check the BIOS setup menu to determine this. Plug in the power cord to provide power to the modem. See All Modems for further instructions.

Internal Modems (on ISA bus)

(See PCI Modems for the PCI bus) If the modem says it will only work under MS Windows, you are out of luck. If you already have 2 serial ports, make this the 3rd serial port (ttyS2 = COM3). Find an unused IRQ number to use. An unused one is often IRQ5 for the 2nd parallel port or a sound card (which you may not have). Then set the jumpers (or the like) on the internal modem to the unused IRQ and IO address 3E8 (ttyS2) .

"Or the like" (in the previous sentence) may be a bit tricky. If the modem is a Plug and Play (PnP) for the ISA bus, the equivalent probably can be done using the "isapnp" program which comes with "isapnptools". See "man isapnp" or the FAQ for it. See also "Plug-and-Play-HOWTO. With a PnP-BIOS you may be able to tell the CMOS setup menu that you don't have a PnP OS and then the BIOS may set a suitable IRQ and IO address in the modem card. I once thought that if you configured the modem under Windows 9x then Windows would be nice and put that configuration info (IRQ and IO address) into the BIOS's flash memory so that it could be used by the BIOS each time you start Linux. But it may not do this so try it only as a last resort. You might try using ICU utility under Windows.. There may even be a way to disable PnP using software (under Windows) that came with the modem.

Finally you must also find the file where "setserial" is run and add a line something like: "setserial /dev/ttyS2 irq5". See All Modems for further instructions.

All Modems

Plug the modem into a telephone line. Then start up a communication program such as minicom and go to the configuration menu for the serial port. Assign it a high baud rate a few times higher than the bit rate of your modem. See Speed Table for the "best" speeds to use. Tell it the full name of your serial port such as /dev/ttyS1. Set hardware flow control (RTS/CTS). Now you need to save these settings and exit minicom. Then start minicom again, type AT to see if your modem is there and responds with OK. Then go to the dial directory (or menu) and dial a number.

---

2. Modem & Serial Port Basics

You don't have to understand the basics to use and install a modem. But understanding it may help to determine what is wrong if you run into problems. After reading this section, if you want to understand it even better you may want to see How Modems Work in this document (not yet complete). More details on the serial port (including much of this section) will be found in Serial-HOWTO.

2.1 Modem Converts Digital to Analog (and conversely)

Most all telephone main lines are digital already but the lines leading to your house (or business) are usually analog which means that they were designed to transmit a voltage wave which is an exact replica of the sound wave coming out of your mouth. Such a voltage wave is called "analog". If viewed on an oscilloscope it looks like a sine wave of varying frequency and amplitude. A digital signal is like a square wave. For example 3 v (volts) might be a 1-bit and 0 v could be a 0-bit. For most serial ports (used by external modems) +12 v is a 0-bit and -12 v is a 1-bit (some are + or - 5 v).

To send data from your computer over the phone line, the modem takes the digital signal from your computer and converts it to "analog". It does this by both creating an analog sine wave and then "MODulating" it. Since the result still represents digital data, it could also be called a digital signal instead of analog. But it looks something like an analog signal and almost everyone calls it analog. At the other end of the phone line another modem "DEModulates" this signal and the pure digital signal is recovered. Put together the "mod" and "dem" parts of the two words above and you get "modem" (if you drop one of the two d's). A "modem" is thus a MODulator-DEModulator. Just what modulation is may be found in the section Modulation Details.

2.2 What is a Serial Port ?

Intro to Serial

The serial port is an I/O (Input/Output) device. Since modems have a serial port between them and the computer, it's necessary to understand the serial port as well as the modem.

Most PC's have one or two serial ports. Each has a 9-pin connector (sometimes 25-pin) on the back of the computer. Computer programs can send data (bytes) to the transmit pin (output) and receive bytes from the receive pin (input). The other pins are for control purposes and ground.

The serial port is much more than just a connector. It converts the data from parallel to serial and changes the electrical representation of the data. Inside the computer, data bits flow in parallel (using many wires at the same time). Serial flow is a stream of bits over a single wire (such as on the transmit or receive pin

TOC

of the serial connector). For the serial port to create such a flow, it must convert data from parallel (in the computer) to serial on the transmit pin (and conversely).

Most of the electronics of the serial port is found in a computer chip (or a section of a chip) known as a UART. For more details on UARTs see the section What Are UARTs? How Do They Affect Performance?. But you may want to finish this section first so that you will hopefully understand how the UART fits into the overall scheme of things.

Pins and Wires

Old PC's used 25 pin connectors but only about 9 pins were actually used so today most connectors are only 9-pin. Each of the 9 pins usually connects to a wire. Besides the two wires used for transmitting and receiving data, another pin (wire) is signal ground. The voltage on any wire is measured with respect to this ground. Thus the minimum number of wires to use for 2-way transmission of data is 3. Except that it has been known to work with no signal ground wire but with degraded performance and sometimes with errors.

There are still more wires which are for control purposes (signalling) only and not for sending bytes. All of these signals could have been shared on a single wire, but instead, there is a separate dedicated wire for every type of signal. Some (or all) of these control wires are called "modem control lines". Modem control wires are either in the asserted state (on) of +12 volts or in the negated state (off) of -12 volts. One of these wires is to signal the computer to stop sending bytes out the serial port cable. Conversely, another wire signals the device attached to the serial port to stop sending bytes to the computer. If the attached device is a modem, other wires may tell the modem to hang up the telephone line or tell the computer that a connection has been made or that the telephone line is ringing (someone is attempting to call in). See the Serial-HOWTO: Pinout and Signals for more details.

Internal Modem Contains Serial Port

For an internal modem there is no 9-pin connector but the behavior is almost exactly as if the above mentioned cable wires existed. Instead of a a 12 volt signal in a wire giving the state of a modem control line, the internal modem may just use a status bit in its own memory (a register) to determine the state of this non-existent "wire". The internal modem's serial port looks just like a real serial port to the computer. It even includes the speed limits that one may set at ordinary serial ports such as 115200 bits/sec. Unfortunately for Linux, many internal modems today don't work exactly this way but instead use MS Windows software to help do their job. They will not work under Linux. See Avoid: winmodems.

2.3 IO Address & IRQ

Since the computer needs to communicate with each serial port, the operating system must know that each serial port exists and where it is (its I/O address). It also needs to know which wire (IRQ number) the serial port must use to request service from the computer's CPU. It requests service by sending an interrupt on this wire. Thus every serial port device must store in its non-volatile memory both its I/O address and its Interrupt ReQuest number: IRQ. See Interrupts. For the PCI bus it doesn't work exactly

this way since the PCI bus has its own system of interrupts. But since the PCI-aware BIOS sets up cl to map these PCI interrupts to IRQs, it seemingly behaves just as described above except that sharing interrupts is allowed (2 or more devices may use the same IRQ number).

I/O addresses are not the same as memory addresses. When an I/O addresses is put onto the computer's address bus, another wire is energized. This both tells main memory to ignore the address and tells all devices which have I/O addresses (such as the serial port) to listen to the address to see if it matches the device's. If the address matches, then the I/O device reads the data on the data bus.

2.4 Names: ttyS0, ttyS1, etc.

The serial ports are named ttyS0, ttyS1, etc. (and usually correspond respectively to COM1, COM2, etc. in DOS/Windows). The /dev directory has a special file for each port. Type "ls /dev/ttyS*" to see them. Just because there may be (for example) a ttyS3 file, doesn't necessarily mean that there exists a phyical serial port there.

Which one of these names (ttyS0, ttyS1, etc.) refers to which physical serial port is determined as follows. The serial driver (software) maintains a table showing which I/O address corresponds to which ttyS. This mapping of names (such as ttyS1) to I/O addresses (and IRQ's) may be both set and viewed by the "setserial" command. See What is Setserial. This does `not` set the I/O address and IRQ in the hardware itself (which is set by jumpers or by plug-and-play software). Thus what physical port corresponds to say ttyS1 depends both on what the serial driver thinks (per setserial) and what is set in the hardware. If a mistake has been made, the phyical port may not correspond to any name (such as ttyS2) and thus it can't be used. See Serial Port Devices /dev/ttyS2, etc. for more details>

2.5 Interrupts

Bytes come in over the phone line to the modem, are converted from analog to digital by the modem and passed along to the serial port on their way to their destination inside your computer. When the serial port receives several bytes (may be set to 1, 4, 8, or 14) into its buffer, it signals the CPU to fetch it (them) by sending an electrical signal known as an interrupt on a certain wire. On the ISA bus, each device usually has it own private interrupt line (wire). This interrupt will also be sent if there is an unexpected delay while waiting for the next byte to arrive (known as a timeout). Thus if the bytes are being received slowly (such as someone typing on a terminal keyboard) there may be an interrupt issued for every byte received. For some UART chips if 4 bytes could have received, but none of these 4 show up, then the port gives up waiting for the next byte and issues an interrupt to fetch the 1 or more bytes waiting in the buffer.

Each interrupt conductor (inside the computer) has a number (IRQ) and the serial port must know which conductor to use to signal on. For example, ttyS0 normally uses IRQ number 4 known as IRQ4 (or IRQ 4). A list of them and more will be found in "man setserial" (search for "Configuring Serial Ports"). Interrupts are issued whenever the serial port needs to get the CPU's attention. It's important to do this in a timely manner since the buffer inside the serial port can hold only 16 (1 in old modems) incoming bytes. If the CPU fails to remove such received bytes promptly, then there will not be any space left for any more incoming bytes and the small buffer may overflow (overrun). Bytes will be lost.

For an external modem, there is no way (such as flow control) to stop the flow rapidly enough to prevent this. For an internal modem the 16-byte buffer is on the same card and a good modem will not write to it if it's full. Thus a good internal modem will not overrun the 16-byte buffers and this is one advantage of an internal modem.

Interrupts are also issued when the serial port has just sent out all 16 of its bytes from its small transmit buffer out the external cable. It then has space for 16 more outgoing bytes. The interrupt is to notify the CPU of that fact so that it may put more bytes in the small transmit buffer to be transmitted. Also, when a modem control line changes state an interrupt is issued.

The buffers mentioned above are all hardware buffers. The serial port also has large buffers in main memory. This will be explained later

Interrupts convey a lot of information but only indirectly. The interrupt itself just tells a chip called the interrupt controller that a certain serial port needs attention. The interrupt controller then signals the CPU. The CPU runs a special program to service the serial port. That program is called an interrupt service routine (part of the serial driver software). It tries to find out what has happened at the serial port and then deals with the problem such a transferring bytes from (or to) the serial port's hardware buffer. This program can easily find out what has happened since the serial port has registers at IO addresses known to the the serial driver software. These registers contain status information about the serial port. The software reads these registers and by inspecting the contents, finds out what has happened and takes appropriate action.

2.6 Data Compression (by the Modem)

Before continuing with the basics of the serial port, one needs to understand about something done by the modem: data compression. In some cases this task is actually done by software run on the computer's CPU but unfortunately at present, such software only works for MS Windows. The discussion here will be for the case where the modem itself does the compression since this is what must happen in order for the modem to work under Linux.

In order to send data faster over the phone line, one may compress (encode it) using a custom encoding scheme which itself depends on the data. The encoded data is smaller than the original (less bytes) and can be sent over the Internet in less time. This process is called "data compression".

If you download files from the Internet, they are likely already compressed and it is not feasible for the modem to try to compress them further. Your modem may sense that what is passing thru has already been compressed and refrain from trying a compress it any more. If you are receiving data which has been compressed by the other modem, your modem will decompress it and create many more bytes than were sent over the phone line. Thus the flow of data from your modem into your computer will be higher than the flow over the phone line to you. The ratio of this flow is called the compression ratio. Compression ratios as high as 4 are possible, but not very likely.

TOC

2.7 Error Correction

Similar to data compression, modems may be set to do error correction. While there is some overhead cost involved which slows down the byte/sec flow rate, the fact that error correction strips off start and stop bits actually increases the data byte/sec flow rate.

For the serial port's interface with the external world, each 8-bit byte has 2 extra bits added to it: a start-bit and a stop-bit. Without error correction, these extra start and stop bits usually go right thru the modem and out over the phone lines. But when error correction is enabled, these extra bits are stripped off and the 8-bit bytes are put into packets. This is more efficient and results in higher byte/sec flow in spite of the fact that there are a few more bytes added for packet headers and error correction purposes.

2.8 Data Flow (Speeds)

Data (bytes representing letters, pictures, etc.) flows from your computer to your modem and then out on the telephone line (and conversely). Flow rates (such as 56k (56000) bits/sec) are (incorrectly) called "speed". But almost everyone says "speed" instead of "flow rate". If there were no data compression the flow rate from the computer to the modem would be about the same as the flow rate over the telephone line.

Actually there are two different speeds to consider at your end of the phone line:

- The speed on the phone line itself (DCE speed) modem-to-modem
- The speed from your computer's serial port to your modem (DTE speed)

When you dial out and connect to another modem on the other end of the phone line, your modem often sends you a message like "CONNECT 28800" or "CONNECT 115200". What do these mean? Well, its either the DCE speed or the DTE speed. If it's higher than the advertised modem speed it must be the DTE modem-to-computer speed. This is the case for the 115200 speed shown above. The 28800 must be a DCE (modem-to-modem) speed since the serial port has no such speed. One may configure the modem to report either speed. Some modems report both speeds and report the modem-to-modem speed as (for example): CARRIER 28800.

If you have an internal modem you would not expect that there would be any speed limit on the DTE speed from your modem to your computer since you modem is inside your computer and is almost part of your computer. But there is since the modem contains a dedicated serial port within it.

It's important to understand that the average speed is often less than the specified speed, especially on the short DTE computer-to-modem line. Waits (or idle time) result in a lower average speed. These waits may include long waits of perhaps a second due to Flow Control. At the other extreme there may be very short waits (idle time) of several micro-seconds separating the end of one byte and the start of the next byte. In addition, modems will fallback to lower speeds if the telephone line conditions are less than pristine.

For a discussion of what DTE speed is best to use see section What Speed Should I Use.

2.9 Flow Control

TOC

Flow control means the ability to stop the flow of bytes in a wire. It also includes provisions to restart the flow without any loss of bytes. Flow control is needed for modems to allow a jump in instantaneous flow rates.

Example of Flow Control

For example, consider the case where you connect a 36.6k modem to your serial port that simply sends and receives bytes over the phone line at exactly 36.6k bits per second (bps). It's not doing any data compression or error correction. You have set the serial port speed to 115,200 bits/sec (bps), and you are sending data from your computer to the phone line. Then the flow from the your computer to your modem is at 115.2k bps. However the flow from your modem out the phone line is at best only 33.6k bps. Since a faster flow (115.2k) is going into your modem than is coming out of it, the modem is storing the excess flow (115.2k -33.6k = 81.6k bps) in one of its buffers. This buffer would eventually overrun (run out of storage space) unless the 115.2k flow is stopped.

But now flow control comes to the rescue. When the modem's buffer is almost full, the modem sends a stop signal to the serial port. The serial port passes on the stop signal on to the device driver and the 115.2k bps flow is halted. Then the modem continues to send out data at 33.6k bps drawing on the data it previous accumulated in its (the modem's) buffer. Since nothing is coming into the buffer, the level of bytes in it starts to drop. When almost no bytes are left in the buffer, the modem sends a start signal to the serial port and the 115.2k flow from the computer to the modem resumes. In effect, flow control creates an average flow rate (in this case 33.6k) which is significantly less than the "on" flow rate of 115.2k bps. This is "start-stop" flow control.

The above is a simple example of flow control for flow from the computer to a modem , but there is also flow control which is used for the opposite direction of flow: from a modem (or other device) to a computer. Each direction of flow involve 3 buffers: 1. in the modem 2. in the uart (serial port) 3. in main memory managed by the serial driver. Flow control protects certain buffers from overflowing. The small serial hardware buffers are not protected in this way but rely instead on a fast response to the interrupts they issue. This is the essence of flow control but there are still some more details.

You don't often need flow control in the direction from the modem to a PC. For complex example of a case where it's needed see "Complex Flow Control Example" in the Serial-HOWTO. But if you don't have a high enough speed set between the modem and the computer (serial port speed) then you do need to slow down the flow from the modem to the PC. To do this you must stop the incoming flow of bytes over the telephone line. Your modem must tell the other modem to stop sending. See Modem-to-Modem Flow Control

Hardware vs. Software Flow Control

If feasible best to use "hardware" flow control that uses two dedicated "modem control" wires to send the "stop" and "start" signals. Software flow control uses the main receive and transmit wires to send the start and stop signals. It uses the ASCII control characters DC1 (start) and DC3 (stop) for this purpose. They are just inserted into the regular stream of data. Software flow control is not only slower in reacting

but also does not allow the sending of binary data unless special precautions are taken. Since binary will likely contain DC1 and DC3, special means must be taken to distinguish between a DC3 that means a flow control stop and a DC3 that is part of the binary code. Likewise for DC1. To get software flow control to work for binary data requires both modem (hardware) and software support.

Symptoms of No Flow Control

Understanding flow-control theory can be of practical use. For example I used my modem to access the Internet and it seemed to work fine. But after a few months I tried to send long files from my PC to an ISP and a huge amount of retries and errors resulted (but eventually Kermit could send a long file after many retries). Receiving in the other direction (from my ISP to me) worked fine. The problem turned out to be a hardware defect in my modem that had resulted in disabling flow control. My modem's buffer was overflowing (overrunning) on long outgoing files since no "stop" signal was ever sent to the computer to halt sending to the modem. There was no problem in the direction from the modem to my computer since the capacity (say 115.2k) was always higher than the flow over the telephone line. The fix was to enable flow control by putting an enable-flow-control command for the modem last in the init string.

Modem-to-Modem Flow Control

This is the flow control of the data sent over the telephone lines between two modems. Practically speaking, it only exists when you have error correction set. Actually, even without error correction it's possible to enable software flow control between modems but it may interfere with sending binary data so it's not often used.

2.10 Data Flow Path; Buffers

Although much has been explained about this including flow control, a pair of 16-byte serial port buffers (in the hardware), and a pair of larger buffers inside a modem there is still another pair of buffers. These are large buffers (perhaps 8k) in main memory also known as serial port buffers. When an application program sends bytes to the serial port (and modem) they first get stashed in the the transmit serial port buffer in main memory. The pair consists of both this transmit buffer and a receive buffer for the opposite direction of byte-flow.

The serial device driver takes out say 16 bytes from this transmit buffer, one byte at a time and puts them into the 16-byte transmit buffer in the serial hardware for transmission. Once in that transmit buffer, there is no way to stop them from being transmitted. They are then transmitted to the modem which also has a fair sized (say 1k) buffer. When the device driver (on orders from flow control) stops the flow of outgoing bytes from the computer, what it actually stops is the flow of outgoing bytes from the large transmit buffer in main memory. Even after this has happened and the flow to the modem has stopped, an application program may keep sending bytes to the 8k transmit buffer until it becomes fill.

When it gets fill, the application program can't send any more bytes to it (a "write" statement in a C_program blocks) and the application program temporarily stops running and waits until some buffer space becomes available. Thus a flow control "stop" is ultimately able to stop the program that is sending the bytes. Even though this program stops, the computer does not necessarily stop computing. It may

switch to running other processes while it's waiting at a flow control stop. The above was a little oversimplified since there is another alternative of having the application program itself do something else while it is waiting to "write".

2.11 Modem Commands

Commands to the modem are sent to it from the communication software over the same conductor as used to send data. The commands are short ASCII strings. Examples are "AT&K3" for enabling hardware flow control (RTS/CTS) between your computer and modem; and "ATDT5393401 for Dialing the number 5393401. Note all commands are prefaced by "AT". Some commands such as enabling flow control help configure the modem. Other commands such as dialing a number actually do something. There are about a hundred or so different possible commands. When your communication software starts running, it first sends an "init" string of commands to the modem to configure it. All commands are sent on the ordinary data line before the modem dials (or receives a call).

Once the modem is connected to another modem (on-line mode), everything that is sent from your computer to your modem goes directly to the other modem and is not interpreted by the modem as a command. There is a way to "escape" from this mode of operation and go back to command mode where everything sent to the modem will be interpreted as a command. The computer just sends "+++" with a specified time spacing before and after it. If this time spacing is correct, the modem reverts to command mode. Another way to do this is by a signal on a certain modem control line.

There are a number of lists of modem commands on the Internet. The section Web Sites has links to a couple of such web-sites. Different models and brands of modems do not use exactly the same set of such commands. So what works for one modem might not work for another. Some common command (not guaranteed to work on all modems) are listed in this HOWTO in the section Modem Configuration

2.12 Serial Software: Device Driver Module

The device driver for the serial port is the software that operates the serial port. It is now provided as a serial module. This module will normally get loaded automatically if it's needed. The kernel 2.2 + will do this. In earlier kernels, you had to have `kerneld` running in order to do auto-load modules on demand. Otherwise the serial module needed to be explicitly listed in /etc/modules. Before modules became popular with Linux, the serial driver was usually built into the kernel.

When the serial module is loaded it displays a message on the screen about the existing serial ports (often showing a wrong IRQ). But once the module is used by `setserial` to tell the device driver the (hopefully) correct IRQ then you should see a second display with the correct IRQ, etc. See What is Setserial for more info on `setserial`.)

One may modify the driver by editing the kernel source code. Much of the serial driver is found in the file serial.c. For details regarding writing of programs for the serial port see Serial-Programming-HOWTO (currently being revised by Vern Hoxie).

3. Configuring Modems (including the serial port)

If you want to use a modem only for MS Windows/Dos, then you can just install almost any modem and it will work OK. With a Linux PC it's not usually this easy unless you use an external modem. All external modems should work OK (even if they are labeled "Plug and Play") But most new internal modems are Plug-and-Play (PnP) and have PnP serial ports. You may need to use the Linux "isapnp" program to configure these PnP serial ports. See the Plug-and-Play-HOWTO for more information.

3.1 Configuring Overview

Since each modem has an associated serial port there are two parts to configuring a modem:

- Configuring the modem itself: Done by the communication program
- Configuring the modem's serial port: Done only partly by the communication program

Most of the above configuring (but not necessarily most of the effort) is done by the communication program that you use with the modem such as `minicom` or `seyon`, or by the PPP (usually needed for use of a web browser). If you use the modem for dial-in, then the `getty` program which you use to present outsiders with a login-prompt, will help configure. Thus to configure the modem (and much of the serial port) you need to configure the communication program (or PPP or `getty`).

But note that not all of the configuration of the serial port is done by the communication program (or getty). The remaining configuring is simple to state (but sometimes difficult to do). It mainly consists of setting the IO address of the port and its IRQ number. In fact, plug-and-play could set these without you doing a thing. But there's a serious problem: Linux (as of early 1999) doesn't support plug-and-play very well. This may create a difficult problem for you. The next section covers the configuring of the serial port.

4. Configuring the Serial Port

4.1 Configuring Overview

In many cases, configuring will happen automatically and you have nothing to do. But sometimes you need to configure (or just want to check out the configuration). If so, first you need to know about the two parts to configuring the serial port under Linux:

The first part (low-level configuring) is assigning it an IO address, IRQ, and name (such as ttyS2). This IO-IRQ pair must be set in both the hardware and told to the serial driver. We might just call this "io-irq" configuring for short. The `setserial` is used to tell the driver. PnP methods, jumpers, etc, are used to set the hardware. Details will be supplied later. If you need to configure but don't understand certain details it's easy to get into trouble.

The second part (high-level configuring) is assigning it a speed (such as 38.4K bits/sec), selecting flow control, etc. This is often done by communication programs such as PPP, minicom, or by getty (which you may run on the port so that others may log into it). This high-level configuring may also be done with the `stty` program. `stty` is also useful to view the current status if you're having problems and to reset the terminal interface to close to "normal" by typing something like ``stty sane". See also the Serial-HOWTO section: "Stty". When Linux starts, some effort is made to detect and configure (low-level) a few serial ports. Exactly what happens depends on your BIOS, hardware, Linux distribution, etc. If the serial ports work OK, there may be no need for you to do any configuring. Application programs often do the high-level configuring but you may need to supply them some information so that they can do it. With Plug-and-Play serial ports (often built into an internal modem), the situation has become more complex. Here are cases when you need to do low-level configuring (set IRQ and IO addresses):

- You want to use more than 2 serial ports
- Installing a new serial port (such as an internal modem)
- Having problems with serial port(s)

For kernel 2.2+ you may be able to use more that 2 serial ports without low-level configuring by sharing interrupts but this may be no easier than low-level configuring. See Interrupt sharing and Kernels 2.2+

The low-level configuring (setting the IRQ and IO address) seems to cause people more trouble (than high-level), although for many it's fully automatic and there is no configuring to be done. Thus most all of this section is on that topic. Until the serial driver knows the correct IRQ and IO address the port will probably not work at all. It likely cannot even be found by Linux. Even if it can be found, it may work extremely slow if the IRQ is wrong. See Slow: Text appears on the screen slowly after long delays.

In the Wintel world, the IO address and IRQ are called "resources" and we are thus configuring certain resources. In review, the low-level configuring consists of putting two values (an IRQ number and IO address) into two places:

1. the device driver (often by running "`setserial`" at boot-time)
2. memory registers of the serial port hardware itself

TOC

You may watch the start-up (= boot-time) messages. They are usually correct. But if you're having problems there's a good chance that some of these messages don't show the true configuration of the hardware (and are not supposed to). See I/O Address & IRQ: Boot-time messages.

4.2 Common mistakes made re low-level configuring

Here are some common mistakes people make:

- setserial command: They run it (without the "autoconfig" option) and think it has checked out the hardware (it hasn't).
- setserial messages: They see them displayed on the screen at boot-time, and erroneously think that the result shows how their hardware is actually configured.
- /proc/interrupts: When their serial device isn't in use they don't see its interrupt there, and erroneously conclude that their serial port can't be found (or doesn't have an interrupt set).
- /proc/ioports: People think this shows the hardware configuration when it only shows about the same data (possibly erroneous) as setserial.

4.3 I/O Address & IRQ: Boot-time messages

In many cases your ports will automatically get low-level configured at boot-time (but not always correctly). To see what is happening, look at the start-up messages on the screen. Don't neglect to check the messages from the BIOS before Linux is loaded (no examples shown here). Use shift-PageUp to see the messages after they have flash by. The `dmesg` command may be used at any time to view some of the messages but it often misses important ones. Here's an example of the start-up messages (as of mid 1999). Note that ttyS00 is the same as /dev/ttyS0.

```
At first you see what was detected (but the irq is only a wild guess):

Serial driver version 4.27 with no serial options enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
ttyS01 at 0x02f8 (irq = 3) is a 16550A
ttyS02 at 0x03e8 (irq = 4) is a 16550A

Later you see what was saved, but it's not necessarily correct either:

Loading the saved-state of the serial devices...
/dev/ttyS0 at 0x03f8 (irq = 4) is a 16550A
/dev/ttyS1 at 0x02f8 (irq = 3) is a 16550A
/dev/ttyS2 at 0x03e8 (irq = 5) is a 16550A
```

Note that there is a slight disagreement: The first message shows ttyS2 at irq=4 while the second shows it at irq=5. Your may only have the first message. In most cases the last message is the correct one. But if your having trouble it may be misleading. Before reading the explanation of all of this complexity in the rest of this section, you might just try using your serial port and see if it works OK. If so it may not be essential to read further.

The second message is from the `setserial` program being run at boot-time. It shows what the device driver

thinks is the correct configuration. But this too could be wrong. For example, the irq could actually be set irq=8 in the hardware (both messages wrong). The irq=5 is there because someone incorrectly put this into configuration file (or the like). The fact that Linux sometimes gets IRQs wrong is because it doesn't probe for IRQs. It just assumes the "standard" ones (first message) or accepts what you told it when you configured it (second message). Neither of these is necessarily correct. If the serial driver has the wrong IRQ the serial port is very slow or doesn't seem to work at all.

The first message is a result of Linux probing the serial ports. If a port shows up here it exists but the irq may be wrong. Linux doesn't check IRQs because doing so is not foolproof. It just assumes the IRQs are as shown because they are the "standard" values. Your may check them manually with `setserial` using the `autoconfig` option but this isn't guaranteed to be correct.

The data shown by the BIOS messages (which you see at first) is what is set in the hardware. If your serial port is Plug-and-Play PnP then it's possible that the `isapnp` will run and change these settings. Look for messages about this after Linux starts. The last serial port message shown in the example above should agree with the BIOS messages (as possibly modified by isapnp). If they don't agree then you either need to change the setting in the port hardware or use setserial to tell the driver what is actually set in the hardware.

Also, if you have Plug-and-Play (PnP) serial ports, Linux will not find them unless the IRQ and IO has been set inside the hardware by Plug-and-Play software. This is a common reason why the start-up messages do not show a serial port that physically exists. The PC hardware (a PnP BIOS) may automatically low-level configure this. PnP configuring will be explained later.

4.4 What is the current IO address and IRQ of my Serial Port ?

The previous section indicated how to attempt to do this by looking at the start-up messages. An alternate title for this section is "Other ways of finding out your IO Address and IRQ". If they give you sufficient info then you may not need to read this section. If they don't then there are some other ways to look into this.

There are really two answers to the question "What is my IO and IRQ?" 1. What the device driver thinks has been set (This is what setserial sets and shows). 2. What is actually set in the hardware. They both should be the same. If you're having trouble (including communication programs that can't communicate) it may mean that these two items are not set the same. In other words, this means that the driver has incorrect info on the physical serial port. If the driver has the wrong IO address it will try to send data to a non-existing serial port --or even worse, to an actual device that is not a serial port. If it has the wrong IRQ the driver will not get interrupt service requests from the serial port, resulting in the possible overflow of the serial port's buffer and in very slow response. See Slow: Text appears on the screen slowly after long delays. If it has the wrong model of UART there is also apt to be trouble. To determine if both I0-IRQ pairs are identical you must find out the answers to the questions posed in the next two subsections.

What does the device driver think?

This is easy to find out. Just look at the start-up messages or type "setserial -g /dev/ttyS*". If everything works OK then what it tells you is likely also set in the hardware. There are some other ways to find this info by looking at "files" in the /proc directory. An important reason for understanding these other ways is to warn you that they only show what the device driver thinks. Some people view certain "files" in the /proc directory and erroneously think that what they see is set in the hardware but "it ain't necessarily so".

/proc/ioports will show the IO addresses that the drivers are using. /proc/interrupts shows the IRQs that are used by drivers of currently running processes (that have devices open). Note that in both cases above you are only seeing what the driver thinks and not necessarily what is actually set in the hardware. /proc/interrupts also shows how many interrupts have been issued (often thousands) for each device. You can get a clue from this because if you see a large number of interrupts that have been issued it means that there is a piece of hardware somewhere that is using that interrupt. Sometimes a showing of just a few interrupts doesn't mean that that interrupt is actually being physically generated by any serial port. Thus if you see almost no interrupts for a port that you're trying to use, that interrupt might not be set in the hardware and it implies that the driver is using the wrong interrupt. To view /proc/interrupts to check on a program that you're currently running (such as "minicom") you need to keep the program running while you view it. To do this, try to jump to a shell without exiting the program.

What is set in my serial port hardware ?

How do you find out what IO address and IRQ are actually set in the device hardware? Perhaps the BIOS messages will tell you some info before Linux starts booting. Use the shift-PageUp key to step back thru the boot-time messages and look at the very first ones which are from the BIOS. This is how it was before Linux started. Setserial can't change it but isapnp or pciutils can.

One crude method is try probing with setserial using the "autoconfig" option. You'll need to guess the addresses to probe at. See What is Setserial. For a PCI serial port, look at /proc/pci or /proc/bus/pci/devices. If your serial port is is Plug-and-Play see the next two subsections.

For a port set with jumpers, its how the jumpers were set. If the port is not Plug-and-Play (PnP) but has been setup by using a DOS program then it's set at whatever the person who ran that program set it to.

What is set in my PnP serial port hardware ?

PnP ports don't store their configuration in the hardware when the power is turned off. This is in contrast to Jumpers (non-PnP) which remain the same with the power off. If you have an ISA PnP port, it can reach a state where it doesn't have any IO address or IRQ and is in effect disabled. It should still be possible to find the port using the `pnpdump` program.

For Plug-and-Play (PnP) on the ISA bus one may try the `pnpdump` program (part of `isapnptools`). If you use the --dumpregs option then it should tell you the actual IO address and IRQ set in the port. The address it "trys" is not the device's IO address, but a special

For PnP ports checking on how it's configured under DOS/Windows may not be of much help. Windows stores its configuration info in its Registry which is not used by Linux. It may supply the BIOS's non-volatile memory with some info but it may not be kept in sync with the current Window configuration in the Registry ?? If you let a PnP BIOS automatically do the configuring when you start Linux (and have told the BIOS that you don't have a PnP operating system when running Linux) then Linux should use whatever configuration is in the BIOS's non-volatile memory.

4.5 Choosing Serial IRQs

If you have a true Plug-and-Play set up where either the OS or a PnP BIOS configures all your devices, then you don't choose your IRQs. PnP determines what it thinks is best and assigns them. But if you use the tools in

Linux for Plug-and-Play (isapnp and pcitools) then you have to choose. If you already know what IRQ you want to use you could skip this section except that you may want to know that IRQ 0 has a special use (see following paragraph).

TOC

IRQ 0 is not an IRQ

While IRQ 0 is actually the timer (in hardware) it has a special meaning for setting a serial port with setserial. It tells the driver that there is no interrupt for the port and the driver then will use polling methods. This is quite inefficient but can be tried if there is an interrupt conflict or mis-set interrupt. The advantage of assigning this is that you don't need to know what interrupt is set in the hardware. It should be used only as a temporary expedient until you are able to find a real interrupt to use.

Interrupt sharing and Kernels 2.2+

The general rule is that every device should use a unique IRQ and not share them. But there are situations where sharing is permitted such as with most multi-port boards. Even when it is permitted, it may not be as efficient since every time a shared interrupt is given a check must be made to determine where it came from. Thus if it's feasible, it's nice to allocate every device it's own interrupt.

Prior to kernel 2.2, serial IRQs could be shared with each other only for most multiport boards. Starting with kernel 2.2 serial IRQs may be sometimes shared between all serial ports. In order for sharing to work in 2.2 the kernel must have been compiled with CONFIG_SERIAL_SHARE_IRQ, and the serial port hardware must support sharing (so that if two serial cards put different voltages on the same interrupt wire, only the voltage that means "this is an interrupt" will prevail). Thus even if you have 2.2, it may be best to avoid sharing.

What IRQs to choose?

The serial hardware often has only a limited number of IRQs it can be set at. Also you don't want IRQ conflicts. So there may not be much of a choice. Your PC may normally come with `ttyS0` and `ttyS2` at IRQ 4, and `ttyS1` and `ttyS3` at IRQ 3. Looking at `/proc/interrupts` will show which IRQs are being used by programs currently running. You likely don't want to use one of these. Before IRQ 5 was used for sound cards, it was often used for a serial port.

Here is how Greg (original author of Serial-HOWTO) set his up in /etc/rc.d/rc.serial. rc.serial is a file (shell script) which runs at start-up (it may have a different name of location). For versions of "setserial" after 2.15 it's not always done this way anymore but this example does show the choice of IRQs.

```
/sbin/setserial /dev/ttyS0 irq 3        # my serial mouse
/sbin/setserial /dev/ttyS1 irq 4        # my Wyse dumb terminal
/sbin/setserial /dev/ttyS2 irq 5        # my Zoom modem
/sbin/setserial /dev/ttyS3 irq 9        # my USR modem
```

Standard IRQ assignments:

```
IRQ  0    Timer channel 0 (May mean "no interrupt".  See below.)
IRQ  1    Keyboard
IRQ  2    Cascade for controller 2
IRQ  3    Serial port 2
IRQ  4    Serial port 1
```

```
IRQ  5    Parallel port 2, Sound card
IRQ  6    Floppy diskette
IRQ  7    Parallel port 1
IRQ  8    Real-time clock
IRQ  9    Redirected to IRQ2
IRQ 10    not assigned
IRQ 11    not assigned
IRQ 12    not assigned
IRQ 13    Math coprocessor
IRQ 14    Hard disk controller 1
IRQ 15    Hard disk controller 2
```

There is really no Right Thing to do when choosing interrupts. Just make sure it isn't being used by the motherboard, or any other boards. 2, 3, 4, 5, 7, 10, 11, 12 or 15 are possible choices. Note that IRQ 2 is the same as IRQ 9. You can call it either 2 or 9, the serial driver is very understanding. If you have a very old serial board it may not be able to use IRQs 8 and above.

Make sure you don't use IRQs 1, 6, 8, 13 or 14! These are used by your motherboard. You will make her very unhappy by taking her IRQs. When you are done, double-check `/proc/interrupts` when programs that use interrupts are being run and make sure there are no conflicts.

4.6 Choosing Addresses --Video card conflict with ttyS3

The IO address of the IBM 8514 video board (and others like it) is allegedly 0x?2e8 where ? is 2, 4, 8, or 9. This may conflict (but shouldn't it the serial port is well designed) with the IO address of `ttyS3` at 0x02e8 if the serial port ignores the leading 0 hex digit (many do). That is bad news if you try to use `ttyS3` at this IO address.

In most cases you should use the default addresses if feasible. Addresses shown represent the first address of an 8-byte range. For example 3f8 is really the range 3f8-3ff. Each serial device (as well as other types of devices that use IO addresses) needs its own unique address range. There should be no overlaps (conflicts). Here are the default addresses for the serial ports:

```
ttyS0 address 0x3f8
ttyS1 address 0x2f8
ttyS2 address 0x3e8
ttyS3 address 0x2e8
```

4.7 Set IO Address & IRQ in the hardware (mostly for PnP)

After it's set in the hardware don't forget to insure that it also gets set in the driver by using `setserial`. For non-PnP serial ports they are either set in hardware by jumpers or by running a DOS program ("jumperless") to set them (it may disable PnP). The rest of this subsection is only for PnP serial ports. Here's a list of the possible methods of configuring PnP serial ports:

TOC

- Using a PnP BIOS CMOS setup menu (usually only for external modems on ttyS0 (Com1) and ttyS (Com2))
- Letting a PnP BIOS automatically configure a PnP serial port See Using a PnP BIOS to I0-IRQ Configure
- Doing nothing if you have both a PnP serial port and a PnP Linux operating system (see Plug-and-Play-HOWTO).
- Using `isapnp` for a PnP serial port non-PCI)
- Using pciutils (pcitools) for the PCI bus

The IO address and IRQ must be set (by PnP) in their registers each time the system is powered on since PnP hardware doesn't remember how it was set when the power is shut off. A simple way to do this is to let a PnP BIOS know that you don't have a PnP OS and the BIOS will automatically do this each time you start. This might cause problems in Windows (which is a PnP OS) if you start Windows with the BIOS thinking that Windows is not a PnP OS. See Plug-and-Play-HOWTO.

Plug-and-Play was designed to automate this io-irq configuring, but for Linux at present, it has made life more complicated. The standard kernels for Linux don't support plug-and-play very well. If you use a patch to the Linux kernel to covert it to a plug-and-play operating system, then all of the above should be handled automatically by the OS. But when you want to use this to automate configuring devices other that the serial port, you may find that you'll still have to configure the drivers manually since many Linux drivers are not written to support a Linux PnP OS. If you use `isapnptools` or the BIOS for configuring plug-and-play this will only put the two values into the registers of the serial port section of the modem card and you will likely still need to set up setserial. None of this is easy or very well documented as of early 1999. See Plug-and-Play-HOWTO and the isapnptools FAQ.

Using a PnP BIOS to I0-IRQ Configure

While the explanation of how to use a PnP OS or isapnp for io-irq configuring should come with such software, this is not the case if you want to let a PnP BIOS do such configuring. Not all PnP BIOS can do this. The BIOS usually has a CMOS menu for setting up the first two serial ports. There is often little to choose from. Unless otherwise indicated in menus, these first two ports get set at the standard IO addresses and IRQs. See Serial Port Device Names & Numbers

Whether you like it or not, when you start up a PC a PnP BIOS starts to do PnP (io-irq) configuring of hardware devices. It may do the job partially and turn the rest over to a PnP OS (which you probably don't have) or if thinks you don't have a PnP OS it may fully configure all the PnP devices but not configure the device drivers. This is what you want but it's not always easy to figure out exactly what the PnP BIOS has done.

If you tell the BIOS that you don't have a PnP OS, then the PnP BIOS should do the configuring of all PnP serial ports --not just the first two. If you have MS Windows9x on the same PC, the BIOS might ?? have saved the io-irq configuration used for MS Windows in its non-volatile memory. If you're now reading the latest version of this HOWTO let me know if Windows puts it there. If so the BIOS will use this same configuration for Linux. In this case, if you can find out how MS Windows has set up io-irq then it should be the same under Linux.

If you add a new PnP device, the BIOS should change its PnP configuration to accommodate it. It could even change the io-irq of existing devices if required to avoid any conflicts. For this purpose, it keeps a list of

TOC

non-PnP devices provided that you have told the BIOS how these non-PnP devices are io-irq configured. (way to tell the BIOS this is by running a program called ICU under DOS/Windows.

But how do you find out what the BIOS has done so that you set up the device drivers with this info? The BIOS itself may provide some info, either in it's setup menus of via messages on the screen when you turn on your computer. See What is set in my serial port hardware?

4.8 Giving the IRQ and IO Address to Setserial

Once you've set the IRQ and IO address in the hardware (or arranged for it to be done by PnP) you also need to insure that the "setserial" command is run each time you start Linux. See the subsection Boot-time Configuration

4.9 Other Configuring

Configuring Hardware Flow Control (RTS/CTS)

See Flow Control for an explanation of it. You should always use hardware flow control if possible. Your communication program or "`getty`" should have an option for setting it (and if you're in luck it might be enabled by default). It needs to be set both inside your modem (by an init string or default) and in the device driver. Your communication program should set both of these (if you configure it right).

If none of the above will fully enable hardware flow control. Then you must do it yourself. For the modem, make sure that it's either done by the init string or is on by default. If you need to tell the device driver to do it is best done on startup by putting a file that runs at boot-time. See the subsection Boot-time Configuration You need to add the following to such a file for each serial port (example is ttyS2) you want to enable hardware flow control on:

```
stty crtscts < /dev/ttyS2
```

If you want to see if flow control is enabled do the following: In minicom (or the like) type AT&V to see how the modem is configured and look for &K3 which means hardware flow control. Then see if the device driver knows about it by typing: stty -a < /dev/ttyS2 Look for "crtscts" (without a disabling minus sign).

---

5. Modem Configuration (excluding serial port)

5.1 AT Commands

While the serial port on which a modem resides requires configuring, so does the modem itself. The modem is configured by sending AT commands (or the like) to it on the same serial line that is used to send data.

Most modems use an AT command set. These are cryptic and short ASCII commands where all command strings are prefaced by the letters AT. For example: ATZ&K3 There are two commands here Z and &K3. Unfortunately there are many different variations of the AT command set so that what works for one modem may or may not work for another modem. Thus there is no guarantee that the AT commands given in this section will work on your modem. Another point is that to get the modem to act on the AT command string, a return character must be sent at the end of the string.

Such command strings are either automatically sent to the modem by communication programs or are sent directly by you. Most communication programs provide a screen where you can type commands directly to your modem. This is good for setting up the modem as you can have it remember how it was set even after its powered off.

If you have a manual for your modem you can likely look up the AT command set in it. Otherwise, you may try to find it on the Internet. One may use a search engine and include some actual commands in the search terms to avoid finding sites that just talk about such commands but fail to list them. You might also try a few of the sites listed in the subsection Web Sites

5.2 Init Strings: Saving and Recalling

The examples given in this subsection are from the Hayes AT modem command set. All command strings must be prefaced by the two letters AT (for example: AT&C1&D3). When a modem is powered on, it automatically configures itself with one of the configurations it has stored in its non-volatile memory. If this configuration is satisfactory there is nothing further to do.

If it's not satisfactory, then one may either alter the stored configuration or configure the modem each time you use it by sending it a string of commands known as an "init string" (= initialization string). Normally a a communication program does this. What it sends will depend on how you configured the communications program or what script you wrote for it if you use Kermit. You can usually edit the init string your communication program uses and change it to whatever you want. Sometimes the communications program will let you select the model of your modem and then it will use an init string that it thinks is best for that modem.

The configuration of the modem uses when it's first powered on could be expressed by an init string. You might think of this as the default "string" (called a profile). If your communications program sends the

modem another string (the init string), then this string will modify the default configuration. For example, if the init string only contains two commands, then only those two items will be changed. However, some commands will recall a stored profile from inside the modem so a single such command in the init string can thereby change everything in the configuration.

Modern modems have a few different stored profiles to choose from that are stored in the modem's non-volatile memory (it's still there when you turn it off). In my modem there are two factory profiles (0 and 1, neither of which you can change) and two user defined profiles (0 and 1) that the user may set and store. Your modem may have more. Which one of these user-defined profiles is used at power-up depends on another item stored in the profile. If the command &Y0 is given then in the future profile 0 will be used at power-on. If it's a 1 instead of a 0 then profile 1 will be used at power-on.

There are also commands to recall (use it now) any of the 4 stored profiles. One may put such a command in an init string. Of course if it recalls the same profile as was automatically loaded at power-up, nothing is changed unless the active profile has been modified since power-up. Since it could have been modified It's a good idea to use some kind of an init string even if it does nothing more than recalling a stored profile.

Recalling a saved profile (use 1 instead of 0 for profile 1):
Z0 recalls user-defined profile 0 and resets (hangs up, etc.)
&F0 recalls factory profile 0

Once you have sent commands to the modem to configure it the way you want (including recalling a factory profile and modifying it a little) you may save this as a user-defined profile:
&W0 saves the current configuration to user-profile 0

Many people don't bother saving a good configuration in their modem, but instead, send the modem a longer init string each time the modem is used. Another method is to restore the factory default at the start of the init string and then modify it a little by adding a few other commands to the end of the init string. By doing it this way no one can cause problems by modifying the user-defined profile which is loaded at power-on.

You may pick an init string supplied by someone else that they think is right for your modem. Some communication programs have a library of init strings to select from. The most difficult method (and one which will teach you the most about modems) is to study the modem manual and write one yourself. You could save this configuration inside the modem so that you don't need an init string. A third alternative is to start with an init string someone else wrote, but modify it to suit your purposes.

Now if you look at init strings used by communication programs you may see symbols which are not valid modem commands. These symbols are commands to the communication program itself (such as meaning to pause briefly) and will not be sent to the modem.

5.3 Other Modem Commands

Future editions of Modem-HOWTO may contain more AT commands but the rest of this section is mostly what was in the old Serial-HOWTO. All strings must start with AT. Here's a few Hayes AT codes that should be in the string (if they are not set by using a factory default or by a saved configuration).

```
E1        command echo ON
Q0        result codes are reported
V1        verbose ON
S0=0      never answer (uugetty does this with the WAITFOR option)
```

Here's some more codes concerning modem control lines DCD and DSR:

```
&C1 DCD is on after connect only
&S0 DSR is always on
```

These affect what your modem does when calls start and end. What DTR does may also be set up but it's more complicated.

If your modem does not support a stored profile, you can set these through the `INIT` string in a config file (or the like). Some older modems come with DIP switches that affect register settings. Be sure these are set correctly, too.

Greg Hankins has a collection of modem setups for different types of modems. If you would like to send him your working configuration, please do so: mailto:gregh@cc.gatech.edu You can get these setups at `ftp://ftp.cc.gatech.edu/pub/people/gregh/modem-configs`.

Note: to get his USR Courier V.34 modem to reset correctly when DTR drops, Greg Hankins had to set `&D2` and `S13=1` (this sets bit 0 of register S13). This has been confirmed to work on USR Sportster V.34 modems as well.

Note: some Supra modems treat DCD differently than other modems. If you are using a Supra, try setting `&C0` and *not* `&C1`. You must also set `&D2` to handle DTR correctly.

---

6. Modems for a Linux PC

6.1 External vs. Internal

A modem for a PC may be either internal or external. The internal one is installed inside of your PC (you must remove screws, etc. to install it) and the external one just plugs into a serial port connector on a PC. Internal modems are less expensive, are less likely to overrun, usually use less electricity, and use up no space on your desk. External modems are much easier to install, require less configuration, and have lights which may give you a clue as to what is happening. External modems are easy to move to another computer. Most external modems have no switch to turn off the power supply when not in use and thus are likely to consume a little electricity even when turned off (unless you unplug the power supply from the wall). Each watt they draw costs you about $1/yr. Another possible disadvantage of an external is that you will be forced to use an existing serial port which may not support a speed of over 115,200 k (although as of late 1998 most new internal modems don't either --but some do). If a new internal modem had a 16650 UART it would put less load on the CPU (but almost none do as of late 1998).

Internal modems present a special problem for Linux, but will work just as well as external modems provided you avoid the high percentage of them that will work only for MS Windows, and also provided that you spend time (sometimes a lot of time) to configure them correctly. Some of the modems which will work only under MS Windows are, unfortunately, not labeled as such. If you buy a new one, make sure that you can return it for a refund if it will not work under Linux.

While most new modems are plug-and-play you have various ways to deal with them:

- Use the "isapnp" program
- Have a PnP BIOS do the configuring
- Patch the kernel to create a PnP Linux

Each of the above has shortcomings. Isapnp documentation is difficult to understand although reading the Plug-and-Play-HOWTO (at present incomplete) will aid in understanding it. If you want the PnP BIOS to do the configuring, all you need to do is to make sure that it knows you don't have a PnP operating system. But it may not do it correctly. To find out what it's done see What is set in my serial port hardware?. Patching the kernel can be complicated too and there might not be a patch available against recent kernels.

There are many Linux users that say that it's a lot simpler just to get an external modem and plug it in. But since new peripherals are mostly PnP today, you may eventually need to deal with it, so why delay it? Still, the most expedient (and expensive) solution is an external modem (if you have a free serial port).

6.2 External Modems

TOC

PnP External Modems

Many external modems are labeled "Plug and Play" (PnP) but they should all work fine as non-PnP modems. Since you usually plug the modem into a serial port which has its own IRQ number and IO address, the modem needs no PnP features to set these up. However, the serial port itself may need to be configured (IRQ number and IO address) unless the default configuration is OK.

How can an external modem be called PnP since it can't be configured by PnP? Well, it has a special PnP identification built into it that can be read (thru the serial port) by a PnP operating system. Such an operating system would then know that you have a modem on a certain port and would also know the model number. Then you might not need to configure application programs by telling them what port the modem is on (such as /dev/ttyS2 or COM3). But if you don't have such a PnP operating system you will need to configure your application program manually by giving it the /dev id (such as /dev/ttyS2).

Cabling & Installation

Connecting an external modem is simple compared to connecting most other devices to a serial port that require various types of "null modem" cables. Modems use straight through cable, with no pins crossed over. Most computer stores should have these. Make sure you get the correct gender. If you are using the DB9 or DB25 serial port at your computer, it will always be male which means that the connector on the cable should be female. Hook up your modem to one of your serial ports. If you are willing to accept the default IRQ and IO address of the port you connect it to, then you are ready to start your communication program and configure the modem itself.

What the Lights (LED's) Mean

- TM Test Modem
- AA Auto Answer (If on, your modem will answer an incoming call)
- RD Receive Data line = RxD
- SD Send Data line = TxD
- TR data Terminal Ready = DTR (set by your PC)
- RI Ring Indicator (If on, someone is "ringing" your modem)
- OH Off Hook (If off, your modem has hung up the phone line)
- MR Modem Ready = DSR ??
- EC Error Correction
- DC Data Compression
- HS High Speed (for this modem)

TOC

6.3 Internal Modems

An internal modem is installed in a PC by taking off the cover of the PC and inserting the modem card into a vacant slot on the motherboard. There are modems for the ISA slots and others for the PCI slots. While external modems plug into the serial port (via a short cable) the internal modems have the serial port built into the modem. In other words, the modem card is both a serial port and a modem.

Setting the IO address and IRQ for a serial port was formerly done by jumpers on the card. These are little black rectangular "cubes" about 5x4x2 mm in size which push in over pins on the card. Plug-and-Play modems (actually the serial port part of the modems) don't use jumpers for setting these but instead are configured by sending configuration commands to them (via IO address space on the ISA bus inside the computer). Such configuration commands can be sent by a PnP BIOS, the isapnp program (for the ISA bus only) or by a PnP operating system. The configuring of them is built into Windows 95/98 OSs. Under Linux you have a choice of ways (none of which is always easy) to io-irq configure them:

1. Use "isapnp" which may be run automatically at every boot-time
2. Use a PnP BIOS alone (which runs at every boot-time)
3. Patch Linux to make it a PnP operating system

6.4 Internal Modems You Must Avoid (winmodems, etc.)

A majority of internal modems made after about mid-1998 *don't* work with Linux since they are "winmodems" or the like. Names used include: HSP, HCF, and soft-... modem. Such modems turn over much (or even almost all) of the work of the modem to the main processor (CPU) chip of your computer (such as a Pentium chip). Since only Windows (and not Linux) software is provided to do this, it will not work under Linux. A list of modems which do and don't work under Linux is at Linux modem list

A better term for "winmodem" might be "software modem" or "soft-modem". But since this software is only for MS Windows, the term "winmodem" is now a good one since it also implies it's "Windows-only". The term "Winmodem" is a trademark for a certain type of "winmodem". Here is some more precise terminology regarding "winmodems":

HSP (Host Signal Processor) means that the host processor (your CPU chip) creates the code needed to produce the electrical signal on the phone line. The modem itself just creates whatever electrical waveshape the CPU tells it to. In contrast to this, a "controllerless" modem can create the waveshapes on its own (but can't control the modem). It contains no facilities to deal with bytes being sent and received. It can't compress strings of bytes; it can't check for errors; it can't put them into packets. In other words it can't control the modem but instead has the CPU do all this work using a program for Windows. The Rockwell HCF (Host Controlled Family) does this. If the software that does all this could be ported to Linux and then there wouldn't be this problem. Besides the above, a modem which doesn't simulate a serial port will not work under Linux.

How do you determine if an internal modem will work under Linux? If you don't know the model of the

modem and you also have Windows on your Linux PC, click on the "Modem" icon in the "Control Panel". First check out the modem list on the Web mentioned 3 paragraphs above. If that doesn't wo (or isn't feasible), you can look at the package it came in (or a manual) find the section on the package that says something like "Minimum System Requirements" or just "System Requirements". It may be in fine print. Read it closely. If Windows is listed as one of the requirements then it will likely not work under Linux.

Otherwise, it may work under Linux if it fails to state explicitly that you must have Windows. By saying it's "designed for Windows" it may only mean that it fully supports Microsoft's plug-and-play which is OK since Linux uses the same plug-and-play specs (but doesn't support them very well as yet). Being "designed for Windows" thus gives no clue as to whether or not it will work under Linux. You might check the Website of the manufacturer or inquire via email. I once saw a web-page that specifically stated that one model worked under Linux while implying that another model didn't.

As far as the author knows, there is no effort currently underway to support winmodems in Linux. You might request that modem manufacturers port their code to Linux (or the like).

6.5 Which Internal Modems are Best to Avoid ?

- Avoid: winmodems or the like. They will NOT work at all
- PCI Modems seldom work under Linux
- MWave and DSP Modems might work, but only if you first start Windows/Dos each time you power on your PC
- Modems with RPI (Rockwell) drivers work but with reduced performance

MWave and DSP Modems

Such modems use DSP's (Digital Signal Processors) which are programmed by algorithms which must be downloaded from the hard disk to the DSP's memory just before using the modem. Unfortunately, the downloading is done by Dos/Windows programs so one can't do it from Linux. Ordinary modems that work with Linux often have a DSP too (and may mention this on the packaging), but the program that runs it is stored inside the modem. This is not a "DSP modem" in the sense of this section and should work OK under Linux. An example of a DSP modem is IBM's Aptiva MWAVE.

If a DSP modem modem simulates a serial port, then it is usable with Linux which communicates with modems via the serial port. If you also have Dos/Windows on the same PC you may be able to use the modem: You first install the driver under DOS (using DOS and not Window drivers). Then start Dos/Windows (make sure the modem gets initialized) and without turning off the computer, go into Linux. One way to get to Linux is to use loadlin.exe which is a DOS program that will boot Linux from DOS (See Config-HOWTO). Another way is to just press CTRL-ALT-DEL. The modem remains on the same com port (same IO address) that it used under DOS.

Rockwell (RPI) Drivers

Modems that require Rockwell RPI drivers are not fully usable since the driver software doesn't work on Linux. The RPI does compression and error correction using MS Windows software on your computer's CPU. If you are willing to operate the modem without using the RPI (and have no compression nor error correction) then you may easily disable RPI by sending the modem (via the initialization string) a command to do so each time you power on your modem. On my modem this command is +H0. Not having data compression available may not be much of a handicap since most long files which you download from the Internet are already compressed and attempts at further compression may only slow things down a bit.

PCI Modems

A PCI modem card is one which inserts into a PCI-bus slot on the motherboard of a PC. Unfortunately, it seems that almost all PCI modems will not work under Linux. But since a few people have gotten PCI modems to work under Linux it seems that a tiny minority of such modems may work under Linux. Some claim that no PCI modem will work under Linux but I've seen a few messages claiming it was done (in one case by using jumpers).

After you install a PCI modem, look at /proc/pci. If you see a high main memory address (like 0xfebfff00), it probably works by shared memory which is not supported by Linux. If there is no such high memory address and the only address you see is an IO address under 0xffff ??, then it might work OK. I'm not really sure of the /proc/pci signature to look for to determine whether or not the modem will work under Linux. If this request is still in the latest version of this HOWTO, please send me info on what /proc/pci looks like for PCI modems that work under Linux or let me know what signatures to look for in /proc/pci to determine whether or not the modem will work with Linux.

---

7. Serial Port Devices /dev/ttyS2, etc.

For creating devices in the device directory see the Serial-HOWTO: "Creating Devices In the /dev directory".

7.1 Serial Port Device Names & Numbers

Devices in Linux have major and minor numbers. Each serial port may have 2 possible names in the /dev directory: ttyS and cua. Their drivers behave slightly differently. The cua device is deprecated and will not be used in the future. See The cua Device.

Dos/Windows use the COM name while the `setserial` program uses tty00, tty01, etc. Don't confuse these with dev/tty0, dev/tty1, etc. which are used for the console (your PC monitor) but are not serial ports. The dos names (COM1, etc.) and IO address is shown below for the "standard" case (but yours could be different).

```
        set-                                                    IO
dos    serial          major minor         major minor   address
COM1   tty00  /dev/ttyS0  4,  64;  /dev/cua0  5,  64       3F8
COM2   tty01  /dev/ttyS1  4,  65;  /dev/cua1  5,  65       2F8
COM3   tty02  /dev/ttyS2  4,  66;  /dev/cua2  5,  66       3E8
COM4   tty03  /dev/ttyS3  4,  67;  /dev/cua3  5,  67       2E8
```

Note that all distributions should come with ttyS devices (and cua devices until cua is finally abolished). You can verify this by typing:

```
linux% ls -l /dev/cua*
linux% ls -l /dev/ttyS*
```

7.2 Link ttySN to /dev/modem ?

On some installations, two extra devices will be created, `/dev/modem` for your modem and `/dev/mouse` for your mouse. Both of these are symbolic links to the appropriate device in `/dev` which you specified during the installation (unless you have a bus mouse, then `/dev/mouse` will point to the bus mouse device).

There has been some discussion on the merits of `/dev/mouse` and `/dev/modem`. The use of these links is discouraged. In particular, if you are planning on using your modem for dialin you may run into problems because the lock files may not work correctly if you use `/dev/modem`. Use them if you like, but *be sure they point to the right device*. However, if you change or remove this link, some applications might need reconfiguration.

7.3 The cua Device

TOC

Each ttyS device has a corresponding cua device. But the cua device is to be abolish so it's best to use ttyS (unless cua is required). There is a difference between cua and ttyS but a savvy programmer can make a ttyS port behave just like a cua port so there is no real need for the cua anymore. Except some older programs may need to use the cua.

What's the difference? The main difference between cua and ttyS has to do with what happens in a C-program when an ordinary "open" command tries to open the port. If a cua port has been set to check modem control signals, the port can be opened even if the DCD modem control signal says not to. Astute programming (by adding additional lines to the program) can force a ttyS port to behave this way also. But a cua port can be more easily programmed to open for dialing out on a modem even when the modem fails to assert DCD (since no one has called into it and there's no carrier). That's why cua was once used for dial-out and ttyS used for dial-in.

Starting with Linux kernel 2.2, a warning message will be put in the kernel log when one uses cua. This is an omen that cua is on the way out.

---

8. Interesting Programs You Should Know About

8.1 What is setserial ?

Introduction

Don't ever use `setserial` with Laptops (PCMCIA). `setserial` is a program which allows you to tell the device driver software the I/O address of the serial port, which interrupt (IRQ) is set in the port's hardware, what type of UART you have, etc. It can also show how the driver is currently set. In addition, it can probe the hardware (if certain options are given). Besides the manual for `setserial`, check out info in `/usr/doc/setserial.../` or the like. It should tell you how setserial is handled in your distribution of Linux.

`Setserial` is often run automatically at boot-time by a start-up shell-script. It will only work if the serial module is loaded. If you should for some reason unload the serial module later on, the changes previously made by `setserial` will be forgotten. So `setserial` must be run again to reestablish them. In addition to running via a start-up script, something akin to `setserial` runs when the serial module is loaded. Thus when you watch the start-up messages on the screen it may look like it ran twice, and in fact it has.

With appropriate options, `setserial` can probe (at a given I/O address) for a serial port but you must guess the I/O address. If you ask it to probe for /dev/ttyS2 for example, it will only probe at the address it thinks ttyS2 is at. If you tell setserial that ttyS2 is at a different address, then it will probe at that address, etc. See Probing

Setserial does not set either IRQ's nor I/O addresses in the serial port hardware itself. You must tell setserial the identical values that have been set in the hardware. It's set in the hardware either by jumpers or by plug-and-play. Do not just invent some values that you think would be nice to use and tell them to setserial. However, if you know the I/O address but don't know the IRQ you may command setserial to attempt to determine the IRQ.

You can see a list of possible commands to use (but not the one-letter options such as -v for verbose --which you should normally use when troubleshooting) by just typing `setserial` with no arguments. Note that setserial calls an I/O address a "port". If you type:

```
setserial-g /dev/ttyS*
```

you'll see some info about how that device driver is configured for your ports. Add a "v" to the option "-g" to see more. But this doesn't tell you if the hardware actually has these values set in it. If fact, you can run setserial and assign a purely fictitious I/O address, any IRQ, and whatever uart type you would like to have. Then the next time you type "setserial ..." it will display these bogus values without complaint. Note that assignments made by setserial are lost when the PC is powered down so it is usually run automatically somewhere each time that Linux is booted.

TOC

Probing

In order to try to find out if you have a certain piece of serial hardware you must first know (or guess) its I/O address (or the device driver must have an I/O address for it, likely previously set by setserial). To try to detect the physical hardware use the -v (verbose) and `autoconfig` command to `setserial`. If the resulting message shows a uart type such as 16550A, then you're OK. If instead it shows "`unknown`" for the uart type, then there is likely no serial port at all at that I/O address. Some cheap serial ports don't identify themselves correctly so if you see "`unknown`" you still might have something there.

Besides auto-probing for uart type, setserial can auto-probe for IRQ's but this doesn't always work right either. In versions of setserial >= 2.15, your last probe test may be saved and put into the configuration file `/etc/serial.conf` which will be used next time you start Linux. The script that runs `setserial` at boot-time does not usually probe, but you could change it so that it does. See the next section.

Can Linux Configure The Serial Devices Automagically?

Yes, but ... You may set Linux up to detect and set up the serial devices automatically on startup (probing). It's easy to do for setserial < 2.15. Just add some lines to the file that runs setserial on start-up. See Old configuration method: edit a script For example, for ttyS3 you would add:

```
/sbin/setserial /dev/ttyS3 auto_irq skip_test autoconfig
```

to the file that runs setserial on startup. Do this for every serial port you want to auto configure. Be sure to give a device name that really does exist on your machine.

For versions >= 2.15 (provided your distribution implemented the change, Redhat didn't) it's much harder to do since the file that runs setserial on startup, /etc/init.d/setserial or the like was not intended to be edited by the user. There may be no helpful comments in it like there were in earlier versions.

Boot-time Configuration

When the kernel loads the serial module (or if the "module" is built into the kernel) then only `ttyS{0-3}` are auto-detected and the driver is set to IRQs 4 and 3 (regardless of what the hardware is actually set at). You see this as a boot-time message just like as if `setserial` had been run.. If you use 3 or more ports, this may result in IRQ conflicts.

To fix such conflicts by telling setserial the true IRQs (or for other reasons) there may be a file somewhere that runs `setserial` again. This happens early at boot-time before any process uses the serial port. In fact, your distribution may have set things up so that the setserial program runs automatically from a start-up script at boot-time. More info about how to handle this situation should be found in /usr/doc/setserial.../ or the like.

TOC

New configuration method using /etc/serial.conf

Starting with version 2.15 (1999) of `setserial` the shell-script that runs from a file at boot-time gets its data from a configuration file: /etc/serial.conf. But serial.conf is normally never edited. Instead just use `setserial` on the command line. Normally, what you changed with the setserial command is saved to the configuration file (serial.conf) when you shutdown (normally) or reboot. This only works if "###AUTOSAVE### or the like is on the first line of serial.conf. If you should use `setserial` experimentally and it doesn't work out right, then don't forget to redo it so that the experimental settings don't get saved by mistake. The file most commonly used to run setserial at boot-time (in conformance with the configuration file) is now /etc/init.d/setserial (Debian) or /etc/init.d/serial (Redhat), or etc., but it also should not normally be edited.

To disable a port, use `setserial` to set it to "uart none". The format of /etc/serial.conf appears to be just like that of the parameters placed after "setserial" on the command line with one line for each port. If you don't use autosave, you may edit /etc/serial.conf manually. For 2.15, the Debian distribution installs the system with autosave enabled, but Redhat 6.0 just had a file /usr/doc/setserial-2.15/rc.serial which you have to move to /etc/init.d/.

BUG: As of July 1999 there is a bug/problem since with ###AUTOSAVE### only the setserial parameters displayed by "setserial -G /dev/ttyS?" (where ? = 0, 1, 2, ...) get saved but the other parameters don't get saved. This will only affect a small minority of users since the parameters not saved are seldom used anyway. It's been reported as a bug and may be fixed by now.

In order to force the current settings set by setserial to be saved to the configuration file (serial.conf) without shutting down, do what normally happens when you shutdown: Run the shell-script `/etc/init.d/{set}serial stop`. The "stop" command will save the current configuration but the serial ports still keep working OK.

Old configuration method: edit a script

Prior to 2.15 (1999), or if your distribution chose not to use /etc/serial.conf then you need to look for a file that runs setserial at boot-time. If it doesn't exist, you need to create one (or place the commands in a file that runs early at boot-time). If such a file is currently being used it's likely somewhere in the /etc directory-tree. But Redhat has supplied it in /usr/doc/setserial/ although you need to move it to the /etc tree before using it. You might use "locate" to try to find such a file. For example, you could type: locate "*serial*".

What you are looking for could be named rc.serial, or 0setserial (Debian). If such a file is supplied, it should contain a number of commented-out examples. By uncommenting some of these and/or modifying them, you should be able to set things up correctly. Make sure that you are using a valid path for `setserial`, and a valid device name. You could do a test by executing this file manually (just type its name as the super-user) to see if it works right. Testing like this is a lot faster than doing repeated reboots to get it right. Of course you can also test a single `setserial` command by just typing it on the command line.

The script `/etc/rc.d/rc.serial` was commonly used in the past. The Debian distribution used `/etc/rc.boot/0setserial`. Another file once used is `/etc/rc.d/rc.local` but it's not a

good idea to use this since it may not be run early enough. It's been reported that other processes may to open the serial port before rc.local runs resulting in serial communication failure.

IRQs

By default, both ttyS0 and ttyS2 share IRQ 4, while ttyS0 and ttyS3 share IRQ 3. But sharing serial interrupts is not permitted unless you: 1. have kernel 2.2 or better, and 2. you've complied in support for this, and 3. your serial hardware supports it. See

Interrupt sharing and Kernels 2.2+ If you only have two serial ports, ttyS0 and ttyS1, you're still OK since IRQ sharing conflicts don't exist for non-existent devices.

If you add an internal modem and retain ttyS0 and ttyS1, then you should attempt to find an unused IRQ and set it both on your serial port (or modem card) and then use setserial to assign it to your device driver. If IRQ 5 is not being used for a sound card, this may be one you can use for a modem. To set the IRQ in hardware you may need to use isapnp, a PnP BIOS, or patch Linux to make it PnP. To help you determine which spare IRQ's you might have, type "man setserial" and search for say: "IRQ 11".

----------From TT-HOWTO----------

Setserial tells the serial port driver both the interrupt (IRQ) number and the port IO address. It can tell you what type of UART chips you have if you use the autoconfig parameter. For the serial port to work, the serial module (for the Linux kernel) must be loaded. If you use more than 2 serial ports and want to assign unique IRQ numbers (or use more than 4 serial ports), then you need to use setserial. If your serial port is plug-and-play you should look in the Serial-HOWTO. In addition to the man page for it, but sure to look at /usr/doc/setserial/ if it exists. It may tell you how setserial is run automatically at boot time and how to alter the configuration. Don't ever use setserial with Laptops (PCMCIA bus).

starting with version 2.15 (1999) of `setserial` changes made to setserial on the command line are sometimes automatically saved so that they will remain after you shut off the computer.

Where to Run Setserial ?

Starting with version 2.15 of `setserial` the method of configuration changed. Before 2.15 you configured it by editing a shell-script that ran `setserial`. After 2.15 there's a configuration file /etc/serial.conf which is automatically edited to conform with whatever you do with the `setserial` command.

You may find you have both the old and new way of configuring installed. If the shell-script is named "...pre-2.15" or the like, it's probably not being used anymore so everything should be OK if you keep it.

Prior to version 2.15

Prior to 2.15 there was no configuration file for `setserial`. One needs to edit the shell-script that runs setserial at boot-time. You need to find it or create one if it doesn't exist. Where it is depends on your distribution. It might be in a file named "rc.serial" . In the Debian distribution it was the 0setserial file in /etc/rc.boot. One way to try to find it is to search for it with the "locate" command.

TOC

Version 2.15 and later

You neither edit the file that runs setserial nor the configuration file. Changes made by using `setserial` on the command line are *sometimes* saved in /etc/serial.conf during normal shutdown. This feature is called autosave. It only works if the first line of serial.conf contains ###AUTOSAVE### or the like.

There's a bug I've reported in 2.15 and it's been fixed (but you could still have it): With the bug, seldom used changes made by setserial don't get saved. Only the most important parameters that you see if you give the command "setserial -g /dev/ttyS?" do get saved.

If autosave is enabled, then each time you boot, setserial is run by /etc/init.d/setserial (or "serial", etc.) using the saved configuration.

8.2 What is isapnp ?

`isapnp` is a program to configure Plug-and-Play (PnP) devices on the ISA bus including internal modems. It comes in a package called "isapnptools" and includes another program, "pnpdump" which finds all your ISA PnP devices and shows you options for configuring them in a format which may be added to the PnP configuration file: /etc/isapnp.conf. It may also be used with the --dumpregs option to show the current IO address and IRQ of the modem's serial port. The isapnp command may be put into a startup file so that it runs each time you start the computer and thus will configure ISA PnP devices. It is able to do this even if your BIOS doesn't support PnP. See Plug-and-Play-HOWTO.

8.3 What is wvdialconf ?

`wvdialconf` will try to find which serial port has a modem on it. It also creates a configuration program for the wvdial program. `wvdial` is used for simplified dialing out using the PPP protocol to an ISP. But you don't need to install PPP in order to use `wvdialconf`. It will only find modems which are not in use. It will also automatically devise a "suitable" init strings but sometimes gets it wrong. Since this command has no options, it's simple to use but you must give it the name of a file to put the init string (and other data) into. For example type: wvdialconf my_file_name.

8.4 What is stty ?

`stty` is like setserial but it sets the baud rate and other parameters of a serial port. Typing "stty -a < /dev/ttyS2" should show you how ttyS2 is configured. Most of the settings are for things that you never need to use with modems (such as some used only for old terminals of the 1970s). Your communication package should automatically set up all the setting correctly for modems. But stty is sometimes useful for trouble-shooting.

Two items set by stty are: 1. Hardware flow control by "crtscts" and 2. Ignore the DCD signal from the modem: "clocal". If the modem is not sending a DCD signal and clocal is disabled (stty shows -clocal) then a program may not be able to open the serial port. If the port can't open, the program may just hang, waiting (often in vain) for a DCD signal from the modem.

Minicom sets clocal automatically when it starts up so there is no problem. But version 6.0.192 of K hung when I set -clocal and tried to "set line ..." If -clocal is set and there is no DCD signal then even the "stty" command will hang and there is seemingly no way to set clocal (except by running minicom). But minicom will restore -clocal when it exits. One way to get out of this is to use minicom to send the "AT&C" to the modem (to get the DCD signal) and then exit minicom with no reset so that the DCD signal remains on. Then you may use stty again.

---

9. Trying Out Your Modem (Dialing Out) .

9.1 Are You Ready to Dial Out ?

Once you've plugged in your modem and know which serial port it's on you're ready to try using it. Before you try to get the Internet on it or have people call in to you, first try something simpler like dialing out to some number to see if your modem is working OK. Find a phone number that is connected to a modem. It you don't know what number to call, ask at computer stores for such phone numbers of bulletin boards, etc. or see if a local library has a phone number for their on-line catalog.

Then make sure you are ready to phone. Do you know what serial port (such as ttyS2) your modem is on? You should have found this out when you io-irq configured your serial ports. Have you decided what speed you are going to use for this port? See Speed Table for a quick selection or What Speed Should I Use for more details. If you have no idea what speed to set, just set it a few times faster than the advertised speed of your modem. Also remember that if you see a menu where an option is "hardware flow control" and/or "RTS/CTS" or the like, select it. Is a live telephone cable plugged in to your modem? You may want to connect the cable to a real telephone to make sure that it can produce a dial tone.

Now you need to select a communication (dialing) program to use to dial out. Dialing programs include: minicom, seyon (X-windows), and kermit. See section Communications Programs about some communications programs. Two examples are presented next: Dialing Out with Minicom and Dialing Out with Kermit

9.2 Dialing Out with Minicom

Minicom comes with most Linux distributions. To configure it you should be the root user. Type "minicom -s" to configure. This will take you directly to the configuration (set-up) menus. Alternatively you could just run "minicom" and then type ^A to see the bottom status line. This shows to type ^A Z for help (you've already typed the ^A so just type z). From the help menu go to the Configuration menu.

Most of the options don't need to be set for just simply dialing out. To configure you have to supply a few basic items: the name of the serial port your modem is on such as /dev/ttyS2 and the speed such as 115200. These are set at the serial port menu. Go to it and set them. Also (if possible) set hardware flow control (RTS/CTS). Then save them. When typing in the speed, you should also see something like "8N1" which you should leave alone. It means: 8-bit bytes, No parity, 1 stop-bit appended to each byte. If you can't find the speed you want, a lower speed will always work for a test. Exit (hit return) when done and save the configuration as default (dfl) using the menu. You may want to exit minicom and start it again so it can now find the serial port and initialize the modem, or you could go to help and tell minicom to initialize the modem.

Now you are ready to dial. But first at the main screen you get after you first type "minicom" make sure there's a modem there by typing AT and then hit the <enter> key. It should display OK. If it doesn't something is wrong and there is no point of trying to dial.

If you got the "OK" go back to help and select the dialing directory. You may edit it and type in a phone number, etc. into the directory and then select "dial" to dial it. Alternatively, you may just dial manually (by selecting "manual" and then type the number at the keyboard). If it doesn't work, carefully note any error messages and try to figure out what went wrong.

9.3 Dialing Out with Kermit

TOC

You can find the latest version of `kermit` at http://www.columbia.edu/kermit/. For example, say your modem was on `ttyS3`, and it's speed was 115200 bps. You would do the following:

```
linux# kermit
C-Kermit 6.0.192, 6 Sep 96, for Linux
 Copyright (C) 1985, 1996,
  Trustees of Columbia University in the City of New York.
Default file-transfer mode is BINARY
Type ? or HELP for help.
C-Kermit>set line /dev/ttyS3
C-Kermit>set carrier-watch off
C-Kermit>set speed 115200
/dev/ttyS3, 115200 bps
C-Kermit>c
Connecting to /dev/ttyS3, speed 115200.
The escape character is Ctrl-\ (ASCII 28, FS)
Type the escape character followed by C to get back,
or followed by ? to see other options.
ATE1Q0V1                          ; you type this and then the Enter key
OK                                ; modem should respond with this
```

If your modem responds to `AT` commands, you can assume your modem is working correctly on the Linux side. Now try calling another modem by typing:

```
ATDT7654321
```

where 7654321 is a phone number. Use ATDP instead of ATDT if you have a pulse line. If the call goes through, your modem is working.

To get back to the `kermit` prompt, hold down the Ctrl key, press the backslash key, then let go of the Ctrl key, then press the C key:

```
Ctrl-\-C
(Back at linux)
C-Kermit>quit
linux#
```

This was just a test using the primitive "by-hand" dialing method. The normal method is to let `kermit` do the dialing for you with its built-in modem database and automatic dialing features, for example using a US Robotics (USR) modem:

```
linux# kermit
C-Kermit 6.0.192, 6 Sep 1997, for Linux
 Copyright (C) 1985, 1996,
  Trustees of Columbia University in the City of New York.
Default file-transfer mode is BINARY
Type ? or HELP for help
C-Kermit>set modem type usr        ; Select modem type
```

```
C-Kermit>set line /dev/ttyS3       ; Select communication device
C-Kermit>set speed 115200          ; Set the dialing speed
C-Kermit>dial 7654321              ; Dial
 Number: 7654321
 Device=/dev/ttyS3, modem=usr, speed=115200
 Call completed.<BEEP>
Connecting to /dev/ttyS3, speed 115200
The escape character is Ctrl-\ (ASCII 28, FS).
Type the escape character followed by C to get back,
or followed by ? to see other options.

Welcome to ...

login:
```

10. Dial-In

10.1 Overview

Dial-in is where you set up your PC so that others may dial in to your phone number and use your PC. The "point of view" is your PC. When you dial out from your PC you are also dialing in to another computer (but not dialing in to your own computer)

Dialin works like this. Someone with a modem dials your telephone number. Your modem answers the call and connects. Once the caller is connected, your PC sends a login prompt to the caller. Then the caller logs in and uses your PC. Using your PC may mean that the caller has a shell account and can use your PC just as if they logged in at the console (or text-terminal). It could also mean that they get connected to the Internet. The program that you use at your PC to handle dialin is called `getty`.

Often, after login, another program is run, including programs to connect the caller to the Internet. This HOWTO doesn't cover such programs that run after your login. For connecting to the internet, the PPP program is usually run (see PPP-HOWTO, a new revision is expected soon). Login may be automated so that the communication program automatically types in a "name" and password each time.

10.2 Getty

`getty` is the program you run for dialin. You don't need it for dialout. In addition to presenting a login prompt, it also answers the telephone. Originally getty was used for logging in to a computer from a dumb terminal. It's currently used for logging in to a Linux console). There are a few different getty programs with slightly different names. Only certain ones work with modems for dialin. This `getty` program is usually started at boot-time. It must be called from the /etc/inittab file. You may find an example in this file of a call to getty which you will likely need to edit a bit. If you use a different getty program than the one shown in such an example, then you will need to edit it quite a bit since the options will have a different format.

There are four different getty programs to choose from that may be used with modems for dialin: `mgetty`, `uugetty`, `getty_em`, and `agetty`. A few details are given in the following subsections. `agetty` is the simplest (and weakest) of the four and some consider it mainly for use with directly connected text-terminals. `mgetty` has support for fax and voice mail but `Uugetty` doesn't. `mgetty` allegedly lacks a few of the features of `uugetty`. `getty_em` is a simplified version of `uugetty`. Thus `mgetty` is likely your best choice unless you are already familiar with `uugetty` (or find it difficult to get `mgetty`). The syntax for these getty programs differs, so be sure to check that you are using the correct syntax in `/etc/inittab` for whichever getty you use.

TOC

About mgetty

`mgetty` was written as a replacement for `uugetty` which was in existence long before `mgetty`. Both are for use with modems. Although `mgetty` may be also used for directly connected terminals the documentation for this is hard to pinpoint and mgetty will not (as of mid 1999) support software flow control (used on many terminals) without recompiling. This defect is listed as a bug. In addition to allowing dialup logins, `mgetty` also provides FAX support and auto PPP detection. There is a supplemental program called `vgetty` which handles voicemail for some modems. `mgetty` documentation is good (except for voice mail), and does not need supplementing . Please refer to it for installation instructions. You can find the latest information on `mgetty` at http://www.leo.org/~doering/mgetty/ and http://alpha.greenie.net/mgetty

About uugetty

`getty_ps` contains two programs: `getty` is used for console and terminal devices, and `uugetty` for modems. Greg Hankins (former author of Serial-HOWTO) used `uugetty` so his writings about it are included here. See Uugetty. The other gettys are well covered by the documentation that comes with them.

About getty_em

This is a simplified version of ``uugetty''. It was written by Vern Hoxie after he became fully confused with complex support files needed for getty_ps and uugetty.

It is part of the collection of serial port utilities and information by Vern Hoxie available via ftp from scicom.alphacdc.com. The name of the collection is ``serial_suite.tgz''. When logging into ``scicom'' as "anonymous", you must use your full e-mail address as the password. For example: greg.hankins@cc.gatech.edu

About agetty and mingetty

`agetty` is a simple, completely functional implementation of `getty` which is best suited for virtual consoles or terminals rather than modems. But it works fine with modems under favorable conditions (except you cannot dial out when agetty is running and waiting for a call). `agetty` in the Debian distribution is just named `getty`.

`mingetty` is a small getty that will work only for consoles (monitors) so you can't use it with modems for dialin.

10.3 What Happens when Someone Dials In ?

The caller runs some sort of communication program that dials your telephone number and your telephone rings. There are two different ways that your PC can answer the phone. One way is for the modem to automatically answer the call. The other way is for getty to sense the ringing and send a command to the modem to answer the call. Once the call is answered, your modem sends tones to the

other modem (and conversely). The two modems negotiate how they will communicate and when th done your modem sends a "CONNECTed" message (or the like) to `getty`. When `getty` gets this message, it sends a login prompt out the serial port. Sometimes `getty` just calls on a program named `login` to handle the logging in. `getty` usually starts running at boot-time but it must wait until a connection is made before sending out a "login" prompt.

Now for more details on the two methods of answering the call. By setting the S0 register of the modem to 3, the modem will automatically answer on the 3rd ring. If it's set to 0 then the modem will only answer the call if getty sends it an "A" (= Answer) command while the phone is ringing. Actually an "ATA" is sent since all modem commands are prefixed by "AT". You might think it best to utilize the ability of the modem to automatically answer the call, but it's actually better if `getty` answers it. If the modem doesn't automatically answer, it's called manual answer (even though `getty` automatically handles it).

For the "manual" answer case, `getty` opens the port at boot-time and listens. When the phone rings, a "RING" message is sent to the listening `getty`. Then if `getty` wants to answer this ring, it sends the modem an "ATA" command. The modem then makes a connection and sends a "CONNECT ..." message to `getty` which then sends a login prompt to the caller.

The automatic answer case uses the CD (Carrier Detect) wire from the modem to the serial port to detect when a connection is made. It works like this. At boot-time `getty` tries to open the serial port but the attempt fails since there is normally no CD signal from the modem. Then the `getty` program waits at the open statement in the program until a CD signal appears. When a CD signal arrives (perhaps hours later) then the port is opened and `getty` sends the login prompt. While `getty` is waiting (sleeping) at the open statement, other processes can run since Linux is a multiprocessing operating system. What actually wakes `getty` up is an interrupt which is issued when the CD line from the modem changes state to on.

You may wonder how getty is able to open the serial port in the manual-answer case since there is no CD signal. Well, there's a way to write a program to force the port to open even if there is no CD signal present.

10.4 Why Manual Answer is Best

The difference between the two ways of answering will show itself when the computer happens to be down but the modem is still working. For the manual case, the "RING" message is sent to getty but since the computer is down, getty isn't there and the phone never gets answered. There are no telephone charges when there is no answer. For the automatic answer case, the phone is answered but no login message is ever sent since the computer is down. The phone bill runs up as the waiting continues. If the phone call is toll-free, it doesn't make much difference, although it may be frustrating waiting for a login prompt that never arrives. `mgetty` uses manual answer. `Uugetty` can do this too using a configuration script.

10.5 Callback

Callback is where someone first dials in to your modem. Then, you get a little info from the caller and then call it right back. Why would you want to do this? One reason is to save on telephone bills if you can call the caller cheaper than the caller can call you. Another is to make sure that the caller really is who it claims to be. If a caller calls you and claims to be calling from its usual phone number, then one way to verify this is to actually place a new call to that number.

There's a program for Linux called "callback" that works with mgetty. It's at ftp://ftp.icce.rug.nl/pub/unix/. Step-by-step instructions on how someone installed it (and PPP) is at http://www.stokely.com/unix.serial.port.resources/callback.html

10.6 Voice Mail

Voice mail is like an answering machine run by a computer. To do this you must have a modem that supports "voice" and supporting software. Instead of storing the messages on tape, they are stored in digital format on a disk. When a person phones you, they hear a "greeting" message and can then leave a message for you. More advanced systems would have caller-selectable mail boxes and caller-selectable messages to listen to. Free software is available in Linux for simple answering, but doesn't seem to be available yet for the more advanced stuff.

I know of two different voicemail packages for Linux. One is a very minimal package (see Voicemail Software). The other, more advanced, but currently poorly documented, is `vgetty`. It's an optional addition to the well documented and widely distributed `mgetty` program. It supports ZyXEL-like voice modem commands. In the Debian distribution, you must get the mgetty-voice package in addition to the mgetty package and mgetty-doc package. Obsolete documentation has been removed from mgetty but replacement documentation is lacking (except if you use the -h (help) option when running certain programs, etc.). But one sees postings about using it on the mgetty newsgroup. See About mgetty and >. It seems that `vgetty` is currently not very stable but it's successfully being used and development of it continues. If this is the latest version of this HOWTO can someone who is familiar with vgetty please let me know its current status.

---

11. Uugetty for Dial-In (from the old Serial-HOWTO)

Be aware that you could use mgetty as a (better?) alternative to `uugetty`. `mgetty` is newer and more popular than uugetty. See What is getty? for a brief comparison of these 2 gettys.

11.1 Installing getty_ps

Since uugetty is part of getty_ps you'll first have to install getty_ps. If you don't have it, get the latest version from `metalab.unc.edu:/pub/Linux/system/serial`. In particular, if you want to use high speeds (57600 and 115200 bps), you must get version 2.0.7j or later. You must also have libc 5.x or greater.

By default, `getty_ps` will be configured to be Linux FSSTND (File System Standard) compliant, which means that the binaries will be in `/sbin`, and the config files will be named `/etc/conf.{uu}getty.ttyS`*N*. This is not apparent from the documentation! It will also expect lock files to go in `/var/lock`. Make sure you have the `/var/lock` directory.

If you don't want FSSTND compliance, binaries will go in `/etc`, config files will go in `/etc/default/{uu}getty.ttyS`*N*, and lock files will go in `/usr/spool/uucp`. I recommend doing things this way if you are using UUCP, because UUCP will have problems if you move the lock files to where it isn't looking for them.

`getty_ps` can also use `syslogd` to log messages. See the man pages for `syslogd(1)` and `syslog.conf(5)` for setting up `syslogd`, if you don't have it running already. Messages are logged with priority LOG_AUTH, errors use LOG_ERR, and debugging uses LOG_DEBUG. If you don't want to use `syslogd` you can edit `tune.h` in the `getty_ps` source files to use a log file for messages instead, namely `/var/adm/getty.log` by default.

Decide on if you want FSSTND compliance and syslog capability. You can also choose a combination of the two. Edit the `Makefile`, `tune.h` and `config.h` to reflect your decisions. Then compile and install according to the instructions included with the package.

11.2 Setting up uugetty

With `uugetty` you may dial out with your modem while `uugetty` is watching the port for logins. The original author of this HOWTO said to use /dev/cuaN for dial-out (with uugetty running on /dev/ttyN). but others say you must use /dev/ttyN for both dial-in and dial-out ??

`uugetty` does important lock file checking. Update `/etc/gettydefs` to include an entry for your modem. When you are done editing `/etc/gettydefs`, you can verify that the syntax is correct by doing:

```
linux# getty -c /etc/gettydefs
```

Modern Modems

If you have a 9600 bps or faster modem with data compression, you can lock your serial port to one speed. For example:

```
# 115200 fixed speed
F115200# B115200 CS8 # B115200 SANE -ISTRIP HUPCL #@S @L @B login: #F115200
```

If you have your modem set up to do RTS/CTS hardware flow control, you can add `CRTSCTS` to the entries:

```
# 115200 fixed speed with hardware flow control
F115200# B115200 CS8 CRTSCTS # B115200 SANE -ISTRIP HUPCL CRTSCTS #@S @L @B login:
#F115200
```

Old slow modems

If you have a slow modem (under 9600 bps) Then, instead of one line for a single speed, your need several lines to try a number of speeds. Note the these lines are linked to each other by the last "word" in the line such as #38400. Blank lines are needed between each entry.

```
# Modem entries
115200# B115200 CS8 # B115200 SANE -ISTRIP HUPCL #@S @L @B login: #57600

57600# B57600 CS8 # B57600 SANE -ISTRIP HUPCL #@S @L @B login: #38400

38400# B38400 CS8 # B38400 SANE -ISTRIP HUPCL #@S @L @B login: #19200

19200# B19200 CS8 # B19200 SANE -ISTRIP HUPCL #@S @L @B login: #9600

9600# B9600 CS8 # B9600 SANE -ISTRIP HUPCL #@S @L @B login: #2400

2400# B2400 CS8 # B2400 SANE -ISTRIP HUPCL #@S @L @B login: #115200
```

Login Banner

If you want, you can make `uugetty` print interesting things in the login banner. In Greg's examples, he has the system name, the serial line, and the current bps rate. You can add other things:

```
@B    The current (evaluated at the time the @B is seen) bps rate.
@D    The current date, in MM/DD/YY.
@L    The serial line to which uugetty is attached.
@S    The system name.
@T    The current time, in HH:MM:SS (24-hour).
@U    The number of currently signed-on users.  This is  a
      count of the number of entries in the /etc/utmp file
      that have a non-null ut_name field.
@V    The value of VERSION, as given in the defaults file.
To display a single '@' character, use either '\@' or '@@'.
```

11.3 Customizing uugetty

There are lots of parameters you can tweak for each port you have. These are implemented in separate config files for each port. The file `/etc/conf.uugetty` will be used by *all* instances of `uugetty`, and `/etc/conf.uugetty.ttyS`*N* will only be used by that one port. Sample default config files can be found with the `getty_ps` source files, which come with most Linux distributions. Due to space concerns, they are not listed here. Note that if you are using older versions of `uugetty` (older than 2.0.7e), or aren't using FSSTND, then the default file will be `/etc/default/uugetty.ttyS`*N*. Greg's `/etc/conf.uugetty.ttyS3` looked like this:

```
# sample uugetty configuration file for a Hayes compatible modem to allow
# incoming modem connections
#
# alternate lock file to check... if this lock file exists, then uugetty is
# restarted so that the modem is re-initialized
ALTLOCK=cua3
ALTLINE=cua3
# line to initialize
INITLINE=cua3
```

```
# timeout to disconnect if idle...
TIMEOUT=60
# modem initialization string...
# format: <expect> <send> ... (chat sequence)
INIT="" AT\r OK\r\n
WAITFOR=RING
CONNECT="" ATA\r CONNECT\s\A
# this line sets the time to delay before sending the login banner
DELAY=1
#DEBUG=010
```

Add the following line to your `/etc/inittab`, so that `uugetty` is run on your serial port, substituting in the correct information for your environment - run-levels (2345 or 345, etc.) config file location, port, speed, and default terminal type:

```
S3:2345:respawn:/sbin/uugetty -d /etc/default/uugetty.ttyS3 ttyS3 F115200 vt100
```

Restart `init`:

```
linux# init q
```

For the speed parameter in your `/etc/inittab`, you want to use the highest bps rate that your modem supports.

Now Linux will be watching your serial port for connections. Dial in from another machine and login to you Linux system.

`uugetty` has a lot more options, see the man page for `uugetty)` (often just called `getty`) for a full description. Among other things there is a scheduling feature, and a ringback feature.

---

12. What Speed Should I Use with My Modem?

By "speed" we really mean the "data flow rate" but almost everybody incorrectly calls it speed. For all modern modems you have no choice of the speed that the modem uses on the telephone line since it will automatically choose the highest possible speed that is possible under the circumstances. But you do have a choice as to what speed will be used between your modem and your computer. This is sometimes called "DTE speed" where "DTE" stands for Data Terminal Equipment (Your computer is a DTE.) You need to set this speed high enough so this part of the signal path will not be a bottleneck. The setting for the DTE speed is the maximum speed of this link. Most of the time it will likely operate at lower speeds.

For an external modem, DTE speed is the speed (in bits/sec) of the flow over the cable between you modem and PC. For an internal modem, it's the same idea since the modem also emulates a serial port. It may seem ridiculous having a speed limit on communication between a computer and a modem card that is directly connected inside the computer to a much higher speed bus. But it's that way since the modem card probably includes a dedicated serial port which does have speed limits (and settable speeds).

12.1 Speed and Data Compression

What speed do you choose? If it were not for "data compression" one might try to choose a DTE speed exactly the same as the modem speed. Data compression takes the bytes sent to the modem from your computer and encodes them into a fewer number of bytes. For example, if the flow (speed) from the PC to the modem was 20,000 bytes/sec (bps) and the compression ratio was 2 to 1, then only 10,000 bytes/sec would flow over the telephone line. Thus for a 2:1 compression ratio you need to set the speed double the maximum modem speed on the phone line. If the compression ratio is 3 to 1 you need to set it 3 times faster.

12.2 Where do I Set Speed ?

This DTE speed is normally set by a menu in your communications program or by an option given to the getty command if someone is dialing in. You can't set the DCE modem-to-modem speed.

12.3 Can't Set a High Enough Speed

You need to find out the highest speed supported by your hardware. As of late 1998 most hardware only supported speeds up to 115.2K bps. A few 56K internal modems support 230.4K bps. Recent Linux kernels support high speeds (over 115.2K) but you might have difficulty using it because of one or both of the following reasons:

1. The application program (or stty) will not accept the high speed.
2. Setserial has a default speed of 115,200 (but this default is easy to change)

TOC

How speed is set in hardware: the divisor and baud_base

Here's a list of commonly used divisors and their corresponding speeds (assuming a maximum speed of 115,200): 1 (115.2K), 2 (57.6K), 3 (38.4K), 6 (19.2K), 12 (9.6K), 24 (4.8K), 48 (2.4K), 96 (1.2K), etc. The serial driver sets the speed in the hardware by sending the hardware only a "divisor" (a positive integer). This "divisor" divides the maximum speed of the hardware resulting in a slower speed (except a divisor of 1 obviously tells the hardware to run at maximum speed).

Normally, if you specify a speed of 115.2K (in your communication program or by stty) then the serial driver sets the port hardware to divisor 1 which obviously sets the highest speed. If you happen to have hardware with a maximum speed of say 230.4K, then specifying 115.2K will result in divisor 1 and will actually give you 230.4K. This is double the speed that you set. In fact, for any speed you set, the actual speed will be double. If you had hardware that could run at 460.8K then the actual speed would be quadruple what you set.

Work-arounds for setting speed

To correct this accounting (but not always fix the problem) you may use "setserial" to change the baud_base to the actual maximal speed of your port such as 230.4K. Then if you set the speed (by your application or by stty) to 230.4K, a divisor of 1 will be used and you'll get the same speed as you set. PROBLEM: stty and many communication programs (as of mid 1999) still have 115.2K as their maximum speed setting and will not let you set 230.4K, etc. So in these cases one solution is not to change anything with `setserial` but mentally keep in mind that the actual speed is always double what you set.

There's another work-around which is not much better. To use it you set the baud_base (with setserial) to the maximal speed of your hardware. This corrects the accounting so that if you set say 115.2K you actually get 115.2K. Now you still have to figure out how to set the highest speed if your communication program (or the like) will not let you do it. Fortunately, setserial has a way to do this: use the "spd_cust" parameter with "divisor 1". Then when you set the speed to 38400 in a communication program, the divisor will be set to 1 in the port and it will operate at maximum speed. For example:
setserial /dev/ttyS2 spd_cust baud_base 230400 divisor 1
Don't try using "divisor" for any other purpose other than the special use illustrated above (with spd_cust).

If there are two or more high speeds that you want to use that your communication program can't set, then it's not quite as easy as above. But the same principles apply. You could just keep the default baud_base and understand that when you set a speed you are really only setting a divisor. So your actual speed will always be your maximum speed divided by whatever divisor is set by the serial driver. See How speed is set in hardware: the divisor and baud_base

Crystal frequency is not baud_base

Note that the baud_base setting is usually much lower than the frequency of the crystal oscillator in the hardware since the crystal frequency is often divided by 16 in the hardware to get the actual top speed. The reason the crystal frequency needs to be higher is so that this high crystal speed can be used to take a

number of samples of each bit to determine if it's a 1 or a 0.

12.4 Speed Table

It's best to have at least a 16650 UART for a 56K modem but few modems support it. Second best is a 16550 that has been tweaked to give 230,400 bps. Here are some suggested speeds to set your serial line if your modem speed is:

- 56K (V.90) use 115200 bps or 230400 bps (a few % faster ?)
- 28.8K (V.34), 33.6K (V.34) use 115200 bps
- 14400 bps (V.32bis), with V.42bis data compression, use 57600 bps
- 9600 bps (V.32), with V.42bis data compression, use 38400 bps
- slower than a 9600 bps (V.32) modem, set your speed to the highest speed your modem supports.

13. Communications Programs And Utilities

PPP is by far the most widely used. It's used for Internet access. For dialing out to public libraries, bulletin boards, etc. `minicom` is the most popular followed by `Seyon` (X-Windows only) and `Kermit`.

13.1 Minicom vs. Kermit

Minicom is only a communications program while Kermit is both a communications program and a file transfer protocol. But one may use the Kermit protocol from within Minicom (provided one has Kermit installed on one's PC) . Minicom is menu based while Kermit is command line based (interactive at the special Kermit prompt). While the Kermit program is free software, the documentation is not all free. There is no detailed manual supplied and it is suggested that you purchase a book as the manual. However Kermit has interactive online help which tells all but lacks tutorial explanations for the beginner. Commands may be put in a script file so you don't have to type them over again each time. Kermit (as a communications program) is more powerful than Minicom.

Although all Minicom documentation is free, it's not as extensive as Kermit's. Since permission is required to include Kermit in a commercial distribution, and since the documentation is not entirely free, some distributions don't include Kermit. In my opinion it's easier to set up Minicom and there is less to learn.

13.2 Lists of Programs

Here is a list of some communication software you can choose from, available via FTP, if they didn't come with your distribution. I would like comparative comments on the dialout programs. Are the least popular ones obsolete?

Least Popular Dialout

- `ecu` - a communications program
- `pcomm` - `procomm`-like communications program with zmodem
- `xc` - xcomm communication package

Most Popular Dialout

- `minicom` - `telix`-like communications program. Supports scripts, zmodem, kermit
- C-Kermit - portable, scriptable, serial and TCP/IP communications including file transfer, character-set translation, and zmodem support
- `seyon` - X based communication program

TOC

Fax

- `efax` a small fax program
- `hylafax` a large fax program based on the client-server model.
- `mgetty+fax` handles fax stuff and login for dial-ins

Voicemail Software

- mvm is a Minimal VoiceMail for Linux
- vgetty is an extension to mgetty that handles voicemail for some modems. It should come with recent releases of mgetty.

Dial-in (uses getty)

- `mgetty+fax` is for modems and is well documented (except for voicemail as of early 1999). It also handles fax stuff and provides an alternative to `uugetty`. It's incorporating voicemail (using vgetty) features. See About mgetty
- `uugetty` is also for modems. It comes as a part of the `ps_getty` package. See About getty_ps

Other

- `callback` is where you dial out to a remote modem and then that modem hangs up and calls you back (to save on phone bills).
- `SLiRP` and `term` provide a PPP-like service that you can run in user space on a remote computer with a shell account. See term and SLiRP for more details
- `ZyXEL` is a control program for ZyXEL U-1496 modems. It handles dialin, dialout, dial back security, FAXing, and voice mailbox functions.
- SLIP and PPP software can be found at `ftp://metalab.unc.edu/pub/Linux/system/network/serial`.
- Other things can be found on `ftp://metalab.unc.edu/pub/Linux/system/serial` and `ftp://metalab.unc.edu/pub/Linux/apps/serialcomm` or one of the many mirrors. These are the directories where serial programs are kept.

13.3 SLiRP and term

`SLiRP` and `term` are programs which are of use if you only have a dial-up shell account on a Unix-like machine and want to get the equivalent of a PPP account (or the like) without being authorized to have it (possibly because you don't want to pay extra for it, etc.). `SLiRP` is more popular than `term` which is almost obsolete.

To use `SLiRP` you install it in your shell account on the remote computer. Then you dial up the account and run SLiRP on the remote and PPP on your local PC. You now have a PPP connection over which

you may run a web browser on your local PC such as Netscape, etc. There may be some problems as SLiRP is not as good as a real PPP account. Some accounts may provide SLiRP since it saves on IP addresses (You have no IP address while using SLiRP).

`term` is something like SLiRP only you need to run `term` on both the local and remote computer. There is no PPP on the phone line since `term` uses its own protocol. To use `term` from your PC you need to use a term-aware version of ftp to do ftp, etc. Thus it's easier to use SLiRP since the ordinary version of ftp works fine with SLiRP. There is an unmaintained Term HOWTO.

---

14. What Are UARTs? How Do They Affect Performance?

14.1 Introduction to UARTS

(This section is also in the Serial-HOWTO)

UARTs (**U**niversal **A**synchronous **R**eceiver **T**ransmitter) are serial chips on your PC motherboard (or on an internal modem card). The UART function may also be done on a chip that does other things as well. On older computers like many 486's, the chips were on the disk IO controller card. Still older computer have dedicated serial boards.

The UART's purpose is to convert bytes from the PC's parallel bus to a serial bit-stream. The cable going out of the serial port is serial and has only one wire for each direction of flow. The serial port sends out a stream of bits, one bit at a time. Conversely, the bit stream that enters the serial port via the external cable is converted to parallel bytes that the computer can understand. UARTs deal with data in byte sized pieces, which is conveniently also the size of ASCII characters.

Say you have a terminal hooked up to your PC. When you type a character, the terminal gives that character to it's transmitter (also a UART). The transmitter sends that byte out onto the serial line, one bit at a time, at a specific rate. On the PC end, the receiving UART takes all the bits and rebuilds the (parallel) byte and puts it in a buffer.

Along with converting between serial and parallel, the UART does some other things as a byproduct (side effect) of it's primary task. The voltage used to represent bits is also converted (changed). Extra bits (called start and stop bits) are added to each byte before it is transmitted. See the Serial-HOWTO section, ``Voltage Waveshapes'' for details. Also, while the flow rate (in bytes/sec) on the parallel bus inside the computer is very high, the flow rate out the UART on the serial port side of it is much lower. The UART has a fixed set of rates (speeds) which it can use at its serial port interface.

14.2 Two Types of UARTs

There are two basic types of UARTs: dumb UARTS and FIFO UARTS. Dumb UARTs are the 8250, 16450, early 16550, and early 16650. They are obsolete but if you understand how they work it's easy to understand how the modern ones work with FIFO UARTS (late 16550, 16550A, 16c552, late 16650, 16750, and 16C950).

There is some confusion regarding 16550. Early models had a bug and worked properly only as 16450's (no FIFO). Later models with the bug fixed were named 16550A but many manufacturers did not accept the name change and continued calling it a 16550. Most all 16550's in use today are like 16550A's. Linux will report it as being a 16550A even though your hardware manual (or a label note) says it's a 16550. A similar situation exists for the 16650 (only it's worse since the manufacturer allegedly didn't admit

anything was wrong). Linux will report a late 16650 as being a 16650V2. If it reports it as 16650 it is news and only is used as if it had a one-byte buffer.

TOC

14.3 FIFOs

To understand the differences between dumb and FIFO (First In, First Out queue discipline) first let's examine what happens when a UART has sent or received a byte. The UART itself can't do anything with the data passing thru it, it just receives and sends it. For the original dumb UARTS, the CPU gets an interrupt from the serial device every time a byte has been sent or received. The CPU then moves the received byte out of the UART's buffer and into memory somewhere, or gives the UART another byte to send. The 8250 and 16450 UARTs only have a 1 byte buffer. That means, that every time 1 byte is sent or received, the CPU is interrupted. At low transfer rates, this is OK. But, at high transfer rates, the CPU gets so busy dealing with the UART, that is doesn't have time to adequately tend to other tasks. In some cases, the CPU does not get around to servicing the interrupt in time, and the byte is overwritten, because they are coming in so fast. This is called an "overrun" or "overflow".

That's where the FIFO UARTs are useful. The 16550A (or 16550) FIFO chip comes with 16 byte FIFO buffers. This means that it can receive up to 14 bytes (or send 16 bytes) before it has to interrupt the CPU. Not only can it wait for more bytes, but the CPU then can transfer all 14 (or more) bytes at a time. Although the interrupt threshold (trigger level) may be set at 8 instead of 14, this is still a significant advantage over the other UARTs, which only have 1 byte buffers. The CPU receives less interrupts, and is free to do other things. Data is not lost, and everyone is happy.

While most PC's only have a 16550 with 16-byte buffers, better UARTS have even larger buffers. Note that the interrupt is issued slightly before the buffer get full (at say a "trigger level" of 14 bytes for a 16-byte buffer). This allows room for a few more bytes to be received during the time that the interrupt is being serviced. The trigger level may be set to various permitted values by kernel software. A trigger level of 1 will be almost like a dumb UART (except that it still has room for 15 more bytes after it issues the interrupt).

If you type something while visiting a BBS, the characters you type go out thru the serial port. Your typed characters that you see on the screen are what was echoed back thru the telephone line thru your modem and then thru your serial port to the screen. If you had a 16-byte buffer on the serial port which held back characters until it had 14 of them, you would need to type many characters before you could see what you typed (before they appeared on the screen). This would be very confusing but there is a "timeout" to prevent this. Thus you normally see a character on the screen just as soon as you type it.

The "timeout" works like this for the receive UART buffer: If characters arrive one after another, then an interrupt is issued only when say the 14th character reaches the buffer. But if a character arrives and the next character doesn't arrive soon thereafter, then an interrupt is issued. This happens even though there are not 14 characters in the buffer (there may only be one character in it). Thus when what you type goes thru this buffer, it acts almost like a 1-byte buffer even though it is actually a 16-byte buffer (unless your typing speed is a hundred times faster than normal). There is also "timeout" for the transmit buffer as well.

14.4 UART Model Numbers

TOC

Here's a list of UARTs. *TL* is *T*rigger *L*evel

- 8250, 16450, early 16550: Obsolete with 1-byte buffers
- 16550, 16550A, 16c552: 16-byte buffers, TL=1,4,8,14
- 16650: 32-byte buffers. Speed up to 460.8 Kbps
- 16750: 64-byte buffer for send, 56-byte for receive. Speed up to 921.6 Kbps
- Hayes ESP: 1K-byte buffers.

The obsolete ones are only good for modems no higher than 14.4k (DTE speeds up to 38400 bps). For modern modems you need at least a 16550 (and not an early 16550). For V.90 56k modems, it may be a several percent faster with a 16650 (especially if you are downloading uncompressed files). The main advantage of the 16650 is its larger buffer size as the extra speed isn't needed unless the modem compression ratio is high. Some 56k internal modems may come with a 16650 ??

Non-UART, and intelligent multiport boards use DSP chips to do additional buffering and control, thus relieving the CPU even more. For example, the Cyclades Cyclom, and Stallion EasyIO boards use a Cirrus Logic CD1400 RISC UART, and many boards use 80186 CPUs or even special RISC CPUs, to handle the serial IO.

Most newer PC's (486's, Pentiums, or better) come with 16550A's (usually called just 16550's). If you have something really old the chip may unplug so that you may be able to upgrade by buying a 16550A chip and replacing your existing 16450 UART. If the functionality has been put on another type of chip, you are out of luck. If the UART is socketed, then upgrading is easy (if you can find a replacement). The new and old are pin-to-pin compatible. It may be more feasible to just buy a new serial board on the Internet (few retail stores stock them today).

---

15. Troubleshooting

15.1 I can't get 56K on my 56K modem

There must be very low noise on the line for it to work at even close to 56K. Some phone lines are so bad that the speeds obtainable are much slower than 56K (like 28.8K or even slower). Sometimes extension phones connected to the same line can cause problems. To check this you might try connecting your modem directly to the point where the telephone line enters the building with the feed for everything else on that line disconnected (if others can tolerate that).

15.2 Uploading (downloading) files is broken/slow

Flow control (both at your PC and/or modem) may not be enabled. If you have set a high DTE speed (like 115.2K) then flow from your modem to your PC may work OK but a lot of flow in the other direction will not all get thru due to the telephone line bottleneck. This will result in many errors and the resending of packets. It may thus take far too long to send a file. In some cases, files don't make it thru at all. If you're downloading long uncompressed files or web pages (and your modem uses data compression) or you've set a low DTE speed, then downloading may also be broken due to no flow control.

15.3 For Dial-in I Keep Getting "line NNN of inittab invalid"

Make sure you are using the correct syntax for your version of `init`. The different `init`'s that are out there use different syntax in the `/etc/inittab` file. Make sure you are using the correct syntax for your version of `getty`.

15.4 When I Try To Dial Out, It Says "/dev/ttySN: Device or resource busy"

This problem can arise when DCD or DTR are not implemented correctly. DCD should only be on (asserted) when there is an actual connection (ie someone has dialed in), not when `getty` is watching the port. Check to make sure that your modem is configured to only assert DCD when there is a connection. DTR should be on (asserted) whenever something is using, or watching the line, like `getty`, `kermit`, or some other comm program.

Another common cause of ``device busy'' errors, is that you set up your serial port with an interrupt already taken by something else. As each device initializes, it asks Linux for permission to use its

hardware interrupt. Linux keeps track of which interrupt is assigned to whom, and if your interrupt is already taken, your device won't be able to initialize properly. The device really doesn't have much of any way to tell you that this happened, except that when you try to use it, it will return a ``device-busy'' error. Check the interrupts on all of your boards (serial, ethernet, SCSI, etc.). Look for IRQ conflicts.

15.5 I Keep Getting: ``Id "S3" respawning too fast: disabled for 5 minutes''

Id "S3" is just an example. In this case look on the line which starts with "S3" in `/etc/inittab` . This is causing the problem. Make sure the syntax for this line is correct and that the device (ttyS3) exists and can be found.

Make sure your modem is configured correctly. Look at registers `E` and `Q`. This can occur when your modem is chatting with `getty`.

If you use uugetty, verify that your `/etc/gettydefs` syntax is correct by doing the following:

```
linux# getty -c /etc/gettydefs
```

This can also happen when the `uugetty` initialization is failing. See section uugetty Still Doesn't Work.

15.6 My Modem is Hosed after Someone Hangs Up, or uugetty doesn't respawn

This can happen when your modem doesn't reset when DTR is dropped. Greg Hankins saw his RD and SD LEDs go crazy when this happened. You need to have your modem reset. Most Hayes compatible modems do this with `&D3`, but on his USR Courier, he had to set `&D2` and `S13=1`. Check your modem manual (if you have one).

15.7 uugetty Still Doesn't Work

There is a `DEBUG` option that comes with `getty_ps`. Edit your config file `/etc/conf.{uu}getty.ttyS`*N* and add `DEBUG=`*NNN*. Where *NNN* is one of the following combination of numbers according to what you are trying to debug:

```
D_OPT   001          option settings
D_DEF   002          defaults file processing
D_UTMP  004          utmp/wtmp processing
D_INIT  010          line initialization (INIT)
D_GTAB  020          gettytab file processing
D_RUN   040          other runtime diagnostics
D_RB    100          ringback debugging
D_LOCK  200          uugetty lockfile processing
```

```
    D_SCH   400             schedule processing
    D_ALL   777             everything
```

Setting `DEBUG=010` is a good place to start.

If you are running `syslogd`, debugging info will appear in your log files. If you aren't running `syslogd` info will appear in `/tmp/getty:ttyS`*N* for debugging `getty` and `/tmp/uugetty:ttyS`*N* for `uugetty`, and in `/var/adm/getty.log`. Look at the debugging info and see what is going on. Most likely, you will need to tune some of the parameters in your config file, and reconfigure your modem.

You could also try `mgetty`. Some people have better luck with it.

15.8 My Modem is Physically There but Can't be Found

If you know about the existing serial ports before you installed an internal modem, then the problem is to find the new serial port. This is covered in the next section. This section is about finding out which serial port has the modem on it.

There's a program that looks for modems on commonly used serial ports called "wvdialconf". Just type "wvdialconf <a-new-file-name>". It will create the new file as a configuration file but you don't need this file unless you are going to use "wvdial" for dialing. See What is wvdialconf ?

Your problem could be due to a winmodem (or the like) which can't be used with Linux. See Avoid: winmodems. The "setserial program may be used to detect serial ports but will not detect modems on them. Thus "wvdialconf" is best to try first.

Another way to see if there's a modem on a port is to start "minicom" on the port (go to the setup menus with ^AO). Then type "AT" and you should see OK (or 0 if it's set for "digit result codes"). If it takes many seconds to get a response (including only the cursor moving down one line) then see Slow: Text appears on the screen slowly after long delays

15.9 The following subsections are in both the Serial and Modem HOWTOs:

15.10 My Serial Port is Physically There but Can't be Found

Check the BIOS menus and BIOS messages. If it's an ISA bus PnP serial port, try "pnpdump --dumpregs" and/or see Plug-and-Play-HOWTO. For the PCI bus look at /proc/pci. You may try probing with setserial. See Probing If nothing seems to get thru the port it may be there but have a bad interrupt. See Slow: Text appears on the screen slowly after long delays

TOC

15.11 Slow: Text appears on the screen slowly aft long delays

It's likely mis-set/conflicting interrupts. Here are some of the symptoms which will happen the first time you try to use a modem, terminal, or printer. In some cases you type something but nothing appears on the screen until many seconds later. Only the last character typed may show up (and it may be just an invisible <return> character but you notice the cursor jumps down one line). In other cases where a lot of data should appear on the screen, only a batch of about 16 characters appear. Then there is a long wait of many seconds for the next batch of characters. You might also get "input overrun" error messages (or find them in logs).

For more details on the symptoms and why this happens see the Serial-HOWTO section: "Interrupt Problem Details".

As a final check to see if it really is an interrupt problem, set the IRQ to zero with "setserial". This will tell the driver not to use interrupts but to use inefficient (but faster) polling methods. If this seemingly fixes the "slow" problem then you had an interrupt problem but should still try to fix it since polling uses excessive computer resources.

Checking to find the interrupt conflict may not be easy since looking in the /proc directory may be misleading. Make sure there are no IRQs being illegally shared. Check all your boards (serial, ethernet, SCSI, etc...). Make sure the jumper (or PnP) settings, and the `setserial` parameters are correct for all your serial devices. Also check `/proc/ioports` and `/proc/interrupts` and `/proc/pci` for conflicts. For more details see Serial-HOWTO: Interrupt Problem Details. If it involves Plug-and-Play devices, see also Plug-and-Play-HOWTO.

15.12 The Startup Screen Show Wrong IRQs for the Serial Ports.

Linux does not do any IRQ detection on startup. When the serial module loads it only does serial device detection. Thus, disregard what it says about the IRQ, because it's just assuming the standard IRQs. This is done, because IRQ detection is unreliable, and can be fooled. But if and when setserial runs from a start-up script, it changes the IRQ's and displays the new (and hopefully correct) state on on the startup screen. If the wrong IRQ is not corrected by a later display on the screen, then you've got a problem.

So, even though I have my `ttyS2` set at IRQ 5, I still see

```
ttyS02 at 0x03e8 (irq = 4) is a 16550A
```

at first when Linux boots. (Older kernels may show "ttyS02" as "tty02") You have to use `setserial` to tell Linux the IRQ you are using.

15.13 "Cannot open /dev/ttyS?: Permission denied"

TOC

Check the file permissions on this port with "ls -l /dev/ttyS?"_ If you own the ttyS? then you need read and write permissions: crw with the c (Character device) in col. 1. It you don't own it then it should show rw- in cols. 8 & 9 which means that everyone has read and write permission on it. Use "chmod" to change permissions. There are more complicated ways to get access like belonging to a "group" that has group permission.

15.14 "Operation not supported by device" (error message) for ttySx

This means that an operation requested by setserial, stty, etc. couldn't be done because the kernel doesn't support doing it. Formerly this was often due to the "serial" module not being loaded. But with the advent of PnP, it may likely mean that there is no modem (or other serial device) at the address where the driver (and setserial) thinks it is. If there is no modem there, commands (for operations) sent to that address obviously don't get done. See What is set in my serial port hardware?

If the "serial" module wasn't loaded but "lsmod" shows you it's now loaded it might be the case that it's loaded now but wasn't loaded when you got the error message. In many cases the module will automatically loaded when needed (if it can be found). To force loading of the "serial" module it may be listed in the file: /etc/modules.conf or /etc/modules. The actual module should reside in: /lib/modules/.../misc/serial.o.

15.15 "Cannot create lockfile. Sorry" (error message)

When a port is "opened" by a program a lockfile is created in /var/lock/. Wrong permissions for the lock directory will not allow a lockfile to be created there. Use "ls -ld /var/lock" to see if the permissions are OK: usually rwx for everyone (repeated 3 times). If it's wrong, use "chmod" to fix it. Of course, if there is no "lock" directory no lockfile can be created there. For more info on lockfiles see the Serial-HOWTO subsection: "What Are Lock Files".

15.16 Software which may help

- `modemstat` and `statserial` show the current state of various modem signal lines (such as DTR, CTS, etc.)
- `irqtune` will give serial port interrupts higher priority to improve performance.
- `hdparm` for hard-disk tuning may help some more.

---

16. Flash Upgrades

Many modems can be upgraded by reprogramming their flash memories with an upgrade program which you get from the Internet. By sending this "program" from the PC via the serial port to the modem, the modem will store this program in its non-volatile memory (it's still there when the power is turned off). The instructions on installing it are usually on how to do in under Windows so you'll need to figure out how to do the equivalent under Linux (unless you want to install the upgrade under Windows). Sending the program to the modem is often called a download.

If the latest version of this HOWTO still contains this request (see New Versions of this HOWTO) please send me your experiences with installing such upgrades that will be helpful to others.

Here's the general idea of doing an upgrade. First, there may be a command that you need to send your modem to tell it that what follows is a flash ROM upgrade. In one case this was AT** You can do this by starting a communications program (such as minicom) and type. First type AT <enter> to see if your modem is there and answers "OK".

Next, you need to send an file (sometimes two files) directly to the modem. Communication programs (such as minicom) often use zmodem or kermit to send files to the modem (and beyond) but these put the file into packets which append headers and you want the exact file sent to the modem, not a modified one. But the kermit communications program has a "transmit" command that will send the file directly (without using the kermit packets) so this is one way to send a file directly. Minicom didn't have this feature in 1998.

Another way to send the file(s) would be to escape from the communications program to the shell (in minicom this is ^AJ) and then: `cat upgrade_file_name > /dev/ttyS2` (if your serial port is ttyS2). Then go back to the communication program (type fg at the command line prompt in minicom) to see what happened.

Here's an example session for a certain Rockwell modem (C-a is ^A):

```
- Run minicom
- Type AT** : see "Download initiated .."
- C-a J
- cat FLASH.S37 > /dev/modem
- fg : see "Download flash code .."
- C-a J
- cat 283P1722.S37 > /dev/modem
- fg : see "Device successfully programmed"
```

17. Other Sources of Information

17.1 Misc

- man pages for: `agetty(8)`, `getty(1m)`, `gettydefs(5)`, `init(1)`, `isapnp(8)`, `login(1)`, `mgetty(8)`, `setserial(8)`
- Your modem manual (if it exists). Some modems come without manuals.
- Serial Suite by Vern Hoxie is a collection of blurbs about the care and feeding of the Linux serial port plus some simple programs.
- The Linux serial mailing list. To join, send email to `majordomo@vger.rutgers.edu`, with ``subscribe linux-serial'' in the message body. If you send ``help'' in the message body, you get a help message. The server also serves many other Linux lists. Send the ``lists'' command for a list of mailing lists.

17.2 Books

I've been unable to find a good up-to-date book on modems.

- The Complete Modem Reference by Gilbert Held, 1997. Contains too much info about obsolete topics. More up-to-date info may be found on the Internet.
- Modems For Dummies by Tina Rathbone, 1996. (Have never seen it.)

17.3 HOWTOs

- Cable-Modem mini-howto
- ISDN Howto (not a LDP Howto) http://www.suse.de/Support/sdb_e/isdn.html: drivers for ISDN "Modems". Much related info on this is in German.
- Modems-HOWTO: In French (Not used in creating this Modem-HOWTO)
- NET-3-HOWTO: all about networking, including SLIP, CSLIP, and PPP
- PPP-HOWTO: help with PPP including modem set-up
- Serial-HOWTO has info on Multiport Serial Cards used for both terminals and banks of modems. Covers the serial port in more detail than in the HOWTO.
- Serial-Programming-HOWTO: for some aspects of serial-port programming
- Text-Terminal-HOWTO: (including connecting up with modems)
- UUCP-HOWTO: for information on setting up UUCP

17.4 Usenet newsgroups

- comp.os.linux.answers FAQs, How-To's, READMEs, etc. about Linux.
- comp.os.linux.hardware Hardware compatibility with the Linux operating system.
- comp.os.linux.setup Linux installation and system administration.
- comp.dcom.modems Modems for all OS's

17.5 Web Sites

- Hayes AT modem commands Technical Reference for Hayes (tm) Modem Users
- Rockwell-based modem commands
- A white paper discussing serial communications and multiport serial boards is available from Cyclades at `http://www.cyclades.com`.
- Modem FAQs:
  Navas 28800 Modem FAQ
- Curt's High Speed Modem Page
- Much info on 56k modems 56k Modem = v.Unreliable
- Links to modem manufacturers
- Identifying modems by FCC ID
- Partial list of modems which work/don't_work under Linux modem list

---

18. Appendix A: How Modems Work (technical) (unfinished)

18.1 Modulation Details

Intro to Modulation

Modulation is the conversion of a digital signal represented by binary bits (0 or 1) into an analog signal something like a sine wave. The modulated signal consists pure sine wave "carrier" signal which is modified to convey information. A pure carrier sine wave, unchanging in frequency and voltage, provides no flow of information at all (except that a carrier is present). To make it convey information we modify (or modulate) this carrier. There are 3 basic types of modulation: frequency, amplitude, and phase. They will be explained next.

Frequency Modulation

The simplest modulation method is frequency modulation. Frequency is measured in cycles per second (of a sine wave). It's the count of the number of times the sine wave shape repeats itself in a second. This is the same as the number of times it reaches it peak value in a second. The word "Hertz" (abbreviated Hz) is used to mean "cycles per second".

A simple example of frequency modulation is where one frequency means a 0 and another means a 1. For example, for some obsolete 300 baud modems 1070 Hz meant a binary 0 while 1270 Hz meant a binary 1. This was called "frequency shift keying". Instead of just two possible frequencies, more could be used to allow more information to be transmitted. If we had 4 different frequencies (call them A, B, C, and D) then each frequency could stand for a pair of bits. For example, to send 00 one would use frequency A. To send 01, use frequency B; for 10 use C; for 11 use D. In like manner, by using 8 different frequencies we could send 3 bits with each shift in frequency. Each time we double the number of possible frequencies we increase the number of bits it can represent by 1.

Amplitude Modulation

Once one understands frequency modulation example above including the possibilities of representing a few bits by a single shift in frequency, it's easier to understand both amplitude modulation and phase modulation. For amplitude modulation, one just changes the height (voltage) of the sine wave analogous to changing the frequency of the sine wave. For a simple case there could only be 2 allowed amplitude levels, one representing a 0-bit and another representing a 1-bit. As explained for the case of frequency modulation, having more possible amplitudes will result in more information being transmitted.

TOC

Phase Modulation

To change the phase of a sine wave at a certain instant of time, we stop sending this old sine wave and immediately begin sending a new sine wave of the same frequency and amplitude. If we started sending the new sine wave at the same voltage level (and slope) as existed when we stopped sending the old sine wave, there would be no change in phase (and no detectable change at all). But suppose that we started up the new sine wave at a different point on the sine wave curve. Then there would likely be a sudden voltage jump at the point in time where the old sine wave stopped and the new sine wave began. This is a phase shift and it's measured in degrees (deg.) A 0 deg. (or a 360 deg.) phase shift means no change at all while a 180 deg. phase shift just reverses the voltage (and slope) of the sine wave. Put another way, a 180 deg. phase shift just skips over a half-period (180 deg.) at the point of transition. Of course we could just skip over say 90 deg. or 135 deg. etc. As in the example for frequency modulation, the more possible phase shifts, the more bits a single shift in phase can represent.

Combination Modulation

Instead of just selecting either frequency, amplitude, or phase modulation, we may chose to combine modulation methods. Suppose that we have 256 possible frequencies and thus can send a byte (8 bits) for each shift in frequency (since 2 to the 8 power is 256). Suppose also that we have another 256 different amplitudes so that each shift in amplitude represents a byte. Also suppose there are 256 possible phase shifts. Then a certain points in time we may make a shift in all 3 things: frequency, amplitude and phase. This would send out 3 bytes for each such transition.

No modulation method in use today actually does this. It's not practical due to the relatively long time it would take to detect all 3 types of changes. But what is quite common is the simultaneous change in both phase and amplitude. This is called phase-amplitude modulation (sometimes also called quadrature amplitude modulation = QAM). This method is used for the common modem speeds of 14.4k, 28.8k, and 33.6k. The only significant case where this modulation method is not used today is for 56k modems. But even 56k modems exclusively use QAM (phase-amplitude modulation) in the direction from your PC out the telephone line. Sometimes even the other direction will also fall back to QAM when line conditions are not good enough. Thus QAM (phase-amplitude modulation) still remains the most widely used method on ordinary telephone lines.

56k Modems (v.90)

The modulation method used above 33.6k is entirely different than the common phase-amplitude modulation. Since ordinary telephone calls are converted to digital signals at the local offices of the telephone company, the fastest speed that you can send digital data by an ordinary telephone call is the same speed that the telephone company uses over its digital portion of the phone call transmission. What is this speed? Well, it's close to 64Kbps. It would be 64k but sometimes bits are "stolen" for the purposes of signalling. But if the phone Co. knows that the link is digital, bits may not get stolen. The case of 64k will be presented and then it will be explained why the actual speed is lower (56k or less --usually significantly less).

Thus 64k is the absolute top speed possible for an ordinary telephone call using the digital portion of the circuit that was designed to send digital encodings of the human voice. In order to use 64k, the modem

must know exactly how the telephone company is doing its digital encoding of the analog signals. Th task is far too complicated if both sides of a telephone call have only an analog interface to the telep company. But if one side has a digital interface, then it's possible (at least in one direction). Thus if your ISP has a digital interface to the phone company, the ISP may send out a certain digital signal over the phone lines toward your PC. The digital signal from the ISP gets converted to analog at the local telephone office near your PC's location (perhaps near your home). Then it's your modem's task to try to figure out exactly what that digital signal was. If it can do this then transmission at 64k (the speed of the telephone company's digital signal) is possible in this direction.

What method does the telephone company use to digitally encode analog signals? It uses a method of sampling the amplitude of the analog signal at a rate of 8000 samples per second. Each sample amplitude is encoded as a 8-bit (ASCII-like) byte. (Note: 8 x 8000 = 64k) This is called "Pulse Code Modulation" = PCM. These bytes are then sent digitally on the telephone company's digital circuits where many calls share a single circuit using a time-sharing scheme known as "time division multiplexing". Then finally at the local telephone office near your home, the digital signal is de-multiplexed resulting in the same digital signal as was originally created by PCM. This signal is then converted back to analog and sent to your home. Each 8-bit byte creates a certain amplitude of the analog signal. Your modem's task is to determine just what that PCM 8-bit byte was based on the amplitude it detects.

This is (sort of) "amplitude demodulation" but not really. It's not amplitude demodulation because there is no carrier. Actually, it's called "modulus conversion" which is the inverse of PCM. In order to determine the digital codes the telephone Co. used to create the analog signal, the modem must sample this analog signal amplitude at exactly the same points in time the phone Co. used when it created the analog signal. To do this a timing signal is generated from a residual 4k Hz signal on the phone line. The setting of amplitudes to go out to your home/office at 8k settings/sec sort of creates a 4k signal. Suppose every other amplitude was of opposite polarity. Then there would be a 4k Hz sine-like wave created. Each amplitude is in a sense a 8-bit symbol and when to sample amplitudes is known as "symbol timing".

Now the encoding of amplitudes in PCM is not linear and at low amplitudes the differences between possible amplitude is quite small. So to make the distinction of amplitudes that are close to each other easier, certain amplitudes are not used. This give a larger delta between possible amplitudes and makes correct detection of them by your modem easier. If half the amplitude levels were not used by v.90 and it would be tantamount to each symbol (allowed amplitude level) representing 7 bits instead of 8. This is where 56k comes from: 7 bits/symbol x 8k symbols/sec = 56k bps. Of course each symbol is actually generated by 8-bits but only 128 bytes of the possible 256 bytes are actually used. There is a code table mapping these 128 8-bit bytes to 128 7-bit bytes.

But it's a little more complicated that this. If the line conditions are not nearly perfect, then even fewer possible levels (symbols) are used resulting in speeds under 56k. Also due to government rules prohibiting high power levels on phone lines, certain high amplitudes levels can't be used resulting in only about 53.3k at best for "56k" modems.

Note that the digital part of the telephone network is bi-directional. Two such circuits are used for a phone call, one in each direction. The direction from your home/office to the ISP uses the conventional phase-amplitude modulation scheme with a maximum of 36.6k (and not 53.3k).

---

19. Appendix B: "baud" vs. "bps"

19.1 A simple example

``baud'' and ``bps'' are perhaps one of the most misused terms in the computing and telecommunications field. Many people use these terms interchangeably, when in fact they are not! bps is simply the number of bits transmitted per second. The baud rate is a measure of how many times per second a signal changes (or could change). For a typical serial port a 1-bit is -12 volts and a 0-bit is +12 v (volts). If the bps is 38,400 a sequence of 010101... would also be 38,400 baud since the voltage shifts back and forth from positive to negative to positive ... and there are 38,400 shifts per second. For another sequence say 111000111... there will be fewer shifts of voltage since for three 1's in sequence the voltage just stays at -12 volts yet we say that its still 38,400 baud since there is a possibility that the number of changes per second will be that high.

Looked at another way, put an imaginary tic mark separating each bit (even thought the voltage may not change). 38,400 baud then means 38,400 tic marks per second. The tic marks at at the instants of permitted change and are actually marked by a synchronized clock signal generated in the hardware but not sent over the external cable.

Suppose that a "change" may have more than the two possible outcomes of the previous example (of +-12 v). Suppose it has 4 possible outcomes, each represented by a unique voltage level. Each level may represent a pair of bits (such as 01). For example, -12v could be 00, -6v 01, +6v 10 and +12v 11. Here the bit rate is double the baud rate. For example, 3000 changes per second will generate 2 bits for each change resulting in 6000 bits per second (bps). In other words 3000 baud results in 6000 bps.

19.2 Real examples

The above example is overly simple. Real examples are more complicated but based on the same idea. This explains how a modem running at 2400 baud, can send 14400 bps (or higher). The modem achieves a bps rate greater than baud rate by encoding many bits in each signal change (or transition). Thus, when 2 or more bits are encoded per baud, the bps rate exceeds the baud rate. If your modem-to-modem connection is at 14400 bps, it's going to be sending 6 bits per signal transition (or symbol) at 2400 baud. A speed of 28800 bps is obtained by 3200 baud at 9 bits/baud. When people misuse the word baud, they may mean the modem speed (such as 33.6K).

Common modem bps rates were formerly 50, 75, 110, 300, 1200, 2400, 9600. These were also the bps rates over the serial_port-to-modem cables. Today the bps modem-to-modem rates are 14.4K, 28.8K, 33.6K, and 56K, but the common rates over the serialPort-to-modem cables are not the same but are: 19.2K, 38.4K, 57.6K, 115.2K. Using modems with V.42bis compression (max 4:1 compression), rates up to 115.2K bps are possible for 33.6K modems (230.4K is possible for 56K modems).

Except for the 56k modems, most modems run at 2400, 3000, or 3200 baud. Because of the bandwidth

limitations on voice-grade phone lines, baud rates greater than 2400 are harder to achieve, and only v under conditions of pristine phone line quality.

How did this confusion between bps and baud start? Well, back when antique low speed modems were high speed modems, the bps rate actually did equal the baud rate. One bit would be encoded per phase change. People would use bps and baud interchangeably, because they were the same number. For example, a 300 bps modem also had a baud rate of 300. This all changed when faster modems came around, and the bit rate exceeded the baud rate. ``baud'' is named after Emile Baudot, the inventor of the asynchronous telegraph printer. One way this problem gets resolved is to use the term "symbol rate" instead of "baud" and thus avoid using the term "baud".

---

20. Appendix C: Terminal Server Connection

This section was adapted from Text-Terminal-HOWTO.

A terminal server is something like an intelligent switch that can connect many modems (or terminals) to one or more computers. It's not a mechanical switch so it may change the speeds and protocols of the streams of data that go thru it. A number of companies make terminal servers: Xyplex, Cisco, 3Com, Computone, Livingston, etc. There are many different types and capabilities. Another HOWTO is needed to compare and describe them (including the possibility of creating your own terminal server with a Linux PC). Most are used for modem connections rather than directly connected terminals.

One use for them is to connect many modems (or terminals) to a high speed network which connects to host computers. Of course the terminal server must have the computing power and software to run network protocols so it is in some ways like a computer. The terminal server may interact with the user and ask what computer to connect to, etc. or it may connect without asking. One may sometimes send jobs to a printer thru a terminal server.

A PC today has enough computing power to act like a terminal server except that each serial port should have its own hardware interrupt. PC's only have a few spare interrupts for this purpose and since they are hard-wired you can't create more by software. A solution is to use an advanced multiport serial card which has its own system of interrupts (or on lower cost models, shares one of the PC's interrupts between a number of ports). See Serial-HOWTO for more info. If such a PC runs Linux with getty running on many serial ports it might be thought of as a terminal server. It is in effect a terminal server if it's linked to other PC's over a network and if its job is mainly to pass thru data and handle the serial port interrupts every 14 (or so) bytes. Software called "radius" is sometimes used.

Today real terminal servers serve more than just terminals. They also serve PC's which emulate terminals, and are sometimes connected to a bank of modems connected to phone lines. Some even include built-in modems. If a terminal (or PC emulating one) is connected directly to a modem, the modem at the other end of the line could be connected to a terminal server. In some cases the terminal server by default expects the callers to use PPP packets, something that real text terminals don't generate.

21. Appendix D: Other Types of Modems

This HOWTO currently only deals with the common type of modem used to connect PC's to ordinary analog telephone lines. There are various other types of modems, including devices called modems that are not really modems.

21.1 Digital-to-Digital "Modems"

The standard definition of a modem is sometimes broadened to include "digital" modems. Today direct digital service is now being provided to many homes and offices so a computer there sends out digital signals directly (well almost) into the telephone lines. But a device is still needed to convert the computer digital signal into type allowed on telephone circuits and this device is sometimes called a modem. The next 2 sections: ISDN and DSL concern digital-to-digital "modems".

21.2 ISDN "Modems"

The "modem" is really a Terminal Adapter (TA). A Debian package "isdnutils" is available. There is a ISDN Howto in German with an English translation: http://www.suse.de/Support/sdb_e/isdn.html. It's put out by the SuSE distribution of Linux and likely is about drivers available in that distribution. There is an isdn4linux package and a newsgroup: de.alt.comm.isdn4linux. Many of the postings are in German. You might try using a search engine (such as DejaNews) to find "isdn4linux".

21.3 Digital Subscriber Line (DSL)

DSL uses the existing twisted pair line from your home (etc.) to the local telephone office. This can be used if your telephone line can accept higher speeds than an ordinary modem (say 56k) sends over it. It replaces the analog-to-digital converter at the local telephone office with a converter which can accept a much faster flow of data (in a different format of course). The device which converts the digital signals from your computer to the signal used to represent digital data on the local telephone line is also called a modem. This document presently does not cover the special aspects of these modems.

END OF Modem-HOWTO

HOWTO: Multi Disk System Tuning

Stein Gjoen, sgjoen@nyx.net

v0.22i, 27 May 1999

This document describes how best to use multiple disks and partitions for a Linux system. Although some of this text is Linux specific the general approach outlined here can be applied to many other multi tasking operating systems.

1. Introduction

- 1.1 Copyright
- 1.2 Disclaimer
- 1.3 News
- 1.4 Credits

2. Structure

- 2.1 Logical structure
- 2.2 Document structure
- 2.3 Reading plan

3. Drive technologies

- 3.1 Drives
- 3.2 Geometry
- 3.3 Media
- 3.4 Interfaces
- 3.5 Cabling
- 3.6 Host Adapters
- 3.7 Multi Channel Systems
- 3.8 Multi Board Systems

- 3.9 Speed Comparison
- 3.10 Benchmarking
- 3.11 Comparisons
- 3.12 Future Development
- 3.13 Recommendations

4. File System Structure

- 4.1 File System Features
- 4.2 Explanation of Terms

5. File Systems

- 5.1 General Purpose File Systems
- 5.2 Microsoft File Systems
- 5.3 Logging and Journaling File Systems
- 5.4 Read-only File Systems
- 5.5 Networking File Systems
- 5.6 Special File Systems
- 5.7 File System Recommendations

6. Technologies

- 6.1 RAID
- 6.2 Volume Management
- 6.3 Linux `md` Kernel Patch
- 6.4 Compression
- 6.5 ACL
- 6.6 `cachefs`
- 6.7 Translucent or Inheriting File Systems
- 6.8 Physical Track Positioning
- 6.9 Stacking
- 6.10 Recommendations

TOC

7. Other Operating Systems

- 7.1 DOS
- 7.2 Windows
- 7.3 OS/2
- 7.4 NT
- 7.5 Sun OS

8. Clusters

9. Mount Points

10. Considerations and Dimensioning

- 10.1 Home Systems
- 10.2 Servers
- 10.3 Pitfalls

11. Disk Layout

- 11.1 Selection for Partitioning
- 11.2 Mapping Partitions to Drives
- 11.3 Sorting Partitions on Drives
- 11.4 Optimizing
- 11.5 Compromises

12. Implementation

- 12.1 Drives and Partitions
- 12.2 Partitioning
- 12.3 Repartitioning
- 12.4 Microsoft Partition Bug
- 12.5 Multiple Devices (`md`)
- 12.6 Formatting

- 12.7 Mounting
- 12.8 `fstab`
- 12.9 Recommendations

13. Maintenance

- 13.1 Backup
- 13.2 Defragmentation
- 13.3 Deletions
- 13.4 Upgrades
- 13.5 Recovery

14. Advanced Issues

- 14.1 Hard Disk Tuning
- 14.2 File System Tuning
- 14.3 Spindle Synchronizing

15. Further Information

- 15.1 News groups
- 15.2 Mailing Lists
- 15.3 HOWTO
- 15.4 Mini-HOWTO
- 15.5 Local Resources
- 15.6 Web Pages
- 15.7 Search Engines

16. Getting Help

17. Concluding Remarks

- 17.1 Coming Soon
- 17.2 Request for Information
- 17.3 Suggested Project Work

TOC

18. Questions and Answers

19. Bits and Pieces

- 19.1 Swap Partition: to Use or Not to Use
- 19.2 Mount Point and `/mnt`
- 19.3 Power and Heating
- 19.4 Dejanews

20. Appendix A: Partitioning Layout Table: Mounting and Linking

21. Appendix B: Partitioning Layout Table: Numbering and Sizing

22. Appendix C: Partitioning Layout Table: Partition Placement

23. Appendix D: Example: Multipurpose Server

24. Appendix E: Example: Mounting and Linking

25. Appendix F: Example: Numbering and Sizing

26. Appendix G: Example: Partition Placement

27. Appendix H: Example II

28. Appendix I: Example III: SPARC Solaris

29. Appendix J: Example IV: Server with 4 Drives

30. Appendix K: Example V: Dual Drive System

31. Appendix L: Example VI: Single Drive System

1. Introduction

In commemoration of Linux kernel 2.2 release this brand new release is codenamed the **Daniella** release.

New code names will appear as per industry standard guidelines to emphasize the state-of-the-art-ness of this document.

This document was written for two reasons, mainly because I got hold of 3 old SCSI disks to set up my Linux system on and I was pondering how best to utilise the inherent possibilities of parallelizing in a SCSI system. Secondly I hear there is a prize for people who write documents...

This is intended to be read in conjunction with the Linux Filesystem Structure Standard (FSSTND). It does not in any way replace it but tries to suggest where physically to place directories detailed in the FSSTND, in terms of drives, partitions, types, RAID, file system (fs), physical sizes and other parameters that should be considered and tuned in a Linux system, ranging from single home systems to large servers on the Internet.

The followup to FSSTND is called the Filesystem Hierarchy Standard (FHS) and covers more than Linux alone. FHS version 2.0 has been released but there are still a few issues to be dealt with and even longer before this new standard will have an impact on actual distributions. FHS is not yet used in any distributions but Debian has announced they will use it in Debian 2.1 which is their next distribution.

It is also a good idea to read the Linux Installation guides thoroughly and if you are using a PC system, which I guess the majority still does, you can find much relevant and useful information in the FAQs for the newsgroup comp.sys.ibm.pc.hardware especially for storage media.

This is also a learning experience for myself and I hope I can start the ball rolling with this HOWTO and that it perhaps can evolve into a larger more detailed and hopefully even more correct HOWTO.

First of all we need a bit of legalese. Recent development shows it is quite important.

1.1 Copyright

This HOWTO is copyrighted 1996 Stein Gjoen.

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

TOC

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact the Linux HOWTO coordinator, at linux-howto@metalab.unc.edu via email.

1.2 Disclaimer

Use the information in this document at your own risk. I disavow any potential liability for the contents of this document. Use of the concepts, examples, and/or other content of this document is entirely at your own risk.

All copyrights are owned by their owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

You are strongly recommended to take a backup of your system before major installation and backups at regular intervals.

1.3 News

This release features a major restructuring and more additions than I can list here especially on added file system support.

This HOWTO now uses indexing and is based on SGMLtools version 1.0.5 and the old version will therefore not format this document properly.

Also quite new is a number of new translations available.

On the development front people are concentrating their energy towards completing Linux 2.2 and until that is released there is not going to be much news on disk technology for Linux.

Debian 2.1 is readying for release and as I use Debian for my test systems I will make more updates when I upgrade.

The latest version number of this document can be gleaned from my plan entry if you finger my Nyx account.

Also, the latest version will be available on my web space on nyx in a number of formats:

- HTML.
- plain ASCII text.
- compressed postscript.
- SGML source.

A European mirror of the Multi Disk HOWTO just went on line.

1.4 Credits

TOC

In this version I have the pleasure of acknowledging even more people who have contributed in one way or another:

```
ronnej (at ) ucs.orst.edu
cm (at) kukuruz.ping.at
armbru (at) pond.sub.org
R.P.Blake (at) open.ac.uk
neuffer (at) goofy.zdv.Uni-Mainz.de
sjmudd (at) redestb.es
nat (at) nataa.fr.eu.org
sundbyk (at) oslo.geco-prakla.slb.com
ggjoeen (at) online.no
mike (at) i-Connect.Net
roth (at) uiuc.edu
phall (at) ilap.com
szaka (at) mirror.cc.u-szeged.hu
CMckeon (at) swcp.com
kris (at) koentopp.de
edick (at) idcomm.com
pot (at) fly.cnuce.cnr.it
earl (at) sbox.tu-graz.ac.at
ebacon (at) oanet.com
vax (at) linkdead.paranoia.com
tschenk (at) theoffice.net
pjfarley (at) dorsai.org
jean (at) stat.ubc.ca
johnf (at) whitsunday.net.au
clasen (at) unidui.uni-duisburg.de
eeslgw (at) ee.surrey.asc.uk
adam (at) onshore.com
anikolae (at) wega-fddi2.rz.uni-ulm.de
cjaeger (at) dwave.net
eperezte (at) c2i.net
```

Special thanks go to `nakano (at) apm.seikei.ac.jp` for doing the Japanese translation, general contributions as well as contributing an example of a computer in an academic setting, which is included at the end of this document.

There are now many new translations available and special thanks go to the translators for the job and the input they have given:

- German Translation by `chewie (at) nuernberg.netsurf.de`
- Swedish Translation by `jonah (at) swipnet.se`

- French Translation by `Patrick.Loiseleur (at) lri.fr`

Also DPT is acknowledged for sending me documentation on their controllers as well as permission to quote from the material. These quotes have been approved before appearing here and will be clearly labelled. No quotes as of yet but that is coming.

Not many still, so please read through this document, make a contribution and join the elite. If I have forgotten anyone, please let me know.

New in this version is an appendix with a few tables you can fill in for your system in order to simplify the design process.

Any comments or suggestions can be mailed to my mail address on nyx: sgjoen@nyx.net.

So let's cut to the chase where `swap` and `/tmp` are racing along hard drive...

---

2. Structure

As this type of document is supposed to be as much for learning as a technical reference document I have rearranged the structure to this end. For the designer of a system it is more useful to have the information presented in terms of the goals of this exercise than from the point of view of the logical layer structure of the devices themselves. Nevertheless this document would not be complete without such a layer structure the computer field is so full of, so I will include it here as an introduction to how it works.

It is a long time since the *mini* in mini-HOWTO could be defended as proper but I am convinced that this document is as long as it needs to be in order to make the right design decisions, and not longer.

2.1 Logical structure

This is based on how each layer access each other, traditionally with the application on top and the physical layer on the bottom. It is quite useful to show the interrelationship between each of the layers used in controlling drives.

```
_____________________________________________________________
|__     File structure          ( /usr /tmp etc)        __|
|__     File system             (ext2fs, vfat etc)      __|
|__     Volume management       (AFS)                   __|
|__     RAID, concatenation     (md)                    __|
|__     Device driver           (SCSI, IDE etc)         __|
|__     Controller              (chip, card)            __|
|__     Connection              (cable, network)        __|
|__     Drive                   (magnetic, optical etc) __|
-------------------------------------------------------------
```

In the above diagram both volume management and RAID and concatenation are optional layers. The 3 lower layers are in hardware. All parts are discussed at length later on in this document.

2.2 Document structure

Most users start out with a given set of hardware and some plans on what they wish to achieve and how big the system should be. This is the point of view I will adopt in this document in presenting the material, starting out with hardware, continuing with design constraints before detailing the design strategy that I have found to work well. I have used this both for my own personal computer at home, a multi purpose server at work and found it worked quite well. In addition my Japanese co-worker in this project have applied the same strategy on a server in an academic setting with similar success.

Finally at the end I have detailed some configuration tables for use in your own design. If you have any comments regarding this or notes from your own design work I would like to hear from you so this

document can be upgraded.

2.3 Reading plan

Although not the biggest HOWTO it is nevertheless rather big already and I have been requested to make a reading plan to make it possible to cut down on the volume

Expert

(aka the elite). If you are familiar with Linux as well as disk drive technologies you will find most of what you need in the appendices. Additionally you are recommended to read the FAQ and the Bits'n'pieces chapter.

Experienced

(aka Competent). If you are familiar with computers in general you can go straight to the chapters on technologies and continue from there on.

Newbie

(mostly harmless). You just have to read the whole thing. Sorry. In addition you are also recommended to read all the other disk related HOWTOs.

---

3. Drive technologies

A far more complete discussion on drive technologies for IBM PCs can be found at the home page of The Enhanced IDE/Fast-ATA FAQ which is also regularly posted on Usenet News. Here I will just present what is needed to get an understanding of the technology and get you started on your setup.

3.1 Drives

This is the physical device where your data lives and although the operating system makes the various types seem rather similar they can in actual fact be very different. An understanding of how it works can be very useful in your design work. Floppy drives fall outside the scope of this document, though should there be a big demand I could perhaps be persuaded to add a little here.

3.2 Geometry

Physically disk drives consists of one or more platters containing data that is read in and out using sensors mounted on movable heads that are fixed with respects to themselves. Data transfers therefore happens across all surfaces simultaneously which defines a cylinder of tracks. The drive is also divided into sectors containing a number of data fields.

Drives are therefore often specified in terms of its geometry: the number of Cylinders, Heads and Sectors (CHS).

For various reasons there is now a number of translations between

- the physical CHS of the drive itself
- the logical CHS the drive reports to the BIOS or OS
- the logical CHS used by the OS

Basically it is a mess and a source of much confusion. For more information you are strongly recommended to read the *Large Disk mini-HOWTO*

3.3 Media

The media technology determines important parameters such as read/write rates, seek times, storage size as well as if it is read/write or read only.

Magnetic Drives

This is the typical read-write mass storage medium, and as everything else in the computer world, comes in many flavours with different properties. Usually this is the fastest technology and offers read/write capability. The platter rotates with a constant angular velocity (CAV) with a variable physical sector density for more efficient magnetic media area utilisation. In other words, the number of bits per unit length is kept roughly constant by increasing the number of logical sectors for the outer tracks.

Typical values for rotational speeds are 4500 and 5400 RPM, though 7200 is also used. Very recently also 10000 RPM has entered the mass market. Seek times are around 10 ms, transfer rates quite variable from one type to another but typically 4-40 MB/s. With the extreme high performance drives you should remember that performance costs more

electric power which is dissipated as heat, see the point on Power and Heating.

Note that there are several kinds of transfers going on here, and that these are quoted in different units. First of all there is the platter-to-drive cache transfer, usually quoted in Mbits/s. Typical values here is about 50-250 Mbits/s. The second stage is from the built in drive cache to the adapter, and this is typically quoted in MB/s, and typical quoted values here is 3-40 MB/s. Note, however, that this assumed data is already in the cache and hence for maximum readout speed from the drive the effective transfer rate will decrease dramatically.

Optical Drives

Optical read/write drives exist but are slow and not so common. They were used in the NeXT machine but the low speed was a source for much of the complaints. The low speed is mainly due to the thermal nature of the phase change that represents the data storage. Even when using relatively powerful lasers to induce the phase changes the effects are still slower than the magnetic effect used in magnetic drives.

Today many people use CD-ROM drives which, as the name suggests, is read-only. Storage is about 650 MB, transfer speeds are variable, depending on the drive but can exceed 1.5 MB/s. Data is stored on a spiraling single track so it is not useful to talk about geometry for this. Data density is constant so the drive uses constant linear velocity (CLV). Seek is also slower, about 100 ms, partially due to the spiraling track. Recent, high speed drives, use a mix of CLV and CAV in order to maximize performance. This also reduces access time caused by the need to reach correct rotational speed for readout.

A new type (DVD) is on the horizon, offering up to about 18 GB on a single disk.

Solid State Drives

This is a relatively recent addition to the available technology and has been made popular especially in portable computers as well as in embedded systems. Containing no movable parts they are very fast both in terms of access and transfer rates. The most popular type is flash RAM, but also other types of RAM is used. A few years ago many had great hopes for magnetic bubble memories but it turned out to be relatively expensive and is not that common.

In general the use of RAM disks are regarded as a bad idea as it is normally more sensible to add more RAM to the motherboard and let the operating system divide the memory pool into buffers, cache, program and data areas. Only in very special cases, such as real time systems with short time margins, can RAM disks be a sensible solution.

Flash RAM is today available in several 10's of megabytes in storage and one might be tempted to use it for fast, temporary storage in a computer. There is however a huge snag with this: flash RAM has a finite life time in terms of the number of times you can rewrite data, so putting `swap`, `/tmp` or `/var/tmp` on such a device will certainly shorten its lifetime dramatically. Instead, using flash RAM for directories that are read often but rarely written to, will be a big performance win.

In order to get the optimum life time out of flash RAM you will need to use special drivers that will use the RAM evenly and minimize the number of block erases.

This example illustrates the advantages of splitting up your directory structure over several devices.

Solid state drives have no real cylinder/head/sector addressing but for compatibility reasons this is simulated by the driver to give a uniform interface to the operating system.

3.4 Interfaces

There is a plethora of interfaces to chose from widely ranging in price and performance. Most motherboards today include IDE interface which are part of modern chipsets.

TOC

Many motherboards also include a SCSI interface chip made by Symbios (formerly NCR) and that is connected directly to the PCI bus. Check what you have and what BIOS support you have with it.

MFM and RLL

Once upon a time this was the established technology, a time when 20 MB was awesome, which compared to todays sizes makes you think that dinosaurs roamed the Earth with these drives. Like the dinosaurs these are outdated and are slow and unreliable compared to what we have today. Linux does support this but you are well advised to think twice about what you would put on this. One might argue that an emergency partition with a suitable vintage of DOS might be fitting.

ESDI

Actually, ESDI was an adaptation of the very widely used SMD interface used on "big" computers to the cable set used with the ST506 interface, which was more convenient to package than the 60-pin + 26-pin connector pair used with SMD. The ST506 was a "dumb" interface which relied entirely on the controller and host computer to do everything from computing head/cylinder/sector locations and keeping track of the head location, etc. ST506 required the controller to extract clock from the recovered data, and control the physical location of detailed track features on the medium, bit by bit. It had about a 10-year life if you include the use of MFM, RLL, and ERLL/ARLL modulation schemes. ESDI, on the other hand, had intelligence, often using three or four separate microprocessors on a single drive, and high-level commands to format a track, transfer data, perform seeks, and so on. Clock recovery from the data stream was accomplished at the drive, which drove the clock line and presented its data in NRZ, though error correction was still the task of the controller. ESDI allowed the use of variable bit density recording, or, for that matter, any other modulation technique, since it was locally generated and resolved at the drive. Though many of the techniques used in ESDI were later incorporated in IDE, it was the increased popularity of SCSI which led to the demise of ESDI in computers. ESDI had a life of about 10 years, though mostly in servers and otherwise "big" systems rather than PC's.

IDE and ATA

Progress made the drive electronics migrate from the ISA slot card over to the drive itself and Integrated Drive Electronics was borne. It was simple, cheap and reasonably fast so the BIOS designers provided the kind of snag that the computer industry is so full of. A combination of an IDE limitation of 16 heads together with the BIOS limitation of 1024 cylinders gave us the infamous 504 MB limit. Following the computer industry traditions again, the snag was patched with a kludge and we got all sorts of translation schemes and BIOS bodges. This means that you need to read the installation documentation very carefully and check up on what BIOS you have and what date it has as the BIOS has to tell Linux what size drive you have. Fortunately with Linux you can also tell the kernel directly what size drive you have with the drive parameters, check the documentation for LILO and Loadlin, thoroughly. Note also that IDE is equivalent to ATA, AT Attachment. IDE uses CPU-intensive Programmed Input/Output (PIO) to transfer data to and from the drives and has no capability for the more efficient Direct Memory Access (DMA) technology. Highest transfer rate is 8.3 MB/s.

EIDE, Fast-ATA and ATA-2

These 3 terms are roughly equivalent, fast-ATA is ATA-2 but EIDE additionally includes ATAPI. ATA-2 is what most use these days which is faster and with DMA. Highest transfer rate is increased to 16.6 MB/s.

Ultra-ATA

A new, faster DMA mode that is approximately twice the speed of EIDE PIO-Mode 4 (33 MB/s). Disks with and without Ultra-ATA can be mixed on the same cable without speed penalty for the faster adapters. The Ultra-ATA

interface is electrically identical with the normal Fast-ATA interface, including the maximum cable length.

ATAPI

The ATA Packet Interface was designed to support CD-ROM drives using the IDE port and like IDE it is cheap and simple.

SCSI

The Small Computer System Interface is a multi purpose interface that can be used to connect to everything from drives, disk arrays, printers, scanners and more. The name is a bit of a misnomer as it has traditionally been used by the higher end of the market as well as in work stations since it is well suited for multi tasking environments.

The standard interface is 8 bits wide and can address 8 devices. There is a wide version with 16 bit that is twice as fast on the same clock and can address 16 devices. The host adapter always counts as a device and is usually number 7. It is also possible to have 32 bit wide busses but this usually requires a double set of cables to carry all the lines.

The old standard was 5 MB/s and the newer fast-SCSI increased this to 10 MB/s. Recently ultra-SCSI, also known as Fast-20, arrived with 20 MB/s transfer rates for an 8 bit wide bus. New low voltage differential (LVD) signalling allows these high speeds as well as much longer cabling than before.

Even more recently an even faster standard has been proposed: SCSI 160/m which is capable of a monstrous 160 MB/s over a 16 bit wide bus. Support is scarce yet but for a few 10000 RPM drives that can transfer 40 MB/s sustained. Putting 6 such drives on a RAID will keep such a bus saturated and also saturate most PCI busses. Obviously this is only for the very highest end servers per today.

The higher performance comes at a cost that is usually higher than for (E)IDE. The importance of correct termination and good quality cables cannot be overemphasized. SCSI drives also often tend to be of a higher quality than IDE drives. Also adding SCSI devices tend to be easier than adding more IDE drives: Often it is only a matter of plugging or unplugging the device; some people do this without powering down the system. This feature is most convenient when you have multiple systems and you can just take the devices from one system to the other should one of them fail for some reason.

There is a number of useful documents you should read if you use SCSI, the SCSI HOWTO as well as the SCSI FAQ posted on Usenet News.

SCSI also has the advantage you can connect it easily to tape drives for backing up your data, as well as some printers and scanners. It is even possible to use it as a very fast network between computers while simultaneously share SCSI devices on the same bus. Work is under way but due to problems with ensuring cache coherency between the different computers connected, this is a non trivial task.

SCSI numbers are also used for arbitration. If several drives request service, the drive with the lowest number is given priority.

3.5 Cabling

I do not intend to make too many comments on hardware but I feel I should make a little note on cabling. This might seem like a remarkably low technological piece of equipment, yet sadly it is the source of many frustrating problems. At todays high speeds one should think of the cable more of a an RF device with its inherent demands on impedance matching. If you do not take your precautions you will get a much reduced reliability or total failure. Some SCSI host adapters are more sensitive to this than others.

Shielded cables are of course better than unshielded but the price is much higher. With a little care you can get good performance from a cheap unshielded cable.

- For Fast-ATA and Ultra-ATA, the maximum cable length is specified as 45cm (18"). The data lines of both channels are connected on many boards, though, so they count as **one** cable. In any case EIDE cables should as short as possible. If there are mysterious crashes or spontaneous changes of data, it is well worth investigating your cabling. Try a lower PIO mode or disconnect the second channel and see if the problem still occurs.
- Use as short cable as possible, but do not forget the 30 cm minimum separation for ultra SCSI and 60 cm separation for differential SCSI.
- Avoid long stubs between the cable and the drive, connect the plug on the cable directly to the drive without an extension.
- SCSI Cabling limitations:

```
Bus Speed (MHz)          |    Max Length (m)
-------------------------------------------------
 5                       |        6
10  (fast)               |        3
20  (fast-20 / ultra)    |        3 (max 4 devices), 1.5 (max 8 devices)
xx  (differential)       |       25 (max 16 devices
-------------------------------------------------
```

- Use correct termination for SCSI devices and at the correct positions: both ends of the SCSI chain. Remember the host adapter itself may have on board termination.
- Do not mix shielded or unshielded cabling, do not wrap cables around metal, try to avoid proximity to metal parts along parts of the cabling. Any such discontinuities can cause impedance mismatching which in turn can cause reflection of signals which increases noise on the cable. This problems gets even more severe in the case of multi channel controllers. Recently someone suggested wrapping bubble plastic around the cables in order to avoid too close proximity to metal, a real problem inside crowded cabinets.

More information on SCSI cabling and termination can be found at other web pages around the net.

3.6 Host Adapters

This is the other end of the interface from the drive, the part that is connected to a computer bus. The speed of the computer bus and that of the drives should be roughly similar, otherwise you have a bottleneck in your system. Connecting a RAID 0 disk-farm to a ISA card is pointless. These days most computers come with 32 bit PCI bus capable of 132 MB/s transfers which should not represent a bottleneck for most people in the near future.

As the drive electronic migrated to the drives the remaining part that became the (E)IDE interface is so small it can easily fit into the PCI chip set. The SCSI host adapter is more complex and often includes a small CPU of its own and is therefore more expensive and not integrated into the PCI chip sets available today. Technological evolution might change this.

Some host adapters come with separate caching and intelligence but as this is basically second guessing the operating system the gains are heavily dependent on which operating system is used. Some of the more primitive ones, that shall remain nameless, experience great gains. Linux, on the other hand, have so much smarts of its own that the gains are much smaller.

Mike Neuffer, who did the drivers for the DPT controllers, states that the DPT controllers are intelligent enough that given enough cache memory it will give you a big push in performance and suggests that people who have experienced little gains with smart controllers just have not used a sufficiently intelligent caching controller.

TOC

3.7 Multi Channel Systems

In order to increase throughput it is necessary to identify the most significant bottlenecks and then eliminate them. In some systems, in particular where there are a great number of drives connected, it is advantageous to use several controllers working in parallel, both for SCSI host adapters as well as IDE controllers which usually have 2 channels built in. Linux supports this.

Some RAID controllers feature 2 or 3 channels and it pays to spread the disk load across all channels. In other words, if you have two SCSI drives you want to RAID and a two channel controller, you should put each drive on separate channels.

3.8 Multi Board Systems

In addition to having both a SCSI and an IDE in the same machine it is also possible to have more than one SCSI controller. Check the SCSI-HOWTO on what controllers you can combine. Also you will most likely have to tell the kernel it should probe for more than just a single SCSI or a single IDE controller. This is done using kernel parameters when booting, for instance using LILO. Check the HOWTOs for SCSI and LILO for how to do this.

Multi board systems can offer significant speed gains if you configure your disks right, especially for RAID0. Make sure you interleave the controllers as well as the drives, so that you add drives to the md RAID device in the right order. If controller 1 is connected to drives `sda` and `sdc` while controller 2 is connected to drives `sdb` and `sdd` you will gain more paralellicity by adding in the order of `sda - sdc - sdb - sdd` rather than `sda - sdb - sdc - sdd` because a read or write over more then one cluster will be more likely to span two controllers.

The same methods can also be applied to IDE. Most motherboards come with typically 4 IDE ports:

- `hda` primary controller
- `hdb` primary slave
- `hdc` secondary controller
- `hdd` secondary slave

where the two primaries share one flat cable and the secondaries share another cable. Modern chipsets keep these independent. Therefore it is best to RAID in the order `hda - hdc - hdb - hdd` as this will most likely parallelise both channels.

3.9 Speed Comparison

The following tables are given just to indicate what speeds are possible but remember that these are the theoretical maximum speeds. All transfer rates are in MB per second and bus widths are measured in bits.

Controllers

```
IDE              :        8.3 - 16.7
Ultra-ATA        :       33

SCSI             :
                         Bus width (bits)

Bus Speed (MHz)          |        8      16      32
---------------------------------------------------
```

TOC

```
 5                        |        5      10      20
10  (fast)                |       10      20      40
20  (fast-20 / ultra)     |       20      40      80
40  (fast-40 / ultra-2)   |       40      80      --
-----------------------------------------------------
```

Bus Types

```
ISA             :        8-12
EISA            :       33
VESA            :       40    (Sometimes tuned to 50)

PCI
                        Bus width (bits)

Bus Speed (MHz)         |       32      64
-----------------------------------------------------
33                      |       132     264
66                      |       264     528
-----------------------------------------------------
```

3.10 Benchmarking

This is a very, very difficult topic and I will only make a few cautious comments about this minefield. First of all, it is more difficult to make comparable benchmarks that have any actual meaning. This, however, does not stop people from trying...

Instead one can use benchmarking to diagnose your own system, to check it is going as fast as it should, that is, not slowing down. Also you would expect a significant increase when switching from a simple file system to RAID, so a lack of performance gain will tell you something is wrong.

When you try to benchmark you should not hack up your own, instead look up `iozone` and `bonnie` and read the documentation very carefully. In particular make sure your buffer size is bigger than your RAM size, otherwise you test your RAM rather than your disks which will give you unrealistically high performance.

A very simple benchmark can be obtained using `hdparm -tT` which can be used both on IDE and SCSI drives.

For more information on benchmarking and software for a number of platforms, check out ACNC benchmark page.

3.11 Comparisons

SCSI offers more performance than EIDE but at a price. Termination is more complex but expansion not too difficult. Having more than 4 (or in some cases 2) IDE drives can be complicated, with wide SCSI you can have up to 15 per adapter. Some SCSI host adapters have several channels thereby multiplying the number of possible drives even further.

For SCSI you have to dedicate one IRQ per host adapter which can control up to 15 drives. With EIDE you need one IRQ for each drive which can cause conflict.

RLL and MFM is in general too old, slow and unreliable to be of much use.

TOC

3.12 Future Development

SCSI-3 is under way and will hopefully be released soon. Faster devices are already being announced, recently an 80 MB/s and then a 160 MB/s monster specification has been proposed. These are based around the Ultra-2 standard (which used a 40 MHz clock) combined with a 16 bit cable.

Some manufacturers already announce SCSI-3 devices but this is currently rather premature as the standard is not yet firm. As the transfer speeds increase the saturation point of the PCI bus is getting closer. Currently the 64 bit version has a limit of 264 MB/s. The PCI transfer rate will in the future be increased from the current 33 MHz to 66 MHz, thereby increasing the limit to 528 MB/s.

Another trend is for larger and larger drives. I hear it is possible to get 55 GB on a single drive though this is rather expensive. Currently the optimum storage for your money is about 6.4 GB but also this is continuously increasing. The introduction of DVD will in the near future have a big impact, with nearly 20 GB on a single disk you can have a complete copy of even major FTP sites from around the world. The only thing we can be reasonably sure about the future is that even if it won't get any better, it will definitely be bigger.

Addendum: soon after I first wrote this I read that the maximum useful speed for a CD-ROM was 20x as mechanical stability would be too great a problem at these speeds. About one month after that again the first commercial 24x CD-ROMs were available... Currently you can get 40x and no doubt higher speeds are in the pipeline.

3.13 Recommendations

My personal view is that EIDE or Ultra ATA is the best way to start out on your system, especially if you intend to use DOS as well on your machine. If you plan to expand your system over many years or use it as a server I would strongly recommend you get SCSI drives. Currently wide SCSI is a little more expensive. You are generally more likely to get more for your money with standard width SCSI. There is also differential versions of the SCSI bus which increases maximum length of the cable. The price increase is even more substantial and cannot therefore be recommended for normal users.

In addition to disk drives you can also connect some types of scanners and printers and even networks to a SCSI bus.

Also keep in mind that as you expand your system you will draw ever more power, so make sure your power supply is rated for the job and that you have sufficient cooling. Many SCSI drives offer the option of sequential spin-up which is a good idea for large systems. See also Power and Heating.

---

4. File System Structure

Linux has been multi tasking from the very beginning where a number of programs interact and run continuously. It is therefore important to keep a file structure that everyone can agree on so that the system finds data where it expects to. Historically there has been so many different standards that it was confusing and compatibility was maintained using symbolic links which confused the issue even further and the structure ended looking like a maze.

In the case of Linux a standard was fortunately agreed on early on called the *File Systems Standard* (FSSTND) which today is used by all main Linux distributions.

Later it was decided to make a successor that should also support operating systems other than just Linux, called the *Filesystem Hierarchy Standard* (FHS) at version 2.1 currently. This standard is is under continuous development and will soon be adopted by Linux distributions.

I recommend not trying to roll your own structure as a lot of thought has gone into the standards and many software packages comply with the standards. Instead you can read more about this at the FHS home page.

This HOWTO endeavours to comply with FSSTND and will follow FHS when distributions become available.

4.1 File System Features

The various parts of FSSTND have different requirements regarding speed, reliability and size, for instance losing root is a pain but can easily be recovered. Losing `/var/spool/mail` is a rather different issue. Here is a quick summary of some essential parts and their properties and requirements. Note that this is just a guide, there can be binaries in `etc` and `lib` directories, libraries in `bin` directories and so on.

Swap

Speed

Maximum! Though if you rely too much on swap you should consider buying some more RAM. Note, however, that on many PC motherboards the cache will not work on RAM above 128 MB.

Size

Similar as for RAM. Quick and dirty algorithm: just as for tea: 16 MB for the machine and 2 MB for each user. Smallest kernel run in 1 MB but is tight, use 4 MB for general work and light applications, 8 MB for X11 or GCC or 16 MB to be comfortable. (The author is known to brew a rather powerful cuppa tea...)

Some suggest that swap space should be 1-2 times the size of the RAM, pointing out that the locality of the programs determines how effective your added swap space is. Note that using the same algorithm as for 4BSD is slightly incorrect as Linux does not allocate space for pages in core.

A more thorough approach is to consider swap space plus RAM as your total working set, so if you

TOC

know how much space you will need at most, you subtract the physical RAM you have and that is swap space you will need.

There is also another reason to be generous when dimensioning your swap space: memory leaks. Ill behaving programs that do not free the memory they allocate for themselves are said to have a memory leak. This allocation remains even after the offending program has stopped so this is a source of memory consumption. Once all physical RAM and swap space are exhausted the only solution is to reboot and start over. Thankfully such programs are not too common but should you come across one you will find that extra swap space will buy you extra time between reboots.

Also remember to take into account the type of programs you use. Some programs that have large working sets, such as finite element modeling (FEM) have huge data structures loaded in RAM rather than working explicitly on disk files. Data and computing intensive programs like this will cause excessive swapping if you have less RAM than the requirements.

Other types of programs can lock their pages into RAM. This can be for security reasons, preventing copies of data reaching a swap device or for performance reasons such as in a real time module. Either way, locking pages reduces the remaining amount of swappable memory and can cause the system to swap earlier then otherwise expected.

In `man 8 mkswap` it is explained that each swap partition can be a maximum of just under 128 MB in size for 32-bit machines and just under 256 MB for 64-bit machines.

Reliability

Medium. When it fails you know it pretty quickly and failure will cost you some lost work. You save often, don't you?

Note 1

Linux offers the possibility of interleaved swapping across multiple devices, a feature that can gain you much. Check out "`man 8 swapon`" for more details. However, software raiding `swap` across multiple devices adds more overheads than you gain.

Thus the `/etc/fstab` file might look like this:

```
/dev/sda1        swap              swap     pri=1            0        0
/dev/sdc1        swap              swap     pri=1            0        0
```

Remember that the `fstab` file is *very* sensitive to the formatting used, read the man page carefully and do *not* just cut and paste the lines above.

Note 2

Some people use a RAM disk for swapping or some other file systems. However, unless you have some very unusual requirements or setups you are unlikely to gain much from this as this cuts into the memory available for caching and buffering.

Note 2b

There is once exception: on a number of badly designed motherboards the on board cache memory is not able to cache all the RAM that can be addressed. Many older motherboards could accept 128 MB RAM but only cache the lower 64 MB. In such cases it would improve the performance if you used the upper (uncached) 64 MB RAM for RAMdisk based swap or other temporary storage.

TOC

Temporary Storage (/tmp and /var/tmp)

Speed

Very high. On a separate disk/partition this will reduce fragmentation generally, though `ext2fs` handles fragmentation rather well.

Size

Hard to tell, small systems are easy to run with just a few MB but these are notorious hiding places for stashing files away from prying eyes and quota enforcement and can grow without control on larger machines. Suggested: small home machine: 8 MB, large home machine: 32 MB, small server: 128 MB, and large machines up to 500 MB (The machine used by the author at work has 1100 users and a 300 MB `/tmp` directory). Keep an eye on these directories, not only for hidden files but also for old files. Also be prepared that these partitions might be the first reason you might have to resize your partitions.

Reliability

Low. Often programs will warn or fail gracefully when these areas fail or are filled up. Random file errors will of course be more serious, no matter what file area this is.

Files

Mostly short files but there can be a huge number of them. Normally programs delete their old `tmp` files but if somehow an interruption occurs they could survive. Many distributions have a policy regarding cleaning out `tmp` files at boot time, you might want to check out what your setup is.

Note1

In FSSTND there is a note about putting `/tmp` on RAM disk. This, however, is not recommended for the same reasons as stated for swap. Also, as noted earlier, do not use flash RAM drives for these directories. One should also keep in mind that some systems are set to automatically clean `tmp` areas on rebooting.

Note2

Older systems had a `/usr/tmp` but this is no longer recommended and for historical reasons a symbolic link now makes it point to one of the other `tmp` areas.

(* That was 50 lines, I am home and dry! *)

Spool Areas (/var/spool/news and /var/spool/mail)

Speed

High, especially on large news servers. News transfer and expiring are disk intensive and will benefit from fast drives. Print spools: low. Consider RAID0 for news.

Size

For news/mail servers: whatever you can afford. For single user systems a few MB will be sufficient if you read continuously. Joining a list server and taking a holiday is, on the other hand, not a good idea. (Again the machine I use at work has 100 MB reserved for the entire `/var/spool`)

Reliability

TOC

Mail: very high, news: medium, print spool: low. If your mail is very important (isn't it always?) consider RAID for reliability.

Files

Usually a huge number of files that are around a few KB in size. Files in the print spool can on the other hand be few but quite sizable.

Note

Some of the news documentation suggests putting all the `.overview` files on a drive separate from the news files, check out all news FAQs for more information. Typical size is about 3-10 percent of total news spool size.

Home Directories (`/home`)

Speed

Medium. Although many programs use `/tmp` for temporary storage, others such as some news readers frequently update files in the home directory which can be noticeable on large multiuser systems. For small systems this is not a critical issue.

Size

Tricky! On some systems people pay for storage so this is usually then a question of finance. Large systems such as nyx.net (which is a free Internet service with mail, news and WWW services) run successfully with a suggested limit of 100 KB per user and 300 KB as enforced maximum. Commercial ISPs offer typically about 5 MB in their standard subscription packages.

If however you are writing books or are doing design work the requirements balloon quickly.

Reliability

Variable. Losing `/home` on a single user machine is annoying but when 2000 users call you to tell you their home directories are gone it is more than just annoying. For some their livelihood relies on what is here. You do regular backups of course?

Files

Equally tricky. The minimum setup for a single user tends to be a dozen files, 0.5 - 5 KB in size. Project related files can be huge though.

Note1

You might consider RAID for either speed or reliability. If you want extremely high speed and reliability you might be looking at other operating system and hardware platforms anyway. (Fault tolerance etc.)

Note2

Web browsers often use a local cache to speed up browsing and this cache can take up a substantial amount of space and cause much disk activity. There are many ways of avoiding this kind of performance hits, for more information see the sections on Home Directories and WWW.

Note3

Users often tend to use up all available space on the `/home` partition. The Linux Quota subsystem is

capable of limiting the number of blocks and the number of inode a single user ID can allocate on per-filesystem basis. See the Linux Quota mini-HOWTO by Albert M.C. Tam `bertie (at) scn.org` for details on setup.

Main Binaries (`/usr/bin` and `/usr/local/bin`)

Speed

Low. Often data is bigger than the programs which are demand loaded anyway so this is not speed critical. Witness the successes of live file systems on CD ROM.

Size

The sky is the limit but 200 MB should give you most of what you want for a comprehensive system. A big system, for software development or a multi purpose server should perhaps reserve 500 MB both for installation and for growth.

Reliability

Low. This is usually mounted under root where all the essentials are collected. Nevertheless losing all the binaries is a pain...

Files

Variable but usually of the order of 10 - 100 KB.

Libraries (`/usr/lib` and `/usr/local/lib`)

Speed

Medium. These are large chunks of data loaded often, ranging from object files to fonts, all susceptible to bloating. Often these are also loaded in their entirety and speed is of some use here.

Size

Variable. This is for instance where word processors store their immense font files. The few that have given me feedback on this report about 70 MB in their various `lib` directories. A rather complete Debian 1.2 installation can take as much as 250 MB which can be taken as an realistic upper limit. The following ones are some of the largest disk space consumers: GCC, Emacs, TeX/LaTeX, X11 and perl.

Reliability

Low. See point Main binaries.

Files

Usually large with many of the order of 1 MB in size.

Note

For historical reasons some programs keep executables in the lib areas. One example is GCC which have some huge binaries in the `/usr/lib/gcc/lib` hierarchy.

TOC

Boot

Speed

Quite low: after all booting doesn't happen that often and loading the kernel is just a tiny fraction of the time it takes to get the system up and running.

Size

Quite small, a complete image with some extras fit on a single floppy so 5 MB should be plenty.

Reliability

High. See section below on Root.

Note 1

The most important part about the Boot partition is that on many systems it *must* reside below cylinder 1023. This is a BIOS limitation that Linux cannot get around.

Root

Speed

Quite low: only the bare minimum is here, much of which is only run at startup time.

Size

Relatively small. However it is a good idea to keep some essential rescue files and utilities on the root partition and some keep several kernel versions. Feedback suggests about 20 MB would be sufficient.

Reliability

High. A failure here will possibly cause a fair bit of grief and you might end up spending some time rescuing your boot partition. With some practice you can of course do this in an hour or so, but I would think if you have some practice doing this you are also doing something wrong.

Naturally you do have a rescue disk? Of course this is updated since you did your initial installation? There are many ready made rescue disks as well as rescue disk creation tools you might find valuable. Presumably investing some time in this saves you from becoming a root rescue expert.

Note 1

If you have plenty of drives you might consider putting a spare emergency boot partition on a separate physical drive. It will cost you a little bit of space but if your setup is huge the time saved, should something fail, will be well worth the extra space.

Note 2

For simplicity and also in case of emergencies it is not advisable to put the root partition on a RAID level 0 system. Also if you use RAID for your boot partition you have to remember to have the `md` option turned on for your emergency kernel.

Note 3

For simplicity it is quite common to keep Boot and Root on the same partition. if you do that, then in order to boot from LILO it is important that the essential boot files reside wholly within cylinder 1023.

This includes the kernel as well as files found in `/boot`.

DOS etc.

At the danger of sounding heretical I have included this little section about something many reading this document have strong feelings about. Unfortunately many hardware items come with setup and maintenance tools based around those systems, so here goes.

Speed

Very low. The systems in question are not famed for speed so there is little point in using prime quality drives. Multitasking or multi-threading are not available so the command queueing facility found in SCSI drives will not be taken advantage of. If you have an old IDE drive it should be good enough. The exception is to some degree Win95 and more notably NT which have multi-threading support which should theoretically be able to take advantage of the more advanced features offered by SCSI devices.

Size

The company behind these operating systems is not famed for writing tight code so you have to be prepared to spend a few tens of MB depending on what version you install of the OS or Windows. With an old version of DOS or Windows you might fit it all in on 50 MB.

Reliability

Ha-ha. As the chain is no stronger than the weakest link you can use any old drive. Since the OS is more likely to scramble itself than the drive is likely to self destruct you will soon learn the importance of keeping backups here.

Put another way: "*Your mission, should you choose to accept it, is to keep this partition working. The warranty will self destruct in 10 seconds...*"

Recently I was asked to justify my claims here. First of all I am not calling DOS and Windows sorry excuses for operating systems. Secondly there are various legal issues to be taken into account. Saying there is a connection between the last two sentences are merely the ravings of the paranoid. Surely. Instead I shall offer the esteemed reader a few key words: DOS 4.0, DOS 6.x and various drive compression tools that shall remain nameless.

4.2 Explanation of Terms

Naturally the faster the better but often the happy installer of Linux has several disks of varying speed and reliability so even though this document describes performance as 'fast' and 'slow' it is just a rough guide since no finer granularity is feasible. Even so there are a few details that should be kept in mind:

Speed

This is really a rather woolly mix of several terms: CPU load, transfer setup overhead, disk seek time and transfer rate. It is in the very nature of tuning that there is no fixed optimum, and in most cases price is the dictating factor. CPU load is only significant for IDE systems where the CPU does the transfer itself but is generally low for SCSI, see SCSI documentation for actual numbers. Disk seek time is also small, usually in the millisecond range. This however is not a problem if you use command queueing on SCSI where you then overlap commands keeping the bus busy all the time. News spools are a special case consisting of a huge

number of normally small files so in this case seek time can become more significant.

There are two main parameters that are of interest here:

Seek

is usually specified in the average time take for the read/write head to seek from one track to another. This parameter is important when dealing with a large number of small files such as found in spool files. There is also the extra seek delay before the desired sector rotates into position under the head. This delay is dependent on the angular velocity of the drive which is why this parameter quite often is quoted for a drive. Common values are 4500, 5400 and 7200 RPM (rotations per minute). Higher RPM reduces the seek time but at a substantial cost. Also drives working at 7200 RPM have been known to be noisy and to generate a lot of heat, a factor that should be kept in mind if you are building a large array or "disk farm". Very recently drives working at 10000 RPM has entered the market and here the cooling requirements are even stricter and minimum figures for air flow are given.

Transfer

is usually specified in megabytes per second. This parameter is important when handling large files that have to be transferred. Library files, dictionaries and image files are examples of this. Drives featuring a high rotation speed also normally have fast transfers as transfer speed is proportional to angular velocity for the same sector density.

It is therefore important to read the specifications for the drives very carefully, and note that the maximum transfer speed quite often is quoted for transfers out of the on board cache (burst speed) and *not* directly from the platter (sustained speed). See also section on Power and Heating.

Reliability

Naturally no-one would want low reliability disks but one might be better off regarding old disks as unreliable. Also for RAID purposes (See the relevant information) it is suggested to use a mixed set of disks so that simultaneous disk crashes become less likely.

So far I have had only one report of total file system failure but here unstable hardware seemed to be the cause of the problems.

Disks are cheap these days yet people still underestimate the value of the contents of the drives. If you need higher reliability make sure you replace old drives and keep spares. It is not unusual that drives can work more or less continuous for years and years but what often kills a drive in the end is power cycling.

Files

The average file size is important in order to decide the most suitable drive parameters. A large number of small files makes the average seek time important whereas for big files the transfer speed is more important. The command queueing in SCSI devices is very handy for handling large numbers of small files, but for transfer EIDE is not too far behind SCSI and normally much cheaper than SCSI.

5. File Systems

Over time the requirements for file systems have increased and the demands for large structures, large files, long file names and more has prompted ever more advanced file systems, the system that accesses and organises the data on mass storage. Today there is a large number of file systems to choose from and this section will describe these in detail.

The emphasis is on Linux but with more input I will be happy to add information for a wider audience.

5.1 General Purpose File Systems

Most operating systems usually have a general purpose file system for every day use for most kinds of files, reflecting available features in the OS such as permission flags, protection and recovery.

`minix`

This was the original fs for Linux, back in the days Linux was hosted on minix machines. It is simple but limited in features and hardly every used these days other than in some rescue disks as it is rather compact.

`xiafs` and `extfs`

These are also old and have fallen in disuse and are no longer recommended.

`ext2fs`

This is the established standard for general purpose in the Linux world. It is fast, efficient and mature and is under continuous development and features such as ACL and transparent compression are on the horizon.

For more information check the ext2fs home page.

`ufs`

This is the fs used by BSD and variants thereof. It is mature but also developed for older types of disk drives where geometries were known. The fs uses a number of tricks to optimise performance but as disk geometries are translated in a number of ways the net effect is no longer so optimal.

TOC

efs

The Extent File System (efs) is Silicon Graphics' early file system widely used on IRIX before version 6.0 after which xfs has taken over. While migration to xfs is encouraged efs is still supported and much used on CDs.

There is a Linux driver available in early beta stage, available at Linux extent file system home page.

reiserfs

As of July, 23th 1997 Hans Reiser `reiser (at) RICOCHET.NET` has put up the source to his tree based reiserfs on the web. While his filesystem has some very interesting features and is much faster than `ext2fs`, it is still very experimental and difficult to integrate with the standard kernel. Expect some interesting developments in the future - this is different from your "average log based file system for Linux" project, because Hans already has working code.

enh-fs

Currently in alpha stage the Enhanced File System project aims to combine file system and volume management into a single layer.

5.2 Microsoft File Systems

This company is responsible for a lot, including a number of filesystems that has at the very least caused confusions.

fat

Actually there are 2 `fat`s out there, `fat12` and `fat16` depending on the partition size used but fortunately the difference is so minor that the whole issue is transparent.

On the plus side these are fast and simple and most OSes understands it and can both read and write this fs. And that is about it.

The minus side is limited safety, severely limited permission flags and atrocious scalability. For instance with `fat` you cannot have partitions larger than 2 GB.

fat32

After about 10 years Microsoft realised `fat` was about, well, 10 years behind the times and created this fs which scales reasonably well.

Permission flags are still limited. NT 4.0 cannot read this file system but Linux can.

TOC

vfat

At the same time as Microsoft launched `fat32` they also added support for long file names, known as `vfat`.

Linux reads `vfat` and `fat32` partitions by mounting with type `vfat`.

ntfs

This is the native fs of Win-NT but as complete information is available there is limited support for other OSes.

5.3 Logging and Journaling File Systems

These take a radically different approach to file updates by logging modifications for files in a log and later at some time checkpointing the logs.

Reading is roughly as fast as traditional file systems that always update the files directly. Writing is much faster as only updates are appended to a log. All this is transparent to the user. It is in reliability and particularly in checking file system integrity that these file systems really shine. Since the data before last checkpointing is known to be good only the log has to be checked, and this is much faster than for traditional file systems.

Note that while *logging* filesystems keep track of changes made to both data and inodes, *journaling* filesystems keep track only of inode changes.

Linux has quite a choice in such file systems but none are yet in production quality. Some are also on hold.

- Adam Richter from Yggdrasil posted some time ago that they have been working on a compressed log file based system but that this project is currently on hold. Nevertheless a non-working version is available on their FTP server. Check out the Yggdrasil ftp server where special patched versions of the kernel can be found.
- Another project is the Linux log-structured Filesystem Project which sadly also is on hold. Nevertheless this page contains much information on the topic.
- Finally there is the dtfs -- A Log-Structured Filesystem For Linux which seems to be going strong. Still in alpha but sufficiently complete to make programs run off this file system

5.4 Read-only File Systems

Read-only media has not escaped the ever increasing complexities seen in more general file systems so again there is a large choice to chose from with corresponding opportunities for exciting mistakes.

Most of these are used with the CD-ROM media but also the new DVD can be used and you can even use it through the loopback device on a hard disk file for verifying an image before burning a ROM.

TOC

There is a read-only `romfs` for Linux but as that is not disk related nothing more will be said about here.

High Sierra

This was one of the earliest standards for CD-ROM formats, supposedly named after the hotel where the final agreement took place.

`High Sierra` was so limited in features that new extensions simply had to appear and while there has been no end to new formats the original `High Sierra` remains the common precursor and is therefore still widely supported.

iso9660

The International Standards Organisation made their extensions and formalised the standard into what we know as the `iso9660` standard.

The Linux iso9660 file system supports both High Sierra as well as `Rock Ridge` extensions.

Rock Ridge

Not everyone accepts limits like short filenames and lack of permissions so very soon the `Rock Ridge` extensions appeared to rectify these shortcomings.

Joliet

Microsoft, not be be outdone in the standards extension game, decided it should extend CD-ROM formats with some internationalisation features and called it `Joliet`.

Linux supports this standards in kernels 2.0.34 or newer. You need to enable NLS in order to use it.

Trivia

Joliet is a city outside Chicago; best known for being the site of the prison where Jake was locked up in the movie "Blues Brothers." Rock Ridge (the UNIX extensions to ISO 9660) is named after the (fictional) town in the movie "Blazing Saddles."

UDF

With the arrival of DVD with up to about 17 GB of storage capacity the world seemingly needed another format, this time ambitiously named Universal Disk Format (UDF). This is intended to replace `iso9660` and will be required for DVD.

Currently this is not in the standard Linux kernel but a project is underway to make a UDF driver for Linux. Patches and documentation are available.

5.5 Networking File Systems

TOC

There is a large number of networking technologies available that lets you distribute disks throughout a local or even global networks. This is somewhat peripheral to the topic of this HOWTO but as it can be used with local disks I will cover this briefly. It would be best if someone (else) took this into a separate HOWTO...

NFS

This is one of the earliest systems that allows mounting a file space on one machine onto another. There are a number of problems with NFS ranging from performance to security but it has nevertheless become established.

AFS

This is a system that allows efficient sharing of files across large networks. Starting out as an academic project it is now sold by Transarc whose homepage gives you more details.

Derek Atkins, of MIT, ported AFS to Linux and has also set up the Linux AFS mailing List (linux-afs@mit.edu) for this which is open to the public. Requests to join the list should go to linux-afs-request@mit.edu and finally bug reports should be directed to linux-afs-bugs@mit.edu.

Important: as AFS uses encryption it is restricted software and cannot easily be exported from the US.

IBM who owns Transarc, has announced the availability of the latest version of client as well as server for Linux.

Arla is a free AFS implementation, check the Arla homepage for more information as well as documentation.

Coda

Work has started on a free replacement of AFS and is called Coda.

nbd

The Network Block Device (nbd) is available in Linux kernel 2.2 and later and offers reportedly excellent performance. The interesting thing here is that it can be combined with RAID (see later).

GFS

The Global File System is a new file system designed for storage across a wide area network. it is currently in the early stages and more information will come later.

5.6 Special File Systems

TOC

In addition to the general file systems there is also a number of more specific ones, usually to provide higher performance or other features, usually with a tradeoff in other respects.

`tmpfs` and `swapfs`

For short term fast file storage SunOS offers `tmpfs` which is about the same as the `swapfs` on NeXT. This overcomes the inherent slowness in `ufs` by caching file data and keeping control information in memory. This means that data on such a file system will be lost when rebooting and is therefore mainly suitable for `/tmp` area but not `/var/tmp` which is where temporary data that must survive a reboot, is placed.

SunOS offers very limited tuning for `tmpfs` and the number of files is even limited by total physical memory of the machine.

Linux does not have an equivalent to such file system and it is felt by many that `ext2fs` is fast enough to eliminate the need.

`userfs`

The user file system (`userfs`) allows a number of extensions to traditional file system use such as FTP based file system, compression (`arcfs`) and fast prototyping and many other features. The `docfs` is based on this filesystem. Check the [userfs homepage] for more information.

`devfs`

When disks are added, removed or just fail it is likely that disk device names of the remaining disks will change. For instance if `sdb` fails then the old `sdc` becomes `sdb`, the old `sdc` becomes `sdb` and so on. Note that in this case `hda`, `hdb` etc will remain unchanged. Likewise if a new drive is added the reverse may happen.

There is no guarantee that SCSI ID 0 becomes `sda` and that adding disks in increasing ID order will just add a new device name without renaming previous entries, as some SCSI drivers assign from ID 0 and up while others reverse the scanning order. Likewise adding a SCSI host adapter can also cause renaming.

Generally device names are assigned in the order they are found.

The source of the problem lies in the limited number of bits available for major and minor numbering in the device files used to describe the device itself. You an see these in the /dev directory, info on the numbering and allocation can be found in `man MAKEDEV`. Currently there are 2 solutions to this problem in various stages of development:

scsidev

> works by creating a database of drives and where they belong, check *man scsifs* for more information

devfs

is a more long term project aimed at getting around the whole business of device numbering by making the /dev directory a kernel file system in the same way as /procfs is. More information will appear as it becomes available.

5.7 File System Recommendations

There is a jungle of choices but generally it is recommended to use the general file system that comes with your distribution. If you use `ufs` and have some kind of `tmpfs` available you should first start off with the general file system to get an idea of the space requirements and if necessary buy more RAM to support the size of `tmpfs` you need. Otherwise you will end up with mysterious crashes and lost time.

If you use dual boot and need to transfer data between the two OSes one of the simplest ways is to use an appropriately sized partition formatted with `fat` as most systems can reliably read and write this. Remember the limit of 2 GB for `fat` partitions.

For more information of file system interconnectivity you can check out the file system page.

To avoid total havoc with device renaming if a drive fails check out the scanning order of your system and try to keep your root system on `hda` or `sda` and removable media such as ZIP drives at the end of the scanning order.

---

6. Technologies

In order to decide how to get the most of your devices you need to know what technologies are available and their implications. As always there can be some tradeoffs with respect to speed, reliability, power, flexibility, ease of use and complexity.

Many of the techniques described below can be stacked in a number of ways to maximise performance and reliability, though at the cost of added complexity.

6.1 RAID

This is a method of increasing reliability, speed or both by using multiple disks in parallel thereby decreasing access time and increasing transfer speed. A checksum or mirroring system can be used to increase reliability. Large servers can take advantage of such a setup but it might be overkill for a single user system unless you already have a large number of disks available. See other documents and FAQs for more information.

For Linux one can set up a RAID system using either software (the `md` module in the kernel), a Linux compatible controller card (PCI-to-SCSI) or a SCSI-to-SCSI controller. Check the documentation for what controllers can be used. A hardware solution is usually faster, and perhaps also safer, but comes at a significant cost.

SCSI-to-SCSI

SCSI-to-SCSI controllers are usually implemented as complete cabinets with drives and a controller that connects to the computer with a second SCSI bus. This makes the entire cabinet of drives look like a single large, fast SCSI drive and requires no special RAID driver. The disadvantage is that the SCSI bus connecting the cabinet to the computer becomes a bottleneck.

A significant disadvantage for people with large disk farms is that there is a limit to how many SCSI entries there can be in the `/dev` directory. In these cases using SCSI-to-SCSI will conserve entries.

Usually they are configured via the front panel or with a terminal connected to their on-board serial interface.

Some manufacturers of such systems are CMD and Syred whose web pages describe several systems.

PCI-to-SCSI

PCI-to-SCSI controllers are, as the name suggests, connected to the high speed PCI bus and is therefore not suffering from the same bottleneck as the SCSI-to-SCSI controllers. These controllers require special drivers but you also get the means of controlling the RAID configuration over the network which simplifies management.

Currently only a few families of PCI-to-SCSI host adapters are supported under Linux.

DPT

The oldest and most mature is a range of controllers from DPT including SmartCache I/III/IV and SmartRAID I/III/IV controller families. These controllers are supported by the EATA-DMA driver in the standard kernel. This company also has an informative home page which also describes various general aspects of RAID and SCSI in addition to the product related information.

More information from the author of the DPT controller drivers (EATA* drivers) can be found at his pages on SCSI and DPT.

These are not the fastest but have a good track record of proven reliability.

Note that the maintenance tools for DPT controllers currently run under DOS/Win only so you will need a small DOS/Win partition for some of the software. This also means you have to boot the system into Windows in order to maintain your RAID system.

ICP-Vortex

A very recent addition is a range of controllers from ICP-Vortex featuring up to 5 independent channels and very fast hardware based on the i960 chip. The linux driver was written by the company itself which shows they support Linux.

As ICP-Vortex supplies the maintenance software for Linux it is not necessary with a reboot to other operating systems for the setup and maintenance of your RAID system. This saves you also extra downtime.

Mylex DAC-960

This is one of the latest entries which is out in early beta. More information as well as drivers are available at Dandelion Digital's Linux DAC960 Page.

Compaq Smart-2 PCI Disk Array Controllers

Another very recent entry and currently in beta release is the Smart-2 driver.

Software RAID

A number of operating systems offer software RAID using ordinary disks and controllers. Cost is low and performance for raw disk IO can be very high. As this can be very CPU intensive it increases the load noticeably so if the machine is CPU bound in performance rather then IO bound you might be better off with a hardware PCI-to-RAID controller.

Real cost, performance and especially reliability of software vs. hardware RAID is a very controversial topic. Reliability on Linux systems have been very good so far.

The current software RAID project on Linux is the `md` system (multiple devices) which offers much more than RAID so it is described in more details later.

RAID Levels

RAID comes in many levels and flavours which I will give a brief overview of this here. Much has been written about it and the interested reader is recommended to read more about this in the RAID FAQ.

- RAID *0* is not redundant at all but offers the best throughput of all levels here. Data is striped across a number of drives so read and write operations take place in parallel across all drives. On the other hand if a single drive fail then everything is lost. Did I mention backups?
- RAID *1* is the most primitive method of obtaining redundancy by duplicating data across all drives. Naturally this is massively wasteful but you get one substantial advantage which is fast access. The drive that access the data first wins. Transfers are not any faster than for a single drive, even though you might get some faster read transfers by using one track reading per drive. Also if you have only 2 drives this is the only method of achieving redundancy.
- RAID *2* and *4* are not so common and are not covered here.
- RAID *3* uses a number of disks (at least 2) to store data in a striped RAID 0 fashion. It also uses an additional redundancy disk to store the XOR sum of the data from the data disks. Should the redundancy disk fail, the system can continue to operate as if nothing happened. Should any single data disk fail the system can compute the data on this disk from the information on the redundancy disk and all remaining disks. Any double fault will bring the whole RAID set off-line. RAID 3 makes sense only with at least 2 data disks (3 disks including the redundancy disk). Theoretically there is no limit for the number of disks in the set, but the probability of a fault increases with the number of disks in the RAID set. Usually the upper limit is 5 to 7 disks in a single RAID set. Since RAID 3 stores all redundancy information on a dedicated disk and since this information has to be updated whenever a write to any data disk occurs, the overall write speed of a RAID 3 set is limited by the write speed of the redundancy disk. This,

too, is a limit for the number of disks in a RAID set. The overall read speed of a RAID 3 set with all data disks and running is that of a RAID 0 set with that number of data disks. If the set has to reconstruct data stored on a failed disk from redundant information, the performance will be severely limited: All disks in the set have to be read and XOR-ed to compute the missing information.

- RAID *5* is just like RAID 3, but the redundancy information is spread on all disks of the RAID set. This improves write performance, because load is distributed more evenly between all available disks.

There are also hybrids available based on RAID 0 or 1 and one other level. Many combinations are possible but I have only seen a few referred to. These are more complex than the above mentioned RAID levels.

RAID *0/1* combines striping with duplication which gives very high transfers combined with fast seeks as well as redundancy. The disadvantage is high disk consumption as well as the above mentioned complexity.

RAID *1/5* combines the speed and redundancy benefits of RAID5 with the fast seek of RAID1. Redundancy is improved compared to RAID 0/1 but disk consumption is still substantial. Implementing such a system would involve typically more than 6 drives, perhaps even several controllers or SCSI channels.

6.2 Volume Management

Volume management is a way of overcoming the constraints of fixed sized partitions and disks while still having a control of where various parts of file space resides. With such a system you can add new disks to your system and add space from this drive to parts of the file space where needed, as well as migrating data out from a disk developing faults to other drives before catastrophic failure occurs.

The system developed by Veritas has become the defacto standard for logical volume management.

Volume management is for the time being an area where Linux is lacking.

One is the virtual partition system project VPS that will reimplement many of the volume management functions found in IBM's AIX system. Unfortunately this project is currently on hold.

Another project is the Logical Volume Manager project that is similar to a project by HP.

6.3 Linux `md` Kernel Patch

The Linux Multi Disk (md) provides a number of block level features in various stages of development.

RAID 0 (striping) and concatenation are very solid and in production quality and also RAID 4 and 5 are quite mature.

It is also possible to stack some levels, for instance mirroring (RAID 1) two pairs of drives, each pair set up as striped disks (RAID 0), which offers the speed of RAID 0 combined with the reliability of RAID 1.

In addition to RAID this system offers (in alpha stage) block level volume management and soon also translucent file space. Since this is done on the block level it can be used in combination with any file system, even for `fat` using Wine.

Think very carefully what drives you combine so you can operate all drives in parallel, which gives you better performance and less wear. Read more about this in the documentation that comes with `md`.

Unfortunately the documentation is rather old an in parts misleading and only refers to `md` version 0.35 which uses old style setup. The new system is very different and will soon be released as version 1.0 but is currently undocumented. If you wish to try it out you should follow the `linux-raid` mailing list.

Documentation is improving and a Software RAID HOWTO is in progress.

Hint: if you cannot get it to work properly you have forgotten to set the `persistent-block` flag. Your best documentation is currently the source code.

TOC

6.4 Compression

Disk compression versus file compression is a hotly debated topic especially regarding the added danger of file corruption. Nevertheless there are several options available for the adventurous administrators. These take on many forms, from kernel modules and patches to extra libraries but note that most suffer various forms of limitations such as being read-only. As development takes place at neck breaking speed the specs have undoubtedly changed by the time you read this. As always: check the latest updates yourself. Here only a few references are given.

- DouBle features file compression with some limitations.
- Zlibc adds transparent on-the-fly decompression of files as they load.
- there are many modules available for reading compressed files or partitions that are native to various other operating systems though currently most of these are read-only.
- dmsdos (currently in version 0.9.2.0) offer many of the compression options available for DOS and Windows. It is not yet complete but work is ongoing and new features added regularly.
- `e2compr` is a package that extends `ext2fs` with compression capabilities. It is still under testing and will therefore mainly be of interest for kernel hackers but should soon gain stability for wider use. Check the e2compr homepage for more information. I have reports of speed and good stability which is why it is mentioned here.

6.5 ACL

Access Control List (ACL) offers finer control over file access on a user by user basis, rather than the traditional owner, group and others, as seen in directory listings (`drwxr-xr-x`). This is currently not available in Linux but is expected in kernel 2.3 as hooks are already in place in `ext2fs`.

6.6 `cachefs`

This uses part of a hard disk to cache slower media such as CD-ROM. It is available under SunOS but not yet for Linux.

6.7 Translucent or Inheriting File Systems

This is a copy-on-write system where writes go to a different system than the original source while making it look like an ordinary file space. Thus the file space inherits the original data and the translucent write back buffer can be private to each user.

There is a number of applications:

- updating a live file system on CD-ROM, making it flexible, fast while also conserving space,
- original skeleton files for each new user, saving space since the original data is kept in a single space and shared out,
- parallel project development prototyping where every user can seemingly modify the system globally while not affecting other users.

SunOS offers this feature and this is under development for Linux. There was an old project called the Inheriting File Systems (`ifs`) but this project has stopped. One current project is part of the `md` system and offers block level translucence so it can be applied to any file system.

Sun has an informative page on translucent file system.

6.8 Physical Track Positioning

TOC

This trick used to be very important when drives were slow and small, and some file systems used to take the varying characteristics into account when placing files. Although higher overall speed, on board drive and controller caches and intelligence has reduced the effect of this.

Nevertheless there is still a little to be gained even today. As we know, "*world dominance*" is soon within reach but to achieve this "*fast*" we need to employ all the tricks we can use .

To understand the strategy we need to recall this near ancient piece of knowledge and the properties of the various track locations. This is based on the fact that transfer speeds generally increase for tracks further away from the spindle, as well as the fact that it is faster to seek to or from the central tracks than to or from the inner or outer tracks.

Most drives use disks running at constant angular velocity but use (fairly) constant data density across all tracks. This means that you will get much higher transfer rates on the outer tracks than on the inner tracks; a characteristics which fits the requirements for large libraries well.

Newer disks use a logical geometry mapping which differs from the actual physical mapping which is transparently mapped by the drive itself. This makes the estimation of the "middle" tracks a little harder.

In most cases track 0 is at the outermost track and this is the general assumption most people use. Still, it should be kept in mind that there are no guarantees this is so.

Inner

tracks are usually slow in transfer, and lying at one end of the seeking position it is also slow to seek to.

This is more suitable to the low end directories such as DOS, root and print spools.

Middle

tracks are on average faster with respect to transfers than inner tracks and being in the middle also on average faster to seek to.

This characteristics is ideal for the most demanding parts such as `swap`, `/tmp` and `/var/tmp`.

Outer

tracks have on average even faster transfer characteristics but like the inner tracks are at the end of the seek so statistically it is equally slow to seek to as the inner tracks.

Large files such as libraries would benefit from a place here.

Hence seek time reduction can be achieved by positioning frequently accessed tracks in the middle so that the average seek distance and therefore the seek time is short. This can be done either by using `fdisk` or `cfdisk` to make a partition on the middle tracks or by first making a file (using `dd`) equal to half the size of the entire disk before creating the files that are frequently accessed, after which the dummy file can be deleted. Both cases assume starting from an empty disk.

The latter trick is suitable for news spools where the empty directory structure can be placed in the middle before putting in the data files. This also helps reducing fragmentation a little.

This little trick can be used both on ordinary drives as well as RAID systems. In the latter case the calculation for centring the tracks will be different, if possible. Consult the latest RAID manual.

The speed difference this makes depends on the drives, but a 50 percent improvement is a typical value.

Disk Speed Values

The same mechanical head disk assembly (HDA) is often available with a number of interfaces (IDE, SCSI etc) and the mechanical parameters are therefore often comparable. The mechanics is today often the limiting factor but development is

improving things steadily. There are two main parameters, usually quoted in milliseconds (ms):

- Head movement - the speed at which the read-write head is able to move from one track to the next, called access time. If you do the mathematics and doubly integrate the seek first across all possible starting tracks and then across all possible target tracks you will find that this is equivalent of a stroke across a third of all tracks.
- Rotational speed - which determines the time taken to get to the right sector, called latency.

After voice coils replaced stepper motors for the head movement the improvements seem to have levelled off and more energy is now spent (literally) at improving rotational speed. This has the secondary benefit of also improving transfer rates.

Some typical values:

	Drive type		
Access time (ms)	Fast	Typical	Old
Tract-to-track	<1	2	8
Average seek	10	15	30
End-to-end	10	30	70

This shows that the very high end drives offer only marginally better access times then the average drives but that the old drives based on stepper motors are significantly worse.

Rotational speed (RPM)	3600	4500	4800	5400	7200	10000
Latency (ms)	17	13	12.5	11.1	8.3	6.0

As latency is the average time taken to reach a given sector, the formula is quite simply

```
latency (ms) = 60000 / speed (RPM)
```

Clearly this too is an example of diminishing returns for the efforts put into development. However, what really takes off here is the power consumption, heat and noise.

6.9 Stacking

One of the advantages of a layered design of an operating system is that you have the flexibility to put the pieces together in a number of ways. For instance you can cache a CD-ROM with `cachefs` that is a volume striped over 2 drives. This in turn can be set up translucently with a volume that is NFS mounted from another machine. RAID can be stacked in several layers to offer very fast seek and transfer in such a way that it will work if even 3 drives fail. The choices are many, limited only by imagination and, probably more importantly, money.

6.10 Recommendations

There is a near infinite number of combinations available but my recommendation is to start off with a simple setup without any fancy add-ons. Get a feel for what is needed, where the maximum performance is required, if it is access time or transfer speed that is the bottle neck, and so on. Then phase in each component in turn. As you can stack quite freely you should be able to retrofit most components in as time goes by with relatively few difficulties.

RAID is usually a good idea but make sure you have a thorough grasp of the technology and a solid back up system.

7. Other Operating Systems

Many Linux users have several operating systems installed, often necessitated by hardware setup systems that run under other operating systems, typically DOS or some flavour of Windows. A small section on how best to deal with this is therefore included here.

7.1 DOS

Leaving aside the debate on weather or not DOS qualifies as an operating system one can in general say that it has little sophistication with respect to disk operations. The more important result of this is that there can be severe difficulties in running various versions of DOS on large drives, and you are therefore strongly recommended in reading the *Large Drives mini-HOWTO*. One effect is that you are often better off placing DOS on low track numbers.

Having been designed for small drives it has a rather unsophisticated file system (`fat`) which when used on large drives will allocate enormous block sizes. It is also prone to block fragmentation which will after a while cause excessive seeks and slow effective transfers.

One solution to this is to use a defragmentation program regularly but it is strongly recommended to back up data and verify the disk before defragmenting. All versions of DOS have `chkdsk` that can do some disk checking, newer versions also have `scandisk` which is somewhat better. There are many defragmentation programs available, some versions have one called `defrag`. Norton Utilities have a large suite of disk tools and there are many others available too.

As always there are snags, and this particular snake in our drive paradise is called *hidden files*. Some vendors started to use these for copy protection schemes and would not take kindly to being moved to a different place on the drive, even if it remained in the same place in the directory structure. The result of this was that newer defragmentation programs will not touch any hidden file, which in turn reduces the effect of defragmentation.

Being a single tasking, single threading and single most other things operating system there is very little gains in using multiple drives unless you use a drive controller with built in RAID support of some kind.

There are a few utilities called `join` and `subst` which can do some multiple drive configuration but there is very little gains for a lot of work. Some of these commands have been removed in newer versions.

In the end there is very little you can do, but not all hope is lost. Many programs need fast, temporary storage, and the better behaved ones will look for environment variables called `TMPDIR` or `TEMPDIR` which you can set to point to another drive. This is often best done in `autoexec.bat`.

```
SET TMPDIR=E:/TMP
```

```
SET TEMPDIR=E:/TEMP
```


Not only will this possibly gain you some speed but also it can reduce fragmentation.

There have been reports about difficulties in removing multiple primary partitions using the `fdisk` program that comes with DOS. Should this happen you can instead use a Linux rescue disk with Linux `fdisk` to repair the system.

Don't forget there are other alternatives to DOS, the most well known being DR-DOS from Caldera. This is a direct descendant from DR-DOS from Digital Research. It offers many features not found in the more common DOS, such as multi tasking and long filenames.

Another alternative which also is free is Free DOS which is a project under development. A number of free utilities are also available.

7.2 Windows

Most of the above points are valid for Windows too, with the exception of Windows95 which apparently has better disk handling, which will get better performance out of SCSI drives.

A useful thing is the introduction of long filenames, to read these from Linux you will need the `vfat` file system for mounting these partitions.

Disk fragmentation is still a problem. Some of this can be avoided by doing a defragmentation immediately before and immediately after installing large programs or systems. I use this scheme at work and have found it to work quite well. Purging unused files and emptying the waste basket first can improve defragmentation further.

Windows also use swap drives, redirecting this to another drive can give you some performance gains. There are several mini-HOWTOs telling you how best to share swap space between various operating systems.

The trick of setting `TEMPDIR` can still be used but not all programs will honour this setting. Some do, though. To get a good overview of the settings in the control files you can run `sysedit` which will open a number of files for editing, one of which is the `autoexec` file where you can add the `TEMPDIR` settings.

Much of the temporary files are located in the `/windows/temp` directory and changing this is more tricky. To achieve this you can use `regedit` which is rather powerful and quite capable of rendering your system in a state you will not enjoy, or more precisely, in a state much less enjoyable than windows in general. Registry database error is a message that means seriously bad news. Also you will see that many programs have their own private temporary directories scattered around the system.

Setting the swap file to a separate partition is a better idea and much less risky. Keep in mind that this partition cannot be used for anything else, even if there should appear to be space left there.

It is now possible to read `ext2fs` partitions from Windows, either by mounting the partition using FSDEXT2 or by using a file explorer like tool called Explore2fs.

TOC

7.3 OS/2

The only special note here is that you can get a file system driver for OS/2 that can read an `ext2fs` partition.

7.4 NT

This is a more serious system featuring most buzzwords known to marketing. It is well worth noting that it features software striping and other more sophisticated setups. Check out the drive manager in the control panel. I do not have easy access to NT, more details on this can take a bit of time.

One important snag was recently reported by acahalan at cs.uml.edu : (reformatted from a Usenet News posting)

NT DiskManager has a serious bug that can corrupt your disk when you have several (more than one?) extended partitions. Microsoft provides an emergency fix program at their web site. See the knowledge base for more. (This affects Linux users, because Linux users have extra partitions)

You can now read `ext2fs` partitions from NT using Explore2fs.

7.5 Sun OS

There is a little bit of confusion in this area between Sun OS vs. Solaris. Strictly speaking Solaris is just Sun OS 5.x packaged with Openwindows and a few other things. If you run Solaris, just type `uname -a` to see your version. Parts of the reason for this confusion is that Sun Microsystems used to use an OS from the BSD family, albeight with a few bits and pieces from elsewhere as well as things made by themselves. This was the situation up to Sun OS 4.x.y when they did a "strategic roadmap decision" and decided to switch over to the official Unix, System V, Release 4 (aka SVR5), and Sun OS 5 was created. This made a lot of people unhappy. Also this was bundled with other things and marketed under the name Solaris, which currently stands at release 7 which just recently replaced version 2.6 as the latest and greatest. In spite of the large jump in version number this is actually a minor technical upgrade but a giant leap for marketing.

Sun OS 4

This is quite familiar to most Linux users. The last release is 4.1.4 plus various patches. Note however that the file system structure is quite different and does not conform to FSSTND so any planning must be based on the traditional structure. You can get some information by the man page on this: `man hier`. This is, like most man pages, rather brief but should give you a good start. If you are still confused by the structure it will at least be at a higher level.

TOC

Sun OS 5 (aka Solaris)

This comes with a snazzy installation system that runs under Openwindows, it will help you in partitioning and formatting the drives before installing the system from CD-ROM. It will also fail if your drive setup is too far out, and as it takes a complete installation run from a full CD-ROM in a 1x only drive this failure will dawn on you after too long time. That is the experience we had where I used to work. Instead we installed everything onto one drive and then moved directories across.

The default settings are sensible for most things, yet there remains a little oddity: swap drives. Even though the official manual recommends multiple swap drives (which are used in a similar fashion as on Linux) the default is to use only a single drive. It is recommended to change this as soon as possible.

Sun OS 5 offers also a file system especially designed for temporary files, `tmpfs`. It offers significant speed improvements over `ufs` but does not survive rebooting.

The only comment so far is: beware! Under Solaris 2.0 it seem that creating too big files in `/tmp` can cause a out of swap space kernel panic trap. As the evidence of what has happened is as lost as any data on a RAMdisk after powering down it can be hard to find out what has happened. What is worse, it seems that user space processes can cause this kernel panic and unless this problem is taken care of it is best not to use `tmpfs` in potentially hostile environments.

Also see the notes on tmpfs.

Trivia: There is a movie also called Solaris, a science fiction movie that is very, very long, slow and incomprehensible. This was often pointed out at the time Solaris (the OS) appeared...

BeOS

This operating system is one of the more recent one to arrive and it features a file system that has some database like features.

There is a BFS file system driver being developed for Linux and is available in alpha stage. For more information check the Linux BFS page where patches also are available.

---

8. Clusters

In this section I will briefly touch on the ways machines can be connected together but this is so big a topic it could be a separate HOWTO in its own right, hint, hint. Also, strictly speaking, this section lies outside the scope of this HOWTO, so if you feel like getting fame etc. *you* could contact me and take over this part and turn it into a new document.

These days computers gets outdated at an incredible rate. There is however no reason why old hardware could not be put to good use with Linux. Using an old and otherwise outdated computer as a network server can be both useful in its own right as well as a valuable educational exercise. Such a local networked cluster of computers can take on many forms but to remain within the charter of this HOWTO I will limit myself to the disk strategies. Nevertheless I would hope someone else could take on this topic and turn it into a document on its own.

This is an exciting area of activity today, and many forms of clustering is available today, ranging from automatic workload balancing over local network to more exotic hardware such as Scalable Coherent Interface (SCI) which gives a tight integration of machines, effectively turning them into a single machine. Various kinds of clustering has been available for larger machines for some time and the VAXcluster is perhaps a well known example of this. Clustering is done usually in order to share resources such as disk drives, printers and terminals etc, but also processing resources equally transparently between the computational nodes.

There is no universal definition of clustering, in here it is taken to mean a network of machines that combine their resources to serve users. Admittedly this is a rather loose definition but this will change later.

These days also Linux offers some clustering features but for a starter I will just describe a simple local network. It is a good way of putting old and otherwise unusable hardware to good use, as long as they can run Linux or something similar.

One of the best ways of using an old machine is as a network server in which case the effective speed is more likely to be limited by network bandwidth rather than pure computational performance. For home use you can move the following functionality off to an older machine used as a server:

- news
- mail
- web proxy
- printer server
- modem server (PPP, SLIP, FAX, Voice mail)

You can also `NFS mount` drives from the server onto your workstation thereby reducing drive space requirements. Still read the FSSTND to see what directories should *not* be exported. The best candidates for exporting to all machines are `/usr` and `/var/spool` and possibly `/usr/local` but probably not `/var/spool/lpd`.

Most of the time even slow disks will deliver sufficient performance. On the other hand, if you do processing directly on the disks on the server or have very fast networking, you might want to rethink your strategy and use faster drives. Searching features on a web server or news database searches are two examples of this.

Such a network can be an excellent way of learning system administration and building up your own toaster network, as it often is called. You can get more information on this in other HOWTOs but there are two important things you should keep in mind:

- Do not pull IP numbers out of thin air. Configure your inside net using IP numbers reserved for private use, and use your network server as a router that handles this IP masquerading.
- Remember that if you additionally configure the router as a firewall you might not be able to get to your own data from the outside, depending on the firewall configuration.

The *nyx* network provides an example of a cluster in the sense defined here. It consists of the following machines:

nyx

is one of the two user login machines and also provides some of the networking services.

nox

(aka nyx10) is the main user login machine and is also the mail server.

noc

is a dedicated news server. The news spool is made accessible through NFS mounting to nyx and nox.

arachne

(aka www) is the web server. Web pages are written by NFS mounting onto nox.

There are also some more advanced clustering projects going, notably

- The Beowulf Project
- The Genoa Active Message Machine (GAMMA)

High-tech clustering requires high-tech interconnect, and SCI is one of them. To find out more you can either look up the home page of Dolphin Interconnect Solutions which is one of the main actors in this field, or you can have a look at scizzl.

Centralised mail servers using IMAP are becoming more and more popular as disks become large enough to keep all mail stored indefinitely and also cheap enough to make it a feasible option. Unfortunately it has become clear that `NFS` mounting the mail archives from another machine can cause corruption of the IMAP database as the server software does not handle NFS timeouts too well, and NFS timeouts are a rather common occurrence. Keep therefore the mail archive local to the IMAP server.

---

9. Mount Points

In designing the disk layout it is important not to split off the directory tree structure at the wrong points, hence this section. As it is highly dependent on the FSSTND it has been put aside in a separate section, and will most likely have to be totally rewritten when FHS is adopted in a Linux distribution. In the meanwhile this will do.

Remember that this is a list of where a separation *can* take place, not where it *has* to be. As always, good judgement is always required.

Again only a rough indication can be given here. The values indicate

```
0=don't separate here
1=not recommended
 ...
4=useful
5=recommended
```

In order to keep the list short, the uninteresting parts are removed.

```
Directory   Suitability
/
|
+-bin       0
+-boot      0
+-dev       0
+-etc       0
+-home      5
+-lib       0
+-mnt       0
+-proc      0
+-root      0
+-sbin      0
+-tmp       5
+-usr       5
| \
| +-X11R6     3
| +-bin       3
| +-lib       4
| +-local     4
| | \
```

```
    | | +bin        2
    | | +lib        4
    | +-src       3
    |
    +-var       5
      \
      +-adm       0
      +-lib       2
      +-lock      1
      +-log       0
      +-preserve  1
      +-run       1
      +-spool     4
      | \
      | +-mail      3
      | +-mqueue    3
      | +-news      5
      | +-smail     3
      | +-uucp      3
      +-tmp       5
```

There is of course plenty of adjustments possible, for instance a home user would not bother with splitting off the `/var/spool` hierarchy but a serious ISP should. The key here is *usage*.

QUIZ! Why should `/etc` never be on a separate partition? Answer: Mounting instructions during boot is found in the file `/etc/fstab` so if this is on a separate and unmounted partition it is like the key to a locked drawer is inside that drawer, a hopeless situation. (Yes, I'll do nearly anything to liven up this HOWTO.)

---

10. Considerations and Dimensioning

The starting point in this will be to consider where you are and what you want to do. The typical home system starts out with existing hardware and the newly converted Linux user will want to get the most out of existing hardware. Someone setting up a new system for a specific purpose (such as an Internet provider) will instead have to consider what the goal is and buy accordingly. Being ambitious I will try to cover the entire range.

Various purposes will also have different requirements regarding file system placement on the drives, a large multiuser machine would probably be best off with the `/home` directory on a separate disk, just to give an example.

In general, for performance it is advantageous to split most things over as many disks as possible but there is a limited number of devices that can live on a SCSI bus and cost is naturally also a factor. Equally important, file system maintenance becomes more complicated as the number of partitions and physical drives increases.

10.1 Home Systems

With the cheap hardware available today it is possible to have quite a big system at home that is still cheap, systems that rival major servers of yesteryear. While many started out with old, discarded disks to build a Linux server (which is how this HOWTO came into existence), many can now afford to buy 20 GB disks up front.

Size remains important for some, and here are a few guidelines:

Testing

Linux is simple and you don't even need a hard disk to try it out, if you can get the boot floppies to work you are likely to get it to work on your hardware. If the standard kernel does not work for you, do not forget that often there can be special boot disk versions available for unusual hardware combinations that can solve your initial problems until you can compile your own kernel.

Learning

about operating system is something Linux excels in, there is plenty of documentation and the source is available. A single drive with 50 MB is enough to get you started with a shell, a few of the most frequently used commands and utilities.

Hobby

use or more serious learning requires more commands and utilities but a single drive is still all it takes, 500 MB should give you plenty of room, also for sources and documentation.

Serious

software development or just serious hobby work requires even more space. At this stage you have

probably a mail and news feed that requires spool files and plenty of space. Separate drives for various tasks will begin to show a benefit. At this stage you have probably already gotten hold of few drives too. Drive requirements gets harder to estimate but I would expect 2-4 GB to be plenty, even for a small server.

Servers

come in many flavours, ranging from mail servers to full sized ISP servers. A base of 2 GB for the main system should be sufficient, then add space and perhaps also drives for separate features you will offer. Cost is the main limiting factor here but be prepared to spend a bit if you wish to justify the "S" in ISP. Admittedly, not all do it.

Basically a server is dimensioned like any machine for serious use with added space for the services offered, and tends to be IO bound rather than CPU bound.

With cheap networking technology both for land lines as well as through radio nets, it is quite likely that very soon home users will have their own servers more or less permanently hooked onto the net.

10.2 Servers

Big tasks require big drives and a separate section here. If possible keep as much as possible on separate drives. Some of the appendices detail the setup of a small departmental server for 10-100 users. Here I will present a few consideration for the higher end servers. In general you should not be afraid of using RAID, not only because it is fast and safe but also because it can make growth a little less painful. All the notes below come as additions to the points mentioned earlier.

Popular servers rarely just happens, rather they grow over time and this demands both generous amounts of disk space as well as a good net connection. In many of these cases it might be a good idea to reserve entire SCSI drives, in singles or as arrays, for each task. This way you can move the data should the computer fail. Note that transferring drives across computers is not simple and might not always work, especially in the case of IDE drives. Drive arrays require careful setup in order to reconstruct the data correctly, so you might want to keep a paper copy of your `fstab` file as well as a note of SCSI IDs.

Home Directories

Estimate how many drives you will need, if this is more than 2 I would recommend RAID, strongly. If not you should separate users across your drives dedicated to users based on some kind of simple hashing algorithm. For instance you could use the first 2 letters in the user name, so `jbloggs` is put on `/u/j/b/jbloggs` where `/u/j` is a symbolic link to a physical drive so you can get a balanced load on your drives.

Anonymous FTP

This is an essential service if you are serious about service. Good servers are well maintained, documented, kept up to date, and immensely popular no matter where in the world they are located. The big server ftp.funet.fi is an excellent example of this.

TOC

In general this is not a question of CPU but of network bandwidth. Size is hard to estimate, mainly it's a question of ambition and service attitudes. I believe the big archive at ftp.cdrom.com is a *BSD machine with 50 GB disk. Also memory is important for a dedicated FTP server, about 256 MB RAM would be sufficient for a very big server, whereas smaller servers can get the job done well with 64 MB RAM. Network connections would still be the most important factor.

WWW

For many this is the main reason to get onto the Internet, in fact many now seem to equate the two. In addition to being network intensive there is also a fair bit of drive activity related to this, mainly regarding the caches. Keeping the cache on a separate, fast drive would be beneficial. Even better would be installing a caching proxy server. This way you can reduce the cache size for each user and speed up the service while at the same time cut down on the bandwidth requirements.

With a caching proxy server you need a fast set of drives, RAID0 would be ideal as reliability is not important here. Higher capacity is better but about 2 GB should be sufficient for most. Remember to match the cache period to the capacity and demand. Too long periods would on the other hand be a disadvantage, if possible try to adjust based on the URL. For more information check up on the most used servers such as `Harvest`, Squid and the one from Netscape.

Mail

Handling mail is something most machines do to some extent. The big mail servers, however, come into a class of their own. This is a demanding task and a big server can be slow even when connected to fast drives and a good net feed. In the Linux world the big server at `vger.rutgers.edu` is a well known example. Unlike a news service which is distributed and which can partially reconstruct the spool using other machines as a feed, the mail servers are centralised. This makes safety much more important, so for a major server you should consider a RAID solution with emphasize on reliability. Size is hard to estimate, it all depends on how many lists you run as well as how many subscribers you have.

Note that these days more and more switch over from using `POP` to pull mail to local machine from mail server and instead use `IMAP` to serve mail while keeping the mail archive centralised. This means that mail is no longer spooled in its original sense but often builds up, requiring huge disk space. Also more and more (ab)use mail attachments to send all sorts of things across, even a small word processor document can easily end up over 1 MB. Size your disks generously and keep an eye on how much space is left.

News

This is definitely a high volume task, and very dependent on what news groups you subscribe to. On Nyx there is a fairly complete feed and the spool files consume about 17 GB. The biggest groups are no doubt in the `alt.binary.*` hierarchy, so if you for some reason decide not to get these you can get a good service with perhaps 12 GB. Still others, that shall remain nameless, feel 2 GB is sufficient to claim ISP status. In this case news expires so fast I feel the spelling IsP is barely justified. A full newsfeed means a traffic of a few GB every day and this is an ever growing number.

TOC

Others

There are many services available on the net and even though many have been put somewhat in the shadows by the web. Nevertheless, services like *archie*, *gopher* and *wais* just to name a few, still exist and remain valuable tools on the net. If you are serious about starting a major server you should also consider these services. Determining the required volumes is hard, it all depends on popularity and demand. Providing good service inevitably has its costs, disk space is just one of them.

Server Recommendations

Servers today require large numbers of large disks to function satisfactorily in commercial settings. As mean time between failure (MTBF) decreases rapidly as the number of components increase it is advisable to look into using RAID for protection and use a number of medium sized drives rather than one single huge disk. Also look into the High Availability (HA) project for more information.

10.3 Pitfalls

The dangers of splitting up everything into separate partitions are briefly mentioned in the section about volume management. Still, several people have asked me to emphasize this point more strongly: when one partition fills up it cannot grow any further, no matter if there is plenty of space in other partitions.

In particular look out for explosive growth in the news spool (`/var/spool/news`). For multi user machines with quotas keep an eye on `/tmp` and `/var/tmp` as some people try to hide their files there, just look out for filenames ending in gif or jpeg...

In fact, for single physical drives this scheme offers very little gains at all, other than making file growth monitoring easier (using '`df`') and physical track positioning. Most importantly there is no scope for parallel disk access. A freely available volume management system would solve this but this is still some time in the future. However, when more specialised file systems become available even a single disk could benefit from being divided into several partitions.

---

11. Disk Layout

With all this in mind we are now ready to embark on the layout. I have based this on my own method developed when I got hold of 3 old SCSI disks and boggled over the possibilities.

The tables in the appendices are designed to simplify the mapping process. They have been designed to help you go through the process of optimizations as well as making an useful log in case of system repair. A few examples are also given.

11.1 Selection for Partitioning

Determine your needs and set up a list of all the parts of the file system you want to be on separate partitions and sort them in descending order of speed requirement and how much space you want to give each partition.

The table in Appendix A section is a useful tool to select what directories you should put on different partitions. It is sorted in a logical order with space for your own additions and notes about mounting points and additional systems. It is therefore NOT sorted in order of speed, instead the speed requirements are indicated by bullets ('o').

If you plan to RAID make a note of the disks you want to use and what partitions you want to RAID. Remember various RAID solutions offers different speeds and degrees of reliability.

(Just to make it simple I'll assume we have a set of identical SCSI disks and no RAID)

11.2 Mapping Partitions to Drives

Then we want to place the partitions onto physical disks. The point of the following algorithm is to maximise parallelizing and bus capacity. In this example the drives are A, B and C and the partitions are 987654321 where 9 is the partition with the highest speed requirement. Starting at one drive we 'meander' the partition line over and over the drives in this way:

```
A : 9 4 3
B : 8 5 2
C : 7 6 1
```

This makes the 'sum of speed requirements' the most equal across each drive.

Use the table in Appendix B section to select what drives to use for each partition in order to optimize for paralellicity.

Note the speed characteristics of your drives and note each directory under the appropriate column. Be prepared to shuffle directories, partitions and drives around a few times before you are satisfied.

TOC

11.3 Sorting Partitions on Drives

After that it is recommended to select partition numbering for each drive.

Use the table in Appendix C section to select partition numbers in order to optimize for track characteristics. At the end of this you should have a table sorted in ascending partition number. Fill these numbers back into the tables in appendix A and B.

You will find these tables useful when running the partitioning program (`fdisk` or `cfdisk`) and when doing the installation.

11.4 Optimizing

After this there are usually a few partitions that have to be 'shuffled' over the drives either to make them fit or if there are special considerations regarding speed, reliability, special file systems etc. Nevertheless this gives what this author believes is a good starting point for the complete setup of the drives and the partitions. In the end it is actual use that will determine the real needs after we have made so many assumptions. After commencing operations one should assume a time comes when a repartitioning will be beneficial.

For instance if one of the 3 drives in the above mentioned example is very slow compared to the two others a better plan would be as follows:

```
A : 9 6 5
B : 8 7 4
C : 3 2 1
```

Optimizing by Characteristics

Often drives can be similar in apparent overall speed but some advantage can be gained by matching drives to the file size distribution and frequency of access. Thus binaries are suited to drives with fast access that offer command queueing, and libraries are better suited to drives with larger transfer speeds where IDE offers good performance for the money.

Optimizing by Drive Parallelising

Avoid drive contention by looking at tasks: for instance if you are accessing `/usr/local/bin` chances are you will soon also need files from `/usr/local/lib` so placing these at separate drives allows less seeking and possible parallel operation and drive caching. It is quite possible that choosing what may appear less than ideal drive characteristics will still be advantageous if you can gain parallel operations. Identify common tasks, what partitions they use and try to keep these on separate physical drives.

Just to illustrate my point I will give a few examples of task analysis here.

TOC

Office software

such as editing, word processing and spreadsheets are typical examples of low intensity software both in terms of CPU and disk intensity. However, should you have a single server for a huge number of users you should not forget that most such software have auto save facilities which cause extra traffic, usually on the home directories. Splitting users over several drives would reduce contention.

News

readers also feature auto save features on home directories so ISPs should consider separating home directories

News spools are notorious for their deeply nested directories and their large number of very small files. Loss of a news spool partition is not a big problem for most people, too, so they are good candidates for a RAID 0 setup with many small disks to distribute the many seeks among multiple spindles. It is recommended in the manuals and FAQs for the INN news server to put news spool and `.overview` files on separate drives for larger installations.

There is also a web page dedicated to INN optimising well worth reading.

Database

applications can be demanding both in terms of drive usage and speed requirements. The details are naturally application specific, read the documentation carefully with disk requirements in mind. Also consider RAID both for performance and reliability.

E-mail

reading and sending involves home directories as well as in- and outgoing spool files. If possible keep home directories and spool files on separate drives. If you are a mail server or a mail hub consider putting in- and outgoing spool directories on separate drives.

Losing mail is an extremely bad thing, if you are managing an ISP or major hub. Think about RAIDing your mail spool and consider frequent backups.

Software development

can require a large number of directories for binaries, libraries, include files as well as source and project files. If possible split as much as possible across separate drives. On small systems you can place `/usr/src` and project files on the same drive as the home directories.

Web browsing

is becoming more and more popular. Many browsers have a local cache which can expand to rather large volumes. As this is used when reloading pages or returning to the previous page, speed is quite important here. If however you are connected via a well configured proxy server you do not need more than typically a few megabytes per user for a session. See also the sections on Home Directories and WWW.

11.5 Compromises

TOC

One way to avoid the aforementioned pitfalls is to only set off fixed partitions to directories with a fairly well known size such as swap, `/tmp` and `/var/tmp` and group together the remainders into the remaining partitions using symbolic links.

Example: a slow disk (`slowdisk`), a fast disk (`fastdisk`) and an assortment of files. Having set up `swap` and `tmp` on `fastdisk`; and `/home` and root on slowdisk we have (the fictitious) directories `/a/slow`, `/a/fast`, `/b/slow` and `/b/fast` left to allocate on the partitions `/mnt.slowdisk` and `/mnt.fastdisk` which represents the remaining partitions of the two drives.

Putting `/a` or `/b` directly on either drive gives the same properties to the subdirectories. We could make all 4 directories separate partitions but would lose some flexibility in managing the size of each directory. A better solution is to make these 4 directories symbolic links to appropriate directories on the respective drives.

Thus we make

```
/a/fast point to /mnt.fastdisk/a/fast   or   /mnt.fastdisk/a.fast
/a/slow point to /mnt.slowdisk/a/slow   or   /mnt.slowdisk/a.slow
/b/fast point to /mnt.fastdisk/b/fast   or   /mnt.fastdisk/b.fast
/b/slow point to /mnt.slowdisk/b/slow   or   /mnt.slowdisk/b.slow
```

and we get all fast directories on the fast drive without having to set up a partition for all 4 directories. The second (right hand) alternative gives us a flatter files system which in this case can make it simpler to keep an overview of the structure.

The disadvantage is that it is a complicated scheme to set up and plan in the first place and that all mount points and partitions have to be defined before the system installation.

---

12. Implementation

Having done the layout you should now have a detailled description on what goes where. Most likely this will be on paper but hopefully someone will make a more automated system that can deal with everything from the design, through partitioning to formatting and installation. This is the route one will have to take to realise the design.

Modern distributions come with installation tools that will guide you through partitioning and formatting and also set up `/etc/fstab` for you automatically. For later modifications, however, you will need to understand the underlying mechanisms.

12.1 Drives and Partitions

When you start DOS or the like you will find all partitions labeled `C:` and onwards, with no differentiation on IDE, SCSI, network or whatever type of media you have. In the world of Linux this is rather different. During booting you will see partitions described like this:

```
Dec  6 23:45:18 demos kernel: Partition check:
Dec  6 23:45:18 demos kernel:  sda: sda1
Dec  6 23:45:18 demos kernel:  hda: hda1 hda2
```

SCSI drives are labelled `sda`, `sdb`, `sdc` etc, and (E)IDE drives are labelled `hda`, `hdb`, `hdc` etc. There are also standard names for all devices, full information can be found in `/dev/MAKEDEV` and `/usr/src/linux/Documentation/devices.txt`.

Partitions are labelled numerically for each drive `hda1`, `hda2` and so on. On SCSI drives there can be 15 partitions per drive, on EIDE drives there can be 63 partitions per drive. Both limits exceed what is currently useful for most disks.

These are then mounted according to the file `/etc/fstab` before they appear as a part of the file system.

12.2 Partitioning

First you have to partition each drive into a number of separate partitions. Under Linux there are two main methods, `fdisk` and the more screen oriented `cfdisk`. These are complex programs, read the manual *very* carefully.

Partitions come in 3 flavours, `primary`, `extended` and `logical`. You have to use `primary` partitions for booting, but there is a maximum of 4 primary partitions. If you want more you have to define an `extended` partition within which you define your `logical` partitions.

Each partition has an identifier number which tells the operating system what it is, for Linux the types

TOC

`swap(82)` and `ext2fs(83)` are the ones you will need to know.

There is a readme file that comes with `fdisk` that gives more in-depth information on partitioning.

Someone has just made a *Partitioning HOWTO* which contains excellent, in depth information on the nitty-gritty of partitioning. Rather than repeating it here and bloating this document further, I will instead refer you to it instead.

Redhat has written a screen oriented utility called *Disk Druid* which is supposed to be a user friendly alternative to `fdisk` and `cfdisk` and also automates a few other things. Unfortunately this product is not quite mature so if you use it and cannot get it to work you are well advised to try `fdisk` or `cfdisk`.

The Ranish Partition Manager is another free alternative, while Partition Magic is a popular commercial alternative which also offers some support for resizing `ext2fs` partitions.

Note that Windows will complain if it finds more than one primary partition on a drive. Also it appears to assign drive letters to primary partitions as it finds disks before starting over from the first disk to assign subsequent drive names to logical partitions.

If you want DOS/Windows on your system you should make that partition first, a primary one to boot to, made with the DOS `fdisk` program. Then if you want NT you put that one in. Finally, for Linux, you create those partitions with the Linux `fdisk` program or equivalents. Linux is flexible enough to boot from both primary as well as logical partitions.

12.3 Repartitioning

Sometimes it is necessary to change the sizes of existing partitions while keeping the contents intact. One way is of course to back up everything, recreate new partitions and then restore the old contents, and while this gives your back up system a good test it is also rather time consuming.

Partition resizing is a simpler alternative where a file system is first shrunk to desired volume and then the partition table is updated to reflect the new end of partition position. This process is therefore very file system sensitive.

Repartitioning requires there to be free space at the end of the file space so to ensure you are able to shrink the size you should first defragment your drive and empty any wastebaskets.

Using fips you can resize a `fat` partition, and the latest version 1.6 of `fips` or `fips 2.0` are also able to resize `fat32` partition. Note that these programs actually run under DOS.

Resizing other file systems are much more complicated but one popular commercial system Partition Magic is able to resize more file system types, including `ext2fs` using the `resize2fs` program.

In order to get the most out of `fips` you should first delete unnecessary files, empty wastebaskets etc. before defragmenting your drive. This way you can allocate more space to other partitions. Id the program complains there are still files at the end of your drive it is probably hidden files generated by Microsoft Mirror or Norton Image. These are probably called `image.idx` and `image.dat` and contain backups of some system files.

There are reports that in some Windows defragmentation programs you should make sure the box "allow

TOC

Windows to move files around" is *not* checked, otherwise you will end up with some files in the last cylinder of the partition which will prevent FIPS from reclaiming space.

If you still have unmovable files at the end of your DOS partition you should get the DOS program "showfat" version 3.0 or higher. This shows you what files are where so you can deal with them directly.

Repartitioning is as dangerous process as any other partitioning so you are advised to have a fresh backup handy.

12.4 Microsoft Partition Bug

In Microsoft products all the way up to Win 98 there is a tricky bug that can cause you a bit of trouble: if you have several primary `fat` partitions and the last extended partition is not a `fat` partition the Microsoft system will try to mount the last partition as if it were a FAT partition in place of the last primary FAT partition.

There is more information available on the net on this.

To avoid this you can place a small logical `fat` partition at the very end of your disk.

Since some hardware comes with setup software that is available under DOS only this could come in handy anyway. Notable examples are RAID controllers from DPT and a number of networking cards.

12.5 Multiple Devices (`md`)

Being in a state of flux you should make sure to read the latest documentation on this kernel feature. It is not yet stable, beware.

Briefly explained it works by adding partitions together into new devices `md0`, `md1` etc. using `mdadd` before you activate them using `mdrun`. This process can be automated using the file `/etc/mdtab`.

The latest `md` system uses a /etc/raidtab and a different syntax. Make sure your RAID-tools package matches the `md` version as the internal protocol has changed.

Then you then treat these like any other partition on a drive. Proceed with formatting etc. as described below using these new devices.

There is now also a HOWTO in development for RAID using `md` you should read.

12.6 Formatting

Next comes partition formatting, putting down the data structures that will describe the files and where they are located. If this is the first time it is recommended you use formatting with verify. Strictly speaking it should not be necessary but this exercises the I/O hard enough that it can uncover potential problems, such as incorrect termination, before you store your precious data. Look up the command `mkfs` for more details.

Linux can support a great number of file systems, rather than repeating the details you can read the man page for `fs` which describes them in some details. Note that your kernel has to have the drivers compiled in

TOC

or made as modules in order to be able to use these features. When the time comes for kernel compiling you should read carefully through the file system feature list. If you use `make menuconfig` you can get online help for each file system type.

Note that some rescue disk systems require `minix`, `msdos` and `ext2fs` to be compiled into the kernel.

Also swap partitions have to be prepared, and for this you use `mkswap`.

12.7 Mounting

Data on a partition is not available to the file system until it is mounted on a mount point. This can be done manually using `mount` or automatically during booting by adding appropriate lines to `/etc/fstab`. Read the manual for `mount` and pay close attention to the tabulation.

12.8 `fstab`

During the booting process the system mounts all partitions as described in the `fstab` file which can look something like this:

```
# <file system>   <mount point>   <type>  <options>   <dump>  <pass>
/dev/hda2          /               ext2    defaults    0       1
None               none            swap    sw          0       0
proc               /proc           proc    defaults    0       0
/dev/hda1          /dosc           vfat    defaults    0       1
```

This file is somewhat sensitive to the formatting used so it is best and also most convenient to edit it using one of the editing tools made for this purpose.

Briefly, the fields are partition name, where to mount the partition, type of file system, when to mount, when to dump for backup and when to do `fsck`.

Linux offers the possibility of parallel file checking (`fsck`) but to be efficient it is important not to `fsck` more than one partition on a drive at a time.

For more information refer to the man page for `mount` and `fstab`.

12.9 Recommendations

Having constructed and implemented your clever scheme you are well advised to make a complete record of it all, on paper. After all having all the necessary information on disk is no use if the machine is down.

Partition tables can be damaged or lost, in which case it is excruciatingly important that you enter the exact same numbers into `fdisk` so you can rescue your system. You can use the program `printpar` to make a clear record of the tables. Also write down the SCSI numbers or IDE names for each disk so you can put the system together again in the right order.

---

13. Maintenance

It is the duty of the system manager to keep an eye on the drives and partitions. Should any of the partitions overflow, the system is likely to stop working properly, no matter how much space is available on other partitions, until space is reclaimed.

Partitions and disks are easily monitored using `df` and should be done frequently, perhaps using a cron job or some other general system management tool.

Do not forget the swap partitions, these are best monitored using one of the memory statistics programs such as `free`, `procinfo` or `top`.

Drive usage monitoring is more difficult but it is important for the sake of performance to avoid contention - placing too much demand on a single drive if others are available and idle.

It is important when installing software packages to have a clear idea where the various files go. As previously mentioned GCC keeps binaries in a library directory and there are also other programs that for historical reasons are hard to figure out, X11 for instance has an unusually complex structure.

When your system is about to fill up it is about time to check and prune old logging messages as well as hunt down core files. Proper use of `ulimit` in global shell settings can help saving you from having core files littered around the system.

13.1 Backup

The observant reader might have noticed a few hints about the usefulness of making backups. Horror stories are legio about accidents and what happened to the person responsible when the backup turned out to be non-functional or even non existent. You might find it simpler to invest in proper backups than a second, secret identity.

There are many options and also a mini-HOWTO (`Backup-With-MSDOS`) detailling what you need to know. In addition to the DOS specifics it also contains general information and further leads.

In addition to making these backups you should also make sure you can restore the data. Not all systems verify that the data written is correct and many administrators have started restoring the system after an accident happy in the belief that everything is working, only to discover to their horror that the backups were useless. Be careful.

13.2 Defragmentation

This is very dependent on the file system design, some suffer fast and nearly debilitating fragmentation. Fortunately for us, `ext2fs` does not belong to this group and therefore there has been very little talk about defragmentation tools. It does in fact exist but is hardly ever needed.

If for some reason you feel this is necessary, the quick and easy solution is to do a backup and a restore. If only a small area is affected, for instance the home directories, you could `tar` it over to a temporary area on another partition, *verify* the archive, delete the original and then untar it back again.

13.3 Deletions

Quite often disk space shortages can be remedied simply by deleting unnecessary files that accumulate around the system. Quite often programs that terminate abnormally cause all kinds of mess lying around the oddest places. Normally a core dump results after such an incident and unless you are going to debug it you can simply delete it. These can be found everywhere so you are advised to do a global search for them now and then.

Unexpected termination can also cause all sorts of temporary files remaining in places like `/tmp` or `/var/tmp`, files that are automatically removed when the program ends normally. Rebooting cleans up some of these areas but not necessary all and if you have a long uptime you could end up with a lot of old junk. If space is short you have to delete with care, make sure the file is not in active use first. Utilities like `file` can often tell you what kind of file you are looking at.

Many things are logged when the system is running, mostly to files in the `/var/log` area. In particular the file `/var/log/messages` tends to grow until deleted. It is a good idea to keep a small archive of old log files around for comparison should the system start to behave oddly.

If the mail or news system is not working properly you could have excessive growth in their spool areas, `/var/spool/mail` and `/var/spool/news` respectively. Beware of the overview files as these have a leading dot which makes them invisible to `ls -l`, it is always better to use `ls -Al` which will reveal them.

User space overflow is a particularly tricky topic. Wars have been waged between system administrators and users. Tact, diplomacy and a generous budget for new drives is what is needed. Make use of the message-of-the-day feature, information displayed during login from the `/etc/motd` file to tell users when space is short. Setting the default shell settings to prevent core files being dumped can save you a lot of work too.

Certain kinds of people try to hide files around the system, usually trying to take advantage of the fact that files with a leading dot in the name are invisible to the `ls` command. One common example are files that look like `...` that normally either are not seen, or, when using `ls -al` disappear in the noise of normal files like `.` or `..` that are in every directory. There is however a countermeasure to this, use `ls -Al` that suppresses `.` or `..` but shows all other dot-files.

13.4 Upgrades

No matter how large your drives, time will come when you will find you need more. As technology progresses you can get ever more for your money. At the time of writing this, it appears that 6.4 GB drives gives you the most bang for your bucks.

Note that with IDE drives you might have to remove an old drive, as the maximum number supported on

TOC

your mother board is normally only 2 or some times 4. With SCSI you can have up to 7 for narrow (8-bit) SCSI or up to 15 for wide (15 bit) SCSI, per channel. Some host adapters can support more th single channel and in any case you can have more than one host adapter per system. My personal recommendation is that you will most likely be better off with SCSI in the long run.

The question comes, where should you put this new drive? In many cases the reason for expansion is that you want a larger spool area, and in that case the fast, simple solution is to mount the drive somewhere under `/var/spool`. On the other hand newer drives are likely to be faster than older ones so in the long run you might find it worth your time to do a full reorganizing, possibly using your old design sheets.

If the upgrade is forced by running out of space in partitions used for things like `/usr` or `/var` the upgrade is a little more involved. You might consider the option of a full re-installation from your favourite (and hopefully upgraded) distribution. In this case you will have to be careful not to overwrite your essential setups. Usually these things are in the `/etc` directory. Proceed with care, fresh backups and working rescue disks. The other possibility is to simply copy the old directory over to the new directory which is mounted on a temporary mount point, edit your `/etc/fstab` file, reboot with your new partition in place and check that it works. Should it fail you can reboot with your rescue disk, re-edit `/etc/fstab` and try again.

Until volume management becomes available to Linux this is both complicated and dangerous. Do not get too surprised if you discover you need to restore your system from a backup.

The Tips-HOWTO gives the following example on how to move an entire directory structure across:

```
(cd /source/directory; tar cf - . ) | (cd /dest/directory; tar xvfp -)
```

While this approach to moving directory trees is portable among many Unix systems, it is inconvenient to remember. Also, it fails for deeply nested directory trees when pathnames become to long to handle for tar (GNU tar has special provisions to deal with long pathnames).

If you have access to GNU cp (which is always the case on Linux systems), you could as well use

```
cp -av /source/directory /dest/directory
```

GNU cp knows specifically about symbolic links, FIFOs and device files and will copy them correctly.

Remember that it might not be a good idea to try to transfer `/dev` or `/proc`.

13.5 Recovery

System crashes come in many and entertaining flavours, and partition table corruption always guarantees plenty of excitement. A recent and undoubtedly useful tool for those of us who are happy with the normal level of excitement, is gpart which means "Guess PC-Type hard disk partitions". Useful.

14. Advanced Issues

Linux and related systems offer plenty of possibilities for fast, efficient and devastating destruction. This document is no exception. With power comes dangers and the following sections describe a few more esoteric issues that should not be attempted before reading and understanding the documentation, the issues and the dangers. You should also make a backup. Also remember to try to restore the system from scratch from your backup at least once. Otherwise you might not be the first to be found with a perfect backup of your system and no tools available to reinstall it (or, even more embarrassing, some critical files missing on tape).

The techniques described here are rarely necessary but can be used for very specific setups. Think very clearly through what you wish to accomplish before playing around with this.

14.1 Hard Disk Tuning

The hard drive parameters can be tuned using the `hdparms` utility. Here the most interesting parameter is probably the read-ahead parameter which determines how much prefetch should be done in sequential reading.

If you want to try this out it makes most sense to tune for the characteristic file size on your drive but remember that this tuning is for the *entire* drive which makes it a bit more difficult. Probably this is only of use on large servers using dedicated news drives etc.

For safety the default hdparm settings are rather conservative. The disadvantage is that this mean you can get lost interrupts if you have a high frequency of IRQs as you would when using the serial port and an IDE disk as IRQs from the latter would mask other IRQs. This would be noticeable as less then ideal performance when downloading data from the net to disk. Setting `hdparm -u1 device` would prevent this masking and either improve your performance or, depending on hardware, corrupt the data on your disk. Experiment with caution and fresh backups.

14.2 File System Tuning

Most file systems come with a tuning utility and for `ext2fs` there is the `tune2fs` utility. Several parameters can be modified but perhaps the most useful parameter here is what size should be reserved and who should be able to take advantage of this which could help you getting more useful space out of your drives, possibly at the cost of less room for repairing a system should it crash.

14.3 Spindle Synchronizing

TOC

This should not in itself be dangerous, other than the peculiar fact that the exact details of the connections remain unclear for many drives. The theory is simple: keeping a fixed phase difference between the different drives in a RAID setup makes for less waiting for the right track to come into position for the read/write head. In practice it now seems that with large read-ahead buffers in the drives the effect is negligible.

Spindle synchronisation should not be used on RAID0 or RAID 0/1 as you would then lose the benefit of having the read heads over different areas of the mirrored sectors.

---

15. Further Information

There is wealth of information one should go through when setting up a major system, for instance for a news or general Internet service provider. The FAQs in the following groups are useful:

15.1 News groups

Some of the most interesting news groups are:

- Storage.
- PC storage.
- AFS.
- SCSI.
- Linux setup.

Most newsgroups have their own FAQ that are designed to answer most of your questions, as the name Frequently Asked Questions indicate. Fresh versions should be posted regularly to the relevant newsgroups. If you cannot find it in your news spool you could go directly to the FAQ main archive FTP site. The WWW versions can be browsed at FAQ main archive WWW site.

Some FAQs have their own home site, of particular interest here are

- SCSI FAQ and
- comp.arch.storage FAQ.

15.2 Mailing Lists

These are low noise channels mainly for developers. Think twice before asking questions there as noise delays the development. Some relevant lists are `linux-raid`, `linux-scsi` and `linux-ext2fs`. Many of the most useful mailing lists run on the `vger.rutgers.edu` server but this is notoriously overloaded, so try to find a mirror. There are some lists mirrored at The Redhat Home Page. Many lists are also accessible at linuxhq, and the rest of the web site is a gold mine of useful information.

If you want to find out more about the lists available you can send a message with the line `lists` to the list server at vger.rutgers.edu (majordomo@vger.rutgers.edu). If you need help on how to use the mail server just send the line `help` to the same address. Due to the popularity of this server it is likely it takes a bit to time before you get a reply or even get messages after you send a `subscribe` command.

There is also a number of other majordomo list servers that can be of interest such as the EATA driver list (linux-eata@mail.uni-mainz.de) and the Intelligent IO list linux-i2o@dpt.com.

Mailing lists are in a state of flux but you can find links to a number of interesting lists from the Linu Documentation Homepage.

TOC

15.3 HOWTO

These are intended as the primary starting points to get the background information as well as show you how to solve a specific problem. Some relevant HOWTOs are `Bootdisk`, `Installation`, `SCSI` and `UMSDOS`. The main site for these is the LDP archive at Metalab (formerly known as Sunsite).

There is a a new HOWTO out that deals with setting up a DPT RAID system, check out the DPT RAID HOWTO homepage.

15.4 Mini-HOWTO

These are the smaller free text relatives to the HOWTOs. Some relevant mini-HOWTOs are `Backup-With-MSDOS`, `Diskless`, `LILO`, `Large Disk`, `Linux+DOS+Win95+OS2`, `Linux+OS2+DOS`, `Linux+Win95`, `NFS-Root`, `Win95+Win+Linux`, `ZIP Drive` . You can find these at the same place as the HOWTOs, usually in a sub directory called `mini`. Note that these are scheduled to be converted into SGML and become proper HOWTOs in the near future.

The old `Linux Large IDE mini-HOWTO` is no longer valid, instead read `/usr/src/linux/drivers/block/README.ide` or `/usr/src/linux/Documentation/ide.txt`.

15.5 Local Resources

In most distributions of Linux there is a document directory installed, have a look in the /usr/doc directory. where most packages store their main documentation and README files etc. Also you will here find the HOWTO archive (/usr/doc/HOWTO) of ready formatted HOWTOs and also the mini-HOWTO archive (/usr/doc/HOWTO/mini) of plain text documents.

Many of the configuration files mentioned earlier can be found in the /etc directory. In particular you will want to work with the /etc/fstab file that sets up the mounting of partitions and possibly also /etc/mdtab file that is used for the `md` system to set up RAID.

The kernel source in /usr/src/linux is, of course, the ultimate documentation. In other words, *use the source, Luke*. It should also be pointed out that the kernel comes not only with source code which is even commented (well, partially at least) but also an informative documentation directory. If you are about to ask any questions about the kernel you should read this first, it will save you and many others a lot of time and possibly embarrassment.

Also have a look in your system log file (/var/log/messages) to see what is going on and in particular how the booting went if too much scrolled off your screen. Using `tail -f /var/log/messages` in a separate window or screen will give you a continuous update of what is going on in your system.

TOC

You can also take advantage of the /proc file system that is a window into the inner workings of your system. Use `cat` rather than `more` to view the files as they are reported as being zero length.

15.6 Web Pages

There is a huge number of informative web pages out there and by their very nature they change quickly so don't be too surprised if these links become quickly outdated.

A good starting point is of course the Metalab LDP archive that is a information central for documentation, project pages and much, much more.

- Mike Neuffer, the author of the DPT caching RAID controller drivers, has some interesting pages on SCSI and DPT.
- Software RAID development information can be found at Linux Kernel site along with patches and utilities.
- Disk related information on benchmarking, RAID, reliability and much, much more can be found at Linas Vepstas project page.
- There is also information available on how to RAID the root partition and what software packages are needed to achieve this.
- In depth documentation on ext2fs is also available.
- People who looking for information on VFAT, FAT32 and Joliet could have a look at the development page. These drivers are now in the 2.1.x kernel development series as well as in 2.0.34 and later.
- For more information on booting and also some BSD information have a look at booting information page.

For diagrams and information on all sorts of disk drives, controllers etc. both for current and discontinued lines The Ref is the site you need. There is a lot of useful information here, a real treasure trove. You can also download the database using FTP.

Please let me know if you have any other leads that can be of interest.

15.7 Search Engines

Remember you can also use the web search engines and that some, like

- Altavista
- Excite
- Hotbot

can also search usenet news.

Also remember that Dejanews is a dedicated news searcher that keeps a news spool from early 1995 and onwards.

If you have to ask for help you are most likely to get help in the Linux Setup news group. Due to large workload and a slow network connection I am not able to follow that newsgroup so if you want to contact me you have to do so by e-mail.

---

16. Getting Help

In the end you might find yourself unable to solve your problems and need help from someone else. The most efficient way is either to ask someone local or in your nearest Linux user group, search the web for the nearest one.

Another possibility is to ask on Usenet News in one of the many, many newsgroups available. The problem is that these have such a high volume and noise (called low signal-to-noise ratio) that your question can easily fall through unanswered.

No matter where you ask it is important to ask well or you will not be taken seriously. Saying just *my disk does not work* is not going to help you and instead the noise level is increased even further and if you are lucky someone will ask you to clarify.

Instead describe your problems in some detail that will enable people to help you. The problem could lie somewhere you did not expect. Therefore you are advised to list up the following information on your system:

Hardware

- Processor
- DMA
- IRQ
- Chip set (LX, BX etc)
- Bus (ISA, VESA, PCI etc)
- Expansion cards used (Disk controllers, video, IO etc)

Software

- BIOS (On motherboard and possibly SCSI host adapters)
- LILO, if used
- Linux kernel version as well as possible modifications and patches
- Kernel parameters, if any
- Software that shows the error (with version number or date)

Peripherals

- Type of disk drives with manufacturer name, version and type
- Other relevant peripherals connected to the same busses

As an example of how interrelated these problems are: an old chip set caused problems with a certain combination of video controller and SCSI host adapter.

Remember that booting text is logged to `/var/log/messages` which can answer most of the questions above. Obviously if the drives fail you might not be able to get the log saved to disk but you can at least scroll back up the screen using the `SHIFT` and `PAGE UP` keys. It may also be useful to

include part of this in your request for help but do not go overboard, keep it *brief* as a complete log dumped to Usenet News is more than a little annoying.

17. Concluding Remarks

Disk tuning and partition decisions are difficult to make, and there are no hard rules here. Nevertheless it is a good idea to work more on this as the payoffs can be considerable. Maximizing usage on one drive only while the others are idle is unlikely to be optimal, watch the drive light, they are not there just for decoration. For a properly set up system the lights should look like Christmas in a disco. Linux offers software RAID but also support for some hardware base SCSI RAID controllers. Check what is available. As your system and experiences evolve you are likely to repartition and you might look on this document again. Additions are always welcome.

Finally I'd like to sum up my recommendations:

- Disk is cheap but the data they contain could be much more valuable, use and test your backup system.
- Work is also expensive, make sure you get large enough disks as refitting new or repartitioning old disks takes time.
- Think reliability, replace old disks before they fail.
- Keep a paper copy of your setup, having it all on disk when the machine is down will not help you much.
- Start out with a simple design with a minimum of fancy technology and rather fit it in later. In general adding is easier than replacing, be it disks, technology or other features.

17.1 Coming Soon

There are a few more important things that are about to appear here. In particular I will add more example tables as I am about to set up two fairly large and general systems, one at work and one at home. These should give some general feeling on how a system can be set up for either of these two purposes. Examples of smooth running existing systems are also welcome.

There is also a fair bit of work left to do on the various kinds of file systems and utilities.

There will be a big addition on drive technologies coming soon as well as a more in depth description on using `fdisk`, `cfdisk` and `sfdisk`. The file systems will be beefed up as more features become available as well as more on RAID and what directories can benefit from what RAID level.

There is some minor overlapping with the Linux Filesystem Structure Standard and FHS that I hope to integrate better soon, which will probably mean a big reworking of all the tables at the end of this document.

As more people start reading this I should get some more comments and feedback. I am also thinking of making a program that can automate a fair bit of this decision making process and although it is unlikely to be optimum it should provide a simpler, more complete starting point.

TOC

17.2 Request for Information

It has taken a fair bit of time to write this document and although most pieces are beginning to come together there are still some information needed before we are out of the beta stage.

- More information on swap sizing policies is needed as well as information on the largest swap size possible under the various kernel versions.
- How common is drive or file system corruption? So far I have only heard of problems caused by flaky hardware.
- References to speed and drives is needed.
- Are any other Linux compatible RAID controllers available?
- What relevant monitoring, management and maintenance tools are available?
- General references to information sources are needed, perhaps this should be a separate document?
- Usage of `/tmp` and `/var/tmp` has been hard to determine, in fact what programs use which directory is not well defined and more information here is required. Still, it seems at least clear that these should reside on different physical drives in order to increase paralellicity.

17.3 Suggested Project Work

Now and then people post on comp.os.linux.*, looking for good project ideas. Here I will list a few that comes to mind that are relevant to this document. Plans about big projects such as new file systems should still be posted in order to either find co-workers or see if someone is already working on it.

Planning tools

that can automate the design process outlines earlier would probably make a medium sized project, perhaps as an exercise in constraint based programming.

Partitioning tools

that take the output of the previously mentioned program and format drives in parallel and apply the appropriate symbolic links to the directory structure. It would probably be best if this were integrated in existing system installation software. The drive partitioning setup used in Solaris is an example of what it can look like.

Surveillance tools

that keep an eye on the partition sizes and warn before a partition overflows.

Migration tools

that safely lets you move old structures to new (for instance RAID) systems. This could probably be done as a shell script controlling a back up program and would be rather simple. Still, be sure it is safe and that the changes can be undone.

---

18. Questions and Answers

This is just a collection of what I believe are the most common questions people might have. Give me more feedback and I will turn this section into a proper FAQ.

- Q:How many physical disk drives (spindles) does a Linux system need?

  A: Linux can run just fine on one drive (spindle). Having enough RAM (around 32 MB, and up to 64 MB) to support swapping is a better price/performance choice than getting a second disk. (E)IDE disk is usually cheaper (but a little slower) than SCSI.
- Q: I have a single drive, will this HOWTO help me?

  A: Yes, although only to a minor degree. Still, section Physical Track Positioning will offer you some gains.
- Q: Are there any disadvantages in this scheme?

  A: There is only a minor snag: if even a single partition overflows the system might stop working properly. The severity depends of course on what partition is affected. Still this is not hard to monitor, the command `df` gives you a good overview of the situation. Also check the swap partition(s) using `free` to make sure you are not about to run out of virtual memory.
- Q: OK, so should I split the system into as many partitions as possible for a single drive?

  A: No, there are several disadvantages to that. First of all maintenance becomes needlessly complex and you gain very little in this. In fact if your partitions are too big you will seek across larger areas than needed. This is a balance and dependent on the number of physical drives you have.
- Q: Does that mean more drives allows more partitions?

  A: To some degree, yes. Still, some directories should not be split off from root, check out the file system standards for more details.
- Q: What if I have many drives I want to use?

  A: If you have more than 3-4 drives you should consider using RAID of some form. Still, it is a good idea to keep your root partition on a simple partition without RAID, see section RAID for more details.
- Q: I have installed the latest Windows95 but cannot access this partition from within the Linux system, what is wrong?

  A: Most likely you are using `FAT32` in your windows partition. It seems that Microsoft decided we needed yet another format, and this was introduced in their latest version of Windows95, called OSR2. The advantage is that this format is better suited to large drives.

  You might also be interested to hear that Microsoft NT 4.0 does not support it yet either.

TOC

- Q: I cannot get the disk size and partition sizes to match, something is missing. What has happened?

  A:It is possible you have mounted a partition onto a mount point that was not an empty directory. Mount points are directories and if it is not empty the mounting will mask the contents. If you do the sums you will see the amount of disk space used in this directory is missing from the observed total.

  To solve this you can boot from a rescue disk and see what is hiding behind your mount points and remove or transfer the contents by mounting the offending partition on a temporary mounting point. You might find it useful to have "spare" emergency mounting points ready made.
- Q: It doesn't look like my swap partition is in use, how come?

  A: It is possible that it has not been necessary to swap out, especially if you have plenty of RAM. Check your log files to see if you ran out of memory at one point or another, in that case your swap space should have been put to use. If not it is possible that either the swap partition was not assigned the right number, that you did not prepare it with `mkswap` or that you have not done `swapon` or added it to your fstab.
- Q: What is this nyx that is mentioned several times here?

  A: It is a large free Unix system with currently about 10000 users. I use it for my web pages for this HOWTO as well as a source of ideas for a setup of large Unix systems. It has been running for many years and has a quite stable setup. For more information you can view the Nyx homepage which also gives you information on how to get your own free account.

---

19. Bits and Pieces

This is basically a section where I stuff all the bits I have not yet decided where should go, yet that I feel is worth knowing about. It is a kind of transient area.

19.1 Swap Partition: to Use or Not to Use

In many cases you do not need a swap partition, for instance if you have plenty of RAM, say, more than 64 MB, and you are the sole user of the machine. In this case you can experiment running without a swap partition and check the system logs to see if you ran out of virtual memory at any point.

Removing swap partitions have two advantages:

- you save disk space (rather obvious really)
- you save seek time as swap partitions otherwise would lie in the middle of your disk space.

In the end, having a swap partition is like having a heated toilet: you do not use it very often, but you sure appreciate it when you require it.

19.2 Mount Point and `/mnt`

In an earlier version of this document I proposed to put all permanently mounted partitions under `/mnt`. That, however, is not such a good idea as this itself can be used as a mount point, which leads to all mounted partitions becoming unavailable. Instead I will propose mounting straight from root using a meaningful name like `/mnt.descriptive-name`.

Lately I have become aware that some Linux distributions use mount points at subdirectories *under* `/mnt`, such as `/mnt/floppy` and `/mnt/cdrom`, which just shows how confused the whole issue is. Hopefully FHS should clarify this.

19.3 Power and Heating

Not many years ago a machine with the equivalent power of a modern PC required 3-phase power and cooling, usually by air conditioning the machine room, some times also by water cooling. Technology has progressed very quickly giving not only high speed but also low power components. Still, there is a definite limit to the technology, something one should keep in mind as the system is expanded with yet another disk drive or PCI card. When the power supply is running at full rated power, keep in mind that all this energy is going somewhere, mostly into heat. Unless this is dissipated using fans you will get a serious heating inside the cabinet followed by a reduced reliability and also life time of the electronics. Manufacturers state minimum cooling requirements for their drives, usually in terms of cubic feet per minute (CFM). You are well advised to take this serious.

Keep air flow passages open, clean out dust and check the temperature of your system running. If it i hot to touch it is probably running too hot.

If possible use sequential spin up for the drives. It is during spin up, when the drive platters accelerate up to normal speed, that a drive consumes maximum power and if all drives start up simultaneously you could go beyond the rated power maximum of your power supply.

19.4 Dejanews

This is an Internet system that no doubt most of you are familiar with. It searches and serves *Usenet News* articles from 1995 and to the latest postings and also offers a web based reading and posting interface. There is a lot more, check out Dejanews for more information.

What perhaps is less known, is that they use about 120 Linux SMP computers many of which use the `md` module to manage between 4 and 24 Gig of disk space (over 1200 Gig altogether) for this service. The system is continuously growing but at the time of writing they use mostly dual Pentium Pro 200MHz and Pentium II 300 MHz systems with 256 MB RAM or more.

A production database machine normally has 1 disk for the operating system and between 4 and 6 disks managed by the `md` module where the articles are archived. The drives are connected to BusLogic Model BT-946C and BT-958 PCI SCSI adapters, usually one to a machine.

For the production systems (which are up 365 days a year) the downtime attributable to disk errors is less than 0.25 % (that is a quarter of 1%, not 25%).

Just in case: this is not an advertisement, it is stated as an example of how much is required for what is a major Internet service.

---

20. Appendix A: Partitioning Layout Table: Mounting and Linking

The following table is designed to make layout a simpler paper and pencil exercise. It is probably best to print it out (using NON PROPORTIONAL fonts) and adjust the numbers until you are happy with them.

Mount point is what directory you wish to mount a partition on or the actual device. This is also a good place to note how you plan to use symbolic links.

The size given corresponds to a fairly big Debian 1.2.6 installation. Other examples are coming later.

Mainly you use this table to select what structure and drives you will use, the partition numbers and letters will come from the next two tables.

Directory	Mount point	speed	seek	transfer	size	SIZE
swap	_____	ooooo	ooooo	ooooo	32	____
/	_____	o	o	o	20	____
/tmp	_____	oooo	oooo	oooo		____
/var	_____	oo	oo	oo	25	____
/var/tmp	_____	oooo	oooo	oooo		____
/var/spool	_____					____
/var/spool/mail	_____	o	o	o		____
/var/spool/news	_____	ooo	ooo	oo		____
/var/spool/____	_____	____	____	____		____
/home	_____	oo	oo	oo		____
/usr	_____				500	____
/usr/bin	_____	o	oo	o	250	____
/usr/lib	_____	oo	oo	ooo	200	____
/usr/local	_____					____
/usr/local/bin	_____	o	oo	o		____
/usr/local/lib	_____	oo	oo	ooo		____
/usr/local/____	_____					____
/usr/src	_____	o	oo	o	50	____
DOS	_____	o	o	o		____
Win	_____	oo	oo	oo		____
NT	_____	ooo	ooo	ooo		____
/mnt._____	_____	____	____	____		____
/mnt._____	_____	____	____	____		____
/mnt._____	_____	____	____	____		____

```
/_______________  _____________      _____    _____    _____
/_______________  _____________      _____    _____    _____
/_______________  _____________      _____    _____    _____


Total capacity:
```

21. Appendix B: Partitioning Layout Table: Numbering and Sizing

This table follows the same logical structure as the table above where you decided what disk to use. Here you select the physical tracking, keeping in mind the effect of track positioning mentioned earlier in Physical Track Positioning.

The final partition number will come out of the table after this.

```
  Drive            sda     sdb     sdc     hda     hdb     hdc     ___

SCSI ID          |  __   |  __   |  __   |

Directory
swap             |       |       |       |       |       |       |

/                |       |       |       |       |       |       |

/tmp             |       |       |       |       |       |       |

/var             :       :       :       :       :       :       :
/var/tmp         |       |       |       |       |       |       |
/var/spool       :       :       :       :       :       :       :
/var/spool/mail  |       |       |       |       |       |       |
/var/spool/news  :       :       :       :       :       :       :
/var/spool/____  |       |       |       |       |       |       |

/home            |       |       |       |       |       |       |

/usr             |       |       |       |       |       |       |
/usr/bin         :       :       :       :       :       :       :
/usr/lib         |       |       |       |       |       |       |
/usr/local       :       :       :       :       :       :       :
/usr/local/bin   |       |       |       |       |       |       |
/usr/local/lib   :       :       :       :       :       :       :
/usr/local/____  |       |       |       |       |       |       |
/usr/src         :       :       :       :

DOS              |       |       |       |       |       |       |
Win              :       :       :       :       :       :       :
NT               |       |       |       |       |       |       |

/mnt.___/_____   |       |       |       |       |       |       |
```

```
/mnt.___/_____  :       :       :       :       :       :
/mnt.___/_____  |       |       |       |       |       |
/______________ :       :       :       :       :       :
/______________ |       |       |       |       |       |
/______________ :       :       :       :       :       :


Total capacity:
```

22. Appendix C: Partitioning Layout Table: Partition Placement

This is just to sort the partition numbers in ascending order ready to input to fdisk or cfdisk. Here you take physical track positioning into account when finalizing your design. Unless you get specific information otherwise, you can assume track 0 is the outermost track.

These numbers and letters are then used to update the previous tables, all of which you will find very useful in later maintenance.

In case of disk crash you might find it handy to know what SCSI id belongs to which drive, consider keeping a paper copy of this.

```
          Drive :   sda     sdb     sdc     hda     hdb     hdc     ___

Total capacity: |  ___  |  ___  |  ___  |  ___  |  ___  |  ___  |  ___
SCSI ID         |  __   |  __   |  __   |

Partition

1               |       |       |       |       |       |       |
2               :       :       :       :       :       :       :
3               |       |       |       |       |       |       |
4               :       :       :       :       :       :       :
5               |       |       |       |       |       |       |
6               :       :       :       :       :       :       :
7               |       |       |       |       |       |       |
8               :       :       :       :       :       :       :
9               |       |       |       |       |       |       |
10              :       :       :       :       :       :       :
11              |       |       |       |       |       |       |
12              :       :       :       :       :       :       :
13              |       |       |       |       |       |       |
14              :       :       :       :       :       :       :
15              |       |       |       |       |       |       |
16              :       :       :       :       :       :       :
```

23. Appendix D: Example: Multipurpose Server

The following table is from the setup of a medium sized multipurpose server where I work. Aside from being a general Linux machine it will also be a network related server (DNS, mail, FTP, news, printers etc.) X server for various CAD programs, CD ROM burner and many other things. The files reside on 3 SCSI drives with a capacity of 600, 1000 and 1300 MB.

Some further speed could possibly be gained by splitting `/usr/local` from the rest of the `/usr` system but we deemed the further added complexity would not be worth it. With another couple of drives this could be more worthwhile. In this setup drive sda is old and slow and could just a well be replaced by an IDE drive. The other two drives are both rather fast. Basically we split most of the load between these two. To reduce dangers of imbalance in partition sizing we have decided to keep `/usr/bin` and `/usr/local/bin` in one drive and `/usr/lib` and `/usr/local/lib` on another separate drive which also affords us some drive parallelizing.

Even more could be gained by using RAID but we felt that as a server we needed more reliability than was then afforded by the `md` patch and a dedicated RAID controller was out of our reach.

24. Appendix E: Example: Mounting and Linking

```
Directory        Mount point      speed   seek    transfer        size    SIZE


swap             sdb2, sdc2       ooooo   ooooo   ooooo           32      2x64

/                sda2             o       o       o               20       100

/tmp             sdb3             oooo    oooo    oooo                     300

/var             __________       oo      oo      oo                       ____
/var/tmp         sdc3             oooo    oooo    oooo                     300
/var/spool       sdb1                                                      436
/var/spool/mail  __________       o       o       o                        ____
/var/spool/news  __________       ooo     ooo     oo                       ____
/var/spool/____  __________       ____    ____    ____                     ____

/home            sda3             oo      oo      oo                       400

/usr             sdb4                                             230      200
/usr/bin         __________       o       oo      o               30       ____
/usr/lib         -> libdisk       oo      oo      ooo             70       ____
/usr/local       __________                                                ____
/usr/local/bin   __________       o       oo      o                        ____
/usr/local/lib   -> libdisk       oo      oo      ooo                      ____
/usr/local/____  __________                                                ____
/usr/src         ->/home/usr.src  o       oo      o               10       ____

DOS              sda1             o       o       o                        100
Win              __________       oo      oo      oo                       ____
NT               __________       ooo     ooo     ooo                      ____

/mnt.libdisk     sdc4             oo      oo      ooo                      226
/mnt.cd          sdc1             o       o       oo                       710


Total capacity: 2900 MB
```

25. Appendix F: Example: Numbering and Sizing

Here we do the adjustment of sizes and positioning.

```
Directory          sda     sdb     sdc


swap             |       |   64  |   64  |

/                |  100  |       |       |

/tmp             |       |  300  |       |

/var             :       :       :       :
/var/tmp         |       |       |  300  |
/var/spool       :       :  436  :       :
/var/spool/mail  |       |       |       |
/var/spool/news  :       :       :       :
/var/spool/____  |       |       |       |

/home            |  400  |       |       |

/usr             |       |  200  |       |
/usr/bin         :       :       :       :
/usr/lib         |       |       |       |
/usr/local       :       :       :       :
/usr/local/bin   |       |       |       |
/usr/local/lib   :       :       :       :
/usr/local/____  |       |       |       |
/usr/src         :       :       :       :

DOS              |  100  |       |       |
Win              :       :       :       :
NT               |       |       |       |

/mnt.libdisk     |       |       |  226  |
/mnt.cd          :       :       :  710  :
/mnt.___/_____   |       |       |       |


Total capacity:  |  600  | 1000  | 1300  |
```

26. Appendix G: Example: Partition Placement

This is just to sort the partition numbers in ascending order ready to input to fdisk or cfdisk. Remember to optimize for physical track positioning (not done here).

```
          Drive :   sda     sdb     sdc

Total capacity: |   600 |  1000 |  1300 |

Partition

1               |   100 |   436 |   710 |
2               :   100 :    64 :    64 :
3               |   400 |   300 |   300 |
4               :       :   200 :   226 :
```

27. Appendix H: Example II

The following is an example of a server setup in an academic setting, and is contributed by `nakano (at) apm.seikei.ac.jp`. I have only done minor editing to this section.

`/var/spool/delegate` is a directory for storing logs and cache files of an WWW proxy server program, "delegated". Since I don't notice it widely, there are 1000--1500 requests/day currently, and average disk usage is 15--30% with expiration of caches each day.

`/mnt.archive` is used for data files which are big and not frequently referenced such a s experimental data (especially graphic ones), various source archives, and Win95 backups (growing very fast...).

`/mnt.root` is backup root file system containing rescue utilities. A boot floppy is also prepared to boot with this partition.

```
=================================================
Directory                sda      sdb     hda

swap                     |    64 |    64 |       |
/                        |       |       |    20 |
/tmp                     |       |       |   180 |

/var                     :   300 :       :       :
/var/tmp                 |       |   300 |       |
/var/spool/delegate      |   300 |       |       |

/home                    |       |       |   850 |
/usr                     |   360 |       |       |
/usr/lib                 -> /mnt.lib/usr.lib
/usr/local/lib           -> /mnt.lib/usr.local.lib

/mnt.lib                 |       |   350 |       |
/mnt.archive             :       :  1300 :       :
/mnt.root                |       |    20 |       |

Total capacity:             1024    2034    1050


=================================================
        Drive :            sda     sdb     hda
Total capacity:          |  1024 |  2034 |  1050 |

Partition
1                        |   300 |    20 |    20 |
2                        :    64 :  1300 :   180 :
3                        |   300 |    64 |   850 |
4                        :   360 :   ext :       :
```

```
5                          |        |   300 |        |
6                          :        :   350 :        :


Filesystem          1024-blocks  Used Available Capacity Mounted on
/dev/hda1               19485   10534     7945     57%   /
/dev/hda2              178598      13   169362      0%   /tmp
/dev/hda3              826640  440814   343138     56%   /home
/dev/sda1              306088   33580   256700     12%   /var
/dev/sda3              297925   47730   234807     17%   /var/spool/delegate
/dev/sda4              363272  170872   173640     50%   /usr
/dev/sdb5              297598       2   282228      0%   /var/tmp
/dev/sdb2             1339248  302564   967520     24%   /mnt.archive
/dev/sdb6              323716   78792   228208     26%   /mnt.lib
```

Apparently `/tmp` and `/var/tmp` is too big. These directories shall be packed together into one partition when disk space shortage comes.

`/mnt.lib` is also seemed to be, but I plan to install newer TeX and ghostscript archives, so `/usr/local/lib` may grow about 100 MB or so (since we must use Japanese fonts!).

Whole system is backed up by Seagate Tapestore 8000 (Travan TR-4, 4G/8G).

---

28. Appendix I: Example III: SPARC Solaris

The following section is the basic design used at work for a number of Sun SPARC servers running Solaris 2.5.1 in an industrial development environment. It serves a number of database and cad applications in addition to the normal services such as mail.

Simplicity is emphasized here so `/usr/lib` has not been split off from `/usr`.

This is the basic layout, planned for about 100 users.

Drive:	SCSI 0		SCSI 1	
Partition	Size (MB)	Mount point	Size (MB)	Mount point
0	160	swap	160	swap
1	100	/tmp	100	/var/tmp
2	400	/usr		
3	100	/		
4	50	/var		
5				
6	remainder	/local0	remainder	/local1

Due to specific requirements at this place it is at times necessary to have large partitions available on a short notice. Therefore drive 0 is given as many tasks as feasible, leaving a large `/local1` partition.

This setup has been in use for some time now and found satisfactorily.

For a more general and balanced system it would be better to swap `/tmp` and `/var/tmp` and then move `/var` to drive 1.

29. Appendix J: Example IV: Server with 4 Drives

This gives an example of using all techniques described earlier, short of RAID. It is admittedly rather complicated but offers in return high performance from modest hardware. Dimensioning are skipped but reasonable figures can be found in previous examples.

Partition	sda	sdb	sdc	sdd
1	root	overview	lib	news
2	swap	swap	swap	swap
3	home	/usr	/var/tmp	/tmp
4		spare root	mail	/var

Setup is optimised with respect to track positioning but also for minimising drive seeks.

If you want DOS or Windows too you will have to use `sda1` for this and move the other partitions after that. It will be advantageous to use the swap partitions on `sdb2`, `sdc2` and `sdd2` for Windows swap, `TEMPDIR` and Windows temporary directory under these sessions. A number of other HOWTOs describe how you can make several operating systems coexist on your machine.

For completeness a 4 drive example using several types of RAID is also given which is even more complex than the example above.

Partition	sda	sdb	sdc	sdd
1	boot	overview	news	news
2	overview	swap	swap	swap
3	swap	lib	lib	lib
4	lib	overview	/tmp	/tmp
5	/var/tmp	/var/tmp	mail	/usr
6	/home	/usr	/usr	mail
7	/usr	/home	/var	
8	/ (root)	spare root		

Here all duplicates are parts of a RAID 0 set with two exceptions, swap which is interleaved and home and mail which are implemented as RAID 1 for safety.

Note that boot and root are separated: only the boot file with the kernel has to reside within the 1023 cylinder limit. The rest of the root files can be anywhere and here they are placed on the slowest outermost partition. For simplicity and safety the root partition is not on a RAID system.

With such a complicated comes an equally complicated `fstab` file. The large number of partitions makes it important to do the `fsck` passes in the right order, otherwise the process can take perhaps ten times as long time to complete as the optimal solution.

```
/dev/sda8       /               ?       ?               1 1 (a)
/dev/sdb8       /               ?       noauto          1 2 (b)
/dev/sda1       boot            ?       ?               1 2 (a)
/dev/sdc7       /var            ?       ?               1 2 (c)
/dev/md1        news            ?       ?               1 3 (c+d)
/dev/md2        /var/tmp        ?       ?               1 3 (a+b)
/dev/md3        mail            ?       ?               1 4 (c+d)
/dev/md4        /home           ?       ?               1 4 (a+b)
/dev/md5        /tmp            ?       ?               1 5 (c+d)
/dev/md6        /usr            ?       ?               1 6 (a+b+c+d)
/dev/md7        /lib            ?       ?               1 7 (a+b+c+d)
```

The letters in the brackets indicate what drives will be active for each `fsck` entry and pass. These letters are *not* present in a real `fstab` file. All in all there are 7 passes.

---

30. Appendix K: Example V: Dual Drive System

A dual drive system offers less opportunity for clever schemes but the following should provide a simple starting point.

```
Partition       sda             sdb
                ----            ----
        1       boot            lib
        2       swap            news
        3       /tmp            swap
        4       /usr            /var/tmp
        5       /var            /home
        6       / (root)
```

If you use a dual OS system you have to keep in mind that many other systems must boot from the first partition on the first drive. A simple DOS / Linux system could look like this:

```
Partition       sda             sdb
                ----            ----
        1       DOS             lib
        2       boot            news
        3       swap            swap
        4       /tmp            /var/tmp
        5       /usr            /home
        6       /var            DOSTEMP
        7       / (root)
```

Also remember that DOS and Windows prefer there to be just a single primary partition which has to be the first one where it boots from. As Linux can happily exist in logical partitions this is not a big problem.

31. Appendix L: Example VI: Single Drive System

Although this falls somewhat outside the scope of this HOWTO it cannot be denied that recently some rather large drives have become very affordable. Drives with 10 - 20 GB are becoming common and the question often is how best to partition such monsters. Interestingly enough very few seem to have any problems in filling up such drives and the future looks generally quite rosy for manufacturers planning on even bigger drives.

Opportunities for optimisations are of course even smaller than for 2 drive systems but some tricks can still be used to optimise track positions while minimising head movements.

```
Partition       hda             Size estimate (MB)
                ----            ------------------
         1      DOS             500
         2      boot            20
         3      Winswap         200
         4      data            The bulk of the drive
         5      lib             50 - 500
         6      news            300+
         7      swap            128     (Maximum size for 32-bit CPU)
         8      tmp             300+    (/tmp and /var/tmp)
         9      /usr            50 - 500
        10      /home           300+
        11      /var            50 - 300
        12      mail            300+
        13      dosdata         10      ( Windows bug workaround!)
```

Remember that the `dosdata` partition is a DOS filesystem that must be the very last partition on the drive, otherwise Windows gets confused.

Linux Networking-HOWTO (Previously the Net-3 Howto)

Current Author: {Poet} poet@linuxports.com

v1.5, August 1999

Original Authors: Terry Dawson (main author), VK2KTJ; Alessandro Rubini (maintainer)

The Linux Operating System boasts kernel based networking support written almost entirely from scratch. The performance of the tcp/ip implementation in recent kernels makes it a worthy alternative to even the best of its peers. This document aims to describe how to install and configure the Linux networking software and associated tools.

1. Introduction.

2. Document History

- 2.1 Feedback

3. How to use this HOWTO.

- 3.1 Conventions used in this document

4. General Information about Linux Networking.

- 4.1 A brief history of Linux Networking Kernel Development.
- 4.2 Linux Networking Resources.
- 4.3 Where to get some non-linux-specific network information.

5. Generic Network Configuration Information.

- 5.1 What do I need to start ?
- 5.2 Where should I put the configuration commands ?

- 5.3 Creating your network interfaces.
- 5.4 Configuring a network interface.
- 5.5 Configuring your Name Resolver.
- 5.6 Configuring your loopback interface.
- 5.7 Routing.
- 5.8 Configuring your network servers and services.
- 5.9 Other miscellaneous network related configuration files.
- 5.10 Network Security and access control.

6. IP- and Ethernet-Related Information

- 6.1 Ethernet
- 6.2 EQL - multiple line traffic equaliser
- 6.3 IP Accounting (for Linux-2.0)
- 6.4 IP Accounting (for Linux-2.2)
- 6.5 IP Aliasing
- 6.6 IP Firewall (for Linux-2.0)
- 6.7 IP Firewall (for Linux-2.2)
- 6.8 IPIP Encapsulation
- 6.9 IP Masquerade
- 6.10 IP Transparent Proxy
- 6.11 IPv6
- 6.12 Mobile IP
- 6.13 Multicast
- 6.14 NAT - Network Address Translation
- 6.15 Traffic Shaper - Changing allowed bandwidth
- 6.16 Routing in Linux-2.2

7. Using common PC hardware

- 7.1 ISDN
- 7.2 PLIP for Linux-2.0
- 7.3 PLIP for Linux-2.2
- 7.4 PPP

- 7.5 SLIP client
- 7.6 SLIP server.

8. Other Network Technologies

- 8.1 ARCNet
- 8.2 Appletalk (AF_APPLETALK)
- 8.3 ATM
- 8.4 AX25 (AF_AX25)
- 8.5 DECNet
- 8.6 FDDI
- 8.7 Frame Relay
- 8.8 IPX (AF_IPX)
- 8.9 NetRom (AF_NETROM)
- 8.10 Rose protocol (AF_ROSE)
- 8.11 SAMBA - `NetBEUI', `NetBios', `CIFS' support.
- 8.12 STRIP support (Starmode Radio IP)
- 8.13 Token Ring
- 8.14 X.25
- 8.15 WaveLan Card

9. Cables and Cabling

- 9.1 Serial NULL Modem cable
- 9.2 Parallel port cable (PLIP cable)
- 9.3 10base2 (thin coax) Ethernet Cabling
- 9.4 Twisted Pair Ethernet Cable

10. Glossary of Terms used in this document.

11. Linux for an ISP ?

TOC

12. Acknowledgements

13. Copyright.

1. Introduction.

This is the first release since LinuxPorts has become the author of this document. First let me say that we hope that over the next few months you will find this document to be of use and that we are able to provide accurate and timely information in regards to networking issues with Linux.

This document like the other howto's that we manage is going to become very different, this document will shortly become the Networking-HOWTO not just the Net-3(4) Howto. We will cover such items as PPP, VPN, and others...

2. Document History

The original NET-FAQ was written by Matt Welsh and Terry Dawson to answer frequently asked questions about networking for Linux at a time before the Linux Documentation Project had formally started. It covered the very early development versions of the Linux Networking Kernel. The NET-2-HOWTO superceded the NET-FAQ and was one of the original LDP HOWTO documents, it covered what was called version 2 and later version 3 of the Linux kernel Networking software. This document in turn supercedes it and relates only to version 4 of the Linux Networking Kernel or more specifically kernel releases 2.x and 2.2.x.

Previous versions of this document became quite large because of the enormous amount of material that fell within its scope. To help reduce this problem a number of HOWTO's dealing with specific networking topics have been produced. This document will provide pointers to them where relevant and cover those areas not yet covered by other documents.

2.1 Feedback

We are always interested in feedback. Please contact us at: poet@linuxports.com.

Again, if you find anything erroneous or anything you would like to see added, please contact us.

3. How to use this HOWTO.

This document is organized top-down. The first sections include informative material and can be skipped if you are not interested; what follows is a generic discussion of networking issues, and you must ensure you understand this before proceeding to more specific parts. The rest, ``technology specific'' information is grouped in three main sections: Ethernet and IP-related information, technologies pertaining to widespread PC hardware and seldom-used technologies.

The suggested path through the document is thus the following:

Read the generic sections

These sections apply to every, or nearly every, technology described later and so are very important for you to understand. On the other hand, I expect many of the readers to be already confident with this material.

Consider your network

You should know how your network is, or will be, designed and exactly what hardware and technology types you will be implementing.

Read the ``Ethernet and IP'' section if you are directly connected a LAN or the Internet

This section describes basic Ethernet configuration and the various features that Linux offers for IP networks, like firewalling, advanced routing and so on.

Read the next section if you are interested in low-cost local networks or dial-up connections

The section describes PLIP, PPP, SLIP and ISDN, the widespread technologies used on personal workstations.

Read the technology specific sections related to your requirements

If your needs differ from IP and/or common hardware, the final section covers details specific to non-IP protocols and peculiar communication hardware.

Do the configuration work

You should actually try to configure your network and take careful note of any problems you have.

Look for further help if needed

If you experience problems that this document does not help you to resolve then read the section related to where to get help or where to report bugs.

Have fun!

Networking is fun, enjoy it.

3.1 Conventions used in this document

TOC

No special convention is used here, but you must be warned about the way commands are shown. Following the classic Unix documentation, any command you should type to your shell is prefixed by a prompt. This howto shows "`user%`" as the prompt for commands that do not require superuser privileges, and "`root#`" as the prompt for commands that need to run as root. I chose to use "`root#`" instead of a plain "#" to prevent confusion with snapshots from shell scripts, where the hash mark is used to define comment lines.

When ``Kernel Compile Options'' are shown, they are represented in the format used by *menuconfig*. They should be understandable even if you (like me) are not used to *menuconfig*. If you are in doubt about the options' nesting, running the program once can't but help.

Note that any link to other HOWTO's is local to help you browsing your local copy of the LDP documents, in case you are using the html version of this document. If you don't have a complete set of documents, every HOWTO can be retrieved from `metalab.unc.edu` (directory `/pub/Linux/HOWTO`) and its countless mirrors.

---

4. General Information about Linux Networking.

4.1 A brief history of Linux Networking Kernel Development.

Developing a brand new kernel implementation of the tcp/ip protocol stack that would perform as well as existing implementations was not an easy task. The decision not to port one of the existing implementations was made at a time when there was some uncertainty as to whether the existing implementations may become encumbered by restrictive copyrights because of the court case put by U.S.L. and when there was a lot of fresh enthusiasm for doing it differently and perhaps even better than had already been done.

The original volunteer to lead development of the kernel network code was Ross Biro `<biro@yggdrasil.com>`. Ross produced a simple and incomplete but mostly usable implementation set of routines that were complemented by an ethernet driver for the WD-8003 network interface card. This was enough to get many people testing and experimenting with the software and some people even managed to connect machines in this configuration to live internet connections. The pressure within the Linux community driving development for networking support was building and eventually the cost of a combination of some unfair pressure applied to Ross and his own personal commitments outweighed the benefit he was deriving and he stepped down as lead developer. Ross's efforts in getting the project started and accepting the responsibility for actually producing something useful in such controversial circumstances were what catalyzed all future work and were therefore an essential component of the success of the current product.

Orest Zborowski `<obz@Kodak.COM>` produced the original BSD socket programming interface for the Linux kernel. This was a big step forward as it allowed many of the existing network applications to be ported to linux without serious modification.

Somewhere about this time Laurence Culhane `<loz@holmes.demon.co.uk>` developed the first drivers for Linux to support the SLIP protocol. These enabled many people who did not have access to Ethernet networking to experiment with the new networking software. Again, some people took this driver and pressed it into service to connect them to the Internet. This gave many more people a taste of the possibilities that could be realized if Linux had full networking support and grew the number of users actively using and experimenting with the networking software that existed.

One of the people that had also been actively working on the task of building networking support was Fred van Kempen `<waltje@uwalt.nl.mugnet.org>`. After a period of some uncertainty following Ross's resignation from the lead developer position Fred offered his time and effort and accepted the role essentially unopposed. Fred had some ambitious plans for the direction that he wanted to take the Linux networking software and he set about progressing in those directions. Fred produced a series of networking code called the `NET-2' kernel code (the `NET' code being Ross's) which many people were able to use pretty much usefully. Fred formally put a number of innovations on the

development agenda, such as the dynamic device interface, Amateur Radio AX.25 protocol support a more modularly designed networking implementation. Fred's NET-2 code was used by a fairly large number of enthusiasts, the number increasing all the time as word spread that the software was working. The networking software at this time was still a large number of patches to the standard release of kernel code and was not included in the normal release. The NET-FAQ and subsequent NET-2-HOWTO's described the then fairly complex procedure to get it all working. Fred's focus was on developing innovations to the standard network implementations and this was taking time. The community of users was growing impatient for something that worked reliably and satisfied the 80% of users and, as with Ross, the pressure on Fred as lead developer rose.

Alan Cox `<iialan@www.uk.linux.org>` proposed a solution to the problem designed to resolve the situation. He proposed that he would take Fred's NET-2 code and debug it, making it reliable and stable so that it would satisfy the impatient user base while relieving that pressure from Fred allowing him to continue his work. Alan set about doing this, with some good success and his first version of Linux networking code was called `Net-2D(ebugged)'. The code worked reliably in many typical configurations and the user base was happy. Alan clearly had ideas and skills of his own to contribute to the project and many discussions relating to the direction the NET-2 code was heading ensued. There developed two distinct schools within the Linux networking community, one that had the philosophy of `make it work first, then make it better' and the other of `make it better first'. Linus ultimately arbitrated and offered his support to Alan's development efforts and included Alan's code in the standard kernel source distribution. This placed Fred in a difficult position. Any continued development would lack the large user base actively using and testing the code and this would mean progress would be slow and difficult. Fred continued to work for a short time and eventually stood down and Alan came to be the new leader of the Linux networking kernel development effort.

Donald Becker `<becker@cesdis.gsfc.nasa.gov>` soon revealed his talents in the low level aspects of networking and produced a huge range of ethernet drivers, nearly all of those included in the current kernels were developed by Donald. There have been other people that have made significant contributions, but Donald's work is prolific and so warrants special mention.

Alan continued refining the NET-2-Debugged code for some time while working on progressing some of the matters that remained unaddressed on the `TODO' list. By the time the Linux `1.3.*` kernel source had grown its teeth the kernel networking code had migrated to the NET-3 release on which current versions are based. Alan worked on many different aspects of the networking code and with the assistance of a range of other talented people from the Linux networking community grew the code in all sorts of directions. Alan produced dynamic network devices and the first standard AX.25 and IPX implementations. Alan has continued tinkering with the code, slowly restructuring and enhancing it to the state it is in today.

PPP support was added by Michael Callahan `<callahan@maths.ox.ac.uk>` and Al Longyear `<longyear@netcom.com>` this too was critical to increasing the number of people actively using linux for networking.

Jonathon Naylor `<jsn@cs.nott.ac.uk>` has contributed by significantly enhancing Alan's AX.25 code, adding NetRom and Rose protocol support. The AX.25/NetRom/Rose support itself is quite significant, because no other operating system can boast standard native support for these protocols beside Linux.

There have of course been hundreds of other people who have made significant contribution to the development of the Linux networking software. Some of these you will encounter later in the techno specific sections, other people have contributed modules, drivers, bug-fixes, suggestions, test reports and moral support. In all cases each can claim to have played a part and offered what they could. The Linux kernel networking code is an excellent example of the results that can be obtained from the Linux style of anarchic development, if it hasn't yet surprised you, it is bound to soon enough, the development hasn't stopped.

4.2 Linux Networking Resources.

There are a number of places where you can find good information about Linux networking.

There are a wealth of Consultants available. A listing can be found at LinuxPorts Consultants Database

Alan Cox, the current maintainer of the Linux kernel networking code maintains a world wide web page that contains highlights of current and new developments in linux Networking at: www.uk.linux.org.

Another good place is a book written by Olaf Kirch entitled the `Network Administrators Guide`. It is a work of the Linux Documentation Project and you can read it interactively at Network Administrators Guide HTML version or you can obtain it in various formats by ftp from the metalab.unc.edu LDP ftp archive. Olaf's book is quite comprehensive and provides a good high level overview of network configuration under linux.

There is a newsgroup in the Linux news hierarchy dedicated to networking and related matters, it is: comp.os.linux.networking

There is a mailing list to which you can subscribe where you may ask questions relating to Linux networking. To subscribe you should send a mail message:

```
To: majordomo@vger.rutgers.edu
Subject: anything at all
Message:

subscribe linux-net
```

On the various IRC networks there are often `#linux` channels on which people will be able to answer questions on linux networking.

Please remember when reporting any problem to include as much relevant detail about the problem as you can. Specifically you should specify the versions of software that you are using, especially the kernel version, the version of tools such as *pppd* or *dip* and the exact nature of the problem you are experiencing. This means taking note of the exact syntax of any error messages you receive and of any commands that you are issuing.

4.3 Where to get some non-linux-specific network information.

TOC

If you are after some basic tutorial information on tcp/ip networking generally, then I recommend you take a look at the following documents:

tcp/ip introduction

this document comes as both a text version and a postscript version.

tcp/ip administration

this document comes as both a text version and a postscript version.

If you are after some more detailed information on tcp/ip networking then I highly recommend:

Internetworking with TCP/IP, Volume 1: principles, protocols and architecture, by Douglas E. Comer, ISBN 0-13-227836-7, Prentice Hall publications, Third Edition, 1995.

If you are wanting to learn about how to write network applications in a Unix compatible environment then I also highly recommend:

Unix Network Programming, by W. Richard Stevens, ISBN 0-13-949876-1, Prentice Hall publications, 1990.

A second edition of this book is appearing on the bookshelves; the new book is made up of three volumes: check Prenice-Hall's web site to probe further.

You might also try the comp.protocols.tcp-ip newsgroup.

An important source of specific technical information relating to the Internet and the tcp/ip suite of protocols are RFC's. RFC is an acronym for `Request For Comment' and is the standard means of submitting and documenting Internet protocol standards. There are many RFC repositories. Many of these sites are ftp sites and other provide World Wide Web access with an associated search engine that allows you to search the RFC database for particular keywords.

One possible source for RFC's is at Nexor RFC database.

---

5. Generic Network Configuration Information.

The following subsections you will pretty much need to know and understand before you actually try to configure your network. They are fundamental principles that apply regardless of the exact nature of the network you wish to deploy.

5.1 What do I need to start ?

Before you start building or configuring your network you will need some things. The most important of these are:

Current Kernel source(Optional).

Please note:

The majority of current distributions come with networking enabled, therefore it may not be required to recompile the kernel. If you are running well known hardware you should be just fine. For example: 3COM NIC, NE2000 NIC, or a Intel NIC. However if you find yourself in the position that you do need to update the kernel, the following information is provided.

Because the kernel you are running now might not yet have support for the network types or cards that you wish to use you will probably need the kernel source so that you can recompile the kernel with the appropriate options.

For users of the major distributions such as Redhat, Caldera, Debian, or Suse this no longer holds true. As long as you stay within the mainstream of hardware there should be no need to recompile your kernel unless there is a very specific feature that you need.

You can always obtain the latest kernel source from ftp.cdrom.com. This is not the official site but they have LOTS of bandwidth and ALOT of users allowed. The official site is kernel.org but please use the above if you can. Please remember that ftp.kernel.org is seriously overloaded. Use a mirror.

Normally the kernel source will be untarred into the `/usr/src/linux` directory. For information on how to apply patches and build the kernel you should read the Kernel-HOWTO. For information on how to configure kernel modules you should read the ``Modules mini-HOWTO''. Also, the `README` file found in the kernel sources and the `Documentation` directory are very informative for the brave reader.

Unless specifically stated otherwise, I recommend you stick with the standard kernel release (the one with the even number as the second digit in the version number). Development release kernels (the ones with the odd second digit) may have structural or other changes that may cause problems working with the other software on your system. If you are uncertain that you could resolve those sorts of problems in addition to the potential for there being other software errors, then don't use them.

On the other hand, some of the features described here have been introduced during the development of 2.1 kernels, so you must take your choice: you can stick to 2.0 while wait for 2.2 and an updated distribution with every new tool, or you can get 2.1 and look around for the various support programs needed to exploit the new features. As I write this paragraph, in August 1998, 2.1.115 is current and 2.2 is expected to appear pretty soon.

Current Network tools.

The network tools are the programs that you use to configure linux network devices. These tools allow you to assign addresses to devices and configure routes for example.

Most modern linux distributions are supplied with the network tools, so if you have installed from a distribution and haven't yet installed the network tools then you should do so.

If you haven't installed from a distribution then you will need to source and compile the tools yourself. This isn't difficult.

The network tools are now maintained by Bernd Eckenfels and are available at: ftp.inka.de and are mirrored at: ftp.uk.linux.org.

You can also get the latest RedHat packages from net-tools-1.51-3.i386.rpm

Be sure to choose the version that is most appropriate for the kernel you wish to use and follow the instructions in the package to

install.

To install and configure the version current at the time of the writing you need do the following:

```
user% tar xvfz net-tools-1.33.tar.gz
user% cd net-tools-1.33
user% make config
user% make
root# make install
```

Or to use the Redhat packahges:

```
root# rpm -U net-tools-1.51-3.i386.rpm
```

Additionally, if you intend configuring a firewall or using the IP masquerade feature you will require the *ipfwadm* command. The latest version of it may be obtained from: ftp.xos.nl. Again there are a number of versions available. Be sure to pick the version that most closely matches your kernel. Note that the firewalling features of Linux changed during 2.1 development and has been superceded by *ipchains* in v2.2 of the kernel. *ipfwadm* only applies to version 2.0 of the kernel. The following are known to be distributions with version 2.0 or below of the kernel.

```
Redhat 5.2 or below
Caldera pre version 2.2
Slackware pre version 4.x
Debian pre version 2.x
```

To install and configure the version current at the time of this writing you need to read the IPChains howto located at The Linux Documentation Project

Note that if you run version 2.2 (or late 2.1) of the kernel, *ipfwadm* is not the right tool to configure firewalling. This version of the NET-3-HOWTO currently doesn't deal with the new firewalling setup. If you need more detailed information on ipchains please refer to the above.

Network Application Programs.

The network application programs are programs such as *telnet* and *ftp* and their respective server programs. David Holland has been managing a distribution of the most common of these, which is now maintained by `netbug@ftp.uk.linux.org`. You may obtain the distribution from: ftp.uk.linux.org.

IP Addresses, an Explanation.

Internet Protocol Addresses are composed of four bytes. The convention is to write addresses in what is called `dotted decimal notation'. In this form each byte is converted to a decimal number (0-255) dropping any leading zero's unless the number is zero and written with each byte separated by a `.' character. By convention each interface of a host or router has an IP address. It is legal for the same IP address to be used on each interface of a single machine in some circumstances but usually each interface will have its own address.

Internet Protocol Networks are contiguous sequences of IP addresses. All addresses within a network have a number of digits within the address in common. The portion of the address that is common amongst all addresses within the network is called the `network portion' of the address. The remaining digits are called the `host portion'. The number of bits that are shared by all addresses within a network is called the netmask and it is role of the netmask to determine which addresses belong to the network it is applied to and which don't. For example, consider the following:

```
-----------------  ---------------
Host Address       192.168.110.23
Network Mask       255.255.255.0
```

```
Network Portion    192.168.110.
Host portion                  .23
-----------------  ---------------
Network Address    192.168.110.0
Broadcast Address  192.168.110.255
-----------------  ---------------
```

Any address that is 'bitwise anded' with its netmask will reveal the address of the network it belongs to. The network address is therefore always the lowest numbered address within the range of addresses on the network and always has the host portion of the address coded all zeroes.

The broadcast address is a special address that every host on the network listens to in addition to its own unique address. This address is the one that datagrams are sent to if every host on the network is meant to receive it. Certain types of data like routing information and warning messages are transmitted to the broadcast address so that every host on the network can receive it simultaneously. There are two commonly used standards for what the broadcast address should be. The most widely accepted one is to use the highest possible address on the network as the broadcast address. In the example above this would be `192.168.110.255`. For some reason other sites have adopted the convention of using the network address as the broadcast address. In practice it doesn't matter very much which you use but you must make sure that every host on the network is configured with the same broadcast address.

For administrative reasons some time early in the development of the IP protocol some arbitrary groups of addresses were formed into networks and these networks were grouped into what are called classes. These classes provide a number of standard size networks that could be allocated. The ranges allocated are:

```
-------------------------------------------------------
| Network | Netmask       | Network Addresses           |
| Class   |               |                             |
-------------------------------------------------------
|    A    | 255.0.0.0     | 0.0.0.0    - 127.255.255.255 |
|    B    | 255.255.0.0   | 128.0.0.0  - 191.255.255.255 |
|    C    | 255.255.255.0 | 192.0.0.0  - 223.255.255.255 |
|Multicast| 240.0.0.0     | 224.0.0.0  - 239.255.255.255 |
-------------------------------------------------------
```

What addresses you should use depends on exactly what it is that you are doing. You may have to use a combination of the following activities to get all the addresses you need:

Installing a linux machine on an existing IP network

If you wish to install a linux machine onto an existing IP network then you should contact whoever administers the network and ask them for the following information:

- Host IP Address
- IP network address
- IP broadcast address
- IP netmask
- Router address
- Domain Name Server Address

You should then configure your linux network device with those details. You can not make them up and expect your configuration to work.

Building a brand new network that will never connect to the Internet

If you are building a private network and you never intend that network to be connected to the Internet then you can choose whatever addresses you like. However, for safety and consistency reasons there have been some IP network addresses that have been reserved specifically for this purpose. These are specified in RFC1597 and are as follows:

```
-------------------------------------------------------
|          RESERVED PRIVATE NETWORK ALLOCATIONS       |
```

TOC

```
-------------------------------------------------------------
| Network | Netmask       | Network Addresses                |
| Class   |               |                                  |
-------------------------------------------------------------
|    A    | 255.0.0.0     | 10.0.0.0    - 10.255.255.255     |
|    B    | 255.255.0.0   | 172.16.0.0  - 172.31.255.255     |
|    C    | 255.255.255.0 | 192.168.0.0 - 192.168.255.255    |
-------------------------------------------------------------
```

You should first decide how large you want your network to be and then choose as many of the addresses as you require.

5.2 Where should I put the configuration commands ?

There are a few different approaches to Linux system boot procedures. After the kernel boots, it always executes a program called `*init*'. The *init* program then reads its configuration file called `/etc/inittab` and commences the boot process. There are a few different flavours of *init* around, although everyone is now converging to the System V (Five) flavor, developed by Miguel van Smoorenburg.

Despite the fact that the *init* program is always the same, the setup of system boot is organized in a different way by each distribution.

Usually the `/etc/inittab` file contains an entry looking something like:

```
si::sysinit:/etc/init.d/boot
```

This line specifies the name of the shell script file that actually manages the boot sequence. This file is somewhat equivalent to the `AUTOEXEC.BAT` file in MS-DOS.

There are usually other scripts that are called by the boot script and often the network is configured within one of many of these.

The following table may be used as a guide for your system:

```
-------------------------------------------------------------------------
Distrib. | Interface Config/Routing          | Server Initialization
-------------------------------------------------------------------------
Debian   | /etc/init.d/network               | /etc/rc2.d/*
-------------------------------------------------------------------------
Slackware| /etc/rc.d/rc.inet1                | /etc/rc.d/rc.inet2
-------------------------------------------------------------------------
RedHat   | /etc/rc.d/init.d/network          | /etc/rc.d/rc3.d/*
-------------------------------------------------------------------------
```

Note that Debian and Red Hat use a whole directory to host scripts that fire up system services (and usually information does not lie within these files, for example Red Hat systems store all of system configuration in files under `/etc/sysconfig`, whence it is retrieved by boot scripts). If you want to grasp the details of the boot process, my suggestion is to check */etc/inittab* and the documentation that accompanies *init*. Linux Journal is also going to publish an article about system initialization, and this document will point to it as soon as it is available on the web.

Most modern distributions include a program that will allow you to configure many of the common sorts of network interfaces. If you have one of these then you should see if it will do what you want before attempting a manual configuration.

```
-----------------------------------------
Distrib   | Network configuration program
-----------------------------------------
RedHat    | /usr/bin/netcfg
Slackware | /sbin/netconfig
-----------------------------------------
```

TOC

5.3 Creating your network interfaces.

In many Unix operating systems the network devices have appearances in the */dev* directory. This is not so in Linux. In Linux the network devices are created dynamically in software and do not require device files to be present.

In the majority of cases the network device is automatically created by the device driver while it is initializing and has located your hardware. For example, the ethernet device driver creates `eth[0..n]` interfaces sequentially as it locates your ethernet hardware. The first ethernet card found becomes `eth0`, the second `eth1` etc.

In some cases though, notably *slip* and *ppp*, the network devices are created through the action of some user program. The same sequential device numbering applies, but the devices are not created automatically at boot time. The reason for this is that unlike ethernet devices, the number of active *slip* or *ppp* devices may vary during the uptime of the machine. These cases will be covered in more detail in later sections.

5.4 Configuring a network interface.

When you have all of the programs you need and your address and network information you can configure your network interfaces. When we talk about configuring a network interface we are talking about the process of assigning appropriate addresses to a network device and to setting appropriate values for other configurable parameters of a network device. The program most commonly used to do this is the *ifconfig* (interface configure) command.

Typically you would use a command similar to the following:

```
root# ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up
```

In this case I'm configuring an ethernet interface `` `eth0' `` with the IP address `` `192.168.0.1' `` and a network mask of `` `255.255.255.0' ``. The `` `*up*' `` that trails the command tells the interface that it should become active, but can usually be omitted, as it is the default. To shutdown an interface, you can just call ``` ``ifconfig eth0 down'' ```.

The kernel assumes certain defaults when configuring interfaces. For example, you may specify the network address and broadcast address for an interface, but if you don't, as in my example above, then the kernel will make reasonable guesses as to what they should be based on the netmask you supply and if you don't supply a netmask then on the network class of the IP address configured. In my example the kernel would assume that it is a class-C network being configured on the interface and configure a network address of `` `192.168.0.0' `` and a broadcast address of `` `192.168.0.255' `` for the interface.

There are many other options to the *ifconfig* command. The most important of these are:

up

this option activates an interface (and is the default).

down

this option deactivates an interface.

[-]arp

this option enables or disables use of the address resolution protocol on this interface

[-]allmulti

this option enables or disables the reception of all hardware multicast packets. Hardware multicast enables groups of hosts to receive packets addressed to special destinations. This may be of importance if you are using applications like desktop videoconferencing but is normally not used.

mtu N

this parameter allows you to set the *MTU* of this device.

netmask <addr>

this parameter allows you to set the network mask of the network this device belongs to.

irq <addr>

TOC

this parameter only works on certain types of hardware and allows you to set the IRQ of the hardware of this device.

[-]broadcast [addr]

this parameter allows you to enable and set the accepting of datagrams destined to the broadcast address, or to disable reception of these datagrams.

[-]pointopoint [addr]

this parameter allows you to set the address of the machine at the remote end of a point to point link such as for *slip* or *ppp*.

hw <type> <addr>

this parameter allows you to set the hardware address of certain types of network devices. This is not often useful for ethernet, but is useful for other network types such as AX.25.

You may use the *ifconfig* command on any network interface. Some user programs such as *pppd* and *dip* automatically configure the network devices as they create them, so manual use of *ifconfig* is unnecessary.

5.5 Configuring your Name Resolver.

The `*Name Resolver*' is a part of the linux standard library. Its prime function is to provide a service to convert human-friendly hostnames like ``ftp.funet.fi'` into machine friendly IP addresses such as `128.214.248.6`.

What's in a name ?

You will probably be familiar with the appearance of Internet host names, but may not understand how they are constructed, or deconstructed. Internet domain names are hierarchical in nature, that is, they have a tree-like structure. A `*domain*' is a family, or group of names. A `*domain*' may be broken down into `*subdomain*'. A `*toplevel domain*' is a domain that is not a subdomain. The Top Level Domains are specified in RFC-920. Some examples of the most common top level domains are:

COM

Commercial Organizations

EDU

Educational Organizations

GOV

Government Organizations

MIL

Military Organizations

ORG

Other organizations

NET

Internet-Related Organizations

Country Designator

these are two letters codes that represent a particular country.

For historical reasons most domains belonging to one of the non-country based top level domains were used by organizations within the United States, although the United States also has its own country code `.us'. This is not true any more for `.com` and `.org` domains, which are commonly used by non-us companies.

Each of these top level domains has subdomains. The top level domains based on country name are often next broken down into subdomains based on the `com`, `edu`, `gov`, `mil` and `org` domains. So for example you end up with: `com.au` and `gov.au` for commercial and government organizations in Australia; note that this is not a general rule, as actual policies depend on the naming authority for each domain.

The next level of division usually represents the name of the organization. Further subdomains vary in nature, often the next level of subdomain is based on the departmental structure of the organization but it may be based on any criterion considered reasonable and

TOC

meaningful by the network administrators for the organization.

The very left-most portion of the name is always the unique name assigned to the host machine and is called the `*hostname*', the portion of the name to the right of the hostname is called the `*domainname*' and the complete name is called the `*Fully Qualified Domain Name*'.

To use Terry's host as an example, the fully qualified domain name is ``perf.no.itg.telstra.com.au`'. This means that the host name is ``perf`' and the domain name is ``no.itg.telstra.com.au`'. The domain name is based on a top level domain based on his country, Australia and as his email address belongs to a commercial organization, `.com' is there as the next level domain. The name of the company is (was) `telstra' and their internal naming structure is based on organizational structure, in this case the machine belongs to the Information Technology Group, Network Operations section.

Usually, the names are fairly shorter; for example, my ISP is called ``systemy.it" and my non-profit organization is called ``linux.it", without any `com` and `org` subdomain, so that my own host is just called ``morgana.systemy.it" and `rubini@linux.it` is a valid email address. Note that the owner of a domain has the rights to register hostnames as well as subdomains; for example, the LUG I belong to uses the domain `pluto.linux.it`, because the owners of `linux.it` agreed to open a subdomain for the LUG.

What information you will need.

You will need to know what domain your hosts name will belong to. The name resolver software provides this name translation service by making requests to a `*Domain Name Server*', so you will need to know the IP address of a local nameserver that you can use.

There are three files you need to edit, I'll cover each of these in turn.

/etc/resolv.conf

The `/etc/resolv.conf` is the main configuration file for the name resolver code. Its format is quite simple. It is a text file with one keyword per line. There are three keywords typically used, they are:

domain

this keyword specifies the local domain name.

search

this keyword specifies a list of alternate domain names to search for a hostname

nameserver

this keyword, which may be used many times, specifies an IP address of a domain name server to query when resolving names

An example `/etc/resolv.conf` might look something like:

```
domain maths.wu.edu.au
search maths.wu.edu.au wu.edu.au
nameserver 192.168.10.1
nameserver 192.168.12.1
```

This example specifies that the default domain name to append to unqualified names (ie hostnames supplied without a domain) is `maths.wu.edu.au` and that if the host is not found in that domain to also try the `wu.edu.au` domain directly. Two nameservers entry are supplied, each of which may be called upon by the name resolver code to resolve the name.

/etc/host.conf

The `/etc/host.conf` file is where you configure some items that govern the behaviour of the name resolver code. The format of this file is described in detail in the `resolv+' man page. In nearly all circumstances the following example will work for you:

```
order hosts,bind
multi on
```

TOC

This configuration tells the name resolver to check the `/etc/hosts` file before attempting to query a nameserver and to return all valid addresses for a host found in the `/etc/hosts` file instead of just the first.

/etc/hosts

The `/etc/hosts` file is where you put the name and IP address of local hosts. If you place a host in this file then you do not need to query the domain name server to get its IP Address. The disadvantage of doing this is that you must keep this file up to date yourself if the IP address for that host changes. In a well managed system the only hostnames that usually appear in this file are an entry for the loopback interface and the local hosts name.

```
# /etc/hosts
127.0.0.1       localhost loopback
192.168.0.1     this.host.name
```

You may specify more than one host name per line as demonstrated by the first entry, which is a standard entry for the loopback interface.

Running a name server

If you want to run a local nameserver, you can do it easily. Please refer to the DNS-HOWTO and to any documents included in your version of *BIND* (Berkeley Internet Name Domain).

5.6 Configuring your loopback interface.

The `loopback' interface is a special type of interface that allows you to make connections to yourself. There are various reasons why you might want to do this, for example, you may wish to test some network software without interfering with anybody else on your network. By convention the IP address `127.0.0.1' has been assigned specifically for loopback. So no matter what machine you go to, if you open a telnet connection to `127.0.0.1` you will always reach the local host.

Configuring the loopback interface is simple and you should ensure you do (but note that this task is usually performed by the standard initialization scripts).

```
root# ifconfig lo 127.0.0.1
root# route add -host 127.0.0.1 lo
```

We'll talk more about the *route* command in the next section.

5.7 Routing.

Routing is a big topic. It is easily possible to write large volumes of text about it. Most of you will have fairly simple routing requirements, some of you will not. I will cover some basic fundamentals of routing only. If you are interested in more detailed information then I suggest you refer to the references provided at the start of the document.

Let's start with a definition. What is IP routing ? Here is one that I'm using:

> IP Routing is the process by which a host with multiple network connections decides where to deliver IP datagrams it has received.

It might be useful to illustrate this with an example. Imagine a typical office router, it might have a PPP link off the Internet, a number of ethernet segments feeding the workstations and another PPP link off to another office. When the router receives a datagram on any of its network connections, routing is the mechanism that it uses to determine which interface it should send the datagram to next. Simple hosts also need to route, all Internet hosts have two network devices, one is the loopback interface described above and the other is the one it uses to talk to the rest of the network, perhaps an ethernet, perhaps a PPP or SLIP serial interface.

Ok, so how does routing work ? Each host keeps a special list of routing rules, called a routing table. This table contains rows w typically contain at least three fields, the first is a destination address, the second is the name of the interface to which the data to be routed and the third is optionally the IP address of another machine which will carry the datagram on its next step through the network. In linux you can see this table by using the following command:

```
user% cat /proc/net/route
```

or by using either of the following commands:

```
user% /sbin/route -n
user% netstat -r
```

The routing process is fairly simple: an incoming datagram is received, the destination address (who it is for) is examined and compared with each entry in the table. The entry that best matches that address is selected and the datagram is forwarded to the specified interface. If the gateway field is filled then the datagram is forwarded to that host via the specified interface, otherwise the destination address is assumed to be on the network supported by the interface.

To manipulate this table a special command is used. This command takes command line arguments and converts them into kernel system calls that request the kernel to add, delete or modify entries in the routing table. The command is called `*route*'.

A simple example. Imagine you have an ethernet network. You've been told it is a class-C network with an address of `192.168.1.0`. You've been supplied with an IP address of `192.168.1.10` for your use and have been told that `192.168.1.1` is a router connected to the Internet.

The first step is to configure the interface as described earlier. You would use a command like:

```
root# ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
```

You now need to add an entry into the routing table to tell the kernel that datagrams for all hosts with addresses that match `192.168.1.*` should be sent to the ethernet device. You would use a command similar to:

```
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
```

Note the use of the `-net' argument to tell the route program that this entry is a network route. Your other choice here is a `-host' route which is a route that is specific to one IP address.

This route will enable you to establish IP connections with all of the hosts on your ethernet segment. But what about all of the IP hosts that aren't on your ethernet segment ?

It would be a very difficult job to have to add routes to every possible destination network, so there is a special trick that is used to simplify this task. The trick is called the `default' route. The `default` route matches every possible destination, but poorly, so that if any other entry exists that matches the required address it will be used instead of the `default` route. The idea of the `default` route is simply to enable you to say "and everything else should go here". In the example I've contrived you would use an entry like:

```
root# route add default gw 192.168.1.1 eth0
```

The `gw' argument tells the route command that the next argument is the IP address, or name, of a gateway or router machine which all datagrams matching this entry should be directed to for further routing.

So, your complete configuration would look like:

```
root# ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
root# route add default gw 192.168.1.1 eth0
```

TOC

If you take a close look at your network `rc' files you will find that at least one of them looks very similar to this. This is a very common configuration.

Let's now look at a slightly more complicated routing configuration. Let's imagine we are configuring the router we looked at earlier, the one supporting the PPP link to the Internet and the lan segments feeding the workstations in the office. Lets imagine the router has three ethernet segments and one PPP link. Our routing configuration would look something like:

```
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
root# route add -net 192.168.2.0 netmask 255.255.255.0 eth1
root# route add -net 192.168.3.0 netmask 255.255.255.0 eth2
root# route add default ppp0
```

Each of the workstations would use the simpler form presented above, only the router needs to specify each of the network routes separately because for the workstations the `default` route mechanism will capture all of them letting the router worry about splitting them up appropriately. You may be wondering why the default route presented doesn't specify a `gw'. The reason for this is simple, serial link protocols such as PPP and slip only ever have two hosts on their network, one at each end. To specify the host at the other end of the link as the gateway is pointless and redundant as there is no other choice, so you do not need to specify a gateway for these types of network connections. Other network types such as ethernet, arcnet or token ring do require the gateway to be specified as these networks support large numbers of hosts on them.

So what does the *routed* program do ?

The routing configuration described above is best suited to simple network arrangements where there are only ever single possible paths to destinations. When you have a more complex network arrangement things get a little more complicated. Fortunately for most of you this won't be an issue.

The big problem with `manual routing' or `static routing' as described, is that if a machine or link fails in your network then the only way you can direct your datagrams another way, if another way exists, is by manually intervening and executing the appropriate commands. Naturally this is clumsy, slow, impractical and hazard prone. Various techniques have been developed to automatically adjust routing tables in the event of network failures where there are alternate routes, all of these techniques are loosely grouped by the term `dynamic routing protocols'.

You may have heard of some of the more common dynamic routing protocols. The most common are probably RIP (Routing Information Protocol) and OSPF (Open Shortest Path First Protocol). The Routing Information Protocol is very common on small networks such as small-medium sized corporate networks or building networks. OSPF is more modern and more capable at handling large network configurations and better suited to environments where there is a large number of possible paths through the network. Common implementations of these protocols are: `*routed*' - RIP and `*gated*' - RIP, OSPF and others. The `*routed*' program is normally supplied with your Linux distribution or is included in the `NetKit' package detailed above.

An example of where and how you might use a dynamic routing protocol might look something like the following:

```
     192.168.1.0 /                          192.168.2.0 /
        255.255.255.0                          255.255.255.0
      -                                      -
      |                                      |
      |   /-----\                  /-----\   |
      |   |     |ppp0   //    ppp0|     |   |
eth0  |---|  A  |------//---------|  B  |---| eth0
      |   |     |     //          |     |   |
      |   \-----/                  \-----/   |
      |      \ ppp1             ppp1 /       |
      -       \                     /        -
               \                   /
                \                 /
                 \               /
                  \             /
```

```
                    \           /
                     \         /
                      \       /
                       \     /
                    ppp0\   /ppp1
                       /-----\
                       |     |
                       |  C  |
                       |     |
                       \-----/
                          |eth0
                          |
                     |---------|
                     192.168.3.0 /
                        255.255.255.0
```

We have three routers A, B and C. Each supports one ethernet segment with a Class C IP network (netmask 255.255.255.0). Each router also has a PPP link to each of the other routers. The network forms a triangle.

It should be clear that the routing table at router A could look like:

```
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
root# route add -net 192.168.2.0 netmask 255.255.255.0 ppp0
root# route add -net 192.168.3.0 netmask 255.255.255.0 ppp1
```

This would work just fine until the link between router A and B should fail. If that link failed then with the routing entry shown above hosts on the ethernet segment of A could not reach hosts on the ethernet segment on B because their datagram would be directed to router A's ppp0 link which is broken. They could still continue to talk to hosts on the ethernet segment of C and hosts on the C's ethernet segment could still talk to hosts on B's ethernet segment because the link between B and C is still intact.

But wait, if A can talk to C and C can still talk to B, why shouldn't A route its datagrams for B via C and let C send them to B ? This is exactly the sort of problem that dynamic routing protocols like RIP were designed to solve. If each of the routers A, B and C were running a routing daemon then their routing tables would be automatically adjusted to reflect the new state of the network should any one of the links in the network fail. To configure such a network is simple, at each router you need only do two things. In this case for Router A:

```
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
root# /usr/sbin/routed
```

The `*routed*' routing daemon automatically finds all active network ports when it starts and sends and listens for messages on each of the network devices to allow it to determine and update the routing table on the host.

This has been a very brief explanation of dynamic routing and where you would use it. If you want more information then you should refer to the suggested references listed at the top of the document.

The important points relating to dynamic routing are:

1. You only need to run a dynamic routing protocol daemon when your Linux machine has the possibility of selecting multiple possible routes to a destination. An example of this would be if you plan to use IP Masquerading.
2. The dynamic routing daemon will automatically modify your routing table to adjust to changes in your network.
3. RIP is suited to small to medium sized networks.

5.8 Configuring your network servers and services.

Network servers and services are those programs that allow a remote user to make user of your Linux machine. Server programs listen on network ports. Network ports are a means of addressing a particular service on any particular host and are how a server knows the difference between an incoming telnet connection and an incoming ftp connection. The remote user establishes a network

TOC

connection to your machine and the server program, the network daemon program, listening on that port accepts the connection executes. There are two ways that network daemons may operate. Both are commonly employed in practice. The two ways are:

standalone

the network daemon program listens on the designated network port and when an incoming connection is made it manages the network connection itself to provide the service.

slave to the *inetd* server

the *inetd* server is a special network daemon program that specializes in managing incoming network connections. It has a configuration file which tells it what program needs to be run when an incoming connection is received. Any service port may be configured for either of the tcp or udp protcols. The ports are described in another file that we will talk about soon.

There are two important files that we need to configure. They are the `/etc/services` file which assigns names to port numbers and the `/etc/inetd.conf` file which is the configuration file for the *inetd* network daemon.

`/etc/services`

The `/etc/services` file is a simple database that associates a human friendly name to a machine friendly service port. Its format is quite simple. The file is a text file with each line representing and entry in the database. Each entry is comprised of three fields separated by any number of whitespace (tab or space) characters. The fields are:

```
name      port/protocol        aliases     # comment
```

name

a single word name that represents the service being described.

port/protocol

this field is split into two subfields.

port

a number that specifies the port number the named service will be available on. Most of the common services have assigned service numbers. These are described in `RFC-1340`.

protocol

this subfield may be set to either `tcp` or `udp`. It is important to note that an entry of `18/tcp` is very different from an entry of `18/udp` and that there is no technical reason why the same service needs to exist on both. Normally common sense prevails and it is only if a particular service is available via both `tcp` and `udp` that you will see an entry for both.

aliases

other names that may be used to refer to this service entry.

Any text appearing in a line after a `#' character is ignored and treated as a comment.

An example `/etc/services` file.

All modern linux distributions provide a good `/etc/services` file. Just in case you happen to be building a machine from the ground up, here is a copy of the `/etc/services` file supplied with an old Debian distribution:

```
# /etc/services:
# $Id: services,v 1.3 1996/05/06 21:42:37 tobias Exp $
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1340, ``Assigned Numbers'' (July 1992).  Not all ports
# are included, only the more common ones.
```

TOC

```
tcpmux          1/tcp                           # TCP port service multiplexer
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
qotd            17/tcp          quote
msp             18/tcp                          # message send protocol
msp             18/udp                          # message send protocol
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
telnet          23/tcp
# 24 - private
smtp            25/tcp          mail
# 26 - unassigned
time            37/tcp          timserver
time            37/udp          timserver
rlp             39/udp          resource        # resource location
nameserver      42/tcp          name            # IEN 116
whois           43/tcp          nicname
re-mail-ck      50/tcp                          # Remote Mail Checking Protocol
re-mail-ck      50/udp                          # Remote Mail Checking Protocol
domain          53/tcp          nameserver      # name-domain server
domain          53/udp          nameserver
mtp             57/tcp                          # deprecated
bootps          67/tcp                          # BOOTP server
bootps          67/udp
bootpc          68/tcp                          # BOOTP client
bootpc          68/udp
tftp            69/udp
gopher          70/tcp                          # Internet Gopher
gopher          70/udp
rje             77/tcp          netrjs
finger          79/tcp
www             80/tcp          http            # WorldWideWeb HTTP
www             80/udp                          # HyperText Transfer Protocol
link            87/tcp          ttylink
kerberos        88/tcp          kerberos5 krb5  # Kerberos v5
kerberos        88/udp          kerberos5 krb5  # Kerberos v5
supdup          95/tcp
# 100 - reserved
hostnames       101/tcp         hostname        # usually from sri-nic
iso-tsap        102/tcp         tsap            # part of ISODE.
csnet-ns        105/tcp         cso-ns          # also used by CSO name server
csnet-ns        105/udp         cso-ns
rtelnet         107/tcp                         # Remote Telnet
rtelnet         107/udp
pop-2           109/tcp         postoffice      # POP version 2
pop-2           109/udp
pop-3           110/tcp                         # POP version 3
```

TOC

```
pop-3           110/udp
sunrpc          111/tcp         portmapper      # RPC 4.0 portmapper TCP
sunrpc          111/udp         portmapper      # RPC 4.0 portmapper UDP
auth            113/tcp         authentication tap ident
sftp            115/tcp
uucp-path       117/tcp
nntp            119/tcp         readnews untp   # USENET News Transfer Protocol
ntp             123/tcp
ntp             123/udp                         # Network Time Protocol
netbios-ns      137/tcp                         # NETBIOS Name Service
netbios-ns      137/udp
netbios-dgm     138/tcp                         # NETBIOS Datagram Service
netbios-dgm     138/udp
netbios-ssn     139/tcp                         # NETBIOS session service
netbios-ssn     139/udp
imap2           143/tcp                         # Interim Mail Access Proto v2
imap2           143/udp
snmp            161/udp                         # Simple Net Mgmt Proto
snmp-trap       162/udp         snmptrap        # Traps for SNMP
cmip-man        163/tcp                         # ISO mgmt over IP (CMOT)
cmip-man        163/udp
cmip-agent      164/tcp
cmip-agent      164/udp
xdmcp           177/tcp                         # X Display Mgr. Control Proto
xdmcp           177/udp
nextstep        178/tcp         NeXTStep NextStep       # NeXTStep window
nextstep        178/udp         NeXTStep NextStep       # server
bgp             179/tcp                         # Border Gateway Proto.
bgp             179/udp
prospero        191/tcp                         # Cliff Neuman's Prospero
prospero        191/udp
irc             194/tcp                         # Internet Relay Chat
irc             194/udp
smux            199/tcp                         # SNMP Unix Multiplexer
smux            199/udp
at-rtmp         201/tcp                         # AppleTalk routing
at-rtmp         201/udp
at-nbp          202/tcp                         # AppleTalk name binding
at-nbp          202/udp
at-echo         204/tcp                         # AppleTalk echo
at-echo         204/udp
at-zis          206/tcp                         # AppleTalk zone information
at-zis          206/udp
z3950           210/tcp         wais            # NISO Z39.50 database
z3950           210/udp         wais
ipx             213/tcp                         # IPX
ipx             213/udp
imap3           220/tcp                         # Interactive Mail Access
imap3           220/udp                         # Protocol v3
ulistserv       372/tcp                         # UNIX Listserv
ulistserv       372/udp
#
# UNIX specific services
#
exec            512/tcp
biff            512/udp         comsat
login           513/tcp
who             513/udp         whod
```

TOC

```
shell           514/tcp         cmd             # no passwords used
syslog          514/udp
printer         515/tcp         spooler         # line printer spooler
talk            517/udp
ntalk           518/udp
route           520/udp         router routed   # RIP
timed           525/udp         timeserver
tempo           526/tcp         newdate
courier         530/tcp         rpc
conference      531/tcp         chat
netnews         532/tcp         readnews
netwall         533/udp                         # -for emergency broadcasts
uucp            540/tcp         uucpd           # uucp daemon
remotefs        556/tcp         rfs_server rfs  # Brunhoff remote filesystem
klogin          543/tcp                         # Kerberized `rlogin' (v5)
kshell          544/tcp         krcmd           # Kerberized `rsh' (v5)
kerberos-adm    749/tcp                         # Kerberos `kadmin' (v5)
#
webster         765/tcp                         # Network dictionary
webster         765/udp
#
# From ``Assigned Numbers'':
#
#> The Registered Ports are not controlled by the IANA and on most systems
#> can be used by ordinary user processes or programs executed by ordinary
#> users.
#
#> Ports are used in the TCP [45,106] to name the ends of logical
#> connections which carry long term conversations.  For the purpose of
#> providing services to unknown callers, a service contact port is
#> defined.  This list specifies the port used by the server process as its
#> contact port.  While the IANA can not control uses of these ports it
#> does register or list uses of these ports as a convenience to the
#> community.
#
ingreslock      1524/tcp
ingreslock      1524/udp
prospero-np     1525/tcp                # Prospero non-privileged
prospero-np     1525/udp
rfe             5002/tcp                # Radio Free Ethernet
rfe             5002/udp                # Actually uses UDP only
bbs             7000/tcp                # BBS service
#
#
# Kerberos (Project Athena/MIT) services
# Note that these are for Kerberos v4 and are unofficial.  Sites running
# v4 should uncomment these and comment out the v5 entries above.
#
kerberos4       750/udp         kdc     # Kerberos (server) udp
kerberos4       750/tcp         kdc     # Kerberos (server) tcp
kerberos_master 751/udp                 # Kerberos authentication
kerberos_master 751/tcp                 # Kerberos authentication
passwd_server   752/udp                 # Kerberos passwd server
krb_prop        754/tcp                 # Kerberos slave propagation
krbupdate       760/tcp         kreg    # Kerberos registration
kpasswd         761/tcp         kpwd    # Kerberos "passwd"
kpop            1109/tcp                # Pop with Kerberos
knetd           2053/tcp                # Kerberos de-multiplexor
```

```
zephyr-srv      2102/udp                # Zephyr server
zephyr-clt      2103/udp                # Zephyr serv-hm connection
zephyr-hm       2104/udp                # Zephyr hostmanager
eklogin         2105/tcp                # Kerberos encrypted rlogin
#
# Unofficial but necessary (for NetBSD) services
#
supfilesrv      871/tcp                 # SUP server
supfiledbg      1127/tcp                # SUP debugging
#
# Datagram Delivery Protocol services
#
rtmp            1/ddp                   # Routing Table Maintenance Protocol
nbp             2/ddp                   # Name Binding Protocol
echo            4/ddp                   # AppleTalk Echo Protocol
zip             6/ddp                   # Zone Information Protocol
#
# Debian GNU/Linux services
rmtcfg          1236/tcp                # Gracilis Packeten remote config server
xtel            1313/tcp                # french minitel
cfinger         2003/tcp                # GNU Finger
postgres        4321/tcp                # POSTGRES
mandelspawn     9359/udp        mandelbrot      # network mandelbrot

# Local services
```

In the real world, the actual file is always growing as new services are being created. If you fear your own copy is incomplete, I'd suggest to copy a new `/etc/services` from a recent distribution.

`/etc/inetd.conf`

The `/etc/inetd.conf` file is the configuration file for the *inetd* server daemon. Its function is to tell *inetd* what to do when it receives a connection request for a particular service. For each service that you wish to accept connections for you must tell *inetd* what network server daemon to run and how to run it.

Its format is also fairly simple. It is a text file with each line describing a service that you wish to provide. Any text in a line following a `#' is ignored and considered a comment. Each line contains seven fields separated by any number of whitespace (tab or space) characters. The general format is as follows:

```
service  socket_type  proto  flags  user  server_path  server_args
```

service

is the service relevant to this configuration as taken from the `/etc/services` file.

socket_type

this field describes the type of socket that this entry will consider relevant, allowable values are: `stream`, `dgram`, `raw`, `rdm`, or `seqpacket`. This is a little technical in nature, but as a rule of thumb nearly all `tcp` based services use `stream` and nearly all `udp` based services use `dgram`. It is only very special types of server daemons that would use any of the other values.

proto

the protocol to considered valid for this entry. This should match the appropriate entry in the `/etc/services` file and will typically be either `tcp` or `udp`. Sun RPC (Remote Procedure Call) based servers will use `rpc/tcp` or `rpc/udp`.

flags

there are really only two possible settings for this field. This field setting tells *inetd* whether the network server program frees the socket after it has been started and therefore whether *inetd* can start another one on the next connection request, or whether *inetd* should wait and assume that any server daemon already running will handle the new connection request. Again this is a

TOC

little tricky to work out, but as a rule of thumb all `tcp` servers should have this entry set to `nowait` and most `udp` servers should have this entry set to `wait`. Be warned there are some notable exceptions to this, so let the example guide you if you are not sure.

user

this field describes which user account from `/etc/passwd` will be set as the owner of the network daemon when it is started. This is often useful if you want to safeguard against security risks. You can set the user of an entry to the `nobody` user so that if the network server security is breached the possible damage is minimized. Typically this field is set to `root` though, because many servers require root privileges in order to function correctly.

server_path

this field is pathname to the actual server program to execute for this entry.

server_args

this field comprises the rest of the line and is optional. This field is where you place any command line arguments that you wish to pass to the server daemon program when it is launched.

An example `/etc/inetd.conf`

As for the `/etc/services` file all modern distributions will include a good `/etc/inetd.conf` file for you to work with. Here, for completeness is the `/etc/inetd.conf` file from the Debian distribution.

```
# /etc/inetd.conf:  see inetd(8) for further informations.
#
# Internet server configuration database
#
#
# Modified for Debian by Peter Tobias <tobias@et-inf.fho-emden.de>
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
# Internal services
#
#echo           stream  tcp     nowait  root    internal
#echo           dgram   udp     wait    root    internal
discard         stream  tcp     nowait  root    internal
discard         dgram   udp     wait    root    internal
daytime         stream  tcp     nowait  root    internal
daytime         dgram   udp     wait    root    internal
#chargen        stream  tcp     nowait  root    internal
#chargen        dgram   udp     wait    root    internal
time            stream  tcp     nowait  root    internal
time            dgram   udp     wait    root    internal
#
# These are standard services.
#
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.telnetd
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.ftpd
#fsp    dgram   udp     wait    root    /usr/sbin/tcpd  /usr/sbin/in.fspd
#
# Shell, login, exec and talk are BSD protocols.
#
shell   stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rshd
login   stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rlogind
#exec   stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rexecd
talk    dgram   udp     wait    root    /usr/sbin/tcpd  /usr/sbin/in.talkd
ntalk   dgram   udp     wait    root    /usr/sbin/tcpd  /usr/sbin/in.ntalkd
#
```

TOC

```
# Mail, news and uucp services.
#
smtp    stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.smtpd
#nntp   stream  tcp     nowait  news    /usr/sbin/tcpd  /usr/sbin/in.nntpd
#uucp   stream  tcp     nowait  uucp    /usr/sbin/tcpd  /usr/lib/uucp/uucico
#comsat dgram   udp     wait    root    /usr/sbin/tcpd  /usr/sbin/in.comsat
#
# Pop et al
#
#pop-2  stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.pop2d
#pop-3  stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.pop3d
#
# `cfinger' is for the GNU finger server available for Debian.  (NOTE: The
# current implementation of the `finger' daemon allows it to be run as `root'.)
#
#cfinger stream tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.cfingerd
#finger stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.fingerd
#netstat        stream  tcp     nowait  nobody  /usr/sbin/tcpd  /bin/netstat
#systat stream  tcp     nowait  nobody  /usr/sbin/tcpd  /bin/ps -auwwx
#
# Tftp service is provided primarily for booting.  Most sites
# run this only on machines acting as "boot servers."
#
#tftp   dgram   udp     wait    nobody  /usr/sbin/tcpd  /usr/sbin/in.tftpd
#tftp   dgram   udp     wait    nobody  /usr/sbin/tcpd  /usr/sbin/in.tftpd /boot
#bootps dgram   udp     wait    root    /usr/sbin/bootpd        bootpd -i -t 120
#
# Kerberos authenticated services (these probably need to be corrected)
#
#klogin         stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rlogind
-k
#eklogin        stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rlogind
-k -x
#kshell         stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rshd -k
#
# Services run ONLY on the Kerberos server (these probably need to be corrected)
#
#krbupdate      stream tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/registerd
#kpasswd        stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/kpasswdd
#
# RPC based services
#
#mountd/1       dgram   rpc/udp wait    root    /usr/sbin/tcpd  /usr/sbin/rpc.mountd
#rstatd/1-3     dgram   rpc/udp wait    root    /usr/sbin/tcpd  /usr/sbin/rpc.rstatd
#rusersd/2-3    dgram   rpc/udp wait    root    /usr/sbin/tcpd  /usr/sbin/rpc.rusersd
#walld/1        dgram   rpc/udp wait    root    /usr/sbin/tcpd  /usr/sbin/rpc.rwalld
#
# End of inetd.conf.
ident           stream  tcp     nowait  nobody  /usr/sbin/identd        identd -i
```

5.9 Other miscellaneous network related configuration files.

There are a number of miscellaneous files relating to network configuration under linux that you might be interested in. You may never have to modify these files, but it is worth describing them so you know what they contain and what they are for.

/etc/protocols

The `/etc/protocols` file is a database that maps protocol id numbers against protocol names. This is used by programmers to allow them to specify protocols by name in their programs and also by some programs such as *tcpdump* to allow them to display names instead of numbers in their output. The general syntax of the file is:

```
protocolname  number  aliases
```

The `/etc/protocols` file supplied with the Debian distribution is as follows:

```
# /etc/protocols:
# $Id: protocols,v 1.1 1995/02/24 01:09:41 imurdock Exp $
#
# Internet (IP) protocols
#
#       from: @(#)protocols     5.1 (Berkeley) 4/17/89
#
# Updated for NetBSD based on RFC 1340, Assigned Numbers (July 1992).

ip      0       IP              # internet protocol, pseudo protocol number
icmp    1       ICMP            # internet control message protocol
igmp    2       IGMP            # Internet Group Management
ggp     3       GGP             # gateway-gateway protocol
ipencap 4       IP-ENCAP        # IP encapsulated in IP (officially ``IP'')
st      5       ST              # ST datagram mode
tcp     6       TCP             # transmission control protocol
egp     8       EGP             # exterior gateway protocol
pup     12      PUP             # PARC universal packet protocol
udp     17      UDP             # user datagram protocol
hmp     20      HMP             # host monitoring protocol
xns-idp 22      XNS-IDP         # Xerox NS IDP
rdp     27      RDP             # "reliable datagram" protocol
iso-tp4 29      ISO-TP4         # ISO Transport Protocol class 4
xtp     36      XTP             # Xpress Tranfer Protocol
ddp     37      DDP             # Datagram Delivery Protocol
idpr-cmtp       39      IDPR-CMTP       # IDPR Control Message Transport
rspf    73      RSPF            # Radio Shortest Path First.
vmtp    81      VMTP            # Versatile Message Transport
ospf    89      OSPFIGP         # Open Shortest Path First IGP
ipip    94      IPIP            # Yet Another IP encapsulation
encap   98      ENCAP           # Yet Another IP encapsulation
```

/etc/networks

The `/etc/networks` file has a similar function to that of the `/etc/hosts` file. It provides a simple database of network names against network addresses. Its format differs in that there may be only two fields per line and that the fields are coded as:

```
networkname networkaddress
```

An example might look like:

```
loopnet    127.0.0.0
localnet   192.168.0.0
amprnet    44.0.0.0
```

When you use commands like the *route* command, if a destination is a network and that network has an entry in the `/etc/networks` file then the route command will display that network name instead of its address.

5.10 Network Security and access control.

Let me start this section by warning you that securing your machine and network against malicious attack is a complex art. I do not consider myself an expert in this field at all and while the following mechanisms I describe will help, if you are serious about security then I recommend you do some research of your own into the subject. There are many good references on the Internet relating to the subject, including the Security-HOWTO

An important rule of thumb is: `**Don't run servers you don't intend to use**'. Many distributions come configured with all sorts of services configured and automatically started. To ensure even a minimum level of safety you should go through your `/etc/inetd.conf` file and comment out (*place a `#' at the start of the line*) any entries for services you don't intend to use. Good candidates are services such as: `shell`, `login`, `exec`, `uucp`, `ftp` and informational services such as `finger`, `netstat` and `systat`.

There are all sorts of security and access control mechanisms, I'll describe the most elementary of them.

/etc/ftpusers

The `/etc/ftpusers` file is a simple mechanism that allows you to deny certain users from logging into your machine via ftp. The `/etc/ftpusers` file is read by the ftp daemon program (*ftpd*) when an incoming ftp connection is received. The file is a simple list of those users who are disallowed from logging in. It might looks something like:

```
# /etc/ftpusers - users not allowed to login via ftp
root
uucp
bin
mail
```

/etc/securetty

The `/etc/securetty` file allows you to specify which `tty` devices `root` is allowed to login on. The `/etc/securetty` file is read by the login program (usually */bin/login*). Its format is a list of the tty devices names allowed, on all others `root` login is disallowed:

```
# /etc/securetty - tty's on which root is allowed to login
tty1
tty2
tty3
tty4
```

The *tcpd* hosts access control mechanism.

The *tcpd* program you will have seen listed in the same `/etc/inetd.conf` provides logging and access control mechanisms to services it is configured to protect.

When it is invoked by the *inetd* program it reads two files containing access rules and either allows or denies access to the server it is protecting accordingly.

It will search the rules files until the first match is found. If no match is found then it assumes that access should be allowed to anyone. The files it searches in sequence are: `/etc/hosts.allow`, `/etc/hosts.deny`. I'll describe each of these in turn. For a complete description of this facility you should refer to the appropriate *man* pages (`hosts_access(5)` is a good starting point).

TOC

/etc/hosts.allow

The `/etc/hosts.allow` file is a configuration file of the */usr/sbin/tcpd* program. The `hosts.allow` file contains rules describing which hosts are *allowed* access to a service on your machine.

The file format is quite simple:

```
# /etc/hosts.allow
#
# <service list>: <host list> [: command]
```

`service list`

is a comma delimited list of server names that this rule applies to. Example server names are: `ftpd`, `telnetd` and `fingerd`.

`host list`

is a comma delimited list of host names. You may also use IP addresses here. You may additionally specify hostnames or addresses using wildcard characters to match groups of hosts. Examples include: `gw.vk2ktj.ampr.org` to match a specific host, `.uts.edu.au` to match any hostname ending in that string, `44.` to match any IP address commencing with those digits. There are some special tokens to simplify configuration, some of these are: `ALL` matches every host, `LOCAL` matches any host whose name does not contain a `.' ie is in the same domain as your machine and `PARANOID` matches any host whose name does not match its address (name spoofing). There is one last token that is also useful. The `EXCEPT` token allows you to provide a list with exceptions. This will be covered in an example later.

`command`

is an optional parameter. This parameter is the full pathname of a command that would be executed everytime this rule is matched. It could for example run a command that would attempt to identify who is logged onto the connecting host, or to generate a mail message or some other warning to a system administrator that someone is attempting to connect. There are a number of expansions that may be included, some common examples are: `%h` expands to the name of the connecting host or address if it doesn't have a name, `%d` the daemon name being called.

An example:

```
# /etc/hosts.allow
#
# Allow mail to anyone
in.smtpd: ALL
# All telnet and ftp to only hosts within my domain and my host at home.
telnetd, ftpd: LOCAL, myhost.athome.org.au
# Allow finger to anyone but keep a record of who they are.
fingerd: ALL: (finger @%h | mail -s "finger from %h" root)
```

/etc/hosts.deny

The `/etc/hosts.deny` file is a configuration file of the */usr/sbin/tcpd* program. The `hosts.deny` file contains rules describing which hosts are *disallowed* access to a service on your machine.

A simple sample would look something like this:

```
# /etc/hosts.deny
#
# Disallow all hosts with suspect hostnames
ALL: PARANOID
#
# Disallow all hosts.
ALL: ALL
```

The `PARANOID` entry is really redundant because the other entry traps everything in any case. Either of these entry would make a reasonable default depending on your particular requirement.

Having an `ALL: ALL` default in the `/etc/hosts.deny` and then specifically enabling on those services and hosts that you want in the `/etc/hosts.allow` file is the safest configuration.

/etc/hosts.equiv

The `hosts.equiv` file is used to grant certain hosts and users access rights to accounts on your machine without having to supply a password. This is useful in a secure environment where you control all machines, but is a security hazard otherwise. Your machine is only as secure as the least secure of the trusted hosts. To maximize security, don't use this mechanism and encourage your users not to use the `.rhosts` file as well.

Configure your *ftp* daemon properly.

Many sites will be interested in running an anonymous *ftp* server to allow other people to upload and download files without requiring a specific userid. If you decide to offer this facility make sure you configure the *ftp* daemon properly for anonymous access. Most *man* pages for `ftpd(8)` describe in some length how to go about this. You should always ensure that you follow these instructions. An important tip is to not use a copy of your `/etc/passwd` file in the anonymous account `/etc` directory, make sure you strip out all account details except those that you must have, otherwise you will be vulnerable to brute force password cracking techniques.

Network Firewalling.

Not allowing datagrams to even reach your machine or servers is an excellent means of security. This is covered in depth in the Firewall-HOWTO, and (more concisely) in a later section of this document.

Other suggestions.

Here are some other, potentially religious suggestions for you to consider.

sendmail

despite its popularity the *sendmail* daemon appears with frightening regularity on security warning announcements. Its up to you, but I choose not to run it.

NFS and other Sun RPC services

be wary of these. There are all sorts of possible exploits for these services. It is difficult finding an option to services like NFS, but if you configure them, make sure you are careful with who you allow mount rights to.

---

6. IP- and Ethernet-Related Information

This section covers information specific to Ethernet and IP. These subsections have been grouped together because I think they are the most interesting ones in the formerly-called ``Technology Specific'' Section. Anyone with a LAN should be able to benefit from these goodies.

6.1 Ethernet

Ethernet device names are `eth0', `eth1', `eth2' etc. The first card detected by the kernel is assigned `eth0' and the rest are assigned sequentially in the order they are detected.

By default, the Linux kernel only probes for one Ethernet device, you need to pass command line arguments to the kernel in order to force detection of furter boards.

To learn how to make your ethernet card(s) working under Linux you should refer to the Ethernet-HOWTO.

Once you have your kernel properly built to support your ethernet card then configuration of the card is easy.

Typically you would use something like (which most distributions already do for you, if you configured them to support your ethernet):

```
root# ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up
root# route add -net 192.168.0.0 netmask 255.255.255.0 eth0
```

Most of the ethernet drivers were developed by Donald Becker, `becker@CESDIS.gsfc.nasa.gov`.

6.2 EQL - multiple line traffic equaliser

The EQL device name is `eql'. With the standard kernel source you may have only one EQL device per machine. EQL provides a means of utilizing multiple point to point lines such as PPP, slip or plip as a single logical link to carry tcp/ip. Often it is cheaper to use multiple lower speed lines than to have one high speed line installed.

Kernel Compile Options:

```
Network device support  --->
    [*] Network device support
    <*> EQL (serial line load balancing) support
```

To support this mechanism the machine at the other end of the lines must also support EQL. Linux, Livingstone Portmasters and newer dial-in servers support compatible facilities.

To configure EQL you will need the eql tools which are available from: metalab.unc.edu.

Configuration is fairly straightforward. You start by configuring the eql interface. The eql interface is just like any other network device. You configure the IP address and mtu using the *ifconfig* utility, so something like:

```
root# ifconfig eql 192.168.10.1 mtu 1006
```

Next you need to manually initiate each of the lines you will use. These may be any combination of point to point network devices. How you initiate the connections will depend on what sort of link they are, refer to the appropriate sections for further information.

Lastly you need to associate the serial link with the EQL device, this is called `enslaving' and is done with the *eql_enslave* command as shown:

```
root# eql_enslave eql sl0 28800
root# eql_enslave eql ppp0 14400
```

The `*estimated speed*' parameter you supply *eql_enslave* doesn't do anything directly. It is used by the EQL driver to determine what share of the datagrams that device should receive, so you can fine tune the balancing of the lines by playing with this value.

To disassociate a line from an EQL device you use the *eql_emancipate* command as shown:

```
root# eql_emancipate eql sl0
```

You add routing as you would for any other point to point link, except your routes should refer to the `eql` device rather than the actual serial devices themselves, typically you would use:

```
root# route add default eql
```

The EQL driver was developed by Simon Janes, `simon@ncm.com`.

6.3 IP Accounting (for Linux-2.0)

The IP accounting features of the Linux kernel allow you to collect and analyze some network usage data. The data collected comprises the number of packets and the number of bytes accumulated since the figures were last reset. You may specify a variety of rules to categorize the figures to suit whatever purpose you may have. This option has been removed in kernel 2.1.102, because the old ipfwadm-based firewalling was replaced by ``ipfwchains''.

Kernel Compile Options:

```
Networking options  --->
    [*] IP: accounting
```

After you have compiled and installed the kernel you need to use the *ipfwadm* command to configure IP accounting. There are many different ways of breaking down the accounting information that you might choose. I've picked a simple example of what might be useful to use, you should read the *ipfwadm* man page for more information.

Scenario: You have a ethernet network that is linked to the internet via a PPP link. On the ethernet you have a machine that offers a number of services and that you are interested in knowing how much traffic is generated by each of ftp and world wide web traffic, as well as total tcp and udp traffic.

You might use a command set that looks like the following, which is shown as a shell script:

```
#!/bin/sh
#
# Flush the accounting rules
```

TOC

```
ipfwadm -A -f
#
# Set shortcuts
localnet=44.136.8.96/29
any=0/0
# Add rules for local ethernet segment
ipfwadm -A in  -a -P tcp -D $localnet ftp-data
ipfwadm -A out -a -P tcp -S $localnet ftp-data
ipfwadm -A in  -a -P tcp -D $localnet www
ipfwadm -A out -a -P tcp -S $localnet www
ipfwadm -A in  -a -P tcp -D $localnet
ipfwadm -A out -a -P tcp -S $localnet
ipfwadm -A in  -a -P udp -D $localnet
ipfwadm -A out -a -P udp -S $localnet
#
# Rules for default
ipfwadm -A in  -a -P tcp -D $any ftp-data
ipfwadm -A out -a -P tcp -S $any ftp-data
ipfwadm -A in  -a -P tcp -D $any www
ipfwadm -A out -a -P tcp -S $any www
ipfwadm -A in  -a -P tcp -D $any
ipfwadm -A out -a -P tcp -S $any
ipfwadm -A in  -a -P udp -D $any
ipfwadm -A out -a -P udp -S $any
#
# List the rules
ipfwadm -A -l -n
#
```

The names ``ftp-data'' and ``www'' refer to lines in `/etc/services`. The last command lists each of the Accounting rules and displays the collected totals.

An important point to note when analyzing IP accounting is that **totals for all rules that match will be incremented** so that to obtain differential figures you need to perform appropriate maths. For example if I wanted to know how much data was not ftp nor www I would substract the individual totals from the rule that matches all ports.

```
root# ipfwadm -A -l -n
IP accounting rules
 pkts bytes dir prot source              destination         ports
    0     0 in  tcp  0.0.0.0/0           44.136.8.96/29      * -> 20
    0     0 out tcp  44.136.8.96/29      0.0.0.0/0           20 -> *
   10  1166 in  tcp  0.0.0.0/0           44.136.8.96/29      * -> 80
   10   572 out tcp  44.136.8.96/29      0.0.0.0/0           80 -> *
  252 10943 in  tcp  0.0.0.0/0           44.136.8.96/29      * -> *
  231 18831 out tcp  44.136.8.96/29      0.0.0.0/0            * -> *
    0     0 in  udp  0.0.0.0/0           44.136.8.96/29      * -> *
    0     0 out udp  44.136.8.96/29      0.0.0.0/0           * -> *
    0     0 in  tcp  0.0.0.0/0           0.0.0.0/0           * -> 20
    0     0 out tcp  0.0.0.0/0           0.0.0.0/0           20 -> *
   10  1166 in  tcp  0.0.0.0/0           0.0.0.0/0           * -> 80
   10   572 out tcp  0.0.0.0/0           0.0.0.0/0           80 -> *
  253 10983 in  tcp  0.0.0.0/0           0.0.0.0/0           * -> *
```

```
  231 18831 out tcp  0.0.0.0/0            0.0.0.0/0            * -> *
    0     0 in  udp  0.0.0.0/0            0.0.0.0/0            * -> *
    0     0 out udp  0.0.0.0/0            0.0.0.0/0            * -> *
```

TOC

6.4 IP Accounting (for Linux-2.2)

The new accounting code is accessed via ``IP Firewall Chains''. See the IP chains home page for more information. Among other things, you'll now need to use *ipchains* instead of `ipfwadm` to configure your filters. (From `Documentation/Changes` in the latest kernel sources).

6.5 IP Aliasing

There are some applications where being able to configure multiple IP addresses to a single network device is useful. Internet Service Providers often use this facility to provide a `customized' to their World Wide Web and ftp offerings for their customers. You can refer to the ``IP-Alias mini-HOWTO'' for more information than you find here.

Kernel Compile Options:

```
Networking options  --->
    ....
    [*] Network aliasing
    ....
    <*> IP: aliasing support
```

After compiling and installing your kernel with IP_Alias support configuration is very simple. The aliases are added to virtual network devices associated with the actual network device. A simple naming convention applies to these devices being `<devname>:<virtual dev num>`, e.g. `eth0:0`, `ppp0:10` etc. Note that the the ifname:number device can only be configured *after* the main interface has been set up.

For example, assume you have an ethernet network that supports two different IP subnetworks simultaneously and you wish your machine to have direct access to both, you could use something like:

```
root# ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0

root# ifconfig eth0:0 192.168.10.1 netmask 255.255.255.0 up
root# route add -net 192.168.10.0 netmask 255.255.255.0 eth0:0
```

To delete an alias you simply add a `-' to the end of its name and refer to it and is as simple as:

```
root# ifconfig eth0:0- 0
```

All routes associated with that alias will also be deleted automatically.

6.6 IP Firewall (for Linux-2.0)

IP Firewall and Firewalling issues are covered in more depth in the Firewall-HOWTO. IP Firewalling allows you to secure your machine against unauthorized network access by filtering or allowing datagrams from or to IP addresses that

you nominate. There are three different classes of rules, incoming filtering, outgoing filtering and forwarding filtering. Incoming rules are applied to datagrams that are received by a network device. Outgoing rules are applied to datagrams that are to be transmitted by a network device. Forwarding rules are applied to datagrams that are received and are not for this machine, ie datagrams that would be routed.

Kernel Compile Options:

```
Networking options  --->
    [*] Network firewalls
    ....
    [*] IP: forwarding/gatewaying
    ....
    [*] IP: firewalling
    [ ] IP: firewall packet logging
```

Configuration of the IP firewall rules is performed using the *ipfwadm* command. As I mentioned earlier, security is not something I am expert at, so while I will present an example you can use, you should do your own research and develop your own rules if security is important to you.

Probably the most common use of IP firewall is when you are using your linux machine as a router and firewall gateway to protect your local network from unauthorized access from outside your network.

The following configuration is based on a contribution from Arnt Gulbrandsen, `<agulbra@troll.no>`.

The example describes the configuration of the firewall rules on the Linux firewall/router machine illustrated in this diagram:

```
-                                   -
 \                                  | 172.16.37.0
  \                                 |   /255.255.255.0
   \                  ---------     |
    |  172.16.174.30 | Linux |      |
NET ================|  f/w  |------|    ..37.19
    |    PPP         | router|      |  --------
   /                  ---------     |--| Mail |
  /                                 |  | /DNS |
 /                                  |  --------
-                                   -
```

The following commands would normally be placed in an `rc` file so that they were automatically started each time the system boots. For maximum security they would be performed after the network interfaces are configured, but before the interfaces are actually brought up to prevent anyone gaining access while the firewall machine is rebooting.

```
#!/bin/sh

# Flush the 'Forwarding' rules table
# Change the default policy to 'accept'
#
/sbin/ipfwadm -F -f
/sbin/ipfwadm -F -p accept
#
# .. and for 'Incoming'
#
```

TOC

```
/sbin/ipfwadm -I -f
/sbin/ipfwadm -I -p accept

# First off, seal off the PPP interface
# I'd love to use '-a deny' instead of '-a reject -y' but then it
# would be impossible to originate connections on that interface too.
# The -o causes all rejected datagrams to be logged. This trades
# disk space against knowledge of an attack of configuration error.
#
/sbin/ipfwadm -I -a reject -y -o -P tcp -S 0/0 -D 172.16.174.30

# Throw away certain kinds of obviously forged packets right away:
# Nothing should come from multicast/anycast/broadcast addresses
#
/sbin/ipfwadm -F -a deny -o -S 224.0/3 -D 172.16.37.0/24
#
# and nothing coming from the loopback network should ever be
# seen on a wire
#
/sbin/ipfwadm -F -a deny -o -S 127.0/8 -D 172.16.37.0/24

# accept incoming SMTP and DNS connections, but only
# to the Mail/Name Server
#
/sbin/ipfwadm -F -a accept -P tcp -S 0/0 -D 172.16.37.19 25 53
#
# DNS uses UDP as well as TCP, so allow that too
# for questions to our name server
#
/sbin/ipfwadm -F -a accept -P udp -S 0/0 -D 172.16.37.19 53
#
# but not "answers" coming to dangerous ports like NFS and
# Larry McVoy's NFS extension.  If you run squid, add its port here.
#
/sbin/ipfwadm -F -a deny -o -P udp -S 0/0 53 \
        -D 172.16.37.0/24 2049 2050

# answers to other user ports are okay
#
/sbin/ipfwadm -F -a accept -P udp -S 0/0 53 \
        -D 172.16.37.0/24 53 1024:65535

# Reject incoming connections to identd
# We use 'reject' here so that the connecting host is told
# straight away not to bother continuing, otherwise we'd experience
# delays while ident timed out.
#
/sbin/ipfwadm -F -a reject -o -P tcp -S 0/0 -D 172.16.37.0/24 113

# Accept some common service connections from the 192.168.64 and
# 192.168.65 networks, they are friends that we trust.
#
/sbin/ipfwadm -F -a accept -P tcp -S 192.168.64.0/23 \
```

```
            -D 172.16.37.0/24 20:23

# accept and pass through anything originating inside
#
/sbin/ipfwadm -F -a accept -P tcp -S 172.16.37.0/24 -D 0/0

# deny most other incoming TCP connections and log them
# (append 1:1023 if you have problems with ftp not working)
#
/sbin/ipfwadm -F -a deny -o -y -P tcp -S 0/0 -D 172.16.37.0/24

# ... for UDP too
#
/sbin/ipfwadm -F -a deny -o -P udp -S 0/0 -D 172.16.37.0/24
```

Good firewall configurations are a little tricky. This example should be a reasonable starting point for you. The *ipfwadm* manual page offers some assistance in how to use the tool. If you intend to configure a firewall, be sure to ask around and get as much advice from sources you consider reliable and get someone to test/sanity check your configuration from the outside.

6.7 IP Firewall (for Linux-2.2)

The new firewalling code is accessed via ``IP Firewall Chains''. See the IP chanins home page for more information. Among other things, you'll now need to use *ipchains* instead of `ipfwadm` to configure your filters. (From `Documentation/Changes` in the latest kernel sources).

We are aware that this is a sorely out of date statement and we are currently working on getting this section more current. You can expect a newer version in August of 1999.

6.8 IPIP Encapsulation

Why would you want to encapsulate IP datagrams within IP datagrams? It must seem an odd thing to do if you've never seen an application of it before. Ok, here are a couple of common places where it is used: Mobile-IP and IP-Multicast. What is perhaps the most widely spread use of it though is also the least well known, Amateur Radio.

Kernel Compile Options:

```
Networking options  --->
    [*] TCP/IP networking
    [*] IP: forwarding/gatewaying
    ....
    <*> IP: tunneling
```

IP tunnel devices are called `tunl0', `tunl1' etc.

"But why ?". Ok, ok. Conventional IP routing rules mandate that an IP network comprises a network address and a network mask. This produces a series of contiguous addresses that may all be routed via a single routing entry. This is very convenient, but it means that you may only use any particular IP address while you are connected to the particular piece of network to which it belongs. In most instances this is ok, but if you are a mobile netizen then you may not be able to stay connected to the one place all the time. IP/IP encapsulation (IP tunneling) allows you to overcome this restriction by allowing datagrams destined for your IP address to be wrapped up and redirected to another IP address. If

you know that you're going to be operating from some other IP network for some time you can set up a machine on home network to accept datagrams to your IP address and redirect them to the address that you will actually be using temporarily.

A tunneled network configuration.

```
192.168.1/24                                192.168.2/24

     -                                          -
     |      ppp0 =            ppp0 =            |
     |  aaa.bbb.ccc.ddd  fff.ggg.hhh.iii        |
     |                                          |
     |   /-----\                  /-----\       |
     |   |     |       //         |     |       |
     |---|  A  |------//---------|  B  |---|
     |   |     |     //           |     |       |
     |   \-----/                  \-----/       |
     |                                          |
     -                                          -
```

The diagram illustrates another possible reason to use IPIP encapsulation, virtual private networking. This example presupposes that you have two machines each with a simple dial up internet connection. Each host is allocated just a single IP address. Behind each of these machines are some private local area networks configured with reserved IP network addresses. Suppose that you want to allow any host on network A to connect to any host on network B, just as if they were properly connected to the Internet with a network route. IPIP encapsulation will allow you to do this. Note, encapsulation does not solve the problem of how you get the hosts on networks A and B to talk to any other on the Internet, you still need tricks like IP Masquerade for that. Encapsulation is normally performed by machine functioning as routers.

Linux router `A' would be configured with a script like the following:

```
#!/bin/sh
PATH=/sbin:/usr/sbin
mask=255.255.255.0
remotegw=fff.ggg.hhh.iii
#
# Ethernet configuration
ifconfig eth0 192.168.1.1 netmask $mask up
route add -net 192.168.1.0 netmask $mask eth0
#
# ppp0 configuration (start ppp link, set default route)
pppd
route add default ppp0
#
# Tunnel device configuration
ifconfig tunl0 192.168.1.1 up
route add -net 192.168.2.0 netmask $mask gw $remotegw tunl0
```

Linux router `B' would be configured with a similar script:

```
#!/bin/sh
PATH=/sbin:/usr/sbin
```

```
mask=255.255.255.0
remotegw=aaa.bbb.ccc.ddd
#
# Ethernet configuration
ifconfig eth0 192.168.2.1 netmask $mask up
route add -net 192.168.2.0 netmask $mask eth0
#
# ppp0 configuration (start ppp link, set default route)
pppd
route add default ppp0
#
# Tunnel device configuration
ifconfig tunl0 192.168.2.1 up
route add -net 192.168.1.0 netmask $mask gw $remotegw tunl0
```

The command:

```
route add -net 192.168.1.0 netmask $mask gw $remotegw tunl0
```

reads: `Send any datagrams destined for `192.168.1.0/24` inside an IPIP encap datagram with a destination address of `aaa.bbb.ccc.ddd'.

Note that the configurations are reciprocated at either end. The tunnel device uses the `gw' in the route as the *destination* of the IP datagram in which it will place the datagram it has received to route. That machine must know how to decapsulate IPIP datagrams, that is, it must also be configured with a tunnel device.

A tunneled host configuration.

It doesn't have to be a whole network you route. You could for example route just a single IP address. In that instance you might configure the `tunl` device on the `remote' machine with its home IP address and at the A end just use a host route (and Proxy Arp) rather than a network route via the tunnel device. Let's redraw and modify our configuration appropriately. Now we have just host `B' which to want to act and behave as if it is both fully connected to the Internet and also part of the remote network supported by host `A':

```
192.168.1/24

    -
    |      ppp0 =                ppp0 =
    |  aaa.bbb.ccc.ddd      fff.ggg.hhh.iii
    |
    |   /-----\                 /-----\
    |   |     |       //        |     |
    |---|  A  |------//---------|  B  |
    |   |     |     //          |     |
    |   \-----/                 \-----/
    |                      also: 192.168.1.12
    -
```

Linux router `A' would be configured with:

```
#!/bin/sh
```

```
PATH=/sbin:/usr/sbin
mask=255.255.255.0
remotegw=fff.ggg.hhh.iii
#
# Ethernet configuration
ifconfig eth0 192.168.1.1 netmask $mask up
route add -net 192.168.1.0 netmask $mask eth0
#
# ppp0 configuration (start ppp link, set default route)
pppd
route add default ppp0
#
# Tunnel device configuration
ifconfig tunl0 192.168.1.1 up
route add -host 192.168.1.12 gw $remotegw tunl0
#
# Proxy ARP for the remote host
arp -s 192.168.1.12 xx:xx:xx:xx:xx:xx pub
```

Linux host `B' would be configured with:

```
#!/bin/sh
PATH=/sbin:/usr/sbin
mask=255.255.255.0
remotegw=aaa.bbb.ccc.ddd
#
# ppp0 configuration (start ppp link, set default route)
pppd
route add default ppp0
#
# Tunnel device configuration
ifconfig tunl0 192.168.1.12 up
route add -net 192.168.1.0 netmask $mask gw $remotegwtunl0
```

This sort of configuration is more typical of a Mobile-IP application. Where a single host wants to roam around the Internet and maintain a single usable IP address the whole time. You should refer to the Mobile-IP section for more information on how that is handled in practice.

6.9 IP Masquerade

Many people have a simple dialup account to connect to the Internet. Nearly everybody using this sort of configuration is allocated a single IP address by the Internet Service Provider. This is normally enough to allow only one host full access to the network. IP Masquerade is a clever trick that enables you to have many machines make use of that one IP address, by causing the other hosts to look like, hence the term masquerade, the machine supporting the dialup connection. There is a small caveat and that is that the masquerade function nearly always works only in one direction, that is the masqueraded hosts can make calls out, but they cannot accept or receive network connections from remote hosts. This means that some network services do not work such as *talk* and others such as *ftp* must be configured to operate in passive (PASV) mode to operate. Fortunately the most common network services such as *telnet*, World Wide Web and *irc* do work just fine.

Kernel Compile Options:

TOC

```
Code maturity level options  --->
    [*] Prompt for development and/or incomplete code/drivers
Networking options  --->
    [*] Network firewalls
    ....
    [*] TCP/IP networking
    [*] IP: forwarding/gatewaying
    ....
    [*] IP: masquerading (EXPERIMENTAL)
```

Normally you have your linux machine supporting a slip or PPP dialup line just as it would if it were a standalone machine. Additionally it would have another network device configured, perhaps an ethernet, configured with one of the reserved network addresses. The hosts to be masqueraded would be on this second network. Each of these hosts would have the IP address of the ethernet port of the linux machine set as their default gateway or router.

A typical configuration might look something like this:

```
-                                   -
 \                                  | 192.168.1.0
  \                                 |   /255.255.255.0
   \                 ---------      |
    |                | Linux | .1.1 |
NET ================| masq  |------|
    |    PPP/slip    | router|      |  --------
   /                 ---------      |--| host |
  /                                  |  |      |
 /                                   |  --------
-                                   -
```

Masquerading with IPFWADM

The most relevant commands for this configuration are:

```
# Network route for ethernet
route add -net 192.168.1.0 netmask 255.255.255.0 eth0
#
# Default route to the rest of the internet.
route add default ppp0
#
# Cause all hosts on the 192.168.1/24 network to be masqueraded.
ipfwadm -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0
```

Masquerading with IPCHAINS

This is similar to using IPFWADM but the command structure has changed:

```
# Network route for ethernet
route add -net 192.168.1.0 netmask 255.255.255.0 eth0
#
# Default route to the rest of the internet.
route add default ppp0
```

```
#
# Cause all hosts on the 192.168.1/24 network to be masqueraded.
ipchains -A forward -s 192.168.1.0/24 -j MASQ
```

TOC

You can get more information on the Linux IP Masquerade feature from the IP Masquerade Resource Page. Also, a *very* detailed document about masquesrading is the ``IP-Masquerade mini-HOWTO'' (which also intructs to configure other OS's to run with a Linux masquerade server).

6.10 IP Transparent Proxy

IP transparent proxy is a feature that enables you to redirect servers or services destined for another machine to those services on this machine. Typically this would be useful where you have a linux machine as a router and also provides a proxy server. You would redirect all connections destined for that service remotely to the local proxy server.

Kernel Compile Options:

```
Code maturity level options  --->
        [*] Prompt for development and/or incomplete code/drivers
Networking options  --->
        [*] Network firewalls
        ....
        [*] TCP/IP networking
        ....
        [*] IP: firewalling
        ....
        [*] IP: transparent proxy support (EXPERIMENTAL)
```

Configuration of the transparent proxy feature is performed using the *ipfwadm* command

An example that might be useful is as follows:

```
root# ipfwadm -I -a accept -D 0/0 telnet -r 2323
```

This example will cause any connection attempts to port `telnet` (23) on any host to be redirected to port 2323 on this host. If you run a service on that port, you could forward telnet connections, log them or do whatever fits your need.

A more interesting example is redirecting all `http` traffic through a local cache. However, the protocol used by proxy servers is different from native http: where a client connects to `www.server.com:80` and asks for `/path/page`, when it connects to the local cache it contacts `proxy.local.domain:8080` and asks for `www.server.com/path/page`.

To filter an `http` request through the local proxy, you need to adapt the protocol by inserting a small server, called `transproxy` (you can find it on the world wide web). You can choose to run `transproxy` on port 8081, and issue this command:

```
root# ipfwadm -I -a accept -D 0/0 80 -r 8081
```

The `transproxy` program, then, will receive all connections meant to reach external servers and will pass them to the local proxy after fixing protocol differences.

TOC

6.11 IPv6

Just when you thought you were beginning to understand IP networking the rules get changed! IPv6 is the shorthand notation for version 6 of the Internet Protocol. IPv6 was developed primarily to overcome the concerns in the Internet community that there would soon be a shortage of IP addresses to allocate. IPv6 addresses are 16 bytes long (128 bits). IPv6 incorporates a number of other changes, mostly simplifications, that will make IPv6 networks more managable than IPv4 networks.

Linux already has a working, but not complete, IPv6 implementation in the `2.2.*` series kernels.

If you wish to experiment with this next generation Internet technology, or have a requirement for it, then you should read the IPv6-FAQ which is available from www.terra.net.

6.12 Mobile IP

The term "IP mobility" describes the ability of a host that is able to move its network connection from one point on the Internet to another without changing its IP address or losing connectivity. Usually when an IP host changes its point of connectivity it must also change its IP address. IP Mobility overcomes this problem by allocating a fixed IP address to the mobile host and using IP encapsulation (tunneling) with automatic routing to ensure that datagrams destined for it are routed to the actual IP address it is currently using.

A project is underway to provide a complete set of IP mobility tools for Linux. The Status of the project and tools may be obtained from the: Linux Mobile IP Home Page.

6.13 Multicast

IP Multicast allows an arbitrary number of IP hosts on disparate IP networks to have IP datagrams simultaneously routed to them. This mechanism is exploited to provide Internet wide "broadcast" material such as audio and video transmissions and other novel applications.

Kernel Compile Options:

```
Networking options  --->
        [*] TCP/IP networking
        ....
        [*] IP: multicasting
```

A suite of tools and some minor network configuration is required. Please check the Multicast-HOWTO for more information on Multicast support in Linux.

6.14 NAT - Network Address Translation

The IP Network Address Translation facility is pretty much the standardized big brother of the Linux IP Masquerade facility. It is specified in some detail in RFC-1631 at your nearest RFC archive. NAT provides features that IP-Masquerade does not that make it eminently more suitable for use in corporate firewall router designs and larger scale installations.

An alpha implementation of NAT for Linux 2.0.29 kernel has been developed by Michael.Hasenstein, `Michael.Hasenstein@informatik.tu-chemnitz.de`. Michaels documentation and implementation are available from: Linux IP Network Address Web Page

Newer Linux 2.2.x kernels also include some NAT functionality in the routing algorithm.

TOC

6.15 Traffic Shaper - Changing allowed bandwidth

The traffic shaper is a driver that creates new interface devices, those devices are traffic-limited in a user-defined way, they rely on physical network devices for actual transmission and can be used as outgoing routed for network traffic.

The shaper was introduced in Linux-2.1.15 and was backported to Linux-2.0.36 (it appeared in `2.0.36-pre-patch-2` distributed by Alan Cox, the author of the shaper device and maintainer of Linux-2.0).

The traffic shaper can only be compiled as a module and is configured by the *shapecfg* program with commands like the following:

```
shapecfg attach shaper0 eth1
shapecfg speed shaper0 64000
```

The shaper device can only control the bandwidth of outgoing traffic, as packets are transmitted via the shaper only according to the routing tables; therefore, a ``route by source address'' functionality could help in limiting the overall bandwidth of specific hosts using a Linux router.

Linux-2.2 already has support for such routing, if you need it for Linux-2.0 please check the patch by Mike McLagan, at `ftp.invlogic.com`. Refer to `Documentation`networking/shaper.txt for further information about the shaper.

If you want to try out a (tentative) shaping for incoming packets, try out `rshaper-1.01` (or newer), from ftp.systemy.it.

6.16 Routing in Linux-2.2

The latest versions of Linux, 2.2 offer a lot of flexibility in routing policy. Unfortunately, you have to wait for the next version of this howto, or go read the kernel sources.

---

7. Using common PC hardware

7.1 ISDN

The Integrated Services Digital Network (ISDN) is a series of standards that specify a general purpose switched digital data network. An ISDN `call' creates a synchronous point to point data service to the destination. ISDN is generally delivered on a high speed link that is broken down into a number of discrete channels. There are two different types of channels, the `B Channels' which will actually carry the user data and a single channel called the `D channel' which is used to send control information to the ISDN exchange to establish calls and other functions. In Australia for example, ISDN may be delivered on a 2Mbps link that is broken into 30 discrete 64kbps B channels with one 64kbps D channel. Any number of channels may be used at a time and in any combination. You could for example establish 30 separate calls to 30 different destinations at 64kbps each, or you could establish 15 calls to 15 different destinations at 128kbps each (two channels used per call), or just a small number of calls and leave the rest idle. A channel may be used for either incoming or outgoing calls. The original intention of ISDN was to allow Telecommunications companies to provide a single data service which could deliver either telephone (via digitised voice) or data services to your home or business without requiring you to make any special configuration changes.

There are a few different ways to connect your computer to an ISDN service. One way is to use a device called a `Terminal Adaptor' which plugs into the Network Terminating Unit that you telecommunications carrier will have installed when you got your ISDN service and presents a number of serial interfaces. One of those interfaces is used to enter commands to establish calls and configuration and the others are actually connected to the network devices that will use the data circuits when they are established. Linux will work in this sort of configuration without modification, you just treat the port on the Terminal Adaptor like you would treat any other serial device. Another way, which is the way the kernel ISDN support is designed for allows you to install an ISDN card into your Linux machine and then has your Linux software handle the protocols and make the calls itself.

Kernel Compile Options:

```
ISDN subsystem  --->
        <*> ISDN support
        [ ] Support synchronous PPP
        [ ] Support audio via ISDN
        < > ICN 2B and 4B support
        < > PCBIT-D support
        < > Teles/NICCY1016PC/Creatix support
```

The Linux implementation of ISDN supports a number of different types of internal ISDN cards. These are those listed in the kernel configuration options:

- ICN 2B and 4B
- Octal PCBIT-D
- Teles ISDN-cards and compatibles

Some of these cards require software to be downloaded to them to make them operational. There is a separate utility to do this with.

TOC

Full details on how to configure the Linux ISDN support is available from the `/usr/src/linux/Documentation/isdn/` directory and an FAQ dedicated to *isdn4linux* is available at www.lrz-muenchen.de. (You can click on the english flag to get an english version).

A note about PPP. The PPP suite of protocols will operate over either asynchronous or synchronous serial lines. The commonly distributed PPP daemon for Linux `*pppd*' supports only asynchronous mode. If you wish to run the PPP protocols over your ISDN service you need a specially modified version. Details of where to find it are available in the documentation referred to above.

7.2 PLIP for Linux-2.0

PLIP device names are `plip0', `plip1 and plip2.

Kernel Compile Options:

```
Network device support  --->
    <*> PLIP (parallel port) support
```

plip (Parallel Line IP), is like SLIP, in that it is used for providing a *point to point* network connection between two machines, except that it is designed to use the parallel printer ports on your machine instead of the serial ports (a cabling diagram in included in the cabling diagram section later in this document). Because it is possible to transfer more than one bit at a time with a parallel port, it is possible to attain higher speeds with the *plip* interface than with a standard serial device. In addition, even the simplest of parallel ports, printer ports, can be used in lieu of you having to purchase comparatively expensive 16550AFN UART's for your serial ports. PLIP uses a lot of CPU compared to a serial link and is most certainly not a good option if you can obtain some cheap ethernet cards, but it will work when nothing else is available and will work quite well. You should expect a data transfer rate of about 20 kilobytes per second when a link is running well.

The PLIP device drivers competes with the parallel device driver for the parallel port hardware. If you wish to use both drivers then you should compile them both as modules to ensure that you are able to select which port you want to use for PLIP and which ports you want for the printer driver. Refer to the ``Mudules mini-HOWTO'' for more information on kernel module configuration.

Please note that some laptops use chipsets that will not work with PLIP because they do not allow some combinations of signals that PLIP relies on, that printers don't use.

The Linux *plip* interface is compatible with the *Crynwyr Packet Driver PLIP* and this will mean that you can connect your Linux machine to a DOS machine running any other sort of tcp/ip software via *plip*.

In the 2.0.* series kernel the plip devices are mapped to i/o port and IRQ as follows:

```
device  i/o     IRQ
------  -----   ---
plip0   0x3bc   5
plip1   0x378   7
plip2   0x278   2
```

If your parallel ports don't match any of the above combinations then you can change the IRQ of a port using the *ifconfig* command using the `irq' parameter (be sure to enable IRQ's on your printer ports in your ROM BIOS if it

supports this option). As an alternative, you can specify ``io=" annd ``irq=" options on the *insmod* command line, if you use modules. For example:

```
root# insmod plip.o io=0x288 irq=5
```

PLIP operation is controlled by two timeouts, whose default values are probably ok in most cases. You will probably need to increase them if you have an especially slow computer, in which case the timers to increase are actually on the **other** computer. A program called *plipconfig* exists that allows you to change these timer settings without recompiling your kernel. It is supplied with many Linux distributions.

To configure a *plip* interface, you will need to invoke the following commands (or **add** them to your initialization scripts):

```
root# /sbin/ifconfig plip1 localplip pointopoint remoteplip
root# /sbin/route add remoteplip plip1
```

Here, the port being used is the one at I/O address 0x378; *localplip* amd *remoteplip* are the names or IP addresses used over the PLIP cable. I personally keep them in my `/etc/hosts` database:

```
# plip entries
192.168.3.1   localplip
192.168.3.2   remoteplip
```

The *pointopoint* parameter has the same meaning as for SLIP, in that it specifies the address of the machine at the other end of the link.

In almost all respects you can treat a *plip* interface as though it were a *SLIP* interface, except that neither *dip* nor *slattach* need be, nor can be, used.

Further information on PLIP may be obtained from the ``PLIP mini-HOWTO".

7.3 PLIP for Linux-2.2

During development of the 2.1 kernel versions, support for the parallel port was changed to a better setup.

Kernel Compile Options:

```
General setup  --->
    [*] Parallel port support
Network device support  --->
    <*> PLIP (parallel port) support
```

The new code for PLIP behaves like the old one (use the same *ifconfig* and *route* commands as in the previous section, but initialization of the device is different due to the advanced parallel port support.

The ``first" PLIP device is always called ``plip0", where first is the first device detected by the system, similarly to what happens for Ethernet devices. The actual parallel port being used is one of the available ports, as shown in `/proc/parport`. For example, if you have only one parallel port, you'll only have a directory called

`/proc/parport/0`.

If your kernel didn't detect the IRQ number used by your port, ``insmod plip'' will fail; in this case just write the right number to `/proc/parport/0/irq` and reinvoke *insmod*.

Complete information about parallel port management is available in the file `Documentation/parport.txt`, part of your kernel sources.

7.4 PPP

PPP devices names are `ppp0', `ppp1, etc. Devices are numbered sequentially with the first device configured receiving `0'.

Kernel Compile Options:

```
Networking options  --->
    <*> PPP (point-to-point) support
```

PPP configuration is covered in detail in the PPP-HOWTO.

Maintaining a permanent connection to the net with *pppd*.

If you are fortunate enough to have a semi permanent connection to the net and would like to have your machine automatically redial your PPP connection if it is lost then here is a simple trick to do so.

Configure PPP such that it can be started by the `root` user by issuing the command:

```
# pppd
```

Be sure that you have the `-detach' option configured in your `/etc/ppp/options` file. Then, insert the following line into your `/etc/inittab` file, down with the *getty* definitions:

```
pd:23:respawn:/usr/sbin/pppd
```

This will cause the *init* program to spawn and monitor the *pppd* program and automatically restart it if it dies.

7.5 SLIP client

SLIP devices are named `sl0', `sl1' etc. with the first device configured being assigned `0' and the rest incrementing sequentially as they are configured.

Kernel Compile Options:

```
Network device support  --->
    [*] Network device support
    <*> SLIP (serial line) support
    [ ]  CSLIP compressed headers
    [ ]  Keepalive and linefill
    [ ]  Six bit SLIP encapsulation
```

TOC

SLIP (Serial Line Internet Protocol) allows you to use tcp/ip over a serial line, be that a phone line with a dial u modem, or a leased line of some sort. Of course to use SLIP you need access to a *SLIP-server* in your area. M universities and businesses provide SLIP access all over the world.

Slip uses the serial ports on your machine to carry IP datagrams. To do this it must take control of the serial device. Slip device names are named *sl0*, *sl1* etc. How do these correspond to your serial devices ? The networking code uses what is called an *ioctl* (i/o control) call to change the serial devices into SLIP devices. There are two programs supplied that can do this, they are called *dip* and *slattach*

dip

dip (Dialup IP) is a smart program that is able to set the speed of the serial device, command your modem to dial the remote end of the link, automatically log you into the remote server, search for messages sent to you by the server and extract information for them such as your IP address and perform the *ioctl* necessary to switch your serial port into SLIP mode. *dip* has a powerful scripting ability and it is this that you can exploit to automate your logon procedure.

You can find it at: metalab.unc.edu.

To install it, try the following:

```
user% tar xvzf dip337o-uri.tgz
user% cd dip-3.3.7o
user% vi Makefile
root# make install
```

The `Makefile` assumes the existence of a group called *uucp*, but you might like to change this to either *dip* or *SLIP* depending on your configuration.

slattach

slattach as contrasted with *dip* is a very simple program, that is very easy to use, but does not have the sophistication of *dip*. It does not have the scripting ability, all it does is configure your serial device as a SLIP device. It assumes you have all the information you need and the serial line is established before you invoke it. *slattach* is ideal to use where you have a permanent connection to your server, such as a physical cable, or a leased line.

When do I use which ?

You would use *dip* when your link to the machine that is your SLIP server is a dialup modem, or some other temporary link. You would use *slattach* when you have a leased line, perhaps a cable, between your machine and the server and there is no special action needed to get the link working. See section `Permanent Slip connection' for more information.

Configuring SLIP is much like configuring an Ethernet interface (read section `Configuring an ethernet device' above). However there are a few key differences.

First of all, SLIP links are unlike ethernet networks in that there is only ever two hosts on the network, one at each end of the link. Unlike an ethernet that is available for use as soon are you are cabled, with SLIP, depending on the type of link you have, you may have to initialize your network connection in some special way.

TOC

If you are using *dip* then this would not normally be done at boot time, but at some time later, when you were to use the link. It is possible to automate this procedure. If you are using *slattach* then you will probably want add a section to your *rc.inet1* file. This will be described soon.

There are two major types of SLIP servers: Dynamic IP address servers and static IP address servers. Almost every SLIP server will prompt you to login using a username and password when dialing in. *dip* can handle logging you in automatically.

Static SLIP server with a dialup line and DIP.

A static SLIP server is one in which you have been supplied an IP address that is exclusively yours. Each time you connect to the server, you will configure your SLIP port with that address. The static SLIP server will answer your modem call, possibly prompt you for a username and password, and then route any datagrams destined for your address to you via that connection. If you have a static server, then you may want to put entries for your hostname and IP address (since you know what it will be) into your `/etc/hosts`. You should also configure some other files such as: `rc.inet2`, `host.conf`, `resolv.conf`, `/etc/HOSTNAME` and `rc.local`. Remember that when configuring `rc.inet1`, you don't need to add any special commands for your SLIP connection since it is *dip* that does all of the hard work for you in configuring your interface. You will need to give *dip* the appropriate information and it will configure the interface for you after commanding the modem to establish the call and logging you into your SLIP server.

If this is how your SLIP server works then you can move to section `Using Dip' to learn how to configure *dip* appropriately.

Dynamic SLIP server with a dialup line and DIP.

A *dynamic* SLIP server is one which allocates you an IP address randomly, from a pool of addresses, each time you logon. This means that there is no guarantee that you will have any particular address each time, and that address may well be used by someone else after you have logged off. The network administrator who configured the SLIP server will have assigned a pool of address for the SLIP server to use, when the server receives a new incoming call, it finds the first unused address, guides the caller through the login process and then prints a welcome message that contains the IP address it has allocated and will proceed to use that IP address for the duration of that call.

Configuring for this type of server is similar to configuring for a static server, except that you must add a step where you obtain the IP address that the server has allocated for you and configure your SLIP device with that.

Again, *dip* does the hard work and new versions are smart enough to not only log you in, but to also be able to automatically read the IP address printed in the welcome message and store it so that you can have it configure your SLIP device with it.

If this is how your SLIP server works then you can move to section `Using Dip' to learn how to configure *dip* appropriately.

Using DIP.

As explained earlier, *dip* is a powerful program that can simplify and automate the process of dialing into the SLIP server, logging you in, starting the connection and configuring your SLIP devices with the appropriate *ifconfig* and *route* commands.

Essentially to use *dip* you'll write a `dip script', which is basically a list of commands that *dip* understands that tell *dip* how to perform each of the actions you want it to perform. See `sample.dip` that comes supplied with *dip* to get an idea of how it works. *dip* is quite a powerful program, with many options. Instead of going into all of them

here you should look at the *man* page, README and sample files that will have come with your version of *dip*.

You may notice that the `sample.dip` script assumes that you're using a static SLIP server, so you know what your IP address is beforehand. For dynamic SLIP servers, the newer versions of *dip* include a command you can use to automatically read and configure your SLIP device with the IP address that the dynamic server allocates for you. The following sample is a modified version of the `sample.dip` that came supplied with *dip337j-uri.tgz* and is probably a good starting point for you. You might like to save it as `/etc/dipscript` and edit it to suit your configuration:

```
#
# sample.dip    Dialup IP connection support program.
#
#               This file (should show) shows how to use the DIP
#       This file should work for Annex type dynamic servers, if you
#       use a static address server then use the sample.dip file that
#       comes as part of the dip337-uri.tgz package.
#
#
# Version:      @(#)sample.dip  1.40    07/20/93
#
# Author:       Fred N. van Kempen, <waltje@uWalt.NL.Mugnet.ORG>
#

main:
# Next, set up the other side's name and address.
# My dialin machine is called 'xs4all.hacktic.nl' (== 193.78.33.42)
get $remote xs4all.hacktic.nl
# Set netmask on sl0 to 255.255.255.0
netmask 255.255.255.0
# Set the desired serial port and speed.
port cua02
speed 38400

# Reset the modem and terminal line.
# This seems to cause trouble for some people!
reset

# Note! "Standard" pre-defined "errlevel" values:
#  0 - OK
#  1 - CONNECT
#  2 - ERROR
#
# You can change those grep'ping for "addchat()" in *.c...

# Prepare for dialing.
send ATQ0V1E1X4\r
wait OK 2
if $errlvl != 0 goto modem_trouble
dial 555-1234567
if $errlvl != 1 goto modem_trouble
```

```
# We are connected.  Login to the system.
login:
sleep 2
wait ogin: 20
if $errlvl != 0 goto login_trouble
send MYLOGIN\n
wait ord: 20
if $errlvl != 0 goto password_error
send MYPASSWD\n
loggedin:

# We are now logged in.
wait SOMEPROMPT 30
if $errlvl != 0 goto prompt_error

# Command the server into SLIP mode
send SLIP\n
wait SLIP 30
if $errlvl != 0 goto prompt_error

# Get and Set your IP address from the server.
#   Here we assume that after commanding the SLIP server into SLIP
#   mode that it prints your IP address
get $locip remote 30
if $errlvl != 0 goto prompt_error

# Set up the SLIP operating parameters.
get $mtu 296
# Ensure "route add -net default xs4all.hacktic.nl" will be done
default

# Say hello and fire up!
done:
print CONNECTED $locip ---> $rmtip
mode CSLIP
goto exit

prompt_error:
print TIME-OUT waiting for sliplogin to fire up...
goto error

login_trouble:
print Trouble waiting for the Login: prompt...
goto error

password:error:
print Trouble waiting for the Password: prompt...
goto error
```

```
modem_trouble:
print Trouble occurred with the modem...
error:
print CONNECT FAILED to $remote
quit

exit:
exit
```

The above example assumes you are calling a *dynamic* SLIP server, if you are calling a *static* SLIP server, then the `sample.dip` file that comes with *dip337j-uri.tgz* should work for you.

When *dip* is given the *get $local* command it searches the incoming text from the remote end for a string that looks like an IP address, ie strings numbers separated by `.' characters. This modification was put in place specifically for *dynamic* SLIP servers, so that the process of reading the IP address granted by the server could be automated.

The example above will automatically create a default route via your SLIP link, if this is not what you want, you might have an ethernet connection that should be your default route, then remove the *default* command from the script. After this script has finished running, if you do an *ifconfig* command, you will see that you have a device *sl0*. This is your SLIP device. Should you need to, you can modify its configuration manually, after the *dip* command has finished, using the *ifconfig* and *route* commands. Please note that *dip* allows you to select a number of different protocols to use with the `mode` command, the most common example is *cSLIP* for SLIP with compression. Please note that both ends of the link must agree, so you should ensure that whatever you select agrees with what your server is set to. The above example is fairly robust and should cope with most errors. Please refer to the *dip* man page for more information. Naturally you could, for example, code the script to do such things as redial the server if it doesn't get a connection within a prescribed period of time, or even try a series of servers if you have access to more than one.

Permanent SLIP connection using a leased line and slattach.

If you have a cable between two machines, or are fortunate enough to have a leased line, or some other permanent serial connection between your machine and another, then you don't need to go to all the trouble of using *dip* to set up your serial link. *slattach* is a very simple to use utility that will allow you just enough functionality to configure your connection. Since your connection will be a permanent one, you will want to add some commands to your `rc.inet1` file. In essence all you need to do for a permanent connection is ensure that you configure the serial device to the correct speed and switch the serial device into SLIP mode. *slattach* allows you to do this with one command. **Add** the following to your `rc.inet1` file:

```
#
# Attach a leased line static SLIP connection
#
#  configure /dev/cua0 for 19.2kbps and cslip
/sbin/slattach -p cslip -s 19200 /dev/cua0 &
/sbin/ifconfig sl0 IPA.IPA.IPA.IPA pointopoint IPR.IPR.IPR.IPR up
#
# End static SLIP.
```

Where:

IPA.IPA.IPA.IPA

represents your IP address.

IPR.IPR.IPR.IPR

represents the IP address of the remote end.

slattach allocates the first unallocated SLIP device to the serial device specified. *slattach* starts with *sl0*. Therefore the first *slattach* command attaches SLIP device *sl0* to the serial device specified and *sl1* the next time, etc.

slattach allows you to configure a number of different protocols with the `-p` argument. In your case you will use either *SLIP* or *cSLIP* depending on whether you want to use compression or not. Note: both ends must agree on whether you want compression or not.

7.6 SLIP server.

If you have a machine that is perhaps network connected, that you'd like other people be able to dial into and provide network services, then you will need to configure your machine as a server. If you want to use SLIP as the serial line protocol, then currently you have three options as to how to configure your Linux machine as a SLIP server. My preference would be to use the first presented, *sliplogin*, as it seems the easiest to configure and understand, but I will present a summary of each, so you can make your own decision.

Slip Server using *sliplogin*.

sliplogin is a program that you can use in place of the normal login shell for SLIP users that converts the terminal line into a SLIP line. It allows you to configure your Linux machine as either a *static address server*, users get the same address everytime they call in, or a *dynamic address server*, where users get an address allocated for them which will not necessarily be the same as the last time they called.

The caller will login as per the standard login process, entering their username and password, but instead of being presented with a shell after their login, *sliplogin* is executed which searches its configuration file (`/etc/slip.hosts`) for an entry with a login name that matches that of the caller. If it locates one, it configures the line as an 8bit clean line, and uses an *ioctl* call to convert the line discipline to SLIP. When this process is complete, the last stage of configuration takes place, where *sliplogin* invokes a shell script which configures the SLIP interface with the relevant ip address, netmask and sets appropriate routing in place. This script is usually called `/etc/slip.login`, but in a similar manner to *getty*, if you have certain callers that require special initialization, then you can create configuration scripts called `/etc/slip.login.loginname` that will be run instead of the default specifically for them.

There are either three or four files that you need to configure to get *sliplogin* working for you. I will detail how and where to get the software and how each is configured in detail. The files are:

- `/etc/passwd`, for the dialin user accounts.
- `/etc/slip.hosts`, to contain the information unique to each dial-in user.
- `/etc/slip.login`, which manages the configuration of the routing that needs to be performed for the user.
- `/etc/slip.tty`, which is required only if you are configuring your server for *dynamic address allocation* and contains a table of addresses to allocate
- `/etc/slip.logout`, which contains commands to clean up after the user has hung up or logged out.

TOC

Where to get *sliplogin*

You may already have the *sliplogin* package installed as part of your distribution, if not then *sliplogin* can be obtained from: metalab.unc.edu. The tar file contains both source, precompiled binaries and a *man* page.

To ensure that only authorized users will be able to run *sliplogin* program, you should add an entry to your `/etc/group` file similar to the following:

```
 ..
slip::13:radio,fred
 ..
```

When you install the *sliplogin* package, the `Makefile` will change the group ownership of the *sliplogin* program to `slip`, and this will mean that only users who belong to that group will be able to execute it. The example above will allow only users `radio` and `fred` to execute *sliplogin*.

To install the binaries into your `/sbin` directory and the *man* page into section 8, do the following:

```
# cd /usr/src
# gzip -dc .../sliplogin-2.1.1.tar.gz | tar xvf -
# cd sliplogin-2.1.1
# <..edit the Makefile if you don't use shadow passwords..>
# make install
```

If you want to recompile the binaries before installation, add a `make clean` before the `make install`. If you want to install the binaries somewhere else, you will need to edit the `Makefile` *install* rule.

Please read the `README` files that come with the package for more information.

Configuring `/etc/passwd` for Slip hosts.

Normally you would create some special logins for Slip callers in your `/etc/passwd` file. A convention commonly followed is to use the *hostname* of the calling host with a capital `S' prefixing it. So, for example, if the calling host is called `radio` then you could create a `/etc/passwd` entry that looked like:

```
Sradio:FvKurok73:1427:1:radio SLIP login:/tmp:/sbin/sliplogin
```

It doesn't really matter what the account is called, so long as it is meaningful to you.

Note: the caller doesn't need any special home directory, as they will not be presented with a shell from this machine, so `/tmp` is a good choice. Also note that *sliplogin* is used in place of the normal login shell.

Configuring `/etc/slip.hosts`

The `/etc/slip.hosts` file is the file that *sliplogin* searches for entries matching the login name to obtain configuration details for this caller. It is this file where you specify the ip address and netmask that will be assigned to the caller and configured for their use. Sample entries for two hosts, one a static configuration for host `radio` and another, a dynamic configuration for user host `albert` might look like:

```
#
Sradio   44.136.8.99   44.136.8.100  255.255.255.0  normal      -1
Salbert  44.136.8.99   DYNAMIC       255.255.255.0  compressed  60
```

```
#
```

The `/etc/slip.hosts` file entries are:

1. the login name of the caller.
2. ip address of the server machine, ie this machine.
3. ip address that the caller will be assigned. If this field is coded `DYNAMIC` then an ip address will be allocated based on the information contained in your `/etc/slip.tty` file discussed later. **Note:** you must be using at least version 1.3 of sliplogin for this to work.
4. the netmask assigned to the calling machine in dotted decimal notation eg 255.255.255.0 for a Class C network mask.
5. the slip mode setting which allows you to enable/disable compression and slip other features. Allowable values here are "`normal`" or "`compressed`".
6. a timeout parameter which specifies how long the line can remain idle (no datagrams received) before the line is automatically disconnected. A negative value disables this feature.
7. optional arguments.

Note: You can use either hostnames or IP addresses in dotted decimal notation for fields 2 and 3. If you use hostnames then those hosts must be resolvable, that is, your machine must be able to locate an ip address for those hostnames, otherwise the script will fail when it is called. You can test this by trying trying to telnet to the hostname, if you get the `*Trying nnn.nnn.nnn...*' message then your machine has been able to find an ip address for that name. If you get the message `*Unknown host*', then it has not. If not, either use ip addresses in dotted decimal notation, or fix up your name resolver configuration (See section `Name Resolution`).

The most common slip modes are:

normal

to enable normal uncompressed SLIP.

compressed

to enable van Jacobsen header compression (cSLIP)

Naturally these are mutually exclusive, you can use one or the other. For more information on the other options available, refer to the *man* pages.

Configuring the `/etc/slip.login` file.

After *sliplogin* has searched the `/etc/slip.hosts` and found a matching entry, it will attempt to execute the `/etc/slip.login` file to actually configure the SLIP interface with its ip address and netmask. The sample `/etc/slip.login` file supplied with the *sliplogin* package looks like this:

```
#!/bin/sh -
#
#       @(#)slip.login  5.1 (Berkeley) 7/1/90
#
# generic login file for a SLIP line.  sliplogin invokes this with
# the parameters:
#     $1       $2       $3    $4, $5, $6 ...
#   SLIPunit ttyspeed   pid   the arguments from the slip.host entry
#
/sbin/ifconfig $1 $5 pointopoint $6 mtu 1500 -trailers up
```

```
/sbin/route add $6
arp -s $6 <hw_addr> pub
exit 0
#
```

You will note that this script simply uses the *ifconfig* and *route* commands to configure the SLIP device with its ipaddress, remote ip address and netmask and creates a route for the remote address via the SLIP device. Just the same as you would if you were using the *slattach* command.

Note also the use of *Proxy ARP* to ensure that other hosts on the same ethernet as the server machine will know how to reach the dial-in host. The `<hw_addr>` field should be the hardware address of the ethernet card in the machine. If your server machine isn't on an ethernet network then you can leave this line out completely.

Configuring the `/etc/slip.logout` file.

When the call drops out, you want to ensure that the serial device is restored to its normal state so that future callers will be able to login correctly. This is achieved with the use of the `/etc/slip.logout` file. It is quite simple in format and is called with the same argument as the `/etc/slip.login` file.

```
#!/bin/sh -
#
#               slip.logout
#
/sbin/ifconfig $1 down
arp -d $6
exit 0
#
```

All it does is `down' the interface which will delete the manual route previously created. It also uses the *arp* command to delete any proxy arp put in place, again, you don't need the *arp* command in the script if your server machine does not have an ethernet port.

Configuring the `/etc/slip.tty` file.

If you are using dynamic ip address allocation (have any hosts configured with the `DYNAMIC` keyword in the `/etc/slip.hosts` file, then you must configure the `/etc/slip.tty` file to list what addresses are assigned to what port. You only need this file if you wish your server to dynamically allocate addresses to users.

The file is a table that lists the *tty* devices that will support dial-in SLIP connections and the ip address that should be assigned to users who call in on that port. Its format is as follows:

```
# slip.tty    tty -> IP address mappings for dynamic SLIP
# format: /dev/tty?? xxx.xxx.xxx.xxx
#
/dev/ttyS0      192.168.0.100
/dev/ttyS1      192.168.0.101
#
```

What this table says is that callers that dial in on port `/dev/ttyS0` who have their remote address field in the `/etc/slip.hosts` file set to `DYNAMIC` will be assigned an address of `192.168.0.100`.

TOC

In this way you need only allocate one address per port for all users who do not require an dedicated address themselves. This helps you keep the number of addresses you need down to a minimum to avoid wastage.

Slip Server using *dip*.

Let me start by saying that some of the information below came from the *dip* man pages, where how to run Linux as a SLIP server is briefly documented. Please also beware that the following has been based on the *dip337o-uri.tgz* package and probably will not apply to other versions of *dip*. *dip* has an input mode of operation, where it automatically locates an entry for the user who invoked it and configures the serial line as a SLIP link according to information it finds in the `/etc/diphosts` file. This input mode of operation is activated by invoking *dip* as *diplogin*. This therefore is how you use *dip* as a SLIP server, by creating special accounts where *diplogin* is used as the login shell. The first thing you will need to do is to make a symbolic link as follows:

```
# ln -sf /usr/sbin/dip /usr/sbin/diplogin
```

You then need to add entries to both your `/etc/passwd` and your `/etc/diphosts` files. The entries you need to make are formatted as follows: To configure Linux as a SLIP server with *dip*, you need to create some special SLIP accounts for users, where *dip* (in input mode) is used as the login shell. A suggested convention is that of having all SLIP accounts begin with a capital `S', eg `Sfredm'. A sample `/etc/passwd` entry for a SLIP user looks like:

```
Sfredm:ij/SMxiTlGVCo:1004:10:Fred:/tmp:/usr/sbin/diplogin
^^         ^^        ^^  ^^   ^^   ^^   ^^
|          |         |   |    |    |    \__ diplogin as login shell
|          |         |   |    |    \_______ Home directory
|          |         |   |    \____________ User Full Name
|          |         |   \_________________ User Group ID
|          |         \______________________ User ID
|          \________________________________ Encrypted User Password
\___________________________________________ Slip User Login Name
```

After the user logs in, the *login* program, if it finds and verifies the user ok, will execute the *diplogin* command. *dip*, when invoked as *diplogin* knows that it should automatically assume that it is being used a login shell. When it is started as *diplogin* the first thing it does is use the *getuid()* function call to get the userid of whoever has invoked it. It then searches the `/etc/diphosts` file for the first entry that matches either the userid or the name of the *tty* device that the call has come in on and configures itself appropriately. By judicious decision as to whether to give a user an entry in the `diphosts` file, or whether to let the user be given the default configuration you can build your server in such a way that you can have a mix of static and dynamically assigned address users. *dip* will automatically add a `Proxy-ARP' entry if invoked in input mode, so you do not need to worry about manually adding such entries.

Configuring `/etc/diphosts`

`/etc/diphosts` is used by *dip* to lookup preset configurations for remote hosts. These remote hosts might be users dialing into your linux machine, or they might be for machines that you dial into with your linux machine. The general format for `/etc/diphosts` is as follows:

```
 ..
Suwalt::145.71.34.1:145.71.34.2:255.255.255.0:SLIP uwalt:CSLIP,1006
ttyS1::145.71.34.3:145.71.34.2:255.255.255.0:Dynamic ttyS1:CSLIP,296
```

```
..
```

The fields are:

1. `login name`: as returned by getpwuid(getuid()) or tty name.
2. `unused`: compat. with passwd
3. `Remote Address`: IP address of the calling host, either numeric or by name
4. `Local Address`: IP address of this machine, again numeric or by name
5. `Netmask`: in dotted decimal notation
6. `Comment field`: put whatever you want here.
7. `protocol`: Slip, CSlip etc.
8. `MTU`: decimal number

An example `/etc/net/diphosts` entry for a remote SLIP user might be:

```
Sfredm::145.71.34.1:145.71.34.2:255.255.255.0:SLIP uwalt:SLIP,296
```

which specifies a SLIP link with remote address of 145.71.34.1 and MTU of 296, or:

```
Sfredm::145.71.34.1:145.71.34.2:255.255.255.0:SLIP uwalt:CSLIP,1006
```

which specifies a cSLIP-capable link with remote address 145.71.34.1 and MTU of 1006.

Therefore, all users who you wish to be allowed a statically allocated dial-up IP access should have an entry in the `/etc/diphosts`. If you want users who call a particular port to have their details dynamically allocated then you must have an entry for the `tty` device and do not configure a user based entry. You should remember to configure at least one entry for each `tty` device that your dialup users use to ensure that a suitable configuration is available for them regardless of which modem they call in on.

When a user logs in they will receive a normal login and password prompt at which they should enter their SLIP-login userid and password. If these verify ok then the user will see no special messages and they should just change into SLIP mode at their end. The user should then be able to connect ok and be configured with the relevant parameters from the `diphosts` file.

SLIP server using the *dSLIP* package.

Matt Dillon `<dillon@apollo.west.oic.com>` has written a package that does not only dial-in but also dial-out SLIP. Matt's package is a combination of small programs and scripts that manage your connections for you. You will need to have *tcsh* installed as at least one of the scripts requires it. Matt supplies a binary copy of the *expect* utility as it too is needed by one of the scripts. You will most likely need some experience with *expect* to get this package working to your liking, but don't let that put you off.

Matt has written a good set of installation instructions in the README file, so I won't bother repeating them.

You can get the *dSLIP* package from its home site at: **apollo.west.oic.com**

```
/pub/linux/dillon_src/dSLIP203.tgz
```

or from: **metalab.unc.edu**

```
/pub/Linux/system/Network/serial/dSLIP203.tgz
```

Read the `README` file and create the `/etc/passwd` and `/etc/group` entries **before** doing a `make install`.

---

8. Other Network Technologies

The following subsections are specific to particular network technologies. The information contained in these sections does not necessarily apply to any other type of network technology. The topics are sorted alphabetically.

8.1 ARCNet

ARCNet device names are `arc0e', `arc1e', `arc2e' etc. or `arc0s', `arc1s', `arc2s' etc. The first card detected by the kernel is assigned `arc0e' or `arc0s' and the rest are assigned sequentially in the order they are detected. The letter at the end signifies whether you've selected ethernet encapsulation packet format or RFC1051 packet format.

Kernel Compile Options:

```
Network device support  --->
    [*] Network device support
    <*> ARCnet support
    [ ]   Enable arc0e (ARCnet "Ether-Encap" packet format)
    [ ]   Enable arc0s (ARCnet RFC1051 packet format)
```

Once you have your kernel properly built to support your ethernet card then configuration of the card is easy.

Typically you would use something like:

```
root# ifconfig arc0e 192.168.0.1 netmask 255.255.255.0 up
root# route add -net 192.168.0.0 netmask 255.255.255.0 arc0e
```

Please refer to the `/usr/src/linux/Documentation/networking/arcnet.txt` and `/usr/src/linux/Documentation/networking/arcnet-hardware.txt` files for further information.

ARCNet support was developed by Avery Pennarun, `apenwarr@foxnet.net`.

8.2 Appletalk (AF_APPLETALK)

The Appletalk support has no special device names as it uses existing network devices.

Kernel Compile Options:

```
Networking options  --->
    <*> Appletalk DDP
```

Appletalk support allows your Linux machine to interwork with Apple networks. An important use for this is to share resources such as printers and disks between both your Linux and Apple computers. Additional software is required, this is called *netatalk*. Wesley Craig `netatalk@umich.edu` represents a team called the `Research Systems Unix Group' at the University of Michigan and they have produced the *netatalk* package which provides software that implements the Appletalk protocol stack and some useful utilities. The *netatalk* package will either have been supplied with your Linux distribution, or you will have to ftp it from its home site at the University of Michigan

TOC

To build and install the package do something like:

```
user% tar xvfz .../netatalk-1.4b2.tar.Z
user% make
root# make install
```

You may want to edit the `Makefile' before calling *make* to actually compile the software. Specifically, you might want to change the DESTDIR variable which defines where the files will be installed later. The default of /usr/local/atalk is fairly safe.

Configuring the Appletalk software.

The first thing you need to do to make it all work is to ensure that the appropriate entries in the `/etc/services` file are present. The entries you need are:

```
rtmp  1/ddp   # Routing Table Maintenance Protocol
nbp   2/ddp   # Name Binding Protocol
echo  4/ddp   # AppleTalk Echo Protocol
zip   6/ddp   # Zone Information Protocol
```

The next step is to create the Appletalk configuration files in the `/usr/local/atalk/etc` directory (or wherever you installed the package).

The first file to create is the `/usr/local/atalk/etc/atalkd.conf` file. Initially this file needs only one line that gives the name of the network device that supports the network that your Apple machines are on:

```
eth0
```

The Appletalk daemon program will add extra details after it is run.

Exporting a Linux filesystems via Appletalk.

You can export filesystems from your linux machine to the network so that Apple machine on the network can share them.

To do this you must configure the `/usr/local/atalk/etc/AppleVolumes.system` file. There is another configuration file called `/usr/local/atalk/etc/AppleVolumes.default` which has exactly the same format and describes which filesystems users connecting with guest privileges will receive.

Full details on how to configure these files and what the various options are can be found in the *afpd* man page.

A simple example might look like:

```
/tmp Scratch
/home/ftp/pub "Public Area"
```

Which would export your `/tmp` filesystem as AppleShare Volume `Scratch' and your ftp public directory as AppleShare Volume `Public Area'. The volume names are not mandatory, the daemon will choose some for you, but it won't hurt to specify them anyway.

Sharing your Linux printer across Appletalk.

You can share your linux printer with your Apple machines quite simply. You need to run the *papd* program which is the Appletalk Printer Access Protocol Daemon. When you run this program it will accept requests from your Apple machines and spool the print job to your local line printer daemon for printing.

You need to edit the `/usr/local/atalk/etc/papd.conf` file to configure the daemon. The syntax of this file is the same as that of your usual `/etc/printcap` file. The name you give to the definition is registered with the Appletalk naming protocol, NBP.

A sample configuration might look like:

```
TricWriter:\
   :pr=lp:op=cg:
```

Which would make a printer named `TricWriter' available to your Appletalk network and all accepted jobs would be printed to the linux printer `lp' (as defined in the `/etc/printcap` file) using *lpd*. The entry `op=cg' says that the linux user `cg' is the operator of the printer.

Starting the appletalk software.

Ok, you should now be ready to test this basic configuration. There is an *rc.atalk* file supplied with the *netatalk* package that should work ok for you, so all you should have to do is:

```
root# /usr/local/atalk/etc/rc.atalk
```

and all should startup and run ok. You should see no error messages and the software will send messages to the console indicating each stage as it starts.

Testing the appletalk software.

To test that the software is functioning properly, go to one of your Apple machines, pull down the Apple menu, select the Chooser, click on AppleShare, and your Linux box should appear.

Caveats of the appletalk software.

- You may need to start the Appletalk support before you configure your IP network. If you have problems starting the Appletalk programs, or if after you start them you have trouble with your IP network, then try starting the Appletalk software before you run your `/etc/rc.d/rc.inet1` file.
- The *afpd* (Apple Filing Protocol Daemon) severely messes up your hard disk. Below the mount points it creates a couple of directories called ``.AppleDesktop'' and `Network Trash Folder`. Then, for each directory you access it will create a `.AppleDouble` below it so it can store resource forks, etc. So think twice before exporting `/`, you will have a great time cleaning up afterwards.
- The *afpd* program expects clear text passwords from the Macs. Security could be a problem, so be very careful when you run this daemon on a machine connected to the Internet, you have yourself to blame if somebody nasty does something bad.
- The existing diagnostic tools such as *netstat* and *ifconfig* don't support Appletalk. The raw information is available in the `/proc/net/` directory if you need it.

TOC

More information

For a much more detailed description of how to configure Appletalk for Linux refer to Anders Brownworth *Linux Netatalk-HOWTO* page at thehamptons.com.

8.3 ATM

Werner Almesberger `<werner.almesberger@lrc.di.epfl.ch>` is managing a project to provide Asynchronous Transfer Mode support for Linux. Current information on the status of the project may be obtained from: lrcwww.epfl.ch.

8.4 AX25 (AF_AX25)

AX.25 device names are `sl0', `sl1', etc. in `2.0.*` kernels or `ax0', `ax1', etc. in `2.1.*` kernels.

Kernel Compile Options:

```
Networking options  --->
    [*] Amateur Radio AX.25 Level 2
```

The AX25, Netrom and Rose protocols are covered by the AX25-HOWTO. These protocols are used by Amateur Radio Operators world wide in packet radio experimentation.

Most of the work for implementation of these protocols has been done by Jonathon Naylor, `jsn@cs.nott.ac.uk`.

8.5 DECNet

Support for DECNet is currently being worked on. You should expect it to appear in a late `2.1.*` kernel.

8.6 FDDI

FDDI device names are `fddi0', `fddi1', `fddi2' etc. The first card detected by the kernel is assigned `fddi0' and the rest are assigned sequentially in the order they are detected.

Larry Stefani, `lstefani@ultranet.com`, has developed a driver for the Digital Equipment Corporation FDDI EISA and PCI cards.

Kernel Compile Options:

```
Network device support  --->
    [*] FDDI driver support
    [*] Digital DEFEA and DEFPA adapter support
```

When you have your kernel built to support the FDDI driver and installed, configuration of the FDDI interface is almost identical to that of an ethernet interface. You just specify the appropriate FDDI interface name in the *ifconfig* and *route* commands.

8.7 Frame Relay

TOC

The Frame Relay device names are `dlci00', `dlci01' etc for the DLCI encapsulation devices and `sdla0', `sdla1' etc for the FRAD(s).

Frame Relay is a new networking technology that is designed to suit data communications traffic that is of a `bursty' or intermittent nature. You connect to a Frame Relay network using a Frame Relay Access Device (FRAD). The Linux Frame Relay supports IP over Frame Relay as described in RFC-1490.

Kernel Compile Options:

```
Network device support  --->
    <*> Frame relay DLCI support (EXPERIMENTAL)
    (24)   Max open DLCI
    (8)   Max DLCI per device
    <*>   SDLA (Sangoma S502/S508) support
```

Mike McLagan, `mike.mclagan@linux.org`, developed the Frame Relay support and configuration tools.

Currently the only FRAD supported are the Sangoma Technologies `S502A`, `S502E` and `S508`.

To configure the FRAD and DLCI devices after you have rebuilt your kernel you will need the Frame Relay configuration tools. These are available from ftp.invlogic.com. Compiling and installing the tools is straightforward, but the lack of a top level Makefile makes it a fairly manual process:

```
user% tar xvfz .../frad-0.15.tgz
user% cd frad-0.15
user% for i in common dlci frad; make -C $i clean; make -C $i; done
root# mkdir /etc/frad
root# install -m 644 -o root -g root bin/*.sfm /etc/frad
root# install -m 700 -o root -g root frad/fradcfg /sbin
rppt# install -m 700 -o root -g root dlci/dlcicfg /sbin
```

Note that the previous commands use *sh* syntax, if you use a *csh* flavour instead (like *tcsh*), the *for* loop will look different.

After installing the tools you need to create an `/etc/frad/router.conf` file. You can use this template, which is a modified version of one of the example files:

```
# /etc/frad/router.conf
# This is a template configuration for frame relay.
# All tags are included. The default values are based on the code
# supplied with the DOS drivers for the Sangoma S502A card.
#
# A '#' anywhere in a line constitutes a comment
# Blanks are ignored (you can indent with tabs too)
# Unknown [] entries and unknown keys are ignored
#

[Devices]
Count=1                 # number of devices to configure
```

TOC

```
Dev_1=sdla0             # the name of a device
#Dev_2=sdla1            # the name of a device

# Specified here, these are applied to all devices and can be overridden for
# each individual board.
#
Access=CPE
Clock=Internal
KBaud=64
Flags=TX
#
# MTU=1500              # Maximum transmit IFrame length, default is 4096
# T391=10               # T391 value    5 - 30, default is 10
# T392=15               # T392 value    5 - 30, default is 15
# N391=6                # N391 value    1 - 255, default is 6
# N392=3                # N392 value    1 - 10, default is 3
# N393=4                # N393 value    1 - 10, default is 4

# Specified here, these set the defaults for all boards
# CIRfwd=16             # CIR forward   1 - 64
# Bc_fwd=16             # Bc forward    1 - 512
# Be_fwd=0              # Be forward    0 - 511
# CIRbak=16             # CIR backward  1 - 64
# Bc_bak=16             # Bc backward   1 - 512
# Be_bak=0              # Be backward   0 - 511


#
#
# Device specific configuration
#
#

#
# The first device is a Sangoma S502E
#
[sdla0]
Type=Sangoma            # Type of the device to configure, currently only
                        # SANGOMA is recognized
#
# These keys are specific to the 'Sangoma' type
#
# The type of Sangoma board - S502A, S502E, S508
Board=S502E
#
# The name of the test firmware for the Sangoma board
# Testware=/usr/src/frad-0.10/bin/sdla_tst.502
#
# The name of the FR firmware
# Firmware=/usr/src/frad-0.10/bin/frm_rel.502
#
Port=360                # Port for this particular card
Mem=C8                  # Address of memory window, A0-EE, depending on card
```

```
IRQ=5                   # IRQ number, do not supply for S502A
DLCIs=1                 # Number of DLCI's attached to this device
DLCI_1=16               # DLCI #1's number, 16 - 991
# DLCI_2=17
# DLCI_3=18
# DLCI_4=19
# DLCI_5=20
#
# Specified here, these apply to this device only,
# and override defaults from above
#
# Access=CPE            # CPE or NODE, default is CPE
# Flags=TXIgnore,RXIgnore,BufferFrames,DropAborted,Stats,MCI,AutoDLCI
# Clock=Internal        # External or Internal, default is Internal
# Baud=128              # Specified baud rate of attached CSU/DSU
# MTU=2048              # Maximum transmit IFrame length, default is 4096
# T391=10               # T391 value    5 - 30, default is 10
# T392=15               # T392 value    5 - 30, default is 15
# N391=6                # N391 value    1 - 255, default is 6
# N392=3                # N392 value    1 - 10, default is 3
# N393=4                # N393 value    1 - 10, default is 4

#
# The second device is some other card
#
# [sdla1]
# Type=FancyCard        # Type of the device to configure.
# Board=                # Type of Sangoma board
# Key=Value             # values specific to this type of device


#
# DLCI Default configuration parameters
# These may be overridden in the DLCI specific configurations
#
CIRfwd=64               # CIR forward   1 - 64
# Bc_fwd=16             # Bc forward    1 - 512
# Be_fwd=0              # Be forward    0 - 511
# CIRbak=16             # CIR backward  1 - 64
# Bc_bak=16             # Bc backward   1 - 512
# Be_bak=0              # Be backward   0 - 511

#
# DLCI Configuration
# These are all optional. The naming convention is
# [DLCI_D<devicenum>_<DLCI_Num>]
#

[DLCI_D1_16]
# IP=
# Net=
# Mask=
```

```
# Flags defined by Sangoma: TXIgnore,RXIgnore,BufferFrames
# DLCIFlags=TXIgnore,RXIgnore,BufferFrames
# CIRfwd=64
# Bc_fwd=512
# Be_fwd=0
# CIRbak=64
# Bc_bak=512
# Be_bak=0

[DLCI_D2_16]
# IP=
# Net=
# Mask=
# Flags defined by Sangoma: TXIgnore,RXIgnore,BufferFrames
# DLCIFlags=TXIgnore,RXIgnore,BufferFrames
# CIRfwd=16
# Bc_fwd=16
# Be_fwd=0
# CIRbak=16
# Bc_bak=16
# Be_bak=0
```

When you've built your `/etc/frad/router.conf` file the only step remaining is to configure the actual devices themselves. This is only a little trickier than a normal network device configuration, you need to remember to bring up the FRAD device before the DLCI encapsulation devices. These commands are best hosted in a shell script, due to their number:

```
#!/bin/sh
# Configure the frad hardware and the DLCI parameters
/sbin/fradcfg /etc/frad/router.conf || exit 1
/sbin/dlcicfg file /etc/frad/router.conf
#
# Bring up the FRAD device
ifconfig sdla0 up
#
# Configure the DLCI encapsulation interfaces and routing
ifconfig dlci00 192.168.10.1 pointopoint 192.168.10.2 up
route add -net 192.168.10.0 netmask 255.255.255.0 dlci00
#
ifconfig dlci01 192.168.11.1 pointopoint 192.168.11.2 up
route add -net 192.168.11.0 netmask 255.255.255.0 dlci00
#
route add default dev dlci00
#
```

8.8 IPX (AF_IPX)

The IPX protocol is most commonly utilized in Novell NetWare(tm) local area network environments. Linux includes support for this protocol and may be configured to act as a network endpoint, or as a router for IPX.

Kernel Compile Options:

```
Networking options  --->
    [*] The IPX protocol
    [ ] Full internal IPX network
```

The IPX protocol and the NCPFS are covered in greater depth in the IPX-HOWTO.

8.9 NetRom (AF_NETROM)

NetRom device names are `nr0', `nr1', etc.

Kernel Compile Options:

```
Networking options  --->
    [*] Amateur Radio AX.25 Level 2
    [*] Amateur Radio NET/ROM
```

The AX25, Netrom and Rose protocols are covered by the AX25-HOWTO. These protocols are used by Amateur Radio Operators world wide in packet radio experimentation.

Most of the work for implementation of these protocols has been done by Jonathon Naylor, `jsn@cs.nott.ac.uk`.

8.10 Rose protocol (AF_ROSE)

Rose device names are `rs0', `rs1', etc. in `2.1.*` kernels. Rose is available in the `2.1.*` kernels.

Kernel Compile Options:

```
Networking options  --->
    [*] Amateur Radio AX.25 Level 2
    <*> Amateur Radio X.25 PLP (Rose)
```

The AX25, Netrom and Rose protocols are covered by the AX25-HOWTO. These protocols are used by Amateur Radio Operators world wide in packet radio experimentation.

Most of the work for implementation of these protocols has been done by Jonathon Naylor, `jsn@cs.nott.ac.uk`.

8.11 SAMBA - `NetBEUI', `NetBios', `CIFS' support.

SAMBA is an implementation of the Session Management Block protocol. Samba allows Microsoft and other systems to mount and use your disks and printers.

SAMBA and its configuration are covered in detail in the SMB-HOWTO.

8.12 STRIP support (Starmode Radio IP)

STRIP device names are `st0', `st1', etc.

Kernel Compile Options:

```
Network device support  --->
        [*] Network device support
        ....
        [*] Radio network interfaces
        < > STRIP (Metricom starmode radio IP)
```

STRIP is a protocol designed specifically for a range of Metricom radio modems for a research project being conducted by Stanford University called the MosquitoNet Project. There is a lot of interesting reading here, even if you aren't directly interested in the project.

The Metricom radios connect to a serial port, employ spread spectrum technology and are typically capable of about 100kbps. Information on the Metricom radios is available from the: Metricom Web Server.

At present the standard network tools and utilities do not support the STRIP driver, so you will have to download some customized tools from the MosquitoNet web server. Details on what software you need is available at the: MosquitoNet STRIP Page.

A summary of configuration is that you use a modified *slattach* program to set the line discipline of a serial tty device to STRIP and then configure the resulting `st[0-9]' device as you would for ethernet with one important exception, for technical reasons STRIP does not support the ARP protocol, so you must manually configure the ARP entries for each of the hosts on your subnet. This shouldn't prove too onerous.

8.13 Token Ring

Token ring device names are `tr0', `tr1' etc. Token Ring is an IBM standard LAN protocol that avoids collisions by providing a mechanism that allows only one station on the LAN the right to transmit at a time. A `token' is held by one station at a time and the station holding the token is the only station allowed to transmit. When it has transmitted its data it passes the token onto the next station. The token loops amongst all active stations, hence the name `Token Ring'.

Kernel Compile Options:

```
Network device support  --->
        [*] Network device support
        ....
        [*] Token Ring driver support
        < > IBM Tropic chipset based adaptor support
```

Configuration of token ring is identical to that of ethernet with the exception of the network device name to configure.

8.14 X.25

X.25 is a circuit based packet switching protocol defined by the `C.C.I.T.T.` (a standards body recognized by Telecommunications companies in most parts of the world). An implementation of X.25 and LAPB are being worked on and recent `2.1.*` kernels include the work in progress.

Jonathon Naylor `jsn@cs.nott.ac.uk` is leading the development and a mailing list has been established to discuss Linux X.25 related matters. To subscribe send a message to: `majordomo@vger.rutgers.edu` with the text "`subscribe linux-x25`" in the body of the message.

Early versions of the configuration tools may be obtained from Jonathon's ftp site at ftp.cs.nott.ac.uk.

8.15 WaveLan Card

Wavelan device names are `eth0', `eth1', etc.

Kernel Compile Options:

```
Network device support  --->
        [*] Network device support
        ....
        [*] Radio network interfaces
        ....
        <*> WaveLAN support
```

The WaveLAN card is a spread spectrum wireless lan card. The card looks very like an ethernet card in practice and is configured in much the same way.

You can get information on the Wavelan card from Wavelan.com.

---

9. Cables and Cabling

Those of you handy with a soldering iron may want to build your own cables to interconnect two linux machines. The following cabling diagrams should assist you in this.

9.1 Serial NULL Modem cable

Not all NULL modem cables are alike. Many null modem cables do little more than trick your computer into thinking all the appropriate signals are present and swap transmit and receive data. This is ok but means that you must use software flow control (XON/XOFF) which is less efficient than hardware flow control. The following cable provides the best possible signalling between machines and allows you to use hardware (RTS/CTS) flow control.

```
Pin Name  Pin                                Pin
Tx Data    2  ------------------------------  3
Rx Data    3  ------------------------------  2
RTS        4  ------------------------------  5
CTS        5  ------------------------------  4
Ground     7  ------------------------------  7
DTR        20 -\----------------------------  8
DSR        6  -/
RLSD/DCD   8  ----------------------------/-  20
                                           \-  6
```

9.2 Parallel port cable (PLIP cable)

If you intend to use the PLIP protocol between two machines then this cable will work for you irrespective of what sort of parallel ports you have installed.

```
Pin Name    pin            pin
STROBE      1*
D0->ERROR   2  ----------- 15
D1->SLCT    3  ----------- 13
D2->PAPOUT  4  ----------- 12
D3->ACK     5  ----------- 10
D4->BUSY    6  ----------- 11
D5          7*
D6          8*
D7          9*
ACK->D3     10 ----------- 5
```

```
BUSY->D4     11 ----------- 6
PAPOUT->D2   12 ----------- 4
SLCT->D1     13 ----------- 3
FEED         14*
ERROR->D0    15 ----------- 2
INIT         16*
SLCTIN       17*
GROUND       25 ----------- 25
```

Notes:

- Do not connect the pins marked with an asterisk `*'.
- Extra grounds are 18,19,20,21,22,23 and 24.
- If the cable you are using has a metallic shield, it should be connected to the metallic DB-25 shell at **one end only**.

Warning: A miswired PLIP cable can destroy your controller card. Be very careful and double check every connection to ensure you don't cause yourself any unnecessary work or heartache.

While you may be able to run PLIP cables for long distances, you should avoid it if you can. The specifications for the cable allow for a cable length of about 1 metre or so. Please be very careful when running long plip cables as sources of strong electromagnetic fields such as lightning, power lines and radio transmitters can interfere with and sometimes even damage your controller. If you really want to connect two of your computers over a large distance you really should be looking at obtaining a pair of thin-net ethernet cards and running some coaxial cable.

9.3 10base2 (thin coax) Ethernet Cabling

10base2 is an ethernet cabling standard that specifies the use of 52 ohm coaxial cable with a diameter of about 5 millimeters. There are a couple of important rules to remember when interconnecting machines with 10base2 cabling. The first is that you must use terminators at **both ends** of the cabling. A terminator is a 52 ohm resistor that helps to ensure that the signal is absorbed and not reflected when it reaches the end of the cable. Without a terminator at each end of the cabling you may find that the ethernet is unreliable or doesn't work at all. Normally you'd use `T pieces' to interconnect the machines, so that you end up with something that looks like:

```
|==========T=============T=============T==========T==========|
           |             |             |          |
           |             |             |          |
         -----         -----         -----      -----
         |   |         |   |         |   |      |   |
         -----         -----         -----      -----
```

where the `|' at either end represents a terminator, the `=====' represents a length of coaxial cable with BNC plugs at either end and the `T' represents a `T piece' connector. You should keep the length of cable between the `T piece' and the actual ethernet card in the PC as short as possible, ideally the `T

piece' will be plugged directly into the ethernet card.

9.4 Twisted Pair Ethernet Cable

If you have only two twisted pair ethernet cards and you wish to connect them you do not require a hub. You can cable the two cards directly together. A diagram showing how to do this is included in the Ethernet-HOWTO

---

10. Glossary of Terms used in this document.

The following is a list of some of the most important terms used in this document.

ARP

This is an acronym for the *Address Resolution Protocol* and this is how a network machine associates an IP Address with a hardware address.

ATM

This is an acronym for *Asynchronous Transfer Mode*. An ATM network packages data into standard size blocks which it can convey efficiently from point to point. ATM is a circuit switched packet network technology.

client

This is usually the piece of software at the end of a system where the user is. There are exceptions to this, for example, in the X11 window system it is actually the server with the user and the client runs on the remote machine. The client is the program or end of a system that is receiving the service provided by the server. In the case of *peer to peer* systems such as *slip* or *ppp* the client is taken to be the end that initiates the connection and the remote end, being called, is taken to be the server.

datagram

A datagram is a discrete package of data and headers which contain addresses, which is the basic unit of transmission across an IP network. You might also hear this called a `packet'.

DLCI

The DLCI is the Data Link Connection Identifier and is used to identify a unique virtual point to point connection via a Frame Relay network. The DLCI's are normally assigned by the Frame Relay network provider.

Frame Relay

Frame Relay is a network technology ideally suited to carrying traffic that is of bursty or sporadic nature. Network costs are reduced by having many Frame Relay customer sharing the same network capacity and relying on them wanting to make use of the network at slightly different times.

Hardware address

This is a number that uniquely identifies a host in a physical network at the media access layer. Examples of this are *Ethernet Addresses* and *AX.25 Addresses*.

ISDN

This is an acronym for *Integrated Services Digital Network*. ISDN provides a standardized means by which Telecommunications companies may deliver either voice or data information to a

TOC

customers premises. Technically ISDN is a circuit switched data network.

ISP

This is an acronym of Internet Service Provider. These are organizations or companies that provide people with network connectivity to the Internet.

IP address

This is a number that uniquely identifies a TCP/IP host on the network. The address is 4 bytes long and is usually represented in what is called the "dotted decimal notation", where each byte is represented in decimal from with dots `.' between them.

MSS

The Maximum Segment Size (*MSS*) is the largest quantity of data that can be transmitted at one time. If you want to prevent local fragmentation MSS would equal MTU-IP header.

MTU

The Maximum Transmission Unit (*MTU*) is a parameter that determines the largest datagram than can be transmitted by an IP interface without it needing to be broken down into smaller units. The MTU should be larger than the largest datagram you wish to transmit unfragmented. Note, this only prevents fragmentation locally, some other link in the path may have a smaller MTU and the datagram will be fragmented there. Typical values are 1500 bytes for an ethernet interface, or 576 bytes for a SLIP interface.

route

The *route* is the path that your datagrams take through the network to reach their destination.

server

This is usually the piece of software or end of a system remote from the user. The server provides some service to one or many clients. Examples of servers include *ftp*, *Networked File System*, or *Domain Name Server*. In the case of *peer to peer* systems such as *slip* or *ppp* the server is taken to be the end of the link that is called and the end calling is taken to be the client.

window

The *window* is the largest amount of data that the receiving end can accept at a given point in time.

---

11. Linux for an ISP ?

If you are interested in using Linux for ISP purposes the I recommend you take a look at the Linux ISP homepage for a good list of pointers to information you might need and use.

12. Acknowledgements

I'd like to thank the following people for their contributions to this document (in no particular order): Terry Dawson, Axel Boldt, Arnt Gulbrandsen, Gary Allpike, Cees de Groot, Alan Cox, Jonathon Naylor, Claes Ensson, Ron Nessim, John Minack, Jean-Pierre Cocatrix, Erez Strauss.

13. Copyright.

Copyright Information

The NET-3-HOWTO, information on how to install and configure networking support for Linux.
Copyright (c) 1997 Terry Dawson, 1998 Alessandro Rubini, 1999 {POET} - LinuxPorts

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the: Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

NFS HOWTO

Nicolai Langfeldt `janl@math.uio.no`

v0.7, 3 November 1997

HOWTO set up NFS clients and servers.

1. Preamble

- 1.1 Legal stuff
- 1.2 Other stuff
- 1.3 Dedication

2. README.first

3. Setting up a NFS server

- 3.1 Prerequisites
- 3.2 First step
- 3.3 The portmapper
- 3.4 Mountd and nfsd

4. Setting up a NFS client

- 4.1 Mount options
- 4.2 Optimizing NFS

5. NFS over slow lines

TOC

6. Security and NFS

- 6.1 Client Security
- 6.2 Server security: nfsd
- 6.3 Server security: the portmapper
- 6.4 NFS and firewalls
- 6.5 Summary

7. Mount Checklist

8. FAQs

9. Exporting filesystems

- 9.1 IRIX, HP-UX, Digital-UNIX, Ultrix, SunOS 4 (Solaris 1), AIX
- 9.2 Solaris 2

10. PC-NFS

---

1. Preamble

1.1 Legal stuff

(C)opyright 1997 Nicolai Langfeldt. Do not modify without amending copyright, distribute freely but retain this paragraph. The FAQ section is based on a NFS FAQ compiled by Alan Cox. The Checklist section is based on a mount problem checklist compiled by the IBM Corporation.

1.2 Other stuff

This will never be a finished document, please send me mail about your problems and successes, it can make this a better HOWTO. Please send money, comments and/or questions to janl@math.uio.no. If you send E-mail please *make sure* that the return address is correct and working, I get *a lot* of E-mail and figuring out your e-mail address can be a lot of work. Please.

If you want to translate this HOWTO please notify me so I can keep track of what languages I have been published in :-).

Curses and Thanks to Olaf Kirch who got me to write this and then gave good suggestions for it :-)

This HOWTO covers NFS in the 2.0 versions of the kernel. There are significant enhancements, and changes, of NFS in the 2.1 versions of the kernel.

1.3 Dedication

This HOWTO is dedicated to Anne Line Norheim Langfeldt. Though she will probably never read it since she's not that kind of girl.

2. README.first

NFS, the Network File System has three important characteristics:

- It makes sharing of files over a network possible.
- It mostly works well enough.
- It opens a can of security risks that are well understood by crackers, and easily exploited to get access (read, write and delete) to all your files.

I'll say something on both issues in this HOWTO. Please make sure you read the security section of this HOWTO, and you will be vulnerable to fewer silly security risks. The passages about security will at times be pretty technical and require some knowledge about IP networking and the terms used. If you don't recognize the terms you can either go back and check the networking HOWTO, wing it, or get a book about TCP/IP network administration to familiarize yourself with TCP/IP. That's a good idea anyway if you're administrating UNIX/Linux machines. A very good book on the subject is *TCP/IP Network Administration* by Craig Hunt, published by O'Reilly & Associates, Inc. And after you've read it and understood it you'll have higher value on the job market, you can't loose ;-)

There are two sections to help you troubleshoot NFS, called *Mount Checklist* and *FAQs*. Please refer to them if something dosn't work as advertized.

3. Setting up a NFS server

3.1 Prerequisites

Before you continue reading this HOWTO you will need to be able to telnet back and forth between the machine you're using as server and the client. If that does not work you need to check the networking/NET-2 HOWTO and set up networking properly.

3.2 First step

Before we can do anything else we need a NFS server set up. If you're part of a department or university network there are likely numerous NFS servers already set up. If they will let you get access to them, or indeed, if you're reading this HOWTO to get access to one of them you obviously don't need to read this section and can just skip ahead to the section on setting up a NFS client

If you need to set up a non-Linux box as server you will have to read the system manual(s) to discover how to enable NFS serving and export of file systems through NFS. There is a separate section in this HOWTO on how to do it on many different systems. After you have figured all that out you can continue reading the next section of this HOWTO. Or read more of this section since some of the things I will say are relevant no matter what kind of machine you use as server.

Those of you still reading will need to set up a number of programs.

3.3 The portmapper

The portmapper on Linux is called either `portmap` or `rpc.portmap`. The man page on my system says it is a "DARPA port to RPC program number mapper". It is the first security holes you'll open reading this HOWTO. Description of how to close one of the holes is in the security section. Which I, again, urge you to read.

Start the portmapper. It's either called `portmap` or `rpc.portmap` and it should live in the `/usr/sbin` directory (on some machines it's called rpcbind). You can start it by hand now, but it will need to be started every time you boot your machine so you need to make/edit the rc scripts. Your rc scripts are explained more closely in the init man page, they usually reside in `/etc/rc.d`, `/etc/init.d` or `/etc/rc.d/init.d`. If there is a script called something like `inet` it's probably the right script to edit. But, what to write or do is outside the scope of this HOWTO. Start portmap, and check that it lives by running `ps aux`. It does? Good.

3.4 Mountd and nfsd

TOC

The next programs we need running are mountd and nfsd. But first we'll edit another file. `/etc/exports` this time. Say I want the file system `/mn/eris/local` which lives on the machine `eris` to be available to the machine called `apollon`. Then I'd put this in `/etc/exports` on eris:

```
/mn/eris/local  apollon(rw)
```

The above line gives apollon read/write access to `/mn/eris/local`. Instead of `rw` it could say `ro` which means read only (if you put nothing it defaults to read only). There are other options you can give it, and I will discuss some security related ones later. They are all enumerated in the `exports` man page which you should have read at least once in your life. There are also better ways than listing all the hosts in the exports file. You can for example use net groups if you are running NIS (or NYS) (NIS was known as YP), and always specify domain wild cards and IP-subnets as hosts that are allowed to mount something. But you should consider who can get access to the server in unauthorized ways if you use such blanket authorizations.

Note: This exports file is not the same syntax that other Unixes use. There is a separate section in this HOWTO about other Unixes `exports` files.

Now we're set to start mountd (or maybe it's called `rpc.mountd` and then nfsd (which could be called `rpc.nfsd`). They will both read the exports file.

If you edit `/etc/exports` you will have to make sure nfsd and mountd knows that the files have changed. The traditonal way is to run `exportfs`. Many Linux distributions lack a exportfs program. If you're exportfs-less you can install this script on your machine:

```
#!/bin/sh
killall -HUP /usr/sbin/rpc.mountd
killall -HUP /usr/sbin/rpc.nfsd
echo re-exported file systems
```

Save it in, say, `/usr/sbin/exportfs`, and don't forget to `chmod a+rx` it. Now, whenever you change your exports file, you run exportfs after, as root.

Now you should check that mountd and nfsd are running properly. First with `rpcinfo -p`. It should show something like this:

```
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
```

```
    100005    1   udp    745  mountd
    100005    1   tcp    747  mountd
    100003    2   udp   2049  nfs
    100003    2   tcp   2049  nfs
```

As you see the portmapper has announced it's services, and so has mountd and nfsd.

If you get `rpcinfo: can't contact portmapper: RPC: Remote system error - Connection refused` or something similar instead then the portmapper isn't running. Fix it. If you get `No remote programs registered.` then either the portmapper doesn't want to talk to you, or something is broken. Kill nfsd, mountd, and the portmapper and try the ignition sequence again.

After checking that the portmapper reports the services you can check with ps too. The portmapper will continue to report the services even after the programs that extend them have crashed. So a ps check can be smart if something seems broken.

Of course, you will need to modify your system rc files to start mountd and nfsd as well as the portmapper when you boot. It is very likely that the scripts already exist on your machine, you just have to uncomment the critical section or activate it for the correct init run levels.

Man pages you should be familiar with now: portmap, mountd, nfsd, and exports.

Well, if you did everything exactly like I said you should you're all set to start on the NFS client.

---

4. Setting up a NFS client

First you will need a kernel with the NFS file system either compiled in or available as a module. This is configured before you compile the kernel. If you have never compiled a kernel before you might need to check the kernel HOWTO and figure it out. If you're using a very cool distribution (like Red Hat) and you've never fiddled with the kernel or modules on it (and thus ruined it ;-), nfs is likely automagicaly available to you.

You can now, at a root prompt, enter a appropriate mount command and the file system will appear. Continuing the example in the previous section we want to mount `/mn/eris/local` from eris. This is done with this command:

```
mount -o rsize=1024,wsize=1024 eris:/mn/eris/local /mnt
```

(We'll get back to the rsize and wsize options.) The file system is now available under `/mnt` and you can `cd` there, and `ls` in it, and look at the individual files. You will notice that it's not as fast as a local file system, but a lot more convenient than ftp. If, instead of mounting the file system, mount produces a error message like `mount: eris:/mn/eris/local failed, reason given by server: Permission denied` then the exports file is wrong, or you forgot to run exportfs after editing the exports file. If it says `mount clntudp_create: RPC: Program not registered` it means that nfsd or mountd is not running on the server.

To get rid of the file system you can say

```
umount /mnt
```

To make the system mount a nfs file system upon boot you edit `/etc/fstab` in the normal manner. For our example a line such as this is required:

```
# device       mountpoint     fs-type     options              dump fsckorder
...
eris:/mn/eris/local  /mnt    nfs        rsize=1024,wsize=1024 0    0
...
```

That's all there is too it, almost. Read on please.

4.1 Mount options

There are some options you should consider adding at once. They govern the way the NFS client handles a server crash or network outage. One of the cool things about NFS is that it can handle this gracefully. If you set up the clients right. There are two distinct failure modes:

soft

The NFS client will report and error to the process accessing a file on a NFS mounted file system. Some programs can handle this with composure, most won't. I cannot recommend using this setting.

hard

The program accessing a file on a NFS mounted file system will hang when the server crashes. The process cannot be interrupted or killed unless you also specify `intr`. When the NFS server is back online the program will continue undisturbed from where it were. This is probably what you want. I recommend using `hard,intr` on all NFS mounted file systems.

Picking up the previous example, this is now your fstab entry:

```
# device      mountpoint     fs-type    options                    dump fsckorder
...
eris:/mn/eris/local  /mnt    nfs        rsize=1024,wsize=1024,hard,intr 0 0
...
```

4.2 Optimizing NFS

Normally, if no rsize and wsize options are specified NFS will read and write in chunks of 4096 or 8192 bytes. Some combinations of Linux kernels and network cards cannot handle that large blocks, and it might not be optimal, anyway. So we'll want to experiment and find a rsize and wsize that works and is as fast as possible. You can test the speed of your options with some simple commands. Given the mount command above and that you have write access to the disk you can do this to test the sequential write performance:

```
time dd if=/dev/zero of=/mnt/testfile bs=16k count=4096
```

This creates a 64Mb file of zeroed bytes (which should be large enough that caching is no significant part of any performance perceived, use a larger file if you have a lot of memory). Do it a couple (5-10?) of times and average the times. It is the `elapsed' or `wall clock' time that's most interesting in this connection. Then you can test the read performance by reading back the file:

```
time dd if=/mnt/testfile of=/dev/null bs=16k
```

do that a couple of times and average. Then umount, and mount again with a larger rsize and wsize. They should probably be multiples of 1024, and not larger than 16384 bytes since that's the maximum size in NFS version 2. Directly after mounting with a larger size cd into the mounted file system and do things like ls, explore the fs a bit to make sure everything is as it should. If the rsize/wsize is too large the symptoms are *very* odd and not 100% obvious. A typical symptom is incomplete file lists when doing 'ls', and no error messages. Or reading files failing mysteriously with no error messages. After establishing that the given rsize/wsize works you can do the speed tests again. Different server platforms are likely to have different optimal sizes. SunOS and Solaris is reputedly a lot faster with 4096 byte blocks than with anything else.

Newer Linux kernels (since 1.3 sometime) perform read-ahead for rsizes larger or equal to the machine page size. On Intel CPUs the page size is 4096 bytes. Read ahead will *significantly* increase the NFS read performance. So on a Intel machine you will want 4096 byte rsize if at all possible.

TOC

Remember to edit `/etc/fstab` to reflect the rsize/wsize you found.

A trick to increase NFS write performance is to disable synchronous writes on the server. The NFS specification states that NFS write requests shall not be considered finished before the data written is on a non-volatile medium (normally the disk). This restricts the write performance somewhat, asynchronous writes will speed NFS writes up. The Linux nfsd has never done synchronous writes since the Linux file system implementation does not lend itself to this, but on non-Linux servers you can increase the performance this way with this in your exports file:

```
/dir    -async,access=linuxbox
```

---

or something similar. Please refer to the exports man page on the machine in question. Please note that this increases the risk of data loss.

---

5. NFS over slow lines

Slow lines include Modems, ISDN and quite possibly other long distance connections.

This section is based on knowledge about the used protocols but no actual experiments. My home computer has been down for 6 months (bad HD, low on cash) and so I have had no modem connection to test this with. Please let me hear from you if try this :-)

The first thing to remember is that NFS is a slow protocol. It has high overhead. Using NFS is almost like using kermit to transfer files. It's *slow*. Almost anything is faster than NFS. FTP is faster. HTTP is faster. rcp is faster. ssh is faster.

Still determined to try it out? Ok.

NFS' default parameters are for quite fast, low latency, lines. If you use these default parameters over high latency lines it can cause NFS to report errors, abort operations, pretend that files are shorter than they really are, and act mysteriously in other ways.

The first thing to do is *not* to use the `soft` mount option. This will cause timeouts to return errors to the software, which will, most likely not handle the situation at all well. This is a good way to get for mysterious failures. Instead use the `hard` mount option. When `hard` is active timeouts causes infinite retries instead of aborting whatever it was the software wanted to do. This is what you want. Really.

The next thing to do is to tweak the timeo and retrans mount options. They are described in the nfs(5) man page, but here is a copy:

```
timeo=n          The value in tenths of a second before
                 sending the first retransmission after an
                 RPC timeout.  The default value is 7 tenths
                 of a second.  After the first timeout, the
                 timeout is doubled after each successive
                 timeout until a maximum timeout of 60 sec-
                 onds is reached or the enough retransmis-
                 sions have occured to cause a major time-
                 out.  Then, if the filesystem is hard
                 mounted, each new timeout cascade restarts
                 at twice the initial value of the previous
                 cascade, again doubling at each retransmis-
                 sion.  The maximum timeout is always 60
                 seconds.  Better overall performance may be
                 achieved by increasing the timeout when
                 mounting on a busy network, to a slow
```

```
                    server, or through several routers or gate-
                    ways.

      retrans=n     The number of minor timeouts  and  retrans-
                    missions  that  must  occur  before a major
                    timeout occurs.  The default is 3 timeouts.
                    When a major timeout occurs, the file oper-
                    ation is either aborted or  a  "server  not
                    responding"  message is printed on the con-
                    sole.
```

In other words: If a reply is not received within the 0.7 second (700ms) timeout the NFS client will repeat the request and double the timeout to 1.4 seconds. If the reply does not appear within the 1.4 seconds the request is repeated again and the timeout doubled again, to 2.8 seconds.

A lines speed can be measured with ping with the same packet size as your rsize/wsize options.

```
$ ping -s 8192 lugulbanda
PING lugulbanda.uio.no (129.240.222.99): 8192 data bytes
8200 bytes from 129.240.222.99: icmp_seq=0 ttl=64 time=15.2 ms
8200 bytes from 129.240.222.99: icmp_seq=1 ttl=64 time=15.9 ms
8200 bytes from 129.240.222.99: icmp_seq=2 ttl=64 time=14.9 ms
8200 bytes from 129.240.222.99: icmp_seq=3 ttl=64 time=14.9 ms
8200 bytes from 129.240.222.99: icmp_seq=4 ttl=64 time=15.0 ms

--- lugulbanda.uio.no ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 14.9/15.1/15.9 ms
```

The time here is how long the ping packet took to get back and forth to lugulbanda. 15ms is quite fast. Over a 28.000 bps line you can expect something like 4000-5000ms, and if the line is otherwise loaded this time will be even higher, easily double. When this time is high we say that there is 'high latency'. Generally, for larger packets and for more loaded lines the latency will tend to increase. Increase timeo suitably for your line and load. And since the latency increases when you use the line for other things: If you ever want to use FTP and NFS at the same time you should try measuring ping times while using FTP to transfer files.

---

6. Security and NFS

I am by no means a computer security expert. But I do have a *little* advice for the security conscious. But be warned: This is by no means a complete list of NFS related problems and if you think you're safe once you're read and implemented all this I have a bridge I want to sell you.

This section is probably of no concern if you are on a *closed* network where you trust all the users, and no-one you don't trust can get access to machines on the network. I.e., there should be no way to dial into the network, and it should in no way be connected to other networks where you don't trust everyone using it as well as the security. Do you think I sound paranoid? I'm not at all paranoid. This is just *basic* security advice. And remember, the things I say here is just the start of it. A *secure* site needs a diligent and knowledgeable admin that knows where to find information about current and potential security problems.

NFS has a basic problem in that the client, if not told otherwise, will trust the NFS server and vice versa. This can be bad. It means that if the server's root account is broken into it can be quite easy to break into the client's root account as well. And vice versa. There are a couple of coping strategies for this, which we'll get back to.

Something you should read is the CERT advisories on NFS, most of the text below deals with issues CERT has written advisories about. See ftp.cert.org/01-README for a up to date list of CERT advisories. Here are some NFS related advisories:

```
CA-91:21.SunOS.NFS.Jumbo.and.fsirand                          12/06/91
     Vulnerabilities concerning Sun Microsystems, Inc. (Sun) Network
     File System (NFS) and the fsirand program.  These vulnerabilities
     affect SunOS versions 4.1.1, 4.1, and 4.0.3 on all architectures.
     Patches are available for SunOS 4.1.1.  An initial patch for SunOS
     4.1 NFS is also available. Sun will be providing complete patches
     for SunOS 4.1 and SunOS 4.0.3 at a later date.

CA-94:15.NFS.Vulnerabilities                                  12/19/94
     This advisory describes security measures to guard against several
     vulnerabilities in the Network File System (NFS). The advisory was
     prompted by an increase in root compromises by intruders using tools
     to exploit the vulnerabilities.

CA-96.08.pcnfsd                                               04/18/96
     This advisory describes a vulnerability in the pcnfsd program (also
     known as rpc.pcnfsd). A patch is included.
```

6.1 Client Security

On the client we can decide that we don't want to trust the server too much a couple of ways with options to mount. For example we can forbid suid programs to work off the NFS file system with the `nosuid` option. This is a good idea and you should consider using this with all NFS mounted disks. It means that the server's root user cannot make a suid-root program on the file system, log in to the client as a normal user and then use the suid-root program to

TOC

become root on the client too. We could also forbid execution of files on the mounted file system altogether wi noexec option. But this is more likely to be impractical than nosuid since a file system is likely to at least contain *some* scripts or programs that needs to be executed. You enter these options in the options column, wi h the rsize and wsize, separated by commas.

6.2 Server security: nfsd

On the server we can decide that we don't want to trust the client's root account. We can do that by using the root_squash option in exports:

```
/mn/eris/local apollon(rw,root_squash)
```

Now, if a user with UID 0 on the client attempts to access (read, write, delete) the file system the server substitutes the UID of the servers `nobody' account. Which means that the root user on the client can't access or change files that only root on the server can access or change. That's good, and you should probably use root_squash on all the file systems you export. "But the root user on the client can still use 'su' to become any other user and access and change that users files!" say you. To which the answer is: Yes, and that's the way it is, and has to be with Unix and NFS. This has one important implication: All important binaries and files should be owned by root, and not bin or other non-root account, since the only account the clients root user cannot access is the servers root account. In the NFSd man page there are several other squash options listed so that you can decide to mistrust whomever you (don't) like on the clients. You also have options to squash any UID and GID range you want to. This is described in the Linux NFSd man page.

root_squash is in fact the default with the Linux NFSd, to grant root access to a filesystem use no_root_squash.

Another important thing is to ensure that nfsd checks that all it's requests comes from a privileged port. If it accepts requests from any old port on the client a user with no special privileges can run a program that's is easy to obtain over the Internet. It talks nfs protocol and will claim that the user is anyone the user wants to be. Spooky. The Linux nfsd does this check by default, on other OSes you have to enable this check yourself. This should be described in the nfsd man page for the OS.

Another thing. Never export a file system to 'localhost' or 127.0.0.1. Trust me.

6.3 Server security: the portmapper

The basic portmapper, in combination with nfsd has a design problem that makes it possible to get to files on NFS servers without any privileges. Fortunately the portmapper Linux uses is relatively secure against this attack, and can be made more secure by configuring up access lists in two files.

First we edit /etc/hosts.deny. It should contain the line

```
portmap: ALL
```

which will deny access to *everyone*. That's a bit drastic perhaps, so we open it again by editing /etc/hosts.allow. But first we need to figure out what to put in it. It should basically list all machines that should have access to your portmapper. On a run of the mill Linux system there are very few machines that need any access for any reason. The portmapper administrates nfsd, mountd, ypbind/ypserv, pcnfsd, and 'r' services like

ruptime and rusers. Of these only nfsd, mountd, ypbind/ypserv and perhaps pcnfsd are of any consequence. All machines that needs to access services on your machine should be allowed to do that. Let's say that your machi address is 129.240.223.254 and that it lives on the subnet 129.240.223.0 should have access to it (those are terms introduced by the networking HOWTO, go back and refresh your memory if you need to). Then we write

```
portmap: 129.240.223.0/255.255.255.0
```

in `hosts.allow`. This is the same as the network address you give to route and the subnet mask you give to ifconfig. For the device `eth0` on this machine `ifconfig` should show

```
...
eth0      Link encap:10Mbps Ethernet  HWaddr 00:60:8C:96:D5:56
          inet addr:129.240.223.254  Bcast:129.240.223.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:360315 errors:0 dropped:0 overruns:0
          TX packets:179274 errors:0 dropped:0 overruns:0
          Interrupt:10 Base address:0x320
...
```

and `netstat -rn` should show

```
Kernel routing table
Destination     Gateway         Genmask         Flags Metric Ref Use    Iface
...
129.240.223.0   0.0.0.0         255.255.255.0   U     0      0   174412 eth0
...
```

(Network address in first column).

The `hosts.deny` and `hosts.allow` files are described in the manual pages of the same names.

IMPORTANT: Do *not* put *anything* but *IP NUMBERS* in the portmap lines of these files. Host name lookups can indirectly cause portmap activity which will trigger host name lookups which can indirectly cause portmap activity which will trigger...

The above things should make your server tighter. The only remaining problem (Yeah, right!) is someone breaking root (or boot MS-DOS) on a trusted machine and using that privilege to send requests from a secure port as any user they want to be.

6.4 NFS and firewalls

It's a very good idea to firewall the nfs and portmap ports in your router or firewall. The nfsd operates at port 2049, both udp and tcp protocols. The portmapper at port 111, tcp and udp, and mountd at port 745 and and 747, tcp and udp. Normally. You should check the ports with the `rpcinfo -p` command.

TOC

If on the other hand you want NFS to go through a firewall there are options for newer NFSds and mountds to make them use a specific (nonstandard) port which can be open in the firewall.

6.5 Summary

If you use the hosts.allow/deny, root_squash, nosuid and privileged port features in the portmapper/nfs software you avoid many of the presently known bugs in nfs and can almost feel secure about *that* at least. But still, after all that: When an intruder has access to your network, s/he can make strange commands appear in your `.forward` or mailbox file when `/home` or `/var/spool/mail` are mounted over NFS. For the same reason, you should never access your PGP private key over nfs. Or at least you should know the risk involved. And now you know a bit of it.

NFS and the portmapper makes up a complex subsystem and therefore it's not totally unlikely that new bugs will be discovered, either in the basic design or the implementation we use. There might even be holes known now, which someone is abusing. But that's life. To keep abreast of things like this you should at least read the newsgroups comp.os.linux.announce and comp.security.announce at a absolute minimum.

---

7. Mount Checklist

This section is based on IBM Corp. NFS mount problem checklist. My thanks to them for making it available for this HOWTO. If you experience a problem mounting a NFS filesystem please refer to this list before posting your problem. Each item describes a failure mode and the fix.

1. File system not exported, or not exported to the client in question.

   Fix: Export it

2. Name resolution doesn't jibe with the exports list.

   e.g.: export list says export to `johnmad` but `johnmad`'s name is resolved as `johnmad.austin.ibm.com`. mount permission is denied.

   Fix: Export to both forms of the name.

   It can also happen if the client has 2 interfaces with different names for each of the two adapters and the export only specifies one.

   Fix: export both interfaces.

   This can also happen if the server can't do a lookuphostbyname or lookuphostbyaddr (these are library functions) on the client. Make sure the client can do `host <name>`; `host <ip_addr>`; and that both shows the same machine.

   Fix: straighten out name resolution.

3. The file system was mounted after NFS was started (on that server). In that case the server is exporting underlying mount point, not the mounted filesystem.

   Fix: Shut down NFSd and then restart it.

   Note: The clients that had the underlying mount point mounted will get problems accessing it after the restart.

4. The date is wildly off on one or both machines (this can mess up make)

   Fix: Get the date set right.

   The HOWTO author recommends using NTP to synchronize clocks. Since there are export restrictions on NTP in the US you have to get NTP for debian, redhat or slackware from ftp://ftp.hacktic.nl/pub/replay/pub/linux or a mirror.

5. The server can not accept a mount from a user that is in more than 8 groups.

   Fix: decrease the number of groups the user is in or mount via a different user.

8. FAQs

This is the FAQ section. Most of it was written by Alan Cox.

1. I get a lot of 'stale nfs handle' errors when using Linux as a nfs server.

   This is caused by a bug in some oldish nfsd versions. It is fixed in nfs-server2.2beta16 and later.
2. When I try to mount a file system I get

   ```
   can't register with portmap: system error on send
   ```

   You are probably using a Caldera system. There is a bug in the rc scripts. Please contact Caldera to obtain a fix.
3. Why can't I execute a file after copying it to the NFS server?

   The reason is that nfsd caches open file handles for performance reasons (remember, it runs in user space). While nfsd has a file open (as is the case after writing to it), the kernel won't allow you to execute it. Nfsds newer than spring 95 release open files after a few seconds, older ones would cling to them for days.
4. My NFS files are all read only

   The Linux NFS server defaults to read only. RTFM the ``exports'' and nfsd manual pages. You will need to alter `/etc/exports`.
5. I mount from a linux nfs server and while ls works I can't read or write files.

   On older versions of Linux you must mount a NFS servers with `rsize=1024,wsize=1024`.
6. I mount from a Linux NFS server with a block size of between 3500-4000 and it crashes the Linux box regularly

   Basically don't do it then.
7. Can Linux do NFS over TCP

   No, not at present.
8. I get loads of strange errors trying to mount a machine from a Linux box.

   Make sure your users are in 8 groups or less. Older servers require this.
9. When I reboot my machine it sometimes hangs when trying to unmount a hung NFS server.

   Do **not** unmount NFS servers when rebooting or halting, just ignore them, it will not hurt anything if you don't unmount them. The command is `umount -avt nonfs`.
10. Linux NFS clients are very slow when writing to Sun and BSD systems

    NFS writes are normally synchronous (you can disable this if you don't mind risking losing data).

TOC

Worse still BSD derived kernels tend to be unable to work in small blocks. Thus when you wr
4K of data from a Linux box in the 1K packets it uses BSD does this

```
read 4K page
alter 1K
write 4K back to physical disk
read 4K page
alter 1K
write 4K page back to physical disk
etc..
```

9. Exporting filesystems

The way to export filesytems with NFS is not completely consistent across platforms of course. In this case Linux and Solaris 2 are the deviants. This section lists, superficially the way to do it on most systems. If the kind of system you have is not covered you must check your OS man-pages. Keywords are: nfsd, system administration tool, rc scripts, boot scripts, boot sequence, /etc/exports, exportfs. I'll use one example throughout this section: How to export /mn/eris/local to apollon read/write.

9.1 IRIX, HP-UX, Digital-UNIX, Ultrix, SunOS 4 (Solaris 1), AIX

These OSes use the traditional Sun export format. In `/etc/exports` write:

```
/mn/eris/local -rw=apollon
```

The complete documentation is in the `exports` man page. After editing the file run `exportfs -av` to export the filesystems.

How strict the exportfs command is about the syntax varies. On some OSes you will find that previously entered lines reads:

```
/mn/eris/local apollon
```

or even something degenerate like:

```
/mn/eris/local rw=apollon
```

I recommend being formal. You risk that the next version of `exportfs` if much stricter and then suddenly everything will stop working.

9.2 Solaris 2

TOC

Sun completely re-invented the wheel when they did Solaris 2. So this is completely different from all other OSes. What you do is edit the file `/etc/dfs/dfstab`. In it you place share commands as documented in the share(1M) man page. Like this:

```
share -o rw=apollon -d "Eris Local" /mn/eris/local
```

After editing run the program `shareall` to export the filesystems.

---

10. PC-NFS

You should not run PC-NFS. You should run samba.

Sorry: I don't know anything about PC-NFS. If someone feels like writing something about it please do and I'll include it here.

The Linux NIS(YP)/NYS/NIS+ HOWTO

Thorsten Kukuk

v1.0, 9 March 1999

This document describes how to configure Linux as NIS(YP) or NIS+ client and how to install as NIS server.

1. Introduction

- 1.1 New Versions of this Document
- 1.2 Disclaimer
- 1.3 Feedback and Corrections
- 1.4 Acknowledgements

2. Glossary and General Information

- 2.1 Glossary of Terms
- 2.2 Some General Information

3. NIS, NYS or NIS+ ?

- 3.1 libc 4/5 with traditional NIS or NYS ?
- 3.2 glibc 2 and NIS/NIS+
- 3.3 NIS or NIS+ ?

4. How it works

- 4.1 How NIS works
- 4.2 How NIS+ works

TOC

5. The RPC Portmapper

6. What do you need to set up NIS?

- 6.1 Determine whether you are a Server, Slave or Client.
- 6.2 The Software
- 6.3 The ypbind daemon
- 6.4 Setting up a NIS Client using Traditional NIS
- 6.5 Setting up a NIS Client using NYS
- 6.6 Setting up a NIS Client using glibc 2.x
- 6.7 The nsswitch.conf File
- 6.8 Shadow Passwords with NIS

7. What do you need to set up NIS+ ?

- 7.1 The Software
- 7.2 Setting up a NIS+ client
- 7.3 NIS+, keylogin, login and PAM
- 7.4 The nsswitch.conf File

8. Setting up a NIS Server

- 8.1 The Server Program ypserv
- 8.2 The Server Program yps
- 8.3 The Program rpc.ypxfrd
- 8.4 The Program rpc.yppasswdd

9. Verifying the NIS/NYS Installation

10. Common Problems and Troubleshooting NIS

11. Frequently Asked Questions

1. Introduction

More and more, Linux machines are installed as part of a network of computers. To simplify network administration, most networks (mostly Sun-based networks) run the Network Information Service. Linux machines can take full advantage of existing NIS service or provide NIS service themselves. Linux machines can also act as full NIS+ clients, this support is in beta stage.

This document tries to answer questions about setting up NIS(YP) and NIS+ on your Linux machine. Don't forget to read the section The RPC Portmapper.

The NIS-Howto is edited and maintained by

```
Thorsten Kukuk, <kukuk@suse.de>
```

The primary source of the information for the initial NIS-Howto was from:

```
Andrea Dell'Amico       <adellam@ZIA.ms.it>
Mitchum DSouza          <Mitch.DSouza@NetComm.IE>
Erwin Embsen            <erwin@nioz.nl>
Peter Eriksson          <peter@ifm.liu.se>
```

who we should thank for writing the first versions of this document.

1.1 New Versions of this Document

You can always view the latest version of this document on the World Wide Web via the URL http://www.suse.de/~kukuk/linux/HOWTO/NIS-HOWTO.html.

New versions of this document will also be uploaded to various Linux WWW and FTP sites, including the LDP home page.

Links to translations of this document could be found at http://www.suse.de/~kukuk/linux/nis-howto.html.

1.2 Disclaimer

Although this document has been put together to the best of my knowledge it may, and probably does contain errors. Please read any README files that are bundled with any of the various pieces of software described in this document for more detailed and accurate information. I will attempt to keep this document as error free as possible.

TOC

1.3 Feedback and Corrections

If you have questions or comments about this document, please feel free to mail Thorsten Kukuk, at kukuk@suse.de. I welcome any suggestions or criticisms. If you find a mistake with this document, please let me know so I can correct it in the next version. Thanks.

Please do *not* mail me questions about special problems with your Linux Distribution! I don't know every Linux Distribution. But I will try to add every solution you send me.

1.4 Acknowledgements

We would like to thank all the people who have contributed (directly or indirectly) to this document. In alphabetical order:

```
Byron A Jeff            <byron@cc.gatech.edu>
Markus Rex              <msrex@suse.de>
Miquel van Smoorenburg  <miquels@cistron.nl>
```

Theo de Raadt is responsible for the original yp-clients code. Swen Thuemmler ported the yp-clients code to Linux and also ported the yp-routines in libc (again based on Theo's work). Thorsten Kukuk has written the NIS(YP) and NIS+ routines for GNU libc 2.x from scratch.

---

2. Glossary and General Information

2.1 Glossary of Terms

In this document a lot of acronyms are used. Here are the most important acronyms and a brief explanation:

DBM

DataBase Management, a library of functions which maintain key-content pairs in a data base.

DLL

Dynamically Linked Library, a library linked to an executable program at run-time.

domainname

A name "key" that is used by NIS clients to be able to locate a suitable NIS server that serves that domainname key. Please note that this does not necessarily have anything at all to do with the DNS "domain" (machine name) of the machine(s).

FTP

File Transfer Protocol, a protocol used to transfer files between two computers.

libnsl

Name services library, a library of name service calls (getpwnam, getservbyname, etc...) on SVR4 Unixes. GNU libc uses this for the NIS (YP) and NIS+ functions.

libsocket

Socket services library, a library for the socket service calls (socket, bind, listen, etc...) on SVR4 Unixes.

NIS

Network Information Service, a service that provides information, that has to be known throughout the network, to all machines on the network. There is support for NIS in Linux's standard libc library, which in the following text is referred to as "traditional NIS".

NIS+

Network Information Service (Plus :-), essentially NIS on steroids. NIS+ is designed by Sun Microsystems Inc. as a replacement for NIS with better security and better handling of _large_ installations.

NYS

This is the name of a project and stands for NIS+, YP and Switch and is managed by Peter Eriksson <peter@ifm.liu.se>. It contains among other things a complete reimplementation of the

NIS (= YP) code that uses the Name Services Switch functionality of the NYS library.

NSS

Name Service Switch. The /etc/nsswitch.conf file determines the order of lookups performed when a certain piece of information is requested.

RPC

Remote Procedure Call. RPC routines allow C programs to make procedure calls on other machines across the network. When people talk about RPC they most often mean the Sun RPC variant.

YP

Yellow Pages(tm), a registered trademark in the UK of British Telecom plc.

TCP-IP

Transmission Control Protocol/Internet Protocol. It is the data communication protocol most often used on Unix machines.

2.2 Some General Information

The next four lines are quoted from the Sun(tm) System & Network Administration Manual:

```
"NIS was formerly known as Sun Yellow Pages (YP) but
 the name Yellow Pages(tm) is a registered trademark
 in the United Kingdom of British Telecom plc and may
 not be used without permission."
```

NIS stands for Network Information Service. Its purpose is to provide information, that has to be known throughout the network, to all machines on the network. Information likely to be distributed by NIS is:

- login names/passwords/home directories (/etc/passwd)
- group information (/etc/group)

If, for example, your password entry is recorded in the NIS passwd database, you will be able to login on all machines on the network which have the NIS client programs running.

Sun is a trademark of Sun Microsystems, Inc. licensed to SunSoft, Inc.

---

3. NIS, NYS or NIS+ ?

3.1 libc 4/5 with traditional NIS or NYS ?

The choice between "traditional NIS" or the NIS code in the NYS library is a choice between laziness and maturity vs. flexibility and love of adventure.

The "traditional NIS" code is in the standard C library and has been around longer and sometimes suffers from its age and slight inflexibility.

The NIS code in the NYS library requires you to recompile the libc library to include the NYS code into it (or maybe you can get a precompiled version of libc from someone who has already done it).

Another difference is that the traditional NIS code has some support for NIS Netgroups, which the NYS code doesn't. On the other hand the NYS code allows you to handle Shadow Passwords in a transparent way. The "traditonal NIS" code doesn't support Shadow Passwords over NIS.

3.2 glibc 2 and NIS/NIS+

Forgot all this if you use the new GNU C Library 2.x (aka libc6). It has real NSS (name switch service) support, which makes it very flexible, and contains support for the following NIS/NIS+ maps: aliases, ethers, group, hosts, netgroups, networks, protocols, publickey, passwd, rpc, services and shadow. The GNU C Library has no problems with shadow passwords over NIS.

3.3 NIS or NIS+ ?

The choice between NIS and NIS+ is easy - use NIS if you don't have to use NIS+ or have severe security needs. NIS+ is _much_ more problematic to administer (it's pretty easy to handle on the client side, but the server side is horrible). Another problem is that the support for NIS+ under Linux is still under developement - you need the latest glibc 2.1. There is an unsupported port of the glibc NIS+ support for libc5 as dropin replacement.

4. How it works

4.1 How NIS works

Within a network there must be at least one machine acting as a NIS server. You can have multiple NIS servers, each serving different NIS "domains" - or you can have cooperating NIS servers, where one is the master NIS server, and all the other are so-called slave NIS servers (for a certain NIS "domain", that is!) - or you can have a mix of them...

Slave servers only have copies of the NIS databases and receive these copies from the master NIS server whenever changes are made to the master's databases. Depending on the number of machines in your network and the reliability of your network, you might decide to install one or more slave servers. Whenever a NIS server goes down or is too slow in responding to requests, a NIS client connected to that server will try to find one that is up or faster.

NIS databases are in so-called DBM format, derived from ASCII databases. For example, the files `/etc/passwd` and `/etc/group` can be directly converted to DBM format using ASCII-to-DBM translation software ("makedbm", included with the server software). The master NIS server should have both, the ASCII databases and the DBM databases.

Slave servers will be notified of any change to the NIS maps, (via the "yppush" program), and automatically retrieve the necessary changes in order to synchronize their databases. NIS clients do not need to do this since they always talk to the NIS server to read the information stored in it's DBM databases.

Old ypbind versions do a broadcast to find a running NIS server. This is insecure, due the fact that anyone may install a NIS server and answer the broadcast queries. Newer Versions of ypbind (ypbind-3.3 or ypbind-mt) are able to get the server from a configuration file - thus no need to broadcast.

4.2 How NIS+ works

NIS+ is a new version of the network information nameservice from Sun. The biggest difference between NIS and NIS+ is that NIS+ has support for data encryption and authentication over secure RPC.

The naming model of NIS+ is based upon a tree structure. Each node in the tree corresponds to an NIS+ object, from which we have six types: directory, entry, group, link, table and private.

The NIS+ directory that forms the root of the NIS+ namespace is called the root directory. There are two special NIS+ directories: org_dir and groups_dir. The org_dir directory consists of all administration tables, such as passwd, hosts, and mail_aliases. The groups_dir directory consists of NIS+ group objects which are used for access control. The collection of org_dir, groups_dir and their parent directory is referred to as an NIS+ domain.

5. The RPC Portmapper

To run any of the software mentioned below you will need to run the program /usr/sbin/portmap. Some Linux distributions already have the code in the /sbin/init.d/ or /etc/rc.d/ files to start up this daemon. All you have to do is to activate it and reboot your Linux machine. Read your Linux Distribution Documentation how to do this.

The RPC portmapper (portmap(8)) is a server that converts RPC program numbers into TCP/IP (or UDP/IP) protocol port numbers. It must be running in order to make RPC calls (which is what the NIS/NIS+ client software does) to RPC servers (like a NIS or NIS+ server) on that machine. When an RPC server is started, it will tell portmap what port number it is listening to, and what RPC program numbers it is prepared to serve. When a client wishes to make an RPC call to a given program number, it will first contact portmap on the server machine to determine the port number where RPC packets should be sent.

Normally, standard RPC servers are started by inetd(8), so portmap must be running before inetd is started.

For secure RPC, the portmapper needs the Time service. Make sure, that the Time service is enabled in /etc/inetd.conf on all hosts:

```
#
# Time service is used for clock syncronization.
#
time    stream  tcp     nowait  root    internal
time    dgram   udp     wait    root    internal
```

IMPORTANT: Don't forget to restart inetd after changes on its configuration file !

6. What do you need to set up NIS?

6.1 Determine whether you are a Server, Slave or Client.

To answer this question you have to consider two cases:

1. Your machine is going to be part of a network with existing NIS servers
2. You do not have any NIS servers in the network yet

In the first case, you only need the client programs (ypbind, ypwhich, ypcat, yppoll, ypmatch). The most important program is ypbind. This program must be running at all times, which means, it should always appear in the list of processes. It is a daemon process and needs to be started from the system's startup file (eg. /etc/init.d/nis, /sbin/init.d/ypclient, /etc/rc.d/init.d/ypbind, /etc/rc.local). As soon as ypbind is running your system has become a NIS client.

In the second case, if you don't have NIS servers, then you will also need a NIS server program (usually called ypserv). Section Setting up a NIS Server describes how to set up a NIS server on your Linux machine using the "ypserv" implementation by Peter Eriksson and Thorsten Kukuk. Note that from version 0.14 this implementation supports the master-slave concept talked about in section 4.1.

There is also another free NIS server available, called "yps", written by Tobias Reber in Germany which does support the master-slave concept, but has other limitations and isn't supported since a long time.

6.2 The Software

The system library "/usr/lib/libc.a" (version 4.4.2 and better) or the shared library "/lib/libc.so.x" contain all necessary system calls to succesfully compile the NIS client and server software. For the GNU C Library 2 (glibc 2.x), you also need /lib/libnsl.so.1.

Some people reported that NIS only works with "/usr/lib/libc.a" version 4.5.21 and better so if you want to play it safe don't use older libc's. The NIS client software can be obtained from:

Site	Directory	File Name
ftp.kernel.org	/pub/linux/utils/net/NIS	yp-tools-2.2.tar.gz
ftp.kernel.org	/pub/linux/utils/net/NIS	ypbind-mt-1.4.tar.gz
ftp.kernel.org	/pub/linux/utils/net/NIS	ypbind-3.3.tar.gz
ftp.kernel.org	/pub/linux/utils/net/NIS	ypbind-3.3-glibc5.diff.gz
ftp.uni-paderborn.de	/linux/local/yp	yp-clients-2.2.tar.gz

Once you obtained the software, please follow the instructions which come with the software. yp-clients 2.2 are for use with libc4 and libc5 until 5.4.20. libc 5.4.21 and glibc 2.x needs yp-tools 1.4.1 or later. The new yp-tools 2.2 should work with every Linux libc. Since there was a bug in the NIS code, you shouldn't use libc 5.4.21-5.4.35. Use libc 5.4.36 or later instead, or the most YP programs will not work. ypbind 3.3 will work with all libraries, too. If you use gcc 2.8.x or greater, egcs or glibc 2.x, you should add the ypbind-3.3-glibc5.diff patch to ypbind 3.3. Please never use the ypbind from yp-clients 2.2. ypbind-mt is a new, multithreaded daemon. It needs a Linux 2.2 kernel, and glibc 2.1 or later.

6.3 The ypbind daemon

After you have succesfully compiled the software you are now ready to install it. A suitable place for the ypbind daemon is the directory /usr/sbin. Some people may tell you that you don't need ypbind on a system with NYS. This is wrong. ypwhich and ypcat need it always.

You must do this as root of course. The other binaries (ypwhich, ypcat, yppasswd, yppoll, ypmatch) should go in a directory accessible by all users, normally /usr/bin.

Newer ypbind versions have a configuration file called /etc/yp.conf. You can hardcode a NIS server there - for more info see the manual page for ypbind(8). You also need this file for NYS. An example:

```
ypserver voyager
ypserver defiant
ypserver ds9
```

If the system cam resolv the hostnames without NIS, you may use the name, otherwise you have to use the IP address. ypbind 3.3 has a bug and will only use the last entry (ypserver ds9 in the example). All other entries are ignored. ypbind-mt handle this correct and uses that one, which answerd at first.

It might be a good idea to test ypbind before incorporating it in the startup files. To test ypbind do the following:

- Make sure you have your YP-domain name set. If it is not set then issue the command:

  ```
  /bin/domainname nis.domain
  ```

  where `nis.domain` should be some string _NOT_ normally associated with the DNS-domain name of your machine! The reason for this is that it makes it a little harder for external crackers to retreive the password database from your NIS servers. If you don't know what the NIS domain name is on your network, ask your system/network administrator.
- Start up "/usr/sbin/portmap" if it is not already running.
- Create the directory "/var/yp" if it does not exist.
- Start up "/usr/sbin/ypbind"
- Use the command "rpcinfo -p localhost" to check if ypbind was able to register its service with the portmapper. The output should look like:

  ```
  program vers proto   port
   100000    2   tcp    111  portmapper
   100000    2   udp    111  portmapper
   100007    2   udp    637  ypbind
   100007    2   tcp    639  ypbind
  ```

  or

  ```
  program vers proto   port
   100000    2   tcp    111  portmapper
   100000    2   udp    111  portmapper
   100007    2   udp    758  ypbind
   100007    1   udp    758  ypbind
   100007    2   tcp    761  ypbind
   100007    1   tcp    761  ypbind
  ```

  Depending on the ypbind version you are using.
- You may also run "rpcinfo -u localhost ypbind". This command should produce something like:

  ```
  program 100007 version 2 ready and waiting
  ```

  or

  ```
  program 100007 version 1 ready and waiting
  program 100007 version 2 ready and waiting
  ```

The output depends on the ypbind version you have installed. Important is only the "version 2" message.

At this point you should be able to use NIS client programs like ypcat, etc... For example, "ypcat passwd.byname" will give you the entire NIS password database.

IMPORTANT: If you skipped the test procedure then make sure you have set the domain name, and created the directory

```
/var/yp
```

This directory MUST exist for ypbind to start up succesfully.

To check if the domainname is set correct, use the /bin/ypdomainname from yp-tools 2.2. It uses the yp_get_default_domain() function which is more restrict. It doesn't allow for example the "(none)" domainname, which is the default under Linux and makes a lot of problems.

If the test worked you may now want to change your startupd files so that ypbind will be started at boot time and your system will act as a NIS client. Make sure that the domainname will be set before you start ypbind.

Well, that's it. Reboot the machine and watch the boot messages to see if ypbind is actually started.

6.4 Setting up a NIS Client using Traditional NIS

For host lookups you must set (or add) "nis" to the lookup order line in your /etc/host.conf file. Please read the manpage "resolv+.8" for more details.

Add the following line to /etc/passwd on your NIS clients:

```
+::::::
```

You can also use the + and - characters to include/exclude or change users. If you want to exclude the user guest just add -guest to your /etc/passwd file. You want to use a different shell (e.g. ksh) for the user "linux"? No problem, just add "+linux::::::/bin/ksh" (without the quotes) to your /etc/passwd. Fields that you don't want to change have to be left empty. You could also use Netgroups for user control.

For example, to allow login-access only to miquels, dth and ed, and all members of the sysadmin netgroup, but to have the account data of all other users available use:

```
+miquels:::::::
+ed:::::::
+dth:::::::
+@sysadmins:::::::
-ftp
+:*::::::/etc/NoShell
```

Note that in Linux you can also override the password field, as we did in this example. We also remove the login "ftp", so it isn't known any longer, and anonymous ftp will not work.

The netgroup would look like

```
sysadmins (-,software,) (-,kukuk,)
```

IMPORTANT: The netgroup feature is implemented starting from libc 4.5.26. If you have a version of libc earlier than 4.5.26, every user in the NIS password database can access your linux machine if you run "ypbind" !

TOC

6.5 Setting up a NIS Client using NYS

All that is required is that the NIS configuration file (/etc/yp.conf) points to the correct server(s) for its information. Also, the Name Services Switch configuration file (/etc/nsswitch.conf) must be correctly set up.

You should install ypbind. It isn't needed by the libc, but the NIS(YP) tools need it.

If you wish to use the include/exclude user feature (+/-guest/+@admins), you have to use "passwd: compat" and "group: compat" in nsswitch.conf. Note that there is no "shadow: compat"! You have to use "shadow: files nis" in this case.

The NYS sources are part of the libc 5 sources. When run configure, say the first time "NO" to the "Values correct" question, then say "YES" to "Build a NYS libc from nys".

6.6 Setting up a NIS Client using glibc 2.x

The glibc uses "traditional NIS", so you need to start ypbind. The Name Services Switch configuration file (/etc/nsswitch.conf) must be correctly set up. If you use the compat mode for passwd, shadow or group, you have to add the "+" at the end of this files and you can use the include/exclude user feature. The configuration is excatly the same as under Solaris 2.x.

6.7 The nsswitch.conf File

The Network Services switch file /etc/nsswitch.conf determines the order of lookups performed when a certain piece of information is requested, just like the /etc/host.conf file which determines the way host lookups are performed. For example, the line

```
hosts: files nis dns
```

specifies that host lookup functions should first look in the local /etc/hosts file, followed by a NIS lookup and finally through the domain name service (/etc/resolv.conf and named), at which point if no match is found an error is returned. This file must be readable for every user! You can find more information in the man-page nsswitch.5 or nsswitch.conf.5.

A good /etc/nsswitch.conf file for NIS is:

```
#
# /etc/nsswitch.conf
#
# An example Name Service Switch config file. This file should be
# sorted with the most-used services at the beginning.
#
# The entry '[NOTFOUND=return]' means that the search for an
# entry should stop if the search in the previous entry turned
# up nothing. Note that if the search failed due to some other reason
# (like no NIS server responding) then the search continues with the
# next entry.
#
# Legal entries are:
#
#       nisplus                 Use NIS+ (NIS version 3)
#       nis                     Use NIS (NIS version 2), also called YP
#       dns                     Use DNS (Domain Name Service)
#       files                   Use the local files
#       db                      Use the /var/db databases
#       [NOTFOUND=return]       Stop searching if not found so far
#
```

TOC

```
passwd:     compat
group:      compat
# For libc5, you must use shadow: files nis
shadow:     compat

passwd_compat: nis
group_compat: nis
shadow_compat: nis

hosts:      nis files dns

services:   nis [NOTFOUND=return] files
networks:   nis [NOTFOUND=return] files
protocols:  nis [NOTFOUND=return] files
rpc:        nis [NOTFOUND=return] files
ethers:     nis [NOTFOUND=return] files
netmasks:   nis [NOTFOUND=return] files
netgroup:   nis
bootparams: nis [NOTFOUND=return] files
publickey:  nis [NOTFOUND=return] files
automount:  files
aliases:    nis [NOTFOUND=return] files
```

passwd_compat, group_compat and shadow_compat are only supported by glibc 2.x. If there are no shadow rules in /etc/nsswitch.conf, glibc will use the passwd rule for lookups. There are some more lookup module for glibc like hesoid. For more information, read the glibc documentation.

6.8 Shadow Passwords with NIS

Shadow passwords over NIS are always a bad idea. You loose the security, which shadow gives you, and it is supported by only some few Linux C Libraries. A good way to avoid shadow passwords over NIS is, to put only the local system users in /etc/shadow. Remove the NIS user entries from the shadow database, and put the password back in passwd. So you can use shadow for the root login, and normal passwd for NIS user. This has the advantage that it will work with every NIS client.

Linux

The only Linux libc which supports shadow passwords over NIS, is the GNU C Library 2.x. Linux libc5 has no support for it. Linux libc5 compiled with NYS enabled has some code for it. But this code is badly broken in some cases and doesn't work with all correct shadow entries.

Solaris

Solaris does not support shadow passwords over NIS.

PAM

PAM does not support Shadow passwords over NIS, especially pam_pwdb/libpwdb. This is a big problem for RedHat 5.x users. If you have glibc and PAM, you need to change the /etc/pam.d/* entries. Replace all pam_pwdb rules through pam_unix_* modules. Due a bug in the pam_unix_auth.so module this will not always work.

An example /etc/pam.d/login file looks like:

```
#%PAM-1.0
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_unix_auth.so
auth       required     /lib/security/pam_nologin.so
```

```
account    required     /lib/security/pam_unix_acct.so
password   required     /lib/security/pam_unix_passwd.so
session    required     /lib/security/pam_unix_session.so
```

For auth you need to use the pam_unix_auth.so module, for account the pam_unix_acct.so, for password the pam_unix_passwd.so and for session the pam_unix_session.so module.

7. What do you need to set up NIS+ ?

7.1 The Software

The Linux NIS+ client code was developed for the GNU C library 2. There is also a port for Linux libc5, since most commercial Applications are linked against this library, and you cannot recompile them for using glibc. There are problems with libc5 and NIS+: static programs cannot be linked with it, and programs compiled with this library will not work with other libc5 versions.

You need to retrieve and compile the GNU C Library 2.1 for Intel based platforms, or GNU C Library 2.1.1 for 64bit platforms. As base System you need a glibc based Distribution like Debian 2.x, RedHat 5.x or SuSE Linux 6.x.

For every distribution, you need to recompile the gcc/g++ compiler, libstdc++ and ncures. For Redhat, you need to make a lot of changes of the PAM configuration. For SuSE Linux 6.0, you need to recompile the shadow package.

The NIS+ client software can be obtained from:

Site	Directory	File Name
ftp.funet.fi	/pub/gnu/funet	libc-*, glibc-crypt-*, glibc-linuxthreads-*
ftp.kernel.org	/pub/linux/utils/net/NIS+	nis-utils-19990223.tar.gz
ftp.kernel.org	/pub/linux/utils/net/NIS+	pam_keylogin-1.2.tar.gz

Distributions based on glibc can be fetched from:

Site	Directory
ftp.debian.org	/pub/debian/dists/slink
ftp.redhat.com	/pub/redhat/redhat-5.2
ftp.suse.de	/pub/SuSE-Linux/6.0

For compilation of the GNU C Library please follow the instructions which come with the software. You cam find the patched libc5, based on NYS, and the sources as drop in replacement for the standart libc5 at:

Site	Directory	File Name
ftp.kernel.org	/pub/linux/utils/net/NIS+	libc-5.4.44-nsl-0.4.10.tar.gz

You should also have a look at http://www.suse.de/~kukuk/linux/nisplus.html for more information and the latest sources.

7.2 Setting up a NIS+ client

IMPORTANT: For setting up a NIS+ client read your Solaris NIS+ docs what to do on the server side! This document only describes what to do on the client side!

After installing the new libc and nis-tools, create the credentials for the new client on the NIS+ server. Make sure portmap is running. Then check if your Linux PC has the same time as the NIS+ Server. For secure RPC, you have only a

TOC

small window from about 3 minutes, in which the credentials are valid. A good idea is to run xntpd on every host. A this, run

```
domainname nisplus.domain.
nisinit -c -H <NIS+ server>
```

to initialize the cold start file. Read the nisinit man page for more options. Make sure that the domainname will always be set after a reboot. If you don't know what the NIS+ domain name is on your network, ask your system/network administrator.

Now you should change your /etc/nsswitch.conf file. Make sure that the only service after publickey is nisplus ("publickey: nisplus"), and nothing else!

Then start keyserv and make sure, that it will always be started as first daemon after portmap at boot time. Run

```
keylogin -r
```

to store the root secretkey on your system. (I hope you have added the publickey for the new host on the NIS+ Server?).

"niscat passwd.org_dir" should now show you all entries in the passwd database.

7.3 NIS+, keylogin, login and PAM

When the user logs in, he need to set his secretkey to keyserv. This is done by calling "keylogin". The login from the shadow package will do this for the user, if it was compiled against glibc 2.1. For a PAM aware login, you have to install pam_keylogin-1.2.tar.gz and change the /etc/pam.d/login file to use pam_unix_auth, not pwdb, which doesn't support NIS+. An example:

```
#%PAM-1.0
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_keylogin.so
auth       required     /lib/security/pam_unix_auth.so
auth       required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_unix_acct.so
password   required     /lib/security/pam_unix_passwd.so
session    required     /lib/security/pam_unix_session.so
```

7.4 The nsswitch.conf File

The Network Services switch file /etc/nsswitch.conf determines the order of lookups performed when a certain piece of information is requested, just like the /etc/host.conf file which determines the way host lookups are performed. For example, the line

```
hosts: files nisplus dns
```

specifies that host lookup functions should first look in the local /etc/hosts file, followed by a NIS+ lookup and finally through the domain name service (/etc/resolv.conf and named), at which point if no match is found an error is returned.

A good /etc/nsswitch.conf file for NIS+ is:

```
#
# /etc/nsswitch.conf
#
```

```
# An example Name Service Switch config file. This file should be
# sorted with the most-used services at the beginning.
#
# The entry '[NOTFOUND=return]' means that the search for an
# entry should stop if the search in the previous entry turned
# up nothing. Note that if the search failed due to some other reason
# (like no NIS server responding) then the search continues with the
# next entry.
#
# Legal entries are:
#
#       nisplus                 Use NIS+ (NIS version 3)
#       nis                     Use NIS (NIS version 2), also called YP
#       dns                     Use DNS (Domain Name Service)
#       files                   Use the local files
#       db                      Use the /var/db databases
#       [NOTFOUND=return]       Stop searching if not found so far
#

passwd:     compat
# for libc5: passwd: files nisplus
group:      compat
# for libc5: group: files nisplus
shadow:     compat
# for libc5: shadow: files nisplus

passwd_compat: nisplus
group_compat:  nisplus
shadow_compat: nisplus

hosts:      nisplus files dns

services:   nisplus [NOTFOUND=return] files
networks:   nisplus [NOTFOUND=return] files
protocols:  nisplus [NOTFOUND=return] files
rpc:        nisplus [NOTFOUND=return] files
ethers:     nisplus [NOTFOUND=return] files
netmasks:   nisplus [NOTFOUND=return] files
netgroup:   nisplus
bootparams: nisplus [NOTFOUND=return] files
publickey:  nisplus
automount:  files
aliases:    nisplus [NOTFOUND=return] files
```

8. Setting up a NIS Server

8.1 The Server Program ypserv

This document only describes how to set up the "ypserv" NIS server.

The NIS server software can be found on:

Site	Directory	File Name
ftp.kernel.org	/pub/linux/utils/net/NIS	ypserv-1.3.6.tar.gz

You could also look at http://www.suse.de/~kukuk/linux/nis.html for more information.

The server setup is the same for both traditional NIS and NYS.

Compile the software to generate the `ypserv` and `makedbm` programs. You can configure ypserv to use the securenets file or the tcp_wrappers. The tcp_wrapper is much more flexible, but a lot of people have big problems with it. And some configuration files for tcp_wrappers may cause a memory leak. If you have problems with ypserv compiled for tcp_wrapper, recompile it using the securenets file. ypserv --version tells you, which version you have.

If you run your server as master, determine what files you require to be available via NIS and then add or remove the appropriate entries to the "all" rule in `/var/yp/Makefile`. You always should look at the Makefile and edit the Options at the beginning of the file.

There was one big change between ypserv 1.1 and ypserv 1.2. Since version 1.2, the file handles are cached. This means you have to call makedbm always with the -c option if you create new maps. Make sure, you are using the new `/var/yp/Makefile` from ypserv 1.2 or later, or add the -c flag to makedbm in the Makefile. If you don't do that, ypserv will continue to use the old maps, and not the updated one.

Now edit `/var/yp/securenets` and `/etc/ypserv.conf`. For more information, read the ypserv(8) and ypserv.conf(5) manual pages.

Make sure the portmapper (portmap(8)) is running, and start the server `ypserv`. The command

```
% rpcinfo -u localhost ypserv
```

should output something like

```
program 100004 version 1 ready and waiting
program 100004 version 2 ready and waiting
```

The "version 1" line could be missing, depending on the ypserv version and configuration you are using. It is only necessary if you have old SunOS 4.x clients.

Now generate the NIS (YP) database. On the master, run

```
% /usr/lib/yp/ypinit -m
```

On a slave make sure that `ypwhich -m` works. This means, that your slave must be configured as NIS client before you could run

```
% /usr/lib/yp/ypinit -s masterhost
```

to install the host as NIS slave.

That's it, your server is up and running.

If you have bigger problems, you could start `ypserv` and `ypbind` in debug mode on different xterms. The debug output should show you what goes wrong.

If you need to update a map, run `make` in the `/var/yp` directory on the NIS master. This will update a map if the source file is newer, and push the files to the slave servers. Please don't use `ypinit` for updating a map.

You might want to edit root's crontab *on the slave* server and add the following lines:

```
20 *    * * *    /usr/lib/yp/ypxfr_1perhour
40 6    * * *    /usr/lib/yp/ypxfr_1perday
55 6,18 * * *    /usr/lib/yp/ypxfr_2perday
```

This will ensure that most NIS maps are kept up-to-date, even if an update is missed because the slave was down at the time the update was done on the master.

You can add a slave at every time later. At first, make sure that the new slave server has permissions to contact the NIS master. Then run

```
% /usr/lib/yp/ypinit -s masterhost
```

on the new slave. On the master server, add the new slave server name to `/var/yp/ypservers` and run `make` in `/var/yp` to update the map.

If you want to restrict access for users to your NIS server, you'll have to setup the NIS server as a client as well by running ypbind and adding the plus-entries to /etc/passwd _halfway_ the password file. The library functions will ignore all normal entries after the first NIS entry, and will get the rest of the info through NIS. This way the NIS access rules are maintained. An example:

```
root:x:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:
bin:*:2:2:bin:/bin:
sys:*:3:3:sys:/dev:
sync:*:4:100:sync:/bin:/bin/sync
games:*:5:100:games:/usr/games:
man:*:6:100:man:/var/catman:
lp:*:7:7:lp:/var/spool/lpd:
mail:*:8:8:mail:/var/spool/mail:
news:*:9:9:news:/var/spool/news:
uucp:*:10:50:uucp:/var/spool/uucp:
nobody:*:65534:65534:noone at all,,,,:/dev/null:
+miquels::::::
+:*:::::/etc/NoShell
[ All normal users AFTER this line! ]
tester:*:299:10:Just a test account:/tmp:
miquels:1234567890123:101:10:Miquel van Smoorenburg:/home/miquels:/bin/zsh
```

Thus the user "tester" will exist, but have a shell of /etc/NoShell. miquels will have normal access.

Alternatively, you could edit the `/var/yp/Makefile` file and set NIS to use another source password file. On large systems the NIS password and group files are usually stored in `/etc/yp/`. If you do this the normal tools to administrate the password file such as `passwd`, `chfn`, `adduser` will not work anymore and you need special homemade tools for this.

However, `yppasswd`, `ypchsh` and `ypchfn` will work of course.

TOC

8.2 The Server Program yps

To set up the "yps" NIS server please refer to the previous paragraph. The "yps" server setup is similar, _but_ not exactly the same so beware if you try to apply the "ypserv" instructions to "yps"! "yps" is not supported by any author, and contains some security leaks. You really shouldn't use it !

The "yps" NIS server software can be found on:

```
Site                    Directory                    File Name

ftp.lysator.liu.se      /pub/NYS/servers             yps-0.21.tar.gz
ftp.kernel.org          /pub/linux/utils/net/NIS     yps-0.21.tar.gz
```

8.3 The Program rpc.ypxfrd

rpc.ypxfrd is used for speed up the transfer of very large NIS maps from a NIS master to NIS slave servers. If a NIS slave server receives a message that there is a new map, it will start ypxfr for transfering the new map. ypxfr will read the contents of a map from the master server using the yp_all() function. This process can take several minutes when there are very large maps which have to store by the database library.

The rpc.ypxfrd server speeds up the transfer process by allowing NIS slave servers to simply copy the master server's map files rather than building their own from scratch. rpc.ypxfrd uses an RPC-based file transfer protocol, so that there is no need for building a new map.

rpc.ypxfrd can be started by inetd. But since it starts very slow, it should be started with ypserv. You need to start rpc.ypxfrd only on the NIS master server.

8.4 The Program rpc.yppasswdd

Whenever users change their passwords, the NIS password database and probably other NIS databases, which depend on the NIS password database, should be updated. The program "rpc.yppasswdd" is a server that handles password changes and makes sure that the NIS information will be updated accordingly. rpc.yppasswdd is now integrated in ypserv. You don't need the older, separate yppasswd-0.9.tar.gz or yppasswd-0.10.tar.gz, and you shouldn't use them any longer. The rpc.yppasswdd in ypserv 1.3.2 has full shadow support. yppasswd is now part of yp-tools-2.2.tar.gz.

You need to start rpc.yppasswdd only on the NIS master server. By default, users are not allowed to change their full name or the login shell. You can allow this with the -e chfn or -e chsh option.

If your passwd and shadow files are not in another directory then /etc, you need to add the -D option. For example, if you have put all source files in /etc/yp and wish to allow the user to change his shell, you need to start rpc.yppasswdd with the following parameters:

```
rpc.yppasswdd -D /etc/yp -e chsh
```

or

```
rpc.yppasswdd -s /etc/yp/shadow -p /etc/yp/passwd -e chsh
```

There is nothing more to do. You just need to make sure, that `rpc.yppasswdd` uses the same files as `/var/yp/Makefile`. Errors will be logged using syslog.

---

9. Verifying the NIS/NYS Installation

If everything is fine (as it should be), you should be able to verify your installation with a few simple commands. Assuming, for example, your passwd file is being supplied by NIS, the command

```
% ypcat passwd
```

should give you the contents of your NIS passwd file. The command

```
% ypmatch userid passwd
```

(where userid is the login name of an arbitrary user) should give you the user's entry in the NIS passwd file. The "ypcat" and "ypmatch" programs should be included with your distribution of traditional NIS or NYS.

If a user cannot log in, run the following program on the client:

```
#include <stdio.h>
#include <pwd.h>
#include <sys/types.h>

int
main(int argc, char *argv[])
{
  struct passwd *pwd;

  if(argc != 2)
    {
      fprintf(stderr,"Usage: getwpnam username\n");
      exit(1);
    }

  pwd=getpwnam(argv[1]);

  if(pwd != NULL)
    {
      printf("name.....: [%s]\n",pwd->pw_name);
      printf("password.: [%s]\n",pwd->pw_passwd);
      printf("user id..: [%d]\n", pwd->pw_uid);
      printf("group id.: [%d]\n",pwd->pw_gid);
      printf("gecos....: [%s]\n",pwd->pw_gecos);
      printf("directory: [%s]\n",pwd->pw_dir);
```

```
          printf("shell....: [%s]\n",pwd->pw_shell);
        }
      else
        fprintf(stderr,"User \"%s\" not found!\n",argv[1]);

      exit(0);
    }
```

Running this program with the username as parameter will print all the information the getpwnam function gives back for this user. This should show you which entry is incorrect. The most common problem is, that the password field is overwritten with a "*".

GNU C Library 2.1 (glibc 2.1) comes with a tool called getent. Use this program instead the above on such a system. You could try:

```
getent passwd
```

or

```
getent passwd login
```

---

10. Common Problems and Troubleshooting NIS

Here are some common problems reported by various users:

1. The libraries for 4.5.19 are broken. NIS won't work with it.
2. If you upgrade the libraries from 4.5.19 to 4.5.24 then the su command breaks. You need to get the su command from the slackware 1.2.0 distribution. Incidentally that's where you can get the updated libraries.
3. When a NIS server goes down and comes up again ypbind starts complaining with messages like:

   ```
   yp_match: clnt_call:
                 RPC: Unable to receive; errno = Connection refused
   ```

   and logins are refused for those who are registered in the NIS database. Try to login as root and kill ypbind and start it up again. An update to ypbind 3.3 or higher should also help.
4. After upgrading the libc to a version greater then 5.4.20, the YP tools will not work any longer. You need yp-tools 1.2 or later for libc >= 5.4.21 and glibc 2.x. For earlier libc version you need yp-clients 2.2. yp-tools 2.x should work for all libraries.
5. In libc 5.4.21 - 5.4.35 yp_maplist is broken, you need 5.4.36 or later, or some YP programs like ypwhich will segfault.
6. libc 5 with traditional NIS doesn't support shadow passwords over NIS. You need libc5 + NYS or glibc 2.x.
7. ypcat shadow doesn't show the shadow map. This is correct, the name of the shadow map is shadow.byname, not shadow.
8. Solaris doesn't use always privileged ports. So don't use password mangling if you have a Solaris client.

11. Frequently Asked Questions

Most of your questions should be answered by now. If there are still questions unanswered you might want to post a message to

```
comp.os.linux.networking
```

The Linux Networking Overview HOWTO

Daniel López Ridruejo, ridruejo@esi.us.es

v0.2, 10 July 1998

The purpose of this document is to give an overview of the networking capabilities of the Linux Operating System and provide pointers for further information and implementation details

1. Introduction.

2. Linux.

- 2.1 What is Linux?
- 2.2 What makes Linux different?

3. Networking protocols.

- 3.1 TCP/IP
- 3.2 TCP/IP version 6
- 3.3 IPX/SPX
- 3.4 AppleTalk Protocol Suite
- 3.5 WAN Networking: X.25, Frame-relay, etc...
- 3.6 ISDN
- 3.7 PPP, SLIP, PLIP
- 3.8 Amateur Radio
- 3.9 ATM

4. Networking hardware supported

TOC

5. File Sharing and Printing

- 5.1 Apple environment
- 5.2 Windows Environment
- 5.3 Novell Environment
- 5.4 UNIX Environment

6. Internet/Intranet

- 6.1 Mail
- 6.2 Web Servers
- 6.3 Web Browsers
- 6.4 FTP Servers and clients
- 6.5 News service
- 6.6 Domain Name System
- 6.7 DHCP, bootp
- 6.8 NIS
- 6.9 Authentication

7. Remote execution of applications

- 7.1 Telnet
- 7.2 Remote commands
- 7.3 X-Window

8. Network Interconnection

- 8.1 Router
- 8.2 Bridge
- 8.3 IP-Masquerading
- 8.4 IP-Accounting
- 8.5 IP aliasing
- 8.6 Traffic Shaping
- 8.7 Firewall
- 8.8 Port forwarding

- 8.9 Load Balancing
- 8.10 EQL
- 8.11 Proxy Server
- 8.12 Diald on demand
- 8.13 Tunnelling, mobile IP and virtual private networks

9. Network Management

- 9.1 Network management applications
- 9.2 SNMP

10. Enterprise Linux Networking

- 10.1 High Availability
- 10.2 RAID
- 10.3 Redundant networking

11. Sources of Information

12. Acknowledgements and disclaimer

---

1. Introduction.

The purpose of this document is to give an overview of the networking capabilities of the Linux Operating System. Although one of the strengths of Linux is that there is plenty of information available for nearly every topic, most of it is focused on implementation. Most new Linux users, particularly those coming from a Windows environment, are unaware of the networking possibilities of Linux. This documents aims to show a general picture of such possibilities with a brief description of each one and pointers for further information. The information has been gathered from many sources: how-tos, faqs, projects' web pages and my own hands-on experience. Full credit is given to the authors. Without them and their programs this document would have not been possible or necessary.

2. Linux.

2.1 What is Linux?

The primary author of Linux is Linus Torvalds. Since his original versions, it has been improved by countless numbers of people. It is a clone, written entirely from scratch, of the UNIX operating system. One of the more interesting facts about Linux is that development occurs simultaneously around the world.

Linux has been copyrighted under the terms of the GNU General Public License (GPL). This is a license written by the Free Software Foundation (FSF) that is designed to prevent people from restricting the distribution of software. In brief, it says that although money can be charged for a copy, the person who received the copy can not be prevented from giving it away for free. It also means that the source code must be available. This is useful for programmers. Anybody can modify Linux and even distribute his/her modifications, provided that they keep the code under the same copyright.

2.2 What makes Linux different?

Why work on Linux? Linux is generally cheaper (or at least no more expensive) than other operating systems and is frequently less problematic than many commercial systems. But what makes Linux different is not its price (after all, why would anyone want an OS -even if it is free- if it is not good enough?) but its outstanding capabilities:

- Linux is a true 32-bit multitasking Operating System, robust and capable enough to be used from universities to large corporations.
- It runs from low-end 386 boxes to massive ultra-parallel machines in research centres.
- There are out-of-the-box available versions for Intel/Sparc/Alpha architectures, and experimental support for Power PC and embedded systems among others (SGI, Ultra Sparc, AP1000+, Strong ARM, MIPS R3000/R4000...)
- Finally, when coming to networking, Linux is the choice. Not only because networking is tightly integrated with the O.S. itself and a plethora of applications is freely available, but for the robustness under heavy load that can only be achieved after years of debugging and testing in an Open Source project.

3. Networking protocols.

Linux supports many different networking protocols:

3.1 TCP/IP

The Internet Protocol was originally developed two decades ago for the United States Department of Defense (DoD), mainly for the purpose of interconnecting different brand computers. The TCP/IP suite of protocols allowed, through its layered structure, to insulate applications from networking hardware.

Although it is based in a layered model, it is focused more on delivering interconnectivity than on rigidly adhering to functional layers. This is one of the reasons why TCP/IP has become the de facto standard internetworking protocol as opposed to OSI.

TCP/IP networking has been present in Linux since its beginnings. It has been implemented from scratch. It is one of the most robust, fast and reliable implementations and is one of the key factors for Linux success.

Related How-to: http://sunsite.unc.edu/mdw/HOWTO/NET-3-HOWTO.html

3.2 TCP/IP version 6

IPv6, sometimes also referred as IPng (IP Next Generation) is an upgrade to the IPv4 protocol in order to address many issues. These issues include: shortage of available IP addresses, lack of mechanisms to handle time-sensitive traffic, lack of network layer security, etc

The increasing in address space will be accompanied by an expanded addressing scheme, which will have a great impact at routing performance. A beta implementation already exists for Linux and a production version is expected for the 2.2.0 Linux kernel release.

- Linux IPv6 How-to: http://www.terra.net/ipv6/linux-ipv6.faq.htm
- IPv6 start page: http://playground.sun.com/pub/ipng/html/ipng-main.html

3.3 IPX/SPX

IPX/SPX (Internet Packet Exchange/Sequenced Packet Exchange) is a proprietary protocol stack developed by Novell and based on Xerox Network Systems (XNS) protocol. IPX/SPX became prominent during the early 1980s as an integral part of Novell, Inc.'s NetWare. NetWare became the de facto standard network operating system (NOS) of first generation LANs. Novell complemented its NOS with a business-oriented application suite and client-side connection utilities.

Linux has a very clean IPX/SPX implementation, allowing it to be configured as an:

TOC

- IPX router
- IPX bridge
- NCP client and/or NCP Server (for sharing files)
- Novell Print Client, Novell Print Server

And to:

- Enable PPP/IPX, allowing a Linux box to act a s a PPP server/client
- IPX tunnelling through IP, allowing the connection of two IPX networks through an IP only link

Additionally, Caldera offers commercial support for Novell Netware under Linux. Caldera provides a fully featured Novell NetWare client built on technology licensed from Novell Corporation. The client provides full client access to Novell 3.x and 4.x fileservers and includes features such as NetWare Directory Service (NDS) and RSA encryption.

- IPX How-to: http://sunsite.unc.edu/mdw/HOWTO/IPX-HOWTO.html

3.4 AppleTalk Protocol Suite

Appletalk is the name of Apple's internetworking stack. It allows a peer-to-peer network model which provides basic functionality such as file and printer sharing. Each machine can simultaneously act as a client and a server and the software and hardware necessary are included with every Apple computer.

Linux provides full Appletalk networking. Netatalk is a kernel-level implementation of the AppleTalk Protocol Suite, originally for BSD-derived systems. It includes support for routing AppleTalk, serving Unix and AFS filesystems over AFP (AppleShare), serving Unix printers and accessing AppleTalk printers over PAP.

See section 5.1 for more information.

3.5 WAN Networking: X.25, Frame-relay, etc...

Several third parties provide T-1, T-3, X.25 and Frame Relay products for Linux. Generally special hardware is required for these types of connections. Vendors that provide the hardware also provide the drivers with protocol support.

- WAN resources for Linux: http://www.secretagent.com/networking/wan.html

3.6 ISDN

Linux kernel has built-in ISDN capabilies. Isdn4linux controls ISDN PC cards and can emulate a modem with the Hayes command set ("AT" commands). The possibilities range from simply using a terminal program to connections via HDLC (using included devices) to full connection to the Internet with PPP to audio applications.

- FAQ for isdn4linux: http://tsikora.tiac.net/i4l-faq/eng-i4l-faq.html

TOC

3.7 PPP, SLIP, PLIP

The Linux kernel has built-in support for PPP (Point-to-Point-Protocol), SLIP (Serial Line IP) and PLIP (Parallel Line IP). PPP is the most popular way individual users access their ISP (Internet Service Provider). PLIP allows cheap connection of two machines. It uses a parallel port and a special cable, achieving speeds of 10kBps to 20kBps.

- Linux PPP How-to
- PPP/SLIP emulator
- PLIP information can be found in The Network Administrator Guide

3.8 Amateur Radio

Linux kernel has built-in support for amateur radio protocols.

Especially interesting is the AX.25 support. The AX.25 protocol offers both connected and connectionless modes of operation, and is used either by itself for point-point links, or to carry other protocols such as TCP/IP and NetRom.

It is similar to X.25 level 2 in structure, with some extensions to make it more useful in the amateur radio environment.

- Amateur radio how-to

3.9 ATM

ATM support for Linux is currently in pre-alpha stage. There is an experimental release, which supports raw ATM connections (PVCs and SVCs), IP over ATM, LAN emulation...

- Linux ATM-Linux home page

---

4. Networking hardware supported

Linux supports a great variety of networking hardware, including obsolete one.

Some interesting documents:

- Hardware How-To
- Ethernet How-To

5. File Sharing and Printing

The primary purpose of many PC based Local Area Networks is to provide file and printer sharing services to the users. Linux as a corporate file and print server turns out to be a great solution.

5.1 Apple environment

As outlined in previous sections, Linux supports the Appletalk family of protocols. Linux netatalk allows Macintosh clients to see Linux Systems as another Macintosh on the network, share files and use printers connected to Linux servers.

Netatalk faq and how-to:

- http://thehamptons.com/anders/netatalk/
- http://www.umich.edu/~rsug/netatalk/
- http://www.umich.edu/~rsug/netatalk/faq.html

5.2 Windows Environment

Samba is a suite of applications that allow most Unices (and in particular Linux) to integrate into a Microsoft network both as a client and a server.Acting as a server it allows Windows 95, Windows for Workgroups, DOS and Windows NT clients to access Linux files and printing services. It can completely replace Windows NT for file and printing services, including the automatic downloading of printer drivers to clients. Acting as a client allows the Linux workstation to mount locally exported windows file shares.

According to the SAMBA Meta-FAQ:

```
"Many users report that compared to other SMB implementations Samba is more stable,
faster, and compatible with more clients. Administrators of some large installations
say
that Samba is the only SMB server available which will scale to many tens of
thousands
of users without crashing"
```

- Samba project home page
- Samba How-to
- Printing How-to

5.3 Novell Environment

As stated in previous sections, Linux can be configured to act as a NCP client or server, thus allowing file and printing services over a Novell network for both Novell and Unix clients

- IPX How-to

5.4 UNIX Environment

The preferred way for sharing files in an UNIX networking environment is through NFS. NFS stands for Network File Sharing and it is a protocol originally developed by Sun Microsystems.It is a way to share files between machines as if they were local. A client "mounts" a filesystem "exported" by an NFS server. The mounted filesystem will appear to the client machine as if it was part of the local filesystem.

It is possible to mount the root filesystem at startup time, thus allowing diskless clients to boot up and access all files from a server.

TOC

More clearly, it is possible to have a fully functional computer without a hard disk.

NFS-related documents:

- http://sunsite.unc.edu/mdw/HOWTO/mini/NFS-Root.html
- http://sunsite.unc.edu/mdw/HOWTO/mini/Diskless.html
- http://sunsite.unc.edu/mdw/HOWTO/mini/NFS-Root-Client.html
- http://www.redhat.com/support/docs/rhl/NFS-Tips/NFS-Tips.html
- http://sunsite.unc.edu/mdw/HOWTO/NFS-HOWTO.html

---

6. Internet/Intranet

Linux is a great platform to act as an Intranet / Internet Server. The term Intranet refers to the application of Internet technologies inside an organisation mainly for the purpose of distributing and making available information inside the company. Internet and Intranet services offered by Linux include Mail, News, WWW servers and many more that will be outlined in the next sections.

6.1 Mail

Mail servers

Sendmail is de facto standard mail server program for UNIX platforms. It is robust, scalable and properly confiured and with the necessary hardware, can handle loads of thousands of users without blinking. Other mail servers (MTA Mail Transport Agents) exist as smail and qmail which are designed as replacements for sendmail.

- Sendmail web site
- Smail faq
- Qmail web site

Mail how-tos:

- http://sunsite.unc.edu/mdw/HOWTO/Mail-HOWTO.html
- http://sunsite.unc.edu/mdw/HOWTO/mini/Qmail+MH.html
- http://sunsite.unc.edu/mdw/HOWTO/mini/Sendmail+UUCP.html
- http://sunsite.unc.edu/mdw/HOWTO/mini/Mail-Queue.html

Remote access to mail.

In an organisation or ISP, users will likely access their mail remotely from their desktops. Several alternatives exist in Linux, including POP (Post Office Protocol) and IMAP (Internet Message Access Protocol) servers. POP protocol is usually used to transfer messages from the server to the client. IMAP admits also manipulation of the messages in the server, remote creation and deletion of folders in the server, concurrent access to shared mail folders, etc.

- Brief comparison IMAP and POP

Mail related how-tos:

- http://sunsite.unc.edu/mdw/HOWTO/Mail-HOWTO.html
- http://sunsite.unc.edu/mdw/HOWTO/mini/Cyrus-IMAP.html

TOC

Mail User Agents

There is a number of MUA (Mail User Agents) in Linux, both graphical and text mode. Most widely used include: pine, elm, mutt and Netscape.

- List of mail related software
- http://sunsite.unc.edu/mdw/HOWTO/mini/TkRat.html

Mailing list software

There are many MLM (Mail List Management) programs available for UNIX in general and for Linux in particular.

- A good comparison of existing MLMs may be found at: ftp://ftp.uu.net/usenet/news.answers/mail/list-admin/
- Listserv
- Majordomo home page

Fetch-mail

One mail-related useful utility is fetchmail. Fetchmail is a free, full-featured, robust, well-documented remote-mail retrieval and forwarding utility intended to be used over on-demand TCP/IP links (such as SLIP or PPP connections). It supports every remote-mail protocol now in use on the Internet. It can even support IPv6 and IPSEC.

Fetchmail retrieves mail from remote mail servers and forwards it via SMTP, so it can then be be read by normal mail user agents such as mutt, elm or BSD Mail. It allows all the system MTA's filtering, forwarding, and aliasing facilities to work just as they would on normal mail

Fetchmail can be used as a POP/IMAP-to-SMTP gateway for an entire DNS domain, collecting mail from a single drop box on an ISP and SMTP-forwarding it based on header addresses.

A small company may centralise its mail in a single mailbox, and program fetchmail to collect all the outgoing mail, send it to the internet and retrieve all incoming one.

- Fetch-mail home page

6.2 Web Servers

Most Linux distribution include Apache (http://www.apache.org). Apache is the number one server on the internet (http://www.netcraft.co.uk/survey/) where more than a half of all internet sites are running Apache or one of it derivatives.Apache advantages include its modular design, stability and speed. Given the appropriate hardware and configuration it can support the highest loads: Yahoo, Altavista, GeoCities, Hotmail are based in customized versions of this server.

Optional support for SSL (which enables secure transactions) is also available at:

- http://www.apache-ssl.org/

TOC

- http://raven.covalent.net/
- http://www.c2.net/

Related How-tos:

- http://sunsite.unc.edu/mdw/HOWTO/WWW-HOWTO.html
- http://sunsite.unc.edu/mdw/HOWTO/Virtual-Services-HOWTO.html
- http://sunsite.unc.edu/mdw/HOWTO/Intranet-Server-HOWTO.html
- Web servers for Linux

6.3 Web Browsers

A number of web browsers exist for the Linux platform. Netscape Navigator is one of the choices from the very beginning and the upcoming Mozilla (http://www.mozilla.org) will have a Linux version. Another popular text based web browser is lynx. It is fast and handy when no graphical environment is available.

- Browser software for Linux
- http://sunsite.unc.edu/mdw/HOWTO/mini/Public-Web-Browser.html

6.4 FTP Servers and clients

FTP stands for File Transfer Protocol. An FTP server allows clients to connect to it and retrieve (download) files. Many ftp servers and clients exist for Linux and are usually included with most distributions. There are text-based clients as well as GUI based.FTP related software (servers and clients) for Linux may be found at: http://sunsite.unc.edu/pub/Linux/system/network/file-transfer/

6.5 News service

Usenet (also known as news) is a big bulletin board system that covers all kind of topics and it is organised hierarchically. A network of computers across internet (Usenet) exchange articles through the NNTP protocol. Several implementations exist for Linux. Either for heavy loaded sites or small sites receiving only a few newsgroups.

- INN home page
- Linux news related software

6.6 Domain Name System

A DNS server has the job of translating names (readable by humans) to an IP address. A DNS server does not know all the IP addresses in the world; rather, it is able to request other servers for the unknown addresses. The DNS server will either return the wanted IP address to the user or report that he name cannot be found in the tables.

TOC

Name serving on Unix (and on the vast majority of the Internet) is done by a program called named. is a part of the bind package of The Internet Software Consortium.

- BIND
- DNS How-to

6.7 DHCP, bootp

DHCP and bootp are protocol that allows a client machine to obtain network information (such as their IP number) from a server. Many organisations are starting to use it because it eases network administration especially in large networks or networks which have lots of mobile users.

Related documents:

- DHCP How-to

6.8 NIS

The Network Information Service (NIS) provides a simple network lookup service consisting of databases and processes. Its purpose is to provide information, that has to be known throughout the network, to all machines on the network. It allows for example that one person could log on in any machine in the network running NIS without the need for the adinistrator to add a password entry for him in all the machines, only in the main database.

Related How-To:

- NIS How-to

6.9 Authentication

There are also various ways of authenticating users in mixed networks For Linux/window NT: http://www.mindware.com.au/ftp/smb-NT-verify.1.1.tar.gz The PAM (pluggable authentication module) which is a flexible way of UNIX authentication: PAM library. Finally, LDAP in Linux

---

7. Remote execution of applications

One of the most amazing features of UNIX (yet one of the most unknown to new users) is the great support for remote and distributed execution of applications.

7.1 Telnet

Telnet is a program that allows a person to use a remote computer as if that person was actually at that site. Telnet is one of the most powerful tools for UNIX, allowing for true remote administration. It is also an interesting program from the point of view of users, because it allows remote access to all their files and programs from anywhere in the Internet. Combined with an X server, there is no difference (apart from the delay) between being at the console or in the other side of the planet. Telnet daemons and clients are available with most Linux distributions.

Encrypted remote shell sessions are available through SSH (http://www.cs.hut.fi/ssh/) thus effectively allowing secure remote administration

- Telnet related software

7.2 Remote commands

In Unix, and in particular in Linux, remote commands exist that allow for interaction with other computers from the shell propt. Examples are: rlogin, which allows for login in a remote machine in a similar way to telnet, rcp, that allows for remote transfer of files among machines,etc. Finally, the remote shell command rsh allows the execution of a command on a remote machine without actually logging onto that machine.

7.3 X-Window

The X Window system was developed at MIT in the late 1980s, rapidly becoming the industry standard windowing system for UNIX graphics workstations. The software is freely available, very versatile, and is suitable for a wide range of hardware platforms.Any X window system consists of 2 distinct parts -- the X server and 1 or more X clients. It is important to realise the distinction between the server and the client. The server controls the display directly, and is responsible for all input/output via the keyboard, mouse or display. The clients, on the other hand, do not access the screen directly - they communicate with the server, which handles all input and output. It is the clients which do the "real" computing work - running applications or whatever. The clients communicate with the server, causing the server to open one or more windows to handle input and output for that client.

In short, the X Window system allows an user to log in into a remote machine, execute a process (for example open a web browser) and have the output displayed on his own machine. Because the process is

actually being executed on the server, very little CPU power is needed in the client. Indeed it is possi to have computers whose primary purpose is to act as a pure X-Windows servers and they are known X-terminals.

A free port of the X Window system exists for Linux and can be found at: Xfree It is usually included in most Linux distributions.

Related how-to:

- Remote X Apps How-to> </itemize> <sect1>VNC <p> VNC stands for Virtual Network Computing. It is, in essence, a remote display system which allows to view a computing 'desktop' environment not only on the machine where it is running, but from anywhere on the Internet and from a wide variety of machine architectures. Both client and server exist for Linux as well as many other platforms. It is possible to execute MS-Word in a Windows NT or 95 machine and have the output displayed in a Linux machine. The opposite is also true, it is possible to execute an application in a Linux machine and have the output displayed in any other Linux or Windows machine. A Java client exists, allowing to run the remote display inside a web browser. Finally a port for Linux using the SVGAlib graphics library allows 386s with as little as 4 Mb of RAM to become fully functional X-Terminals <p> <itemize> <item><url name= site" url="http://www.orl.co.uk/vnc">

8. Network Interconnection

Linux networking is rich in features. A Linux box can be configured so it can act as a router, bridge, etc... Some of the available options are described below

8.1 Router

The Linux kernel has built-in support for routing functions. A Linux box can act either as an IP or IPX router for a fraction of the cost of a commercial router. Recent kernels include special options for machines acting primarily as routers:

- Multicasting : Allows the Linux machine to act as a router for IP packets that have several destination addresses. It is needed on the MBONE, a high bandwidth network on top of the Internet which carries audio and video broadcasts
- IP policy routing: Normally, a router decides what to do with a received packet based solely on the packet's final destination address but routing can also take into account the originating address and the network device from which the packet reached it.

There are some related projects which include one aiming at building a complete running Linux router in a floppy disk: Linux router project

8.2 Bridge

The Linux kernel has built-in support for acting as an ethernet bridge, which means that the different Ethernet segments it is connected to will appear as one Ethernet to the participants. Several bridges can work together to create even larger networks of Ethernets using the IEEE802.1 spanning tree algorithm. As this is a standard, Linux bridges will interwork properly with other third party bridge products. Additional packages allow filtering based on IP, IPX or MAC addresses.

Related How-tos:

- Bridge+Firewall
- Bridge

8.3 IP-Masquerading

IP Masquerade is a developing networking function in Linux. If a Linux host is connected to the Internet with IP Masquerade enabled, then computers connecting to it (either on the same LAN or connected with modems) can reach the Internet as well, even though they have no official assigned IP addresses. This allows for reduction of costs, since many people may be able to access the Internet using a single modem connection as well as contributes to increase security (in some way the machine is acting as a firewall.

since unofficial assigned addresses can not be accessed out of that network)

IP masquerade related pages and documents:

- http://www.tor.shaw.wave.ca/~ambrose/
- http://www.indyramp.com/masq/links.pfhtml
- http://sunsite.unc.edu/mdw/HOWTO/mini/IP-Masquerade.html

8.4 IP-Accounting

This option of the Linux kernel keeps track of IP network traffic, packet logging and produces some statistics. A series of rules may be defined so when a certain packet match a pattern a counter is increased, it is accepted/rejected, etc..

8.5 IP aliasing

This feature of the Linux kernel provides the possibility of setting multiple network addresses on the same low-level network device driver (e.g two IP addresses in one Ethernet card). Typically used for services that act differently based on the address they listen on (e.g. "multihosting" or "virtual domains" or "virtual hosting services".

Related How-to:

- IP Aliasing How-to

8.6 Traffic Shaping

The traffic shaper is a virtual network device that makes it possible to limit the rate of outgoing data flow over another network device. This is specially useful in scenarios (as ISP's), in which it is desirable to control and enforce policies regarding how much bandwidth is used by each client. Another alternative (for web services only) may be certain Apache modules which restrict the number of IP connections by client or the bandwith used.

8.7 Firewall

A firewall is a device that protects a private network from the public part (the internet as a whole). It is designed to control the flow of packets based on the source, destination, port and packet type information contained in each packet.

Different firewall toolkits exist for Linux as well as built-in support in the kernel. Other firewalls are TIS and SOCKS. These firewall toolkits are very complete and combined with other tools allow blocking/redirection of all kind of traffic and protocols. Different policies can be implemented via configuration files or GUI programs.

- TIS home page

- SOCKS
- Firewall How-to

8.8 Port forwarding

An increasing number of web sites are becoming interactive by having cgi-bins or Java applets that access some database or other service. Since this access may pose a security problem, the machine containing the database should not be directly connected to the Internet.

Port Forwarding can provide an almost ideal solution to this access problem. On the firewall, IP packets that come in to a specific port number can be re-written and forwarded to the internal server providing the actual service. The reply packets from the internal server are re-written to make it appear that they came from the firewall.

Port forwarding information may be found here

8.9 Load Balancing

Demand for load balancing arise usually in database/web access when many clients make simultaneous requests to a server. It would be desirable to have multiple identical servers and redirect requests to the less loaded server. This can be achieved through Network Address Translation techniques (NAT) from what IP-masquerading is a subset. Network administrators can replace a single server providing Web services - or any other application - with a logical pool of servers sharing a common IP address. Incoming connections are directed to a particular server using one load-balancing algorithm. The virtual server rewrites incoming and outgoing packets so clients have transparent access to the server as if only one did exist

Linux IP-NAT information may be found here

8.10 EQL

EQL is integrated into the Linux kernel. If two serial connections exist to some other computer (this usually requires two modems and two telephone lines) and SLIP or PPP (protocols for sending Internet traffic over telephone lines) are used on them, it is possible to make them behave like one double speed connection using this driver. Naturally, this has to be supported at the other end as well.

- EQL How-to

8.11 Proxy Server

The term proxy means "to do something on behalf of someone else." In networking terms, a proxy server computer can act on the behalf of several clients. An HTTP proxy is a machine that receives requests for web pages from another machine (Machine A). The proxy gets the page requested and returns the result to Machine A. The proxy may have a cache with the requested pages, so if another machine asks for the

same page the copy in the cache will be returned instead. This allows efficient use of bandwidth resources and less response time. As a side effect, as client machines are not directly connected to the outside world this is a way of securing the internal network. A well-configured proxy can be as effective as a good firewall.

Several proxy servers exist for Linux. One popular solution is the Apache proxy module. A more complete and robust implementation of an HTTP proxy is SQUID.

- Apache
- Squid

8.12 Diald on demand

The purpose of dial on demand is to make it transparently appear that the users have a permanent connection to a remote site. Usually, there is a daemon who monitors the traffic of packets and where an interesting packet (interesting is defined usually by a set of rules/priorities/permissions) arrives it establishes a connection with the remote end. When the channel is idle for a certain period of time, it drops the connection.

- Diald How-to

8.13 Tunnelling, mobile IP and virtual private networks

The Linux kernel allows the tunnelling (encapsulation) of protocols. It can do IPX tunnelling through IP, allowing the connection of two IPX networks through an IP only link. It can also do IP-IP tunnelling, which it is essential for mobile IP support, multicast support and amateur radio. (see http://sunsite.unc.edu/mdw/HOWTO/NET-3-HOWTO-6.html#ss6.13)

Mobile IP specifies enhancements that allow transparent routing of IP datagrams to mobile nodes in the Internet. Each mobile node is always identified by its home address, regardless of its current point of attachment to the Internet. While situated away from its home, a mobile node is also associated with a care-of address, which provides information about its current point of attachment to the Internet. The protocol provides for registering the care-of address with a home agent. The home agent sends datagrams destined for the mobile node through a tunnel to the care-of address. After arriving at the end of the tunnel, each datagram is then delivered to the mobile node.

Point-to-Point Tunneling Protocol (PPTP) is a networking technology that allows the use of the Internet as a secure virtual private network (VPN). PPTP is integrated with the Remote Access Services (RAS) server which is built into Windows NT Server. With PPTP, users can dial into a local ISP, or connect directly to the Internet, and access their network as if they were at their desks. PPTP is a closed protocol and its security has recently being compromised. It is highly recomendable to use other Linux based alternatives, since they rely on open standards which have been carefully examined and tested.

- A client implementation of the PPTP for Linux is available here

Mobile IP:

- http://www-uk.hpl.hp.com/people/jt/mip.html
- http://anchor.cs.binghamton.edu/~mobileip/

Virtual Private Networks related documents:

- http://sunsite.unc.edu/mdw/HOWTO/mini/VPN.html
- http://sites.inka.de/sites/bigred/devel/cipe.html

---

9. Network Management

9.1 Network management applications

There is an impressive number of tools focused in network management and remote administration. Some interesting remote administration projects are linuxconf and webmin:

- Webmin
- Linuxconf

Other tools include network traffic analysis tools, network security tools, monitoring tools, configuration tools, etc An archive of many of these tools may be found at Sunsite

9.2 SNMP

The Simple Network Management Protocol is a protocol for Internet network management services. It allows for remote monitoring and configuration of routers, bridges, network cards, switches, etc... There is a large amount of libraries, clients, daemons and SNMP based monitoring programs available for Linux. A good page dealing with SNMP and Linux software may be found at : http://linas.org/linux/NMS.html

10. Enterprise Linux Networking

In certain situations it is necessary for the networking infrastructure to have proper mechanisms to guarantee network availability near 100% of the time. Some related techniques are described in the following sections. Most of the following material can be found at Linas excellent website: http://linas.org/linux/index.html and in the Linux High-Availability how-to

10.1 High Availability

Redundancy is used to prevent the overall IT system from having single points of failure. A server with only one network card or a single SCSI disk has two single points of failure. The objective is to mask unplanned outages from users in a manner to let users continue to work quickly. High availability software are a set of scripts and tools that automatically monitor and detect failures, taking the appropriate steps to restore normal operation and to notifying system administrators.

10.2 RAID

RAID, short for Redundant Array of Inexpensive Disks, is a method whereby information is spread across several disks, using techniques such as disk striping (RAID Level 0) and disk mirroring (RAID level 1) to achieve redundancy, lower latency and/or higher bandwidth for reading and/or writing, and recoverability from hard-disk crashes. Over six different types of RAID configurations have been defined.There are three types of RAID solution options available to Linux users: software RAID, outboard DASD boxes, and RAID disk controllers.

- Software RAID : Pure software RAID implements the various RAID levels in the kernel disk (block device) code.
- Outboard DASD Solutions : DASD (Direct Access Storage Device) are separate boxes that come with their own power supply, provide a cabinet/chassis for holding the hard drives, and appear to Linux as just another SCSI device. In many ways, these offer the most robust RAID solution.
- RAID Disk Controllers: Disk Controllers are adapter cards that plug into the ISA/EISA/PCI bus. Just like regular disk controller cards, a cable attaches them to the disk drives. Unlike regular disk controllers, the RAID controllers will implement RAID on the card itself, performing all necessary operations to provide various RAID levels.

Related How-tos:

- http://sunsite.unc.edu/mdw/HOWTO/mini/DPT-Hardware-RAID.html
- http://sunsite.unc.edu/mdw/HOWTO/Root-RAID-HOWTO.html
- http://sunsite.unc.edu/mdw/HOWTO/mini/Software-RAID.html

RAID at linas.org:

- http://linas.org/linux/raid.html

10.3 Redundant networking

IP Address Takeover (IPAT). When a network adapter card fails, its IP address should be taken by a working network card in the same node or in another node. MAC Address Takeover: when an IP takeover occurs, it should be made sure that all the nodes in the network update their ARP caches (the mapping between IP and MAC addresses).

See the High-Availability How-To for more details:
http://sunsite.unc.edu/pub/Linux/ALPHA/linux-ha/High-Availability-HOWTO.html

---

11. Sources of Information

Where to get further information:

- Linux: http://www.linux.org
- Linux Documentation Project: http://sunsite.unc.edu/mdw/linux.html (check out the Linux Network Administrator Guide)
- Freshmeat: The latest releases of Linux Software. http://www.freshmeat.net
- Linux links: http://www.linuxlinks.com/Networking/

12. Acknowledgements and disclaimer

This document is based on the work of many other people who have made it possible for Linux to be what it is now: one of the best network operating systems. All credit is theirs.A lot of effort has been put into this document to make it simple but accurate and complete but not excessively long. Nevertheless, no liability will be assumed by the author under any circumstance. Use the information contained here at your own risk. Please feel free to e-mail me suggestions, corrections or general comments about the document so I can improve it. Other topics that will probably be included in futures revisions of this document may include radius, web/ftp mirroring tools such as wget, traffic analyzers, CORBA... and many other that may be suggested and suitable. You can reach me here

Finally I would like to thank FinnBjørn av Teilgum for his careful "beta-testing" of this how-to.

Daniel López Ridruejo 9th May 1998

Linux - Optical Disk HOWTO

Skip Rye, abr@preferred.com

v1.6, 11 December 1998

This document describes the installation and configuration of optical disk drives for Linux. Please, if any one has experiences with optical storage under Linux, send it and I will update it in SGML and forward it to the Linux community. Please let me know if it's OK to include your E-mail address!

1. Disclaimer

2. Copyright

- 2.1 LF1000 mini-HOWTO
- 2.2 Optical Disk-HOWTO

3. Phase Change Optical Technology

- 3.1 Introduction
- 3.2 Panasonic LF1000
- 3.3 Additional Configuration concerns by Jeff Rooze

4. Magneto Optical Technology

- 4.1 Introduction
- 4.2 Olympus, Epson, Mitsubishi MK230LK3 - Stephan Shuichi Haupt
- 4.3 Fujitsu DynaMO 640 - Phil Garcia
- 4.4 Panasonic LF-7010 - Philip Kerr

5. Optical jukeboxes

- 5.1 Maxoptix 520 - Zed Shaw

1. Disclaimer

Neither the author nor the distributors, or any other contributor of this HOWTO are in any way responsible for physical, financial, moral or any other type of damage incurred by following the suggestions in this text.

2. Copyright

The "Optical Disk-HOWTO" and "LF1000 mini-HOWTO" are copyrighted.

2.1 LF1000 mini-HOWTO

(C) 1996,1997 by Skip Rye, abr@brspc_0064.msd.ray.com

2.2 Optical Disk-HOWTO

(C) 1997,1998 by Skip Rye, abr@preferred.com

Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged. The author, however, would like to be notified of any such distributions. All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. In other words, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions. In short we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs. Should you have any questions, please contact Greg Hankins, the Linux HOWTO coordinator, at gregh@sunsite.unc.edu. You may finger his address for phone number and additional contact information.

3. Phase Change Optical Technology

3.1 Introduction

Optical Phase Change technology is used to create "In Phase" or "Out of Phase" bits on a special media for phase change writing. The drive uses a LASER of different power levels or LASER intensities to produce this effect. One power level allows the media to flow into a crystalline form while the other creates an "Out of Phase" condition. The crystallized areas reflect the read Lasers beam with a different coefficient of reflectivity than the non-crystallized areas. Thus, data can be read from the disk.

What makes the phase change optical disk special is that it the disk is formated with concentric cylinders or tracks with each track being sectored much like a magnetic disk or read/write optical disk. The tracks are very close so a lot of data can be stored on a disk. This is different from a CD-ROM in that it gives your system the look and feel of a magnetic disk. CD-ROMs have a spiraling track much like a audio record. Having tracks and sectors alone would not make the phase change drive special from optical disk but the drive has some very special properties; The phase change drive allows for direct overwrite of data which magneto optical can't do inexpensively and the media has the very special property of NOT being susceptible to magnetic fields or as sensitive to static discharge which gives the media a very long shelf life.

3.2 Panasonic LF1000

POINTS OF INTEREST

- Read/Write optical disk.
- Can read CD-ROMs at 4X speed.
- Can read Kodak PhotoCDs.
- Media has a 15 Year shelf life.
- SCSI-2 Interface.
- Track/sector format as opposed to CD-ROMs spiraling record format.
- 165ms access time - much better than a tape file restore.
- 650Mb data storage per diskette.
- Diskettes are about $50 each.

THINGS YOU SHOULD KNOW

- Optical disk format not compatible with any other disk drive.
- Vendors don't seem to support UNIX very well - marketing is targeted for DOS/Windows and Macintosh.
- Do NOT purchase the PD drive which uses the parallel port interface - To my knowledge there is no Linux driver for it.

TOC

Installation

The LF1000 is SCSI-2 compatible device. It features a block size of 512 bytes and is compatible with the Linux SCSI drivers. This drive was installed on a PC compatible AMD 100MHZ 486 with an Adaptec 1542C SCSI bus-master controller. To install and mount a disk the following steps were taken;

Installation steps

- Install the drive and set the SCSI address to not interfere with other SCSI devices. Reconnect all cabling.
- Boot the computer. Your SCSI controller should note the new drive.
- During the Linux kernel boot, you should see an additional SCSI device. In my case, having a magnetic system disk for device /dev/sda it shows up as /dev/sdb.
- I did NOT partition the device because fdisk issued an overwrite warning and I did not want to change anything from a dosemu standpoint.
- mkfs -t ext2 /dev/sdb
- mkdir /pd
- mount -t ext2 -o ro,suid,dev,exec,auto,nouser,async /dev/sdb /pd - Read only
- mount -t ext2 -o defaults /dev/sdb /pd - Mount drive W/R

Your ready to "Rock'n'Roll"

Usage hints

- The media which comes with the drive is reported be re-writable about 500,000 times. This means that it is not advisable to install a live operating system such as Linux on the phase change optical drive. These live operating systems tend to cache processes to and from disk. Over time this can easily approach the phase change media life.
- Mount drive read only as much as possible.
- When writing to the drive do so in large chunks. This will help reduce any file fragmentation which will require more read seeks.
- This is however an excellent media for backups, gifs, mpeg or storing large programs which you don't use that often. The restore from backup is much faster that tape. Backups can be performed using the cp -rp command without the need for the ftape driver. This however, will replace symbolic links with the actual file.
- If while using the PD for writing, You find that the file you just wrote to the disk are not there, chances are that the disk write protect tab is in write protect mode and you mounted it in read/write mode.

3.3 Additional Configuration concerns by Jeff Rooze

Hello,

I read your article on configuring the Panasonic LF-1000 for Linux. I have configured my system so that the optical drive has its own device name and the CD-ROM has its own device name. This has allowed me to mount either media at any time. I do not require any media in the drive when I boot Linux. Also I am using the optical drive as an ext2 formatted media.

I had a couple of minor difficulties in doing so.

First, I had configured my hard drive at SCSI ID 6 and my PD at SCSI ID 4. (I wanted to have the hard drive at a higher priority that the PD). This caused a problem with the Linux SCSI driver. The driver scans the SCSI devices from the Lower SCSI id's to the higher (eg: 0 .. 6). Consequently my logical device names were assigned differently depending on which type of media was installed in the PD drive. This caused a big problem. My Linux partition is on my SCSI hard drive and the root device name would change! I corrected this problem by modifying the software in the kernel SCSI driver to scan the devices in reverse order.

Second, the distribution Linux kernel does not scan all SCSI LUNS. The PD/CD drive has a mode that establishes the CD-ROM at LUN 1 and the PD at LUN 0. This mode is selected by the configuration switches on the PD/CD drive. Switch #2 should be down (off?). If this switch is up (on?), the signature of the device is dependent upon the media that is installed and it only reports this device on LUN 0. If no media is installed I think it defaults to CD-ROM. I am using an Future Domain 16-xx SCSI interface card and the software in Linux kernel driver supports an optical device signature when scanning the LUNS. I assume that this is standard for most of the SCSI drivers. I reconfigured the kernel to enable the "scan all LUNS" switch. The kernel then assigns different device names for each device. The following is an excerpt from by boot log. You will note a series of errors in this log. This is because I did not have the optical media installed in the drive and the driver was attempting to look at the partition table to determine the block size. Fortunately it defaults to 512. I am planning on modifying the Future Domain SCSI driver to not do this when it detects the optical device.

```
>  scsi0 <fdomain>: BIOS version 3.2 at 0xde000 using scsi id 7
>  scsi0 <fdomain>: TMC-18C50 chip at 0x140 irq 12
>  scsi0 : Future Domain TMC-16x0 SCSI driver, version 5.28
>  scsi : 1 host.
>    Vendor: CONNER    Model: CP30545 545MB3.5  Rev: A9AF
>    Type:   Direct-Access                      ANSI SCSI revision: 02
>  Detected scsi disk sda at scsi0, id 6, lun 0
>    Vendor: MATSHITA  Model: PD-1 LF-1000      Rev: A109
>    Type:   Optical Device                     ANSI SCSI revision: 02
>  Detected scsi disk sdb at scsi0, id 4, lun 0
>    Vendor: MATSHITA  Model: PD-1 LF-1000      Rev: A109
>    Type:   CD-ROM                             ANSI SCSI revision: 02
>  Detected scsi CD-ROM sr0 at scsi0, id 4, lun 1
>  fdomain: Selection failed
>  scsi : detected 1 SCSI cdrom 2 SCSI disks total.
>  SCSI Hardware sector size is 512 bytes on device sda
>  fdomain: REQUEST SENSE Key = 2, Code = 3a, Qualifier = 0
>  last message repeated 3 times
>  sdb : READ CAPACITY failed.
>  sdb : status = 0, message = 00, host = 0, driver = 28
>  sdb : extended sense code = 2
>  sdb : block size assumed to be 512 bytes, disk size 1GB.
>  .
>  .
>  .
>  Partition check:
```

```
>    sda: sda1 sda2 sda3
>  scsidisk I/O error: dev 0810, sector 0
>    unable to read partition table of device 0810
```

Third, I modified my file system table (/etc/fstab) to list each device but do not attempt to auto mount when booting. I have included an excerpt from my fstab. The most important options are the noauto, rw(ro), and the checkpass flag.

To create a ext2 file system on the PD, I used the command "mkfs.ext2 -i 2048 /dev/sdb".

```
# fstab - List of file systems
#
# device  mount   type          options              dumpfrequency
checkpass
/dev/sdb /optd    ext2   rw,user,suid,noauto,sync,exec,dev,umask=0 0 2
/dev/sr0 /dist  iso9660  ro,user,suid,noauto,sync,exec,dev 0 2
```

After making these changes, I have had no problems with mounting either media. All I need to do is to load the media and type "mount /optd" or "mount /dist" and the system does all the rest.

I hope this information is useful.

```
Jeff
--
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
\ Jeff Rooze -- http://www.treknet.net/~jrooze -- jrooze@treknet.net /
/  If builders built buildings the way some programmers write        \
\  programs, then the first woodpecker that came along would destroy /
/  civilization.                                        GERALD WEINBERG \
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

I tried Jeff's suggestion. Here are the steps I performed;

- Modify my kernel using "make xconfig" in the /usr/src/linux directory and installed it.
- Change the mode jumper on the PD drive to non-DOS mode. I soldered a switch across the mode jumper connections and routed it the the back panel. I figured out which switch position was the open position and labeled this one for DOS. The other position is of course Linux. So before I boot my system I decide which OS I'll be using and set the switch accordingly. History shows it staying in the Linux position more and more.
- Reboot your system. You should now see multiple LUN show up during boot for the PD SCSI device number - It works great!!! If you have an older kernel modify the "/usr/src/linux/drivers/scsi/config.in" file.
- Update the fstab for both CD and PD drives.
- Use appropriate mount command.
- "df" to make sure your ready.

I did try moving my primary SCSI drive to 6 but experienced some difficulties. Can't remember exactly what it was but it may have been that my controller "Adaptec 1542" with "Corel SCSI" requires a bootable disk and SCSI 0 for the BIOS install to work properly with DOS. So I switched it back and enjoyed playing with my

properly install PD drive! With this configuration "workman" - the audio CD player util - works fine.

4. Magneto Optical Technology

4.1 Introduction

Magneto optical drives use a "Far field" magnetic field and a laser to change polarization of a magnetic media. The media is of such a nature that it must be heated to the appropriate temperature before a polarization change can happen - this is where the laser comes into play. A high power write laser is used to heat the disk surface to the appropriate temperature at which time the "Far field" can set the polarization on the disk magnetic surface. After a short period of time the disk surface cools and "locks" the polarization into place. The read back I'm a little fuzzy on - someone please send me the proper wording. I think a low power laser is used for read back and the "H" field of the disk polarization interacts with the "E" and "H" field of the incident laser to produce a reflective polarization which will correspond to the disk bit polarization - I hope this is in the ball-park, it's certainly no home run. Maybe a total strike out.

The use of a laser for polarization change allows the disk bit and track densities to be higher than conventional "Flying" magnetic heads. The "far field" means no more "head crashes" - that is assuming your disk label doesn't peal off during the load or you don't leave one of those sticky pads on the disk cartridge. Most media allows 650 Megs per platter and on some models both sides of the media is used yielding 1.3Gig storage media - you must remove the media and flip it over to use the other 650Megs though.

4.2 Olympus, Epson, Mitsubishi MK230LK3 - Stephan Shuichi Haupt

Stephan Shuichi Haupt <stephan@bios.t.u-tokyo.ac.jp>

```
Hi

I have noticed that there is not much information about
magneto-optical disks in the howto, which may be due to the fact that
these are not very popular in general. In Japan, MO drives are very
common, especially the 3.5' variety with media in 128MB (maybe not
available anymore), 230MB, and recently 640MB sizes. I suppose there
is plenty of info on usage of these drives with Linux in Japanese -
but that does not help most people for some reason ;-) MODs can be
used very much like any removable media and are handy for smaller
backups as the media are relatively inexpensive (about 10US$ / 640MB
as of 10-98). I can only comment on the usage of 230MB drives with
SCSI interface.

Drives used: several, no problems encountered (Olympus, Epson, currently
Mitsubishi MK230LK3). Drives may have strange jumper setting like "Mac
Mode" or such - naturally, disable.
If you decide to get a drive, pay attention the the
```

cache size - It can speed things up enormously, still speed will be soso compared to hard disks, of course.

SCSI controllers: NCR53C810-based (Asus PCI-200), Adaptec APA-1460A, Adaptec AHA2940.

Just install the drive as you would do with an additional SCSI hard disk. It will show up as such. You don't need a disk in the drive when booting.

There are two ways to format the disks:
a) A bit like a floppy. Just run mkfs on the raw device i.e. something like sdb or sdc. I don't recommend this in general (see below).
b) Like a hard disk. Do fdisk on the raw device and then mkfs on the partition as you would for a hard disk (like sdc0, I have never made multiple partitions on a MOD).

What I have not tried is to boot from MOD, yet I cannot see why it should not work. I would only recommend it for emergency system recovery, however, due to MO drive performance.

Note: Purchased disks for Doze or Windog may be formatted "like floppies" and cannot be used with either O(gre)S right away while MODs formatted under linux as hard disks (partition FAT16 / type 6 and mkdosfs) will work fine (only tested with NT 3.5/4.0). Fdisk will issue a warning upon exit that concerned FAT16 partitions and you do better to take it seriously (look at the fdisk man-page). The sector size will not be automatically set properly for mkdosfs. Use "mkdosfs -s 8". That came from some Japanese Web site in mid 1995 (Thanks to Ken Kawabata for finding and deciphering it). Using the vfat file-system with the disks works fine. I have only used FAT/DOSfs or Linux/ext2 formatted disks so far.

Additional Note: The media are probably a bit sensitive. Of course to magnetic fields, but also to mechanical stress, some formats seem to be more fragile than others (Mac format seemingly worst, data loss has occurred when dropping disks during sneaker net traffic).

Though this does not steer anyone through particularly dense jungle, it may be nice for completeness.

Steve

--
**********cut*here*or*do*not**********
S. Shuichi Haupt
email stephan@bios.t.u-tokyo.ac.jp

```
        http://www.bios.t.u-tokyo.ac.jp/~stephan/

---------------- December 11 1998 update from Steve -------------------

OK, some problems will arise with MO disks occasionally. the safest
way to avoid them is not to use the disks "off the shelf".  trying to
mount disks can even result in kernel panics. i accidentally tried to
mount a 640MB disk (format windows95 it said, so maybe FAT32) as -t
vfat, this is not a thing to try.

also, 2.0.x kernels don't support 2048b block size (also 640MB disks).
a patch for 2.0.3x kernels seems to float around somewhere in Japan,
but i have not yet gotten hold of it.  here a link that certainly has
an English description:
http://elektra.e-technik.uni-ulm.de/~mbuck/linux/patches.html
or search the u-tokyo.ac.jp domain. the page of the developers is
hidden somewhere.

the best way to use these 640MB disks is therefore to do fdisk and
mkfs first. i have only done this with mke2fs on type 83 partitions:
mke2fs -b 2048 /dev/sdxy

i will check it out for FAT16 partitions and mkdosfs when i have some
spare time and disks.

my kernel version used is 2.1.124 (for all of the above).

Steve
--
**********************cut*here*or*do*not*******************************
        Stephan Shuichi
office: Dept. for Mechano-Informatics, Yoshizawa Lab.
        Faculty for Engineering, University of Tokyo
        Tel 03-3812-2111 ext 6390, FAX 03-5802-2957
        email stephan@bios.t.u-tokyo.ac.jp
        http://www.bios.t.u-tokyo.ac.jp/~stephan/
private: --
```

4.3 Fujitsu DynaMO 640 - Phil Garcia

pgarcia@execpc.com

```
  You've probably already received a number of messages regarding the
Fujitsu DynaMO 640 - I have the 640SZI, which is the internal version;
the model number given in a SCSI probe is M2513-MCC3064SS.  I recently
installed this drive practically without a hitch.  I say practically
because the sector size of the 640 MB disks is 2048 bytes, which is
```

not supported in the Linux 2.0.x kernel but is supported in the development kernels. A patch for 2.0.x is available at http://wwwcip.informatik.uni-erlangen.de/~orschaer/mo/
-- also at this site is a patched fdisk to use in conjunction with it.

Otherwise, installing the drive was no different from installing a SCSI hard drive. It runs well, and I'm very happy with it.

Phil Garcia

4.4 Panasonic LF-7010 - Philip Kerr

philip_kerr_at_wmc__brsf2@wmcmail.wmc.ac.uk

```
Dear Skip

In your Optical HOWTO, you asked for anyone else's experiences of
installing optical drives under Linux.

Please find below details of how I managed to get a Panasonic LF-7010
(SCSI) working on my Sparc Classic.

I'm using Redhat, 4.2 and 5.1

Regards

Philip Kerr
philip.kerr@wmc.ac.uk


ps I'm now trying to get the drive to work under Solaris 2.6... it's
not an easy a job as it was under Linux!!
------------------------


plugged the drive in (on id5)...

powered up the Sparc...


the following came up....

scsi0 : Sparc ESP100A-FAST
scsi : 1 host.
Vendor: SAMSUNG   Model: WN32162U          Rev: 0100
Type:   Direct-Access                      ANSI SCSI revision: 02
```

```
Detected scsi disk sda at scsi0, channel 0, id 3, lun 0
Vendor: MATSHITA  Model: LF-7010  (00:06)  Rev: 1.42
Type:   Optical Device                      ANSI SCSI revision: 02
Detected scsi removable disk sdb at scsi0, channel 0, id 5, lun 0 scsi
: detected 2 SCSI disks total.
esp0: target 3 [period 100ns offset 15 10.00MHz FAST SCSI-II]
SCSI device sda: hdwr sector= 512 bytes. Sectors= 4236661 [2068 MB]
[2.1 GB]
esp0: target 5 [period 248ns offset 4 4.03MHz synchronous SCSI] sdb :
READ CAPACITY failed.
sdb : status = 0, message = 00, host = 0, driver = 28 sdb : extended
sense code = 2
sdb : block size assumed to be 512 bytes, disk size 1GB.
sunlance.c:v1.9 21/Aug/96 Miguel de Icaza (miguel@nuclecu.unam.mx)
eth0: LANCE 08:00:20:04:3d:cf
eth0: using auto-carrier-detection.
Partition check:
sda: sda1 sda2 sda3 sda4 sda5 sda6 sda7 sda8
sdb:scsidisk I/O error: dev 08:10, sector 0, absolute sector 0 unable
to read partition table
```

I edited my fstab, adding the entry for the drive (on sdb)

```
==========
/etc/fstab
==========
/dev/sda1          /                       ext2    defaults        1 1
/dev/sda2          swap                    swap    defaults        0 0
/dev/fd0           /mnt/floppy             msdos   noauto,user     0 0
/dev/sr0           /mnt/cdrom              iso9660 noauto,ro,user  0 0
/dev/sdb           /mnt/optical            ext2    noauto,rw,user  0 0
none               /proc                   proc    defaults        0 0
```

Then mkfs'ed a blank disc as follows...

```
[root@localhost me]# /sbin/mkfs -t ext2 /dev/sdb

mke2fs 1.10, 24-Apr-97 for EXT2 FS 0.5b, 95/08/09 /dev/sdb is entire
device, not just one partition! Proceed anyway? (y,n) y
Linux ext2 filesystem format
Filesystem label=
118320 inodes, 472448 blocks
23622 blocks (5.00%) reserved for the super user First data block=1
Block size=1024 (log=0)
Fragment size=1024 (log=0)
58 block groups
8192 blocks per group, 8192 fragments per group 2040 inodes per group
Superblock backups stored on blocks:
```

```
8193, 16385, 24577, 32769, 40961, 49153, 57345, 65537, 73729, 81921,
90113, 98305, 106497, 114689, 122881, 131073, 139265,
147457,
155649, 163841, 172033, 180225, 188417, 196609, 204801,
212993, 221185,
229377, 237569, 245761, 253953, 262145, 270337, 278529,
286721, 294913,
303105, 311297, 319489, 327681, 335873, 344065, 352257,
360449, 368641,
376833, 385025, 393217, 401409, 409601, 417793, 425985,
434177, 442369,
450561, 458753, 466945

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

rebooted...

mounted the drive...

I've since then edited the fstab, adding the following mount-point...

/dev/sdb           /mnt/dostical          msdos   noauto,rw,user 0 0

I can now mount ext2 or dos formatted optical carts by mounting either
optical or dostical.
```

5. Optical jukeboxes

I have no experience with optical jukeboxes with Linux!!!! I have had experiences with Optical jukeboxes under HP-UX. In this setup the the jukebox had a SCSI address of it's own. Each slot in the jukebox had an associated LUN number. A device name was assigned for each disk slot A side and B side. The mount command was run against the appropriate device name. I had a jukebox with just one drive and 16 optical disk slots - 20 Gig. I thought it was going to be a real hassle to write a disk mount manager to share this drive among users until I discovered you can mount as many disk as you want and the jukebox driver takes care of arbitration - what a nice feature. Granted, you only want archive type data here and your overall system configuration to be such that not too many processes will be accessing the jukebox at the same time. The disk spin down, carriage load, carriage move, carriage unload, carriage move to the next disk, carriage next disk load, carriage move, optical drive load, and spin up takes about 12 seconds - "seek-from-hell".

5.1 Maxoptix 520 - Zed Shaw

shawz@imap1.asu.edu

Zed's Origional E-Mail - Feb 13 1998

```
Hi,

I was reading your howto (a life saver, thanks) and I was wondering what
kind of jukebox you were running?  I have a Maxoptix 520 Jukebox (20
disks at 2.6G each, nice!) and I would like to connect it to a Linux box
and serve the drives up to my users, but I'm having problems accessing
the individual drives.   Currently I can only access the two drives and
something called MAXLYB which I think is a controller device of some
sort.

Basically, I'm wondering if the jukebox you had was the same or similar
and how you set it up.  I know that you did it under HP-UX, but any help
right now would be nice.  Hey, I'll even let you log into my linux
server if you want to take a look at the jukebox and see what it does.
You can't beat 52Gig of storage!

Anyway, I'd really appreciate your help.


Zed A. Shaw
Application Systems Analyst
Arizona State University
```

Corrospondance with Zed on Mon, 16 Feb 1998:

TOC

> It sounds like your Maxoptix 520 is a jukebox with two physical disk.
Yep, that's the one.

>
> All jukeboxes have a carriage controller. This is probably your MAXLYB
> device.
> ...

What I've come to find out is that Maxoptix is pretty stingy when it comes to drivers. Apparently, they don't make driver software for any of their Jukebox carriage controller interfaces! I don't know how some of these companies stay in business. I'm going to pester them again soon, but you are right, this thing will need a carriage controller driver to operate. The cool thing is that this MX520 (that's the model number of the juke) emulates a whole slew of other carriage controllers, so maybe one of those other guys has a driver. I'll be looking into that too.

>
> You might want to get a-hold of Maxoptix and see if they have a install
> package for your linux kernel version. If not ask them for the programmers
> specification for the carriage controller and maybe we can write one!
>

Hey, if I can't find any driver software, and I can convince Maxoptix to give me the specs, I'd be more than glad to write a driver. I'd could sure use the help too since I haven't got enough time to do it on my own. Also, do you know of anyone else doing this that we might be able to hack off of?

> Any information you find, let me know and we will roll the information
> into the Optical HOWTO, acknowledgments of course!
>

Sure, but let me get some new information first. So far things are looking pretty bleak.

>
> >Basically, I'm wondering if the jukebox you had was the same or similar
> >and how you set it up. I know that you did it under HP-UX, but any help
> >right now would be nice. Hey, I'll even let you log into my linux
> >server if you want to take a look at the jukebox and see what it does.
> >You can't beat 52Gig of storage!
>
> Nice. At home I can use PPP to mount my 84 platter HP-UX jukebox.

> It's slow though - I wish I had it at home.

Oh, I don't have this thing at home. There's no way I could afford the $30,000 my boss paid for this thing. But he's stuck with it and has had it sitting around collecting dust for a year, so he's letting me play with it and try to find a use for it.

I'll get back with you when I have some more information. It should be sometime this week when I find out if I can get it to work or not.

Zed

Oracle Database HOWTO

Paul Haigh, paul@nailed.demon.co.uk

v1.2, 4 August 1998

A guide to installing and configuring the Oracle Database Server on a Linux system

1. Introduction

- 1.1 Version History
- 1.2 Copyright
- 1.3 Disclaimer
- 1.4 Aim of the HOWTO
- 1.5 Requirements
- 1.6 News From Oracle Corporation

2. Installing the Oracle Software

- 2.1 Server Preparation
- 2.2 Installing from CDROM
- 2.3 Post Installation Tasks

3. Creating a Database

- 3.1 Create the Initialisation File
- 3.2 Creating the Database Install Script
- 3.3 Running the Database Installation Script
- 3.4 Starting the Database
- 3.5 Stopping the Database
- 3.6 Create a Default User

TOC

4. Configuring SQL*Net on the Server

- 4.1 `tnsnames.ora`
- 4.2 `listener.ora`
- 4.3 `sqlnet.ora`
- 4.4 Starting and Stopping the Listeners

5. Client Configuration

- 5.1 Windows Clients
- 5.2 Unix Clients

6. Automatic Startup and Shutdown

- 6.1 `dbstart` & `dbstop`
- 6.2 `init.d` & `rc.d`

7. Other Bits

- 7.1 Intelligent Agent

8. Troubleshooting

- 8.1 I cannot create a database when using Oracle 7.2.x.
- 8.2 I'm getting segmentation faults in `svrmgrl` under version 7.3.4.x.

9. Credits

---

1. Introduction

1.1 Version History

- v0.1 - 21 Feb 1998 - Paul Haigh - Original Version.
- v0.2 - 01 Mar 1998 - Paul Haigh - Comments From Proofreaders Added.
- v1.0 - 10 Mar 1998 - Paul Haigh - First Release to LDP.
- v1.1 - 20 Jun 1998 - Paul Haigh - Added troubleshooting section & general tidyup.
- v1.2 - 04 Aug 1998 - Paul Haigh - Added Oracle Corp News & Removed Section on Future Enhancements.

1.2 Copyright

The Oracle Database HOWTO copyright (c) 1998, Paul Haigh.

Like all Linux HOWTO documents, this may be reproduced and distributed in whole or in part, in any medium, physical or electronic, so long as this copyright notice is retained on all copies.

Commercial redistribution is allowed and encouraged; however the author would like to be notified of such distributions. You may translate this HOWTO into any language whatsoever provided that you leave this copyright statement and disclaimer intact, and that you append a notice stating who translated the document.

1.3 Disclaimer

While I have tried to include the most correct and up to date information available to me, I cannot guarantee that usage of information in this document does not result in loss of data or equipment. I provide NO WARRANTY about the information in the HOWTO and I cannot be made liable for any consequences resulting from using the information in this HOWTO.

1.4 Aim of the HOWTO

In this HOWTO I will attempt to cover installation and basic admin of an Oracle database running on a Linux machine. In particular I will cover Oracle server installation, SQL*Net configuration and client configuration.

This document is not an in depth tutorial on using or administering an Oracle database, if that is what you

TOC

are looking for there are great books on those subjects published by O'Reilly and others.

I am also not going to cover the development of Oracle programs under UNIX. If this is absolutley necessary to you then I would recommend that you purchase the SCO development system (with OpenServer 5.x) from SCO, which I am told can be obtained for a very reasonable US $19, from `www.sco.com`.

1.5 Requirements

I am assuming a number of items that you will need for following the HOWTO.

- **Oracle Server CD for SCO Openserver (Version 7.3.3.0.0.)**

  This **must** be a legal copy. Remember that Oracle are a profit making company and charge for their products. If you want a free SQL compliant database use PostgresSQL or similar.

  It is also possible to install oracle, using a 60 day evaluation licence, from a downloadable `tar` file from the Oracle web site. I have not personally tried this and it is completely unverified.

- **A Linux Server**

  You wouldn't be reading this without one...would you?

- **Kernel 2.0.30+**

  I cannot guarantee that these instructions will be accurate for any other Kernel. (Not that I am guaranteeing it for 2.0.30 either...).

- **iBCS**

  It is very important to have this installed and working with the latest possible version for your platform. (I am using iBCS-2.0-10.i386.rpm from Redhat Linux).

- **Lots of disc space**

  600 Mb+ is a reasonable amount. It is possible to install with less but you need to make some sacrifices, and I never like starting with those. However, I will attempt to point out areas in which space can be freed up.

- **32Mb+ Ram**

  I know that this sounds like a lot, especially in Linux terms, but remember that Oracle is a complex piece of software. You wouldn't have the same reservations on SCO!

  I am not saying that Oracle doesn't work with less, just that it is less than Oracle recommend and I wouldn't suggest it.

- **Licenses from Oracle**

  I know that I have already mentioned this but I want to be clear that this is important. Using software from Oracle without a license is illegal.

1.6 News From Oracle Corporation

TOC

Oracle have bowed into pressure from the Linux community. Oracle Corporation have decided to **officially** support Oracle 8 on the Linux (i386) platform. This should be released in December 1998, according to the Oracle website.

Better still Oracle will also be porting `Oracle Applications` to the linux platform. This should be available in the first half of 1999, according to the Oracle website.

References:

- http://www.oracle.com/html/linux.html
- http://www.news.com/News/Item/0,4,24436,00.html
- http://www.zdnet.com/pcweek/news/0720/20morac.html

---

2. Installing the Oracle Software

2.1 Server Preparation

Creating an Oracle User

Unsurprisingly we require a user to hold the Oracle database. Since we have no desire to relink the Oracle kernel (more about that later) we have to accept the Oracle defaults for user name and group name. This includes the user ORACLE and the group DBA.

1. Login as Root
2. Create the oracle user and the group dba.

```
$ groupadd dba
$ useradd  oracle
```

3. Ensure a home directory is created for the user oracle.

```
$ mkdir /home/oracle
$ mkdir /home/oracle/7.3.3.0.0 (Version of Oracle)
$ chown -R oracle.dba /home/oracle
```

2.2 Installing from CDROM

Unfortunately the Oracle Installer on the SCO disc will not work. A variety of problems can be experienced, from core dumps to hangs. As a result we need to copy the files from the CDROM manually and uncompress them:

(Ensure the CDROM is mounted on the system).

1. Log on as Oracle
2. Change directory to `/home/oracle/7.3.3.0.0.`
3. Copy all install files from CDROM

```
$ cp -a /mnt/cdrom/* .
```

4. Un-compress all Oracle files on CDROM.

```
$ find . -name *_ -exec ~/7.3.3.0.0/orainst/oiuncomp {} \;
```

TOC

2.3 Post Installation Tasks

Tasks for Root

Add the following lines to `/etc/profile` or add to the `.profile` for each user who is going to use Oracle.

```
# Oracle Specific
ORACLE_HOME=/home/oracle/7.3.3.0.0
ORACLE_SID=orcl
ORACLE_TERM=vt100
export ORACLE_HOME ORACLE_SID ORACLE_TERM

# Alter path for Oracle
PATH="$PATH:$ORACLE_HOME/bin"
```

We also need to change the owner and permissions of the Oracle ulimit increase utility.

```
$ chown root.root $ORACLE_HOME/bin/osh
$ chmod u+s $ORACLE_HOME/bin/osh
```

Tasks for Oracle

Change permissions for the Oracle files to ensure correct operation.

```
$ chmod +x $ORACLE_HOME/bin/*
$ chmod u+s $ORACLE_HOME/bin/oracle
```

Oracle tools require the messages to be in the `$ORACLE_HOME/tool_name/mesg` directory. So, move the `msb` files from the `msg_ship` directories to the `mesg` directories.

```
$ mv $ORACLE_HOME/plsql/mesg/mesg_ship/* $ORACLE_HOME/plsql/mesg/.
$ mv $ORACLE_HOME/rdbms/mesg/mesg_ship/* $ORACLE_HOME/rdbms/mesg/.
$ mv $ORACLE_HOME/svrmgr/mesg/mesg_ship/* $ORACLE_HOME/svrmgr/mesg/.
```

Create the following directories if they do not exist:

```
$ mkdir $ORACLE_HOME/rdbms/log
$ mkdir $ORACLE_HOME/rdbms/audit
$ mkdir $ORACLE_HOME/network/log
```

Things you can remove

The following directories can safely be removed:

- `$ORACLE_HOME/guicommon2/`
- `$ORACLE_HOME/ctx/`
- `$ORACLE_HOME/md/`
- `$ORACLE_HOME/mlx/`
- `$ORACLE_HOME/precomp/`
- `$ORACLE_HOME/slax/`

3. Creating a Database

Now the Oracle server is installed we need to create a database to test the installation.

`If you are using Oracle 7.2.x or earlier, please read the troubleshooting section below.`

3.1 Create the Initialisation File

Copy the `$ORACLE_HOME/dbs/init.ora` to `$ORACLE_HOME/dbs/initorcl.ora`:

```
$ cd $ORACLE_HOME/dbs
$ cp init.ora initorcl.ora
```

Modify it by adding the following lines:

```
db_name = orcl
COMPATIBLE=7.3.3.0.0
```

3.2 Creating the Database Install Script

Create a script file called makedb.sql in the `$ORACLE_HOME/dbs` directory:

```
connect internal
startup nomount
set echo on
spool makedb.log
create database orcl
        maxinstances 1
        maxlogfiles  8
        datafile '$ORACLE_HOME/dbs/orcl_syst_01.dbf' size 40M reuse
        logfile
                '$ORACLE_HOME/dbs/orcl_redo_01.dbf' size 1M reuse,
                '$ORACLE_HOME/dbs/orcl_redo_02.dbf' size 1M reuse,
                '$ORACLE_HOME/dbs/orcl_redo_03.dbf' size 1M reuse;
@$ORACLE_HOME/rdbms/admin/catalog.sql
create tablespace rollback
        datafile '$ORACLE_HOME/dbs/orcl_roll_01.dbf' size 8.5M reuse;
create tablespace temp
        datafile '$ORACLE_HOME/dbs/orcl_temp_01.dbf' size 5M reuse
```

```
        temporary;
create tablespace users
        datafile '$ORACLE_HOME/dbs/orcl_user_01.dbf' size 10M reuse;
create rollback segment r1 tablespace rollback
        storage ( optimal 5M );
alter rollback segment r1 online;
connect system/manager
@$ORACLE_HOME/rdbms/admin/catdbsyn.sql
connect internal
@$ORACLE_HOME/rdbms/admin/catproc.sql
connect system/manager
@$ORACLE_HOME/sqlplus/admin/pupbld.sql
spool off
exit
```

3.3 Running the Database Installation Script

Start `svrmgrl` and run the script:

```
$ cd $ORACLE_HOME/dbs
$ svrmgrl

Oracle Server Manager Release 2.3.3.0.0 - Production

Copyright (c) Oracle Corporation 1994, 1995. All rights reserved.

Oracle7 Server Release 7.3.3.0.0 - Production Release
PL/SQL Release 2.3.3.0.0 - Production

SVRMGR> connect internal
Connected.
SVRMGR> startup nomount
ORACLE instance started.
Total System Global Area       4313312 bytes
Fixed Size                       41876 bytes
Variable Size                  4140364 bytes
Database Buffers                122880 bytes
Redo Buffers                      8192 bytes
SVRMGR> @makedb
<loads of messages>
SVRMGR> exit
Server Manager complete.
```

3.4 Starting the Database

Firstly, we need to bring up the database by hand (we will automate this later on). To startup an Oracle database we need to issue the `startup` command when connected internally:

```
$ svrmgrl

Oracle Server Manager Release 2.3.3.0.0 - Production

Copyright (c) Oracle Corporation 1994, 1995. All rights reserved.

Oracle7 Server Release 7.3.3.0.0 - Production Release
PL/SQL Release 2.3.3.0.0 - Production

SVRMGR> connect internal
Connected.
SVRMGR> startup
ORACLE instance started.
Total System Global Area       4313316 bytes
Fixed Size                       41876 bytes
Variable Size                  4140368 bytes
Database Buffers                122880 bytes
Redo Buffers                      8192 bytes
Database mounted.
Database opened.
SVRMGR> exit
Server Manager complete.
```

3.5 Stopping the Database

It is worth mentioning here that restarting a Linux server without shutting down the Oracle database first there is a high risk of corrupting the database.

So, before we issue the Linux `shutdown` command it is wise to bring down the database:

```
$ svrmgrl

Oracle Server Manager Release 2.3.3.0.0 - Production

Copyright (c) Oracle Corporation 1994, 1995. All rights reserved.

Oracle7 Server Release 7.3.3.0.0 - Production Release
PL/SQL Release 2.3.3.0.0 - Production

SVRMGR> connect internal
```

```
Connected.
SVRMGR> shutdown
Database closed.
Database dismounted.
ORACLE instance shut down.
SVRMGR> exit
Server Manager complete.
```

3.6 Create a Default User

The database, as created, has a two special users which are automatically created. These are:

Username	Password
SYSTEM	MANAGER
SYS	change_on_install

These users are typically used to hold the standard data dictionary information for the database. It is a good idea to change the passwords from the defaults as soon as possible.

This can be achieved by:

```
sqlplus system/manager

SQL*Plus: Release 3.3.3.0.0 - Production on Sat Feb 21 12:43:33 1998

Copyright (c) Oracle Corporation 1979, 1996.  All rights reserved.


Connected to:
Oracle7 Server Release 7.3.3.0.0 - Production Release

SQL> alter user system identified by <newpassword>;

User altered.

SQL> alter user sys identified by <newpassword>;

User altered.

SQL> exit;
Disconnected from Oracle7 Server Release 7.3.3.0.0 - Production Release
PL/SQL Release 2.3.3.0.0 - Production
```

TOC

Since the user `system/manager` is similar to using `root` on a UNIX machine, we need to create a user less ability to cause damage. (remember to bring up the database before attempting to create a user)

Connect to SQL*Plus and create a user:

```
$ sqlplus system/manager

SQL*Plus: Release 3.3.3.0.0 - Production on Sat Feb 21 12:43:33 1998

Copyright (c) Oracle Corporation 1979, 1996.  All rights reserved.


Connected to:
Oracle7 Server Release 7.3.3.0.0 - Production Release
PL/SQL Release 2.3.3.0.0 - Production

SQL> create user <user> identified by <psw>
  2  default tablespace users
  3  temporary tablespace temp;

User created.

SQL> grant connect, resource to <user>

Grant succeeded.

SQL> exit
Disconnected from Oracle7 Server Release 7.3.3.0.0 - Production Release
PL/SQL Release 2.3.3.0.0 - Production
```

Now that you have a new user on the system you can play with the new system. To login to the Oracle database:

```
$ sqlplus <user>/<password>
```

If this completes with no error messages then you have a working Oracle database. If you never want to connect to this database from anywhere but this server then the job is complete, enjoy!

If, however, like most people you want to configure the networking software so that you can connect from other machines, keep on reading.

4. Configuring SQL*Net on the Server

All of these files configure the Oracle networking software (SQL*Net, aka Net8 for Oracle8). These files should all be created on the server in the `$ORACLE_HOME/network/admin` directory.

4.1 `tnsnames.ora`

The `TNSNAMES.ORA` file identifies services available from the machine. On our instance here we will describe all databases that the server has mounted. For each database instance on your server add a section like below:

```
orcl.world =
  (DESCRIPTION =
    (ADDRESS_LIST =
        (ADDRESS =
          (COMMUNITY = tcp.world)
          (PROTOCOL = TCP)
          (Host = <INSERT HOST NAME OF SERVER HERE> )
          (Port = 1521)
        )
        (ADDRESS =
          (COMMUNITY = tcp.world)
          (PROTOCOL = TCP)
          (Host = <INSERT HOST NAME OF SERVER HERE> )
          (Port = 1526)
        )
    )
    (CONNECT_DATA = (SID = ORCL)
    )
  )
```

4.2 `listener.ora`

The `listener.ora` file contains the descriptions of the services that other machines are allowed to connect to and any configuration that is required for the server listener.

It contains sections for the listener name, listener address, databases served by the listener and configuration parameters.

Here is an example:

```
# Name of listener and addresses to listen on
LISTENER =
        ( ADDRESS_LIST =
                (ADDRESS =
                        (PROTOCOL=tcp)
                        (HOST=<INSERT HOST>)
                        (PORT=1521)
                        (COMMUNITY=UK_SUP_TCPIP)
                )
```

```
                (ADDRESS =
                        (PROTOCOL=ipc)
                        (KEY=700)
                        (COMMUNITY=UK_SUP_TCPIP)
                )
        )

# List of services served by this listener
SID_LIST_LISTENER=
        (SID_LIST=
                (SID_DESC=
                        (SID_NAME=orcl)
                        (ORACLE_HOME=/home/oracle/7.3.3.0.0)
                )
        )

# Start of configuration parameters.
TRACE_LEVEL_LISTENER=OFF
TRACE_FILE_LISTENER = "listener"
LOG_FILE_LISTENER = "listener"
CONNECT_TIMEOUT_LISTENER = 10
STOP_LISTENER = YES
DBA_GROUP = dba
```

4.3 `sqlnet.ora`

The `sqlnet.ora` file contains configuration for the particular node of the network. This is independent of the number of databases or the number of listeners. The most important thing in this file is the `Dead Connection Timeout` configuration variable.

Dead connection timeout checks every incoming process to a database instance and ensures that the client end of it is still responding. If the client (of whatever type) is not responding then the Oracle server shadow process is killed.

This is very useful if you have many clients accessing a database, especially during a developmental phase when those clients are more likely to be failing to exit cleanly from the Oracle database.

Below is a copy of my own `sqlnet.ora` file for you to puruse:

```
TRACE_LEVEL_CLIENT = OFF
sqlnet.expire_time = 30         # The number of seconds between client checks.
names.default_domain = world
name.default_zone = world
```

4.4 Starting and Stopping the Listeners

Now that the configuration of the listeners and SQL*Net is complete we can attempt to connect to the database using the networking software. (Before we were using direct links to the database, whereas here we are simulating a connection from a remote client machine).

To start the listener using the above configuration:

```
$ lsnrctl
```

TOC

```
LSNRCTL for SCO System V/386: Version 2.3.3.0.0 - Production on 23-FEB-98 20:3

Copyright (c) Oracle Corporation 1994.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> start
Starting /home/oracle/7.3.3.0.0/bin/tnslsnr: please wait...

TNSLSNR for SCO System V/386: Version 2.3.3.0.0 - Production
System parameter file is /home/oracle/7.3.3.0.0/network/admin/listener.ora
Log messages written to /home/oracle/7.3.3.0.0/network/log/listener.log
Listening on: (ADDRESS=(PROTOCOL=tcp)(DEV=6)(HOST=192.168.1.1)(PORT=1521))
Listening on: (ADDRESS=(PROTOCOL=ipc)(DEV=10)(KEY=700))

Connecting to
(ADDRESS=(PROTOCOL=tcp)(HOST=magic.com)(PORT=1521)(COMMUNITY=UK_SUP_TCPIP))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for SCO System V/386: Version 2.3.3.0.0 -
Production
Start Date                23-FEB-98 20:38:50
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  OFF
SNMP                      ON
Listener Parameter File   /home/oracle/7.3.3.0.0/network/admin/listener.ora
Listener Log File         /home/oracle/7.3.3.0.0/network/log/listener.log
Services Summary...
  orcl          has 1 service handler(s)
The command completed successfully
LSNRCTL> exit
```

To stop the listeners:

```
$ lsnrctl

LSNRCTL for SCO System V/386: Version 2.3.3.0.0 - Production on 23-FEB-98 20:43:20

Copyright (c) Oracle Corporation 1994.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> stop
Connecting to
(ADDRESS=(PROTOCOL=tcp)(HOST=magic.com)(PORT=1521)(COMMUNITY=UK_SUP_TCPIP))
The command completed successfully
LSNRCTL> exit
```

If you have a DNS setup which doesn't return the IP address for the hostname specified then starting and stopping the listener can take some time (2-3 mins. dependant on the DNS timeout variable). If this is the case, don't worry, be patient.

5. Client Configuration

5.1 Windows Clients

SQL*Net configuration on the PC using newer versions of the Oracle Client Software is very easy. The best (and easiest) way of achiving a fully working client installation is to use the `SQL*Net Easy Configuration` tool supplied by Oracle.

This toolhas a wizard type interface to take you through the installation of the `tnsnames.ora` and `sqlnet.ora` files.

Select "Add Database Alias" and enter a name for the alias when prompted. This alias is the name you will refer to the database instance as, and as such should be the same as the instance name (orcl in this case).

Select TCP/IP as the protocol, and when prompted the hostname of the machine hosting the database and the instance name of the database.

That's it.

However, if you do not have the `SQL*Net Easy Configuration` Tool then don't worry. You can simply create the `tnsnames.ora` and the `sqlnet.ora` files in the `$ORACLE_HOME/network/admin` directory on the client exactly as they are on the server. This will provide an alias the same as on the server (always a good idea anyway).

5.2 Unix Clients

UNIX clients are not very different that windows clients. If you have the `Network Manager` from Oracle then user that in the same way as above, if not then you can, again, just use the same configuration files as the server in the `$ORACLE_HOME/network/admin` directory.

6. Automatic Startup and Shutdown

6.1 dbstart & dbstop

The automatic startup and shutdown of the Oracle database can be achieved (in 7.3.3.0.0) with the files `dbstart` and `dbshut` both provided by Oracle. These files rely on the existance of the file `/etc/oratab` to work (although by altering the `dbshut` and `dbstart` files this can be moved).

The format of the `/etc/oratab` file is as follows:

```
SID:ORACLE_HOME:AUTO
```

An example:

```
orcl:/home/oracle/7.3.3.0.0:Y
leaveup:/home/oracle/7.3.2.1.0:N
```

6.2 init.d & rc.d

To start and stop the database when the machine comes up and goes down by modifying the startup routines for the Linux machine. This is quite easy, although I should point out here that this may change depending on which flavour of Linux (slackware, debian, redhat, etc). I will show examples which work for Redhat Linux 5.0. To modify these for your own flavour of Linux, please see your Linux documentation sets. (Although it should hold true for any Sys V type UNIX).

Firstly, we need to create the script which will run `dbshut` and `dbstart` in the `/etc/rc.d/init.d` directory. Create the following file as `/etc/rc.d/init.d/oracle`:

```
#!/bin/sh
#
# /etc/rc.d/init.d/oracle
# Description: Starts and stops the Oracle database and listeners
# See how we were called.
case "$1" in
  start)
        echo -n "Starting Oracle Databases: "
        echo "-----------------------------------------------------" >>
/var/log/oracle
        date +"! %T %a %D : Starting Oracle Databases as part of system up." >>
/var/log/oracle
        echo "-----------------------------------------------------" >>
/var/log/oracle
        su - oracle -c dbstart >> /var/log/oracle
        echo "Done."
        echo -n "Starting Oracle Listeners: "
        su - oracle -c "lsnrctl start" >> /var/log/oracle
        echo "Done."
        echo ""
```

TOC

```
        echo "-----------------------------------------------------" >>
/var/log/oracle
        date +"! %T %a %D : Finished." >> /var/log/oracle
        echo "-----------------------------------------------------" >>
/var/log/oracle
        touch /var/lock/subsys/oracle
        ;;
  stop)
        echo -n "Shutting Down Oracle Listeners: "
        echo "-----------------------------------------------------" >>
/var/log/oracle
        date +"! %T %a %D : Shutting Down Oracle Databases as part of system down."
>> /var/log/oracle
        echo "-----------------------------------------------------" >>
/var/log/oracle
        su - oracle -c "lsnrctl stop" >> /var/log/oracle
        echo "Done."
        rm -f /var/lock/subsys/oracle
        echo -n "Shutting Down Oracle Databases: "
        su - oracle -c dbshut >> /var/log/oracle
        echo "Done."
        echo ""
        echo "-----------------------------------------------------" >>
/var/log/oracle
        date +"! %T %a %D : Finished." >> /var/log/oracle
        echo "-----------------------------------------------------" >>
/var/log/oracle
        ;;
  restart)
        echo -n "Restarting Oracle Databases: "
        echo "-----------------------------------------------------" >>
/var/log/oracle
        date +"! %T %a %D : Restarting Oracle Databases as part of system up." >>
/var/log/oracle
        echo "-----------------------------------------------------" >>
/var/log/oracle
        su - oracle -c dbstop >> /var/log/oracle
        su - oracle -c dbstart >> /var/log/oracle
        echo "Done."
        echo -n "Restarting Oracle Listeners: "
        su - oracle -c "lsnrctl stop" >> /var/log/oracle
        su - oracle -c "lsnrctl start" >> /var/log/oracle
        echo "Done."
        echo ""
        echo "-----------------------------------------------------" >>
/var/log/oracle
        date +"! %T %a %D : Finished." >> /var/log/oracle
        echo "-----------------------------------------------------" >>
/var/log/oracle
        touch /var/lock/subsys/oracle
        ;;
  *)
        echo "Usage: oracle {start|stop|restart}"
        exit 1
esac
```

TOC

It is worth checking that this file actually correctly stops and starts the databases for your system. Check the log file, `/var/log/oracle` for error messages.

Once this script is working we need to create start and kill symbolic links in the appropriate runlevel directories `/etc/rc.d/rcX.d`.

The following commands will ensure that the databases will come up in runlevels 2,3 and 4:

```
$ ln -s ../init.d/oracle /etc/rc.d/rc2.d/S99oracle
$ ln -s ../init.d/oracle /etc/rc.d/rc3.d/S99oracle
$ ln -s ../init.d/oracle /etc/rc.d/rc4.d/S99oracle
```

To stop the databases on reboot or restart we need the following links:

```
$ ln -s ../init.d/oracle /etc/rc.d/rc0.d/K01oracle          # Halting
$ ln -s ../init.d/oracle /etc/rc.d/rc6.d/K01oracle          # Rebooting
```

7. Other Bits

7.1 Intelligent Agent

If you have a need for the `Oracle Intelligent Agent`, I found that you can run it without any configuration changes. To start the IA try:

```
$ lsnrctl dbsnmp_start
```

To stop the IA try:

```
$ lsnrctl dbsnmp_stop
```

There do not appear to be any messages indicating a sucessful or otherwise start or stop of the intelligent agent. However, the IA responded to Enterprise manager on the client side so I can only assume that it is working.

8. Troubleshooting

See below for various troubleshooting hints.

8.1 I cannot create a database when using Oracle 7.2.x.

The files shipped by Oracle in the 7.2.x product are incorrect in assuming that you want to setup a parallel server configuration. The shipped `init.ora` file has the following line in it:

```
# define parallel server (multi-instance) parameters
ifile = ora_system:initps.ora
```

To fix the problem simply comment it out:

```
# define parallel server (multi-instance) parameters
#ifile = ora_system:initps.ora
```

8.2 I'm getting segmentation faults in `svrmgrl` under version 7.3.4.x.

I've had this problem reported to me by a number of people. Gerald Weber `gerald_weber@master.co.at` solved it:

```
Hi Paul,

first of all thanks for your help,but none of the possible problems you are
thinking about were responsible for my problem.
The problem is the iBCS-emulator itself.
It seems that Oracle performs an sysconf-calls which isn't supported in the
current version of iBCS.
Look at the trace :

<7>[22]615 sysconf(34)
<7>iBCS2 unsupported sysconf call 34
<7>[22]615 sysconf error return linux=-22 -> ibcs=22 <Invalid argument>
<7>[24]615 sysconf(34)
<7>iBCS2 unsupported sysconf call 34
<7>[24]615 sysconf error return linux=-22 -> ibcs=22 <Invalid argument>

Solution:  patching the iBCS-source.apply the following diff-pach :
```

TOC

```
--- sysconf.c   Sun Apr 19 19:19:15 1998
+++ sysconf.c.ori       Sun Apr 19 19:28:45 1998
@@ -60,7 +60,6 @@
 #define _SC_JOB_CONTROL        5
 #define _SC_SAVED_IDS  6
 #define _SC_VERSION    7
-#define _SC_HACK_FOR_ORACLE 34

 #define _SC_PAGESIZE   11

@@ -97,11 +96,6 @@
                case _SC_SAVED_IDS: {
                        return (1);
                }
-
-               case _SC_HACK_FOR_ORACLE: {
-                 return (1);
-               }
-

                case _SC_PAGESIZE: {
                  return PAGE_SIZE;
```

9. Credits

This document was based on a document written by Bob Withers, `bwit@pobox.com`. Additional information taken from documents written by Georg Rehfeld, `rehfeld@wmd.de` and David Mansfield, `david@claremont.com`.

Additional proof reading done by Bob Withers, Mark Watling, `mwatling@mjw-ltd.demon.co.uk`, Peter Sodhi, `petersodhi@unn.unisys.com` and Greg Hankins, `greg.hankins@cc.gatech.edu`.

My thanks go to the tremendous support from all the people involved in this document and the research that has gone into it. Particular thanks to Bob Withers and Mark Watling for the additional comments and help they have provided.

Linux PCI-HOWTO

by Michael Will, Michael.Will@student.uni-tuebingen.de

v0.6g, 30 March 1997

Information on what works with Linux and PCI-boards and what does not. Please get the latest version of this document at The Linux Documentation Project

1. Introduction

2. Why PCI?

- 2.1 General overview
- 2.2 Performance
- 2.3 The onboard-SCSI-II-chip NCR53c810
- 2.4 Drew Eckhardt on PCI-SCSI:
- 2.5 New Alpha Version of the NCR driver
- 2.6 The EATA-DMA driver and the PCI SCSI controllers from DPT
- 2.7 BT-946C fully supported with kernel 1.3.x and newer
- 2.8 Future Domain TMC-3260 PCI SCSI
- 2.9 other thoughts on scsi

3. ASUS-Boards

- 3.1 ASUS and the NMI (Parity) -- impact on Gravis-Ultrasound
- 3.2 Various types of ASUS Boards
- 3.3 Benchmarks on ASUS Mainboards
- 3.4 Detailed information on the old ASUS PCI-I-SP3 with saturn chipset from heinrich@zsv.gmd.de:
- 3.5 Pat Dowler (dowler@pt1B1106.FSH.UVic.CA) with ASUS SP3G

TOC

4. confusion about saturn chipsets

5. Video-Cards

6. Ethernet Cards

- 6.1 3com-3c590-tpo
- 6.2 DEC435 PCI NIC

7. Motherboards

- 7.1 ASUS
- 7.2 Micronics P54i-90
- 7.3 SA486P AIO-II
- 7.4 Sirius SPACE
- 7.5 Gateway-2000
- 7.6 Intel-Premiere
- 7.7 DELL Poweredge SP4100 gbelow@pmail.sams.ch - successful
- 7.8 DELL OptiPlex Gl+ 575
- 7.9 Comtrade Best Buy PCI / PCI48X MB Rev 1.0
- 7.10 IDeal PCI / PCI48X MB Rev 1.0
- 7.11 CMD Tech. PCI IDE / CSA-6400C
- 7.12 GA-486iS (Gigabyte)
- 7.13 GA-586-ID (Gigabyte) 90 Mhz Pentium PCI/EISA Board
- 7.14 ESCOM 486dx2/66 - which board?
- 7.15 J-Bond with i486dx2/66
- 7.16 super micro 011895 03:50 SUPER P54CI-PCI rev 1.3 (Opti)

8. reports on success

- 8.1 GigaByte GA486-AM with AMD Am5x86-133-WB @ 160MHz (40MHz PCI)
- 8.2 California Graphics - Sunray II Pro
- 8.3 Micronics P54i-90 (root@intellibase.gte.com)

- 8.4 Angelo Haritsis (ah@doc.ic.ac.uk) about SA486P AIO-II:
- 8.5 bill.foster@mccaw.com about his Micronics M5Pi
- 8.6 Simon Karpen (karpens@ncssm-server.ncssm.edu) with Micronics M54pi
- 8.7 Goerg von Below (gbelow@pmail.sams.ch) about DELL Poweredge
- 8.8 zenon@resonex.com about Gateway2000 P-66
- 8.9 James D. Levine (jdl@netcom.com) with Gateway2000
- 8.10 hi86@rz.uni-karlsruhe.de with SPACE
- 8.11 grif@cs.ucr.edu with INTEL
- 8.12 Jermoe Meyers (jeromem@amiserv.xnet.com) with Intel Premiere
- 8.13 Timothy Demarest (demarest@rerf.or.jp) Intel Plato Premiere II
- 8.14 heinrich@zsv.gmd.de with ASUS
- 8.15 CARSTEN@AWORLD.aworld.de with ASUS
- 8.16 Lars Heinemann (lars@uni-paderborn.de) with ASUS
- 8.17 Ruediger.Funck@Physik.TU-Muenchen.DE with ASUS
- 8.18 robert logan (rl@de-montfort.ac.uk with GW/2000)
- 8.19 archie@CS.Berkeley.EDU and his friend use ASUS
- 8.20 Michael Will with ASUS-SP3 486 (the old one)
- 8.21 Mike Frisch (mfrisch@saturn.tlug.org) Giga-Byte 486IM
- 8.22 Karl Keyte (kkeyte@esoc.bitnet) Gigabyte GA586 Pentium
- 8.23 kenf@clark.net with G/W 2000
- 8.24 Joerg Wedeck (jw@peanuts.informatik.uni-tuebingen.de) / ESCOM
- 8.25 Ulrich Teichert / ASUS

9. Reports of problems

- 9.1 Compaq PCI systems, especially Presarios
- 9.2 VLSI Wildcat PCI chipset like in Zeos P120 box
- 9.3 dmarples@comms.eee.strathclyde.ac.uk G/W 2000
- 9.4 cip574@wpax01.physik.uni-wuerzburg.de (Frank Hofmann) / ASUS
- 9.5 axel@avalanche.cs.tu-berlin.de (Axel Mahler) / ASUS
- 9.6 Frank Strauss (strauss@dagoba.escape.de) / ASUS
- 9.7 egooch@mc.com / ASUS
- 9.8 Stefan.Dalibor@informatik.uni-erlangen.de / GigaByte
- 9.9 Steve Durst (sdurst@burns.rl.af.mil) with UMC 8500 mainboard

- 9.10 Tom Drabenstott (tldraben@Teleport.Com) with Comtrade / PCI48IX

10. General tips for PCI-Motherboard + Linux NCR PCI SCSI

- 10.1 DON'Ts:
- 10.2 SIMM slots
- 10.3 Praised PCI Pentium motherboard
- 10.4 irq-lines
- 10.5 Info about the different NCR 8xx family scsi chips:
- 10.6 future of 53c8xx
- 10.7 Performance of the 53c810
- 10.8 News about NCR53c825 support
- 10.9 Frederic POTTER (Frederic.Potter@masi.ibp.fr) about Pentium+NCR+Strap_bug
- 10.10 PCIprobe in the latest Linux Kernels by Frederic Potter
- 10.11 Other PCI Devices

11. Conclusion

12. Thanks

13. copyright/legalese

14. GPL - Gnu Public License

---

1. Introduction

Many people, including me, would like to run Linux on a PCI-based machine. Since it is not obvious which PCI motherboards and PCI cards will work with Linux and which do not, I conducted a survey and spent some hours to compile the information contained herein.

If you have information to add, please mail me. If you have questions, feel free to ask.

Help with my style/grammar/language is welcome as well. I am not a native- speaker of English and expect to make occasional mistakes.

Note: "on-board chip" refers to a SCSI chip integrated onto the motherboard rather than on a PCI expansion card.

Also, "quotes" herein may have slight context editing.

2. Why PCI?

2.1 General overview

The PC-architecture has several BUS-Systems to choose from:

ISA

16 or 8bit, cheap, slow (usually 8Mhz), standard, many cards available>

EISA

32bit, expensive, fast, few cards available, fading>

MCA

32 or 16bit ex-IBM-proprietary, fast, becoming rare>

VESA-Local-Bus

32bit, based on 486 architecture, cheap, fast, many cards available>

PCI-Local-Bus

32bit (64 bit coming), cheap, fast, many cards available, nowadays standard>

MCA worked fine, but never achieved much market, being used on only some early IBM PS/2 machines. There were very few cards.

EISA was reliable, but rather expensive, and intended more for servers, than for the average user. It has the next fewest cards available.

VESA-Local-Bus (VLB) had some problems with high bus-speeds, and was not very reliable, but mainly due to its low price and better-than-ISA performance, sold very well. Technically, it's almost a direct map of the 486 processor bus. Most VESA boards should be stable by now. At the beginning of 1996, many 486 motherboards still support VESA, but PCI is growing. VESA busses are tied directly to the speed of the memory bus for 486's, or half the speed for Pentiums.

PCI now has the advantage. Like EISA it is not proprietary. It is as faster than EISA or MCA, and cheaper. Most current Pentium motherboards use the PCI bus; VESA is fading. Virtualy all PCI motherboards and cards sold at the beginning of 1996 are 32 bit, and run at 0-33 MHz.

Currently, most Pentium motherboards run the PCI bus at 1/2 the memory speed (ie: 33 MHz for the 66 MHz memory bus on the P66,P100,P133,P166; 30 MHz for the 60 MHz memory bus on the P60,P90,P120,P150; and 25 Mhz on the 50 MHz memory bus of the P75). This is probably true of Cyrix 6x86 motherboards too. NexGen 5x86 implemention isn't known. The PCI spec does allow the PCI bus to be run asynchronously from the processor, (eg: 33 Mhz bus on P75), but this is not common yet.

PCI 2.1 has been defined, allowing 64 bit PCI, and/or 0-66 MHz operations, but no x86 chipsets yet support these options. 64 bit PCI will probably appear first, in 32/64 bit dual compatible versions. That is, you will be able to mix 32 and 64 bit cards. 66 MHz PCI will take longer, as it's technically demanding, can only support one or maybe two slots per bridge, and may not work well with 33 MHz cards.

PCI is not processor dependent like the VESA Local-Bus. This means you can use the winner-1000-PCI in an Alpha-driven-PCI computer as well as in a i486/Pentium-driven PCI computer, with the appropriate BIOS and

software. Beside Intel and DEC Alpha platforms, PCI is used on some PowerPC's.

Some PCI variations to be aware of: some implementations support "Bus Master" cards in all PCI slots, some in only one slot, and some not at all; some implementations support "bridging" on cards and some do not.

2.2 Performance

taken from Craig Sutphin's early Pro-PCI-Propaganda

> Unlike some local buses, which are aimed at speeding up graphics alone, the PCI Local Bus is a total system solution, providing increased performance for networks, disk drives, full-motion video, graphics and the full range of high-speed peripherals. At 33 MHz, the synchronous PCI Local Bus transfers 32 bits of data at up to 132 Mbytes/sec. A transparent 64-bit extension of the 32-bit data and address buses can double the bus bandwidth (264 Mbytes/sec) and offer forward and backwards compatibility for 32 and 64-bit PCI Local Bus peripherals. Because it is processor-independent, the PCI Local Bus is optimized for I/O functions, enabling the local bus to operate concurrent with the processor/memory subsystem. For users of high-end desktop PC's, PCI makes high reliability, high performance and ease of use more affordable than ever before; no trivial task at 33 MHz bus-clock rates. Variable length linear or toggle mode bursting for both reads and writes improves write dependent graphics performance. By comprehending the loading and frequency requirements of the local bus at the component level, buffers and glue logic are eliminated.

See the chapter about Benchmarks for some crude (and perhaps meaningless) benchmarks on ASUS PCI Boards with 486 and 586.

2.3 The onboard-SCSI-II-chip NCR53c810

One very nice feature of some PCI mother boards is the NCR onboard-SCSI-II-chip, which is said to be as fast as the EISA-Adaptec-1742, but much cheaper. Drivers for DOS/OS2 are available. Drew Eckard has released his version of his NCR53c810-driver, which is in the standard kernel since v1.2.

This works so well I sold my adaptec-1542B-ISA soon after I bought the ASUS SP3-saturn-chipset II PCI board, and found the onboard NCR-SCSI controller to be much faster.

The NCR53c810-chip is onboard on some PCI-motherboards. There are add-on-boards available too, for about US$ 70.00.

There is only one thing I noticed did not work with the NCR-drivers when I tried them. Disconnect/Reconnect did not work, so using a SCSI-tape could be a pain, especially when using "mt erase" or the like blocks the whole SCSI-bus until it has finished. Since this was very unsatisfying for me, I bought one of these nice but expensive DPT PCI SCSI controller and had no such problems anymore.

People have reported this problem has been solved by Drew by now.

FreeBSD does support the NCR53c810 for quite a long time already, including Tagged Command Queues, FAST, WIDE and Disconnect for NCR 53c810, 815, 825. Drew said, it would be possible to adapt the FreeBSD driver to Linux. I somewhere saw some patches to do exactly this, any pointer to the location?

I personaly have the impression there are some important wheels invented more than once because of the differently evolving of FreeBSD and Linux. Some more cooperation could do both systems very well...

TOC

2.4 Drew Eckhardt on PCI-SCSI:

Drew said on end of March 95 about the SCSI on PCI: (slightly edited for clarity in context)

The Adaptec 2940, Buslogic BT946, BT946W, DPT PCI boards, Future Domain 3260, NCR53c810, NCR53c815, NCR53c820, and NCR53c825 all work for some definition of the word works.

- The Adaptec 2940 suffers from the same cabling sensitivity that plagues all recent boards, but otherwise works fine.
- The Future Domain boards are not busmasters, and the driver doesn't support multiple simultaenous commands. If you don't (currently) need multiple simultaneous commands, get a NCR board, which will be cheaper and is busmastering. If you need multiple simultaneous commands, get a Buslogic.
- The Buslogic BT956W will do WIDE SCSI with the Linux drivers (although you can't use targets 8-15), the Adaptec 2940W (with one line patch to the 2940 driver) won't, nor will the NCR53c820 and NCR53c825.
- The NCR boards are dirt cheap (< $ 70 US), are generally quite fast, but the driver currently doesn't support multiple simultaenous commands. Alpha which do neat things like disconnect/reconnect and synchronous transfer are now publicly available, see below.
- Emulux, Forex, and other unmentioned PCI SCSI controllers will not work.

2.5 New Alpha Version of the NCR driver

Well, this is not exactly *that* new anymore, please try to he versions which are in the kernel by version 2.0.x before going for this entry.

Alpha versions of the NCR driver which do neat things like disconnect/reconnect and synchronous transfers are now publically available. Any one interested in playing with them should

- Join the NCR mailing list, by sending mail to majordomo@colorado.edu with subscribe ncr53c810 in the text.
- Get all of the readmes, and latest diffs file from ftp://tsx-11.mit.edu/pub/ALPHA/linux/SCSI/ncr53c810

2.6 The EATA-DMA driver and the PCI SCSI controllers from DPT

The EATA-DMA scsi driver has undergone extensive changes and now also supports PCI SCSI controllers, multiple controllers and all SCSI channels on the multichannel SmartCache/Raid boards in all combinations of WIDE, FAST-20 (ULTRA) and DIFFERENTIAL.

The driver supports all EATA-DMA Protocol (CAM document CAM/89-004 rev. 2.0c) compliant SCSI controllers and has been tested with many of those controllers in mixed combinations.

```
Those are:              (ISA)   (EISA) (PCI)
      DPT Smartcache: PM2011  PM2012B
      Smartcache III: PM2021  PM2022  PM2024
                              PM2122  PM2124
                              PM2322
      Smartcache IV:  PM2041  PM2042  PM2044
                              PM2142  PM2144
                              PM2322
      SmartRAID     : PM3021  PM3122
                              PM3222  PM3224
```

```
                    PM3334
and some controllers from NEC, AT&T, SNI, AST, Olivetti and Alphatron
```

On a "base" DPT card (no caching or RAID module), a MC680x0 controls the bus-mastering DMA chip(s) and the SCSI controller chip. The DPT SCSI card almost works like a SCSI coprocessor.

The DPT card also will emulate an IDE controller/drive (ST506 interface), which enables you to use it with all operating systems even if they don't have an EATA driver.

On a card with the caching module, the 680x0 maintains and manages the on-board cacheing. The DPT card supports up to 64 MB RAM for disk-cacheing.

On a card with the RAID module, the 680x0 also performs the management of the RAID, doing the mirroring on RAID-1, doing the striping and ECC info generation on RAID-5, etc.

The entry level boards utilize a Motorola 68000, the high-end, more raid specific DPT cards use a 68020, 68030 or 68040/40MHz processor.

Official list prices range from $ 265 to $1.645 (January 18, 1996)

Since I've been asked numerous times where you can buy those boards in Europe, I asked DPT to send me a list of their official European distributors. Here is a small excerpt:

```
Austria: Macrotron GmbH            Tel:+43 1 408 15430   Fax:+43 1 408 1545
Denmark: Tallgrass Technologies A/S Tel:+45 86 14 7000   Fax:+45 86 14 7333
Finland: Computer 2000 Finnland OY Tel:+35 80 887 331    Fax:+35 80 887 333 43
France : Chip Technologies         Tel:+33 1 49 60 1011  Fax:+33 1 49 599350
Germany: Akro Datensysteme GmbH    Tel:+49 (0)89 3178701 Fax:+49 (0)89 31787299
Russia : Soft-tronik               Tel:+7 812 315 92 76  Fax:+7 812 311 01 08
U.K.   : Ambar Systems Ltd.        Tel:+44 1296 311 300  Fax:+44 296 479 461
```

"IMHO, the DPT cards are the best-designed SCSI cards available for a PC. And I've written code for just about every type of SCSI card for the PC. (Although, in retrospect, I don't know why!) ;-)" Jon R. Taylor (jtaylor@magicnet.net) President, Visionix, Inc.

The latest version of the EATA-DMA driver and a Slackware bootdisk is available on: ftp.i-Connect.Net:/pub/Local/EATA

Since patchlevel 1.1.81 the driver is included in the standard kernel distribution.

The author can be reached under these addresses: neuffer@mail.uni-mainz.de or mike@i-Connect.Net

2.7 BT-946C fully supported with kernel 1.3.x and newer

There is a driver in the 1.3.x kernels (available as a patch for the 1.2.13 kernel) written by someone associated with buslogic that fully supports the 946C and ALL of it's features including strict round robin, tagged queueing, multiple scatter/gather, multiple mailboxes, IRQ sharing, and yes, 15 devices on Fast/Wide. It is no longer necessary to use any ISA emulation with the driver (no DMA channel, no ISA address), and the driver is /fast/ and /stable/ (it's out of BETA and into full release).

The driver is available on ftp.dandelion.com (the newest version can always be got by doing "get BusLogic*"). It supports ALL BusLogic controllers with the exception of the FlashPoint LT, which uses a different interface. The driver is included in the 1.3.x kernels as standard for BusLogic devices.

2.8 Future Domain TMC-3260 PCI SCSI

TOC

Rik Faith (faith@cs.unc.edu) informed me on Wed, 1 Feb 1995 about the Future Domain TMC-3260 PCI SCSI card being supported by the Future Domain 16x0 SCSI driver. Newer information might be contained in the SCSI-HOWTO.

- Detection is not done well, and does not use standard PCI BIOS detection methods (someone who has a PCI board needs to send me patches to fix this problem). So, you might have to fiddle with the detection routine in the kernel to get it detected.
- The driver still does not support multiple outstanding commands, so your system will hang while your tape rewinds.
- The driver does not support the enhanced pseudo-32bit transfer mode supported by recent Future Domain chips, so you will not get transfer rates as high as under DOS.
- The driver only supports the SCSI-I protocol, so your really fast hard disks will not get used at the highest possible throughput. (Again, fixes for all these problems are solicited -- no one is working on them at this time.)

2.9 other thoughts on scsi

James Soutter (J.K.Soutter1@lut.ac.uk) asked me to add the following information on Fast-Wide-SCSI-2:

> Fast Wide SCSI-2 is sometimes incorrectly called SCSI-3. It differs from the normal Fast SCSI-2 (like the Adapted 1542B?) because it uses a 16 bit data bus rather than the more usual 8 bit bus. This improves the maximum transfer rate from 10 MB/s to 20 MB/s but requires the use of special Fast Wide SCSI-2 drives.
>
> The added performance of Fast Wide SCSI-2 will not necessarily improve the speed of your system. Most hard disk drives have a maximum internal transfer rate of less than 10 MB/s and so one drive alone can not flood a FAST SCSI-2 bus.
>
> In Seagate's Oct 1993 product overview, only one Fast Wide SCSI-2 drive has an internal transfer rate of more than 10 MB/s (the ST12450W). Most of the drives have a maximum internal transfer rate of 6 MB/s or less, although the ST12450W is not the only exception to the rule. In conclusion, Fast Wide SCSI is designed for the file server market and will not necessarily benefit a single user workstation style system.
>
> Rather than buying a PCI system with a SCSI interface on the motherboard, or rather than waiting for the NCR driver, you could purchase a separate PCI based SCSI card. According to Drew, the only PCI SCSI option that stands a chance of working is the Buslogic 946. It purports to be Adaptec 1540 compatible, like the EISA/VESA/ISA boards in the series.
>
> Drew commented that other PCI based SCSI controllers are unlikely to be supported under Linux or the BSD's because the NCR based controllers are cheaper and more prevalent.

I definitly recommend reading the SCSI HOWTO in regards to newer information about PCI SCSI drivers.

Ernst Kloecker (ernst@cs.tu-berlin.de) wrote: (edited)

> Talus Corporation has finished a NS/FIP driver for PCI boards with NCR SCSI. It will be shipping very soon, might even be free because a third party might pay for the work and donate the driver to NeXT.

Not every PCI-Board has got the chip. The old ASUS do, and one of the J-Bond boards does, too. (Most of the boards nowadays (6/95) do expect you to buy the NCR53c810 seperately.) Some vendors provide an alternative as you can read in Drew's text...

The NCR-Chip is clever enough to work with drives formatted by other controllers, and should be no problem.

3. ASUS-Boards

3.1 ASUS and the NMI (Parity) -- impact on Gravis-Ultrasound

The newer trition PCI-Mainboards in 1995 did not seem to support parity-SIMMS anymore. Since I usualy took the cheaper nonparity-SIMMS anyway, I did not consider this a problem until I put the Gravis-Ultrasound into my machine. Under DOS the SBOS-Driver and Setup/Test utility does complain about "nmi procedure disabled on this p.c.". The manual says I'd better get a better mainboard in that case, not very helpful.

The gravis-ultrasound did work nice in the ASUS-SP3 and ASUS-SP4, inspite of this, but the gravis-ultrasound-max I have here got gmod to kernel panic on both boards, and sometimes when playing au-files via /dev/audio did strange things, like playing the rest of an older, previously played sound after the new one. The sounddriver does recommend a buffer of 65536 with the GUS Max instead of the small one like the GUS - why I do not know. I do not have such a problem with the newer ASUS TP4 XE boards, though. Both are equipped with 1M DRAM onboard. These problems are probably not related to the NMI-problem, but because of the sounddriver?

I heard not only ASUS but most of the newer PCI-Mainboards are lacking in parity/NMI-support.

Strange enough - the ASUS-TP4 (Trition Chipset) does work with the GUS Max - it does load the SBOS-Driver. I have to admit, I am confused.

3.2 Various types of ASUS Boards

ASUS SP3 with saturn chipset I (rev. 2) for 486,

- 2 x rs232 with 16550
- NCR53c810 onboard,
- slightly broken saturn-chipset I (rev. 2)

ASUS SP3G with saturn chipset II (rev. 4) for 486,

like SP3, but less buggy saturn chipset

TOC

ASUS SP3-SiS chipset, for 486

like AP4, but newer, SiS chipset, green functions and all the EIDE, rs232 with 2 16550 and centronics. Only 2 SIMM Slots, Does seem to work with AMD486DX4/120, but was not very reliably on NCR53c810 and various operating systems (Windows-NT, Windows95, OS2), after upgrading to a PentiumBoard ASUS SP4, all the problems vanished, so it must have been the board. Still does seem to work nice for Linux, though.

ASUS AP4, for 486, with PCI/ISA/VesaLocalbus

green functions, 1VL, 3 ISA, 4 PCI slots, only EIDE onboard, no fd-controller, no rs232/centronics. Very small size.

does recognice AMD486DX2/66 as DX4/100 only. This can be corrected with soldering one pin (which?) to ground, but I would not recommend a board like this anyway.

The one I tested was broken for OS2 and Linux, but people are said to use it for both.

The VesaLocalbus-Slot is expected to be slower than the normal vesa-localbus boards because of the PCI2VL bridge, but without penalty to the PCI section.

ASUS SP4-SiS, for Pentium90, PCI/ISA

like SP3-SiS, but for Pentium90/100.

ASUS TP4 with Triton chipset and EDO-Support

has the Triton-Chipset for better performance and supports normal PS2-Simms as well as Fast-Page-Mode and EDO modules.

ASUS TP4XE with Triton chipset and additional SRAM/EDORAM support

supports the new EDORAM and upcoming SRAM standards. At least SRAM is said to considerabely increase performance. Did for some reason not accept the 8M PS2-SIMMS working ok in ASUS SP4, after changing them against others, bigger looking ones, (16 chips instead of 8 if I remember right) it worked ok. Has been tested with P90 and P100.

...and many others now.

if you have new information on problems with them, please report.

3.3 Benchmarks on ASUS Mainboards

TOC

I tried to compare the speed of CPUs in two ASUS Mainboards: for 486 I tested the SP3 SiS (the one with one vesa-local-bus slot) and for 586 I tested the ASUS TP4/XE, each with 16M RAM, always the same unloaded system with another CPU, with whetstone and dhrystone.

I must admit, I have not read the benchmarks-faq yet, and will probably edit the section a loot soon. If you have any comments, please mail me.

I am especially confused about the amd486DX4/100 being faster on dhrystones than the DX4/120 version? I did not see that kind of inconsistency on comparing the P90 and P100.

Perhaps this was at fault: when I plugged in the amdDX4-100, I had the board jumpered for DX2-66. While the BIOS did report it as an DX4-100, the board might have used the wrong clockspeeds... but since DX2-66 uses 33Mhz * 2 and DX4 uses 33Mhz * 3, this would have been correct?

The board running with DX4-120 is jumpered to 40Mhz * 3 = 120 Mhz.

Another thing I wonder about is why the whetstones-result does yield so even numbers on some machines?

ASUS SP3 with amd486DX4-100

- Dhrystone time for 500000 passes = 7 by 63559 dhrystones/second
- Whetstone time for 1000 passes = 5 by 200.0000 Whetstones/second

ASUS SP3 with amd486DX4-120

- Dhrystone time for 500000 passes = 8 by 56074 dhrystones/second
- Whetstone time for 1000 passes = 4 by 250.0000 Whetstones/second

ASUS SP3 with intel486DX2-66

- Dhrystone time for 500000 passes = 9 by 50761 dhrystones/second
- Whetstone time for 1000 passes = 7 by 142.8571 Whetstones/second

ASUS TP4/XE with intel586-90

- Dhrystone time for 500000 passes = 4 by 101010 dhrystones/second
- Whetstone time for 1000 passes = 3 by 333.3333 Whetstones/second

ASUS TP4/XE with intel586-100

- Dhrystone time for 500000 passes = 4 by 102040 dhrystones/second
- Whetstone time for 1000 passes = 2 by 500.0000 Whetstones/second

3.4 Detailed information on the old ASUS PCI-I-SP with saturn chipset from heinrich@zsv.gmd.de:

TOC

- 3 PCI, 4 ISA Slots (3x16, 1x8 Bit)
- ZIF Socket for the CPU
- room for 4 72pin-SIMMs (max. 128M)
- Award BIOS in Flash-Eprom
- Onboard: NCR-SCSI, 1par, 2ser (with FIFO), AT-Bus, Floppy

The board does like most in that price class -- write-through cache, no write-back. This should not be significant, maybe 3% of performance.

The BIOS supports scsi-drives under DOS/Windows without additional drivers, but with the board come additional drivers which are said to give better performance, for DOS/Windows(ASPI), OS2, Windows-NT, SCO-Unix, Netware (3.11 and 4, if interpreted correctly)

Gert Doering (gert@greenie.muc.de) was saying the SCO-Unix-driver for the onboard-SCSI-Chip was not working properly. After two or three times doing: "time dd if=/dev/rhd20 of=/dev/null bs=100k count=500" it kernel-paniced...

The trouble some people experienced with this board might be due to them using an outboard Adaptec-SCSI-Controller with "sync negotiation" turned on. (This predates the NCR driver release; hence the use of the Adaptec.) Please check that in the BIOS-Setup of the Adaptec-1542C if you use one and have problems with occasional hangups!

There is a new version of the ASUS-Board which should have definitely less problems. It is called ASUS-PCI-I/SP3G, the G is important. It has the new Saturn-chipset rev. 4 and the bugs should be gone. They use the Saturn-ZX-variant and the new SP3G has fully PCI conforming level-triggered (thus shareable), BIOS-configurable interrupts. It has an on-board PS/2-mouseport, EPA-power-saving-modes and DX4-support, too. It performs excellently. If you can get the German computer magazine C't from July (?), you will find a test report where the ASUS-Board is the best around.

Latest information about ASUS-SP3-G: You might experience crashes when using PCI-to-Memory-Posting. If you disable this, all works perfect. jw@peanuts.informatik.uni-tuebingen.de said he believed it to be a problem of the current Linux-kernel rather than the hardware, because part of the system still works when crashing, looking like a deadlock in the swapper, and OS2/DOS/WINDOZE don't crash at all.

Someone else with a very old ASUS-SP3 (saturn-I chipset) reported crashes with using XFree86, which went away when he installed the very latest betaversion which seems to work around a bit of the problems.

3.5 Pat Dowler (dowler@pt1B1106.FSH.UVic.CA) with ASUS SP3G

TOC

- ASUS SP3G board (it is rev.4 == saturn II)
- AMD DX4-100 CPU (need to set jumper 36 to 1&2 rather than 2&3, otherwise it's set the same as other 486DXn chips)
- 256K cache (comes with 15ns cache :-)
- 16meg RAM (2x8meg)
- ET4000 ISA video card
- quantum IDE hard drive
- SMC Elitel16 combo ethernet card

Unlike some other reports, I find the mouse pointer moves very smoothy under X (just like the ol' 386) - it is jumpy under some, but not all, DOS games though...

Performance is great!! I ran some large floating point tests and found the performance in 3x33 (100MHz) mode to be almost 1.5x that in 2x (66MHz) mode (large being 500x500 doubles - 4meg or so)... I was a little dubious about clock-tripling but I seem to be getting full benefit :-)

The heavily configurable energy star stuff doesn't work with the current AMD DX4 chips - you need an SL chip

I really need a SCSI disk and a PCI video card :-)

(I had a phonecall by a person who had this problem with the buggy SMC FIFO chipset, after using X-window they hung.)

---

4. confusion about saturn chipsets

Pat Duffy (duffy@theory.chem.ubc.ca) said:

```
Saturn I:  these are revisions 1 and 2 of the Saturn chipsets.
Saturn II:  This is also called rev. 4 of the Saturn chipsets.

As far as I know, rev. 3 never actually shipped, and (from a few people who
have it) the SP3G now has rev. 4 (or Saturn II) in it.

Confused?  Well, the only real definitive answer is to get ahold of the board
and run the debug script in the PCI chipset list on it.  As far as I know,
though, the SP3G board is indeed shipping with rev. 4 (Saturn II).
```

5. Video-Cards

Linux people have successfully used # 9 XGE Level 12, ELSA Winner 1000, and S3-928 video cards. The XFree86(tm)-3.1.1 does support boards with the tseng et4000/w32 in accelerated mode, as well as S3 Vision 864 and 964 chipsets including boards like the ELSA Winner 1000Pro and 2000Pro, Number Nine GXE64 and GXE64Pro, Miro Crystal 20SV). Support in the S3 Server for the Chrontel8391 clock chip has been added.

Trio32 and Trio64 S3 Boards like the SPEA V7 Mirage P64 PCI and MIRO Crystal 40SV, are also supported, the Mach32 and Mach64 are supported in accelerated mode, too.

The SVGA Driver

16bpp mode (65K colors instead of the usual 256) support for Mach32 boards as well as 32bpp for some S3 boards and the P9000 boards has been added.

tldraben@teleport.com reported:

- Diamond Stealth W32 (et4000/W32) -- Text mode works, X11 suffered from "pixel dust", unbearable never got it to work and returned it.
- # 9GXE L12 -- Works, virtual consoles corrupted when switched, fixed this with disabling the "fast dram mode" feature in his BIOS. Does not get a dot clock above 85, though.

Genoa Phantom 8900PCI card seems to work well. Genoa Phantom/W32 2MB does not work in an ASUS-Board. Tseng 3000/W32i chipset seems to work well. Spea-v7 mercury-lite works perfectly since XFree86(tm)-2.1.

Spea V7 Mirage P64 PCI 2M with Trio64 works nice since XFree86(tm)-3.1.1

ATI Graphics Ultra Pro for PCI with 2MB VRAM and an ATI68875C DAC run well as dem@skyline.dayton.oh.us tells us: "It's humming right along at 1280x1024 w/256 colors @74Hz non-interlaced. Looks great."

Paradise WD90C33 PCI did lock up on screensaver/X - this has been solved in the newer versions of the kernel. jbauer@badlands.NoDak.edu (John Edward Bauer)

miroChrystal 8S/PCI (1MB) S3 - no problem.

Stephen Tweedie reported his Cirrus Logics 5434 PCI card works well. It is a 64bit with 2M and runs perfectly with the SVGA driver in 8, 16 and 32 bit per pixel.

6. Ethernet Cards

Of course the ISA-ethernet-cards still work, but people are asking for PCI-based ones. The author of many (if not most) ethernet- drivers said the following some time ago (unfortunately I have not managed to contact him about new information):

> From: Donald Becker (becker@cesdis.gsfc.nasa.gov) Subject: PCI ethernet cards supported?
>
> The LANCE code has been extended to handle the PCI version. I hope to get the PCI probe code (about a dozen extra lines in the LANCE driver) into the next kernel version. I'm working on the 32 bit mode code. I haven't yet started the 21040 code.
>
> I'll write drivers for the PCnet32 mode and the DEC 21040. That will cover most of the PCI ethercard market.
>
> file://cesdis.gsfc.nasa.gov/pub/people/becker/whoiam.html

In the new testkernels of 1.1.50 and above, the AMD-singlechip ethernetadapters are supported. With a pentium, they ought to then see 900K/second ftps +(assuming an NCR PCI scsi controller) at about 20% cpu load. (AMD Lance).

Anything based on the AMD PCnet/PCI chip should work at the time being. In the US the Boca board costs under US$ 70

Geoffry Coram reported in the news that he got his 3com 590 TPO to work. He had to get the alpha driver from http://cesdis.gsfc.nasa.gov/linux/drivers. Other drivers would be there as well. Note http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html

Donald Holmgren said he successfully attached his DEC DE435 (PCI) card to the local network on thin coax (BNC). The DE435 driver checks the twisted pair connection first, then switches to the alternate port (jumper selectable as AUI or BNC) if the 10BaseT port fails.

Jim Cusick uses the Boca BEN 1PI card on a thin coax network. It works just fine. You might want to check out: http://cesdis.gsfc.nasa.gov/linux/misc/boca-failure.html for details on the early failures of this card. My second card, after sending one back for replacement, was marked "PN 4186". The old one that did not work was "PN 4185". Mandate this newer model when you order from you vendor. At $ 70, the card is a good deal.

Dave Platt recommends to stay off the Boca BEN1PI card at all costs. It would be unreliable due to design flaws, and Boca seems unable to really fix the problem. The 3Com 3c590 "Vortex" PCI card is available in a combo version (10BaseT, thin coax, and AUI). The Linux driver for this card is not yet part of the release kernel, but is available from http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html and can be patched into the later 1.2.x kernels (as well as 1.3.x) without much difficulty. The Linux driver does not support the interface autodetect feature of this card - you must use the DOS utility to configure the card for the interface you wish to use (thin coax in this case). Once you've done that, the Linux driver

TOC

will use the correct interface.

He has been using a 3c590 for several weeks, and it is working fine.

Dave Kennedy said he got two of the above Boca boards and they work fine under light load, but under heavy work like ftping two 16M files into both directions, they failed. He sent the boards back to Boca for a hardwarefix. After they soldered a couple of things (diodes/resistors) onto the card and sent them back, the cards worked fine regardless of load. The two cards have been in 7/24 use in two P90 systems without problems for 6 months now.

Craig does not recommend it since Boca seems not to follow the AMD specs but he has been running them for 2 weeks without problems. He tested his NFS performance and has been moving large files to and from server (16M, 8M). He also tried to do all his workin localy using his data files mounted by NFS and has had no problems. Performance seems to be 100 percent better (wrt to NFS performance) over his NE2000 ISA board. (editors note: but so would probably have been the ISA SMC Elite Ultra?)

6.1 3com-3c590-tpo

Someone on usenet mentioned ht used the 3Com-3C590-TPO (EtherLink III - PCI). He had to get the "3c59x.c" driver and "vortex.patch" to make it work with his 1.2.8 Linux kernel.

6.2 DEC435 PCI NIC

The DEC435 PCI NIC is said to work great with the drivers included in the Slackwaredistribution - I'd say they are in the standard-kernel?

---

7. Motherboards

The people who answered were using the following boards:

7.1 ASUS

- Ruediger.Funck@Physik.TU-Muenchen.DE - successful.
- strauss@dagoba.escape.de - half-successful, works, but...
- krypton@netzservice.de (Ulrich Teichert), - successful.
- heinrich@zsv.gmd.de - successful
- CARSTEN@AWORLD.aworld.de - successful
- egooch@mc.com - successful - but trouble with the serial port
- archie@CS.Berkeley.EDU and his friend - successful after solving IDE-puzzle
- Lars Heinemann (lars@uni-paderborn.de) successful
- Michael Will (Michael.Will@student.uni-tuebingen.de) - successful.

7.2 Micronics P54i-90

root@intellibase.gte.com succesful bill.foster@mccaw.com successful karpens@ncssm-server.ncssm.edu successful

7.3 SA486P AIO-II

ah@doc.ic.ac.uk successful

7.4 Sirius SPACE

hi86@rz.uni-karlsruhe.de - successful

7.5 Gateway-2000

kenf@clark.net - no problems except the soundcard he tries to swap dmarples@comms.eee.strathclyde.ac.uk - successful, but... robert logan (rl@de-montfort.ac.uk) - flawless. James D. Levine (jdl@netcom.com) - flawless.

TOC

7.6 Intel-Premiere

grif@cs.ucr.edu - successful jeromem@amiserv.xnet.com - successful demarest@rerf.or.jp - successful (Premier-II)

7.7 DELL Poweredge SP4100 gbelow@pmail.sams.ch - successful

7.8 DELL OptiPlex GI+ 575torsten@videonetworks.com - successful when turning off plug and play

7.9 Comtrade Best Buy PCI / PCI48X MB Rev 1.0

tldraben@Teleport.Com - "Works, I believe it has buggy Saturn chipset. I would also like to add: I strongly recommend not buying from Contrade. Their service is horrible. "

7.10 IDeal PCI / PCI48X MB Rev 1.0

tldraben@Teleport.Com - "Did not work with PCI48X motherboard"

7.11 CMD Tech. PCI IDE / CSA-6400C

tldraben@TelePort.com - "Works"

7.12 GA-486iS (Gigabyte)

Stefan.Dalibor@informatik.uni-erlangen.de - success with problems.

7.13 GA-586-ID (Gigabyte) 90 Mhz Pentium PCI/EISA Board

kkeyte@esoc.bitnet - succesful

7.14 ESCOM 486dx2/66 - which board?

Works perfect except the ftape-streamer (archive)

7.15 J-Bond with i486dx2/66

Drew Eckhardt (drew@kinglear.cs.Colorado.EDU) uses Diamond Stealth 64 VRAM with 4M of memory (964 based). It works great, he usualy runs it at 1024x768 72hz in 32bpp; 16 and 8bpp also work. He needed to get the X311u2S3.tgz server from ftp.xfree86.org; people with 968 based Diamond boards will definately need to do this.

7.16 super micro 011895 03:50 SUPER P54CI-PCI rev 1.3 (Opti)

Manuel de Vega Barreiro

- board super micro 011895 03:50 SUPER P54CI-PCI rev 1.3
- Opti chipset: 82c557,82c556,82c558,82c621.
- 4 PCI, 4 ISA Slots (4x16 Bit)
- ZIF Socket for CPU (120,100,90,75 mHz)
- 4 72 pin-SIMMs (max 128Mb)
- cache 256,512,1024 Kb L2-cache
- Ami WinBIOS in Flash-Eprom (101094-VIPER-P)
- onboard: EIDE for 4 drives
- Pentium with 90Mhz, 8M (now 16M) RAM and 256K L2-cache.
- 1 maxtor 540 Mb, 1 st3122A 1Gb
- Number Nine 9GXE64pro with 2Mb
- Sound blaster 16 + cdrom Matsushita
- 17" microscan 5ep ADI monitor

I run linux 1.1.57 (now 1.2.1) without problems. dosemu0.53 work fine (com. software like kermit and xtalk) XFree86 3.1 at 1024x768 resolution

8. reports on success

8.1 GigaByte GA486-AM with AMD Am5x86-133-WB @ 160MHz (40MHz PCI)

GigaByte GA486-AM

- AMD Am5x86-133-WB @ 160MHz (40MHz PCI)
- BIOS as of 11/07/95 (Rev.A)
- 256KB 2nd level cache (15ns)
- 48MB RAM (Mixed 60/70ns)

Hercules Terminator 64/VIDEO (S3 765 or "Trio 64V+")

Sound Blaster 16

- Panasonic CR563 CD-ROM drive

Silicon 4Ser/3Par I/O

- Mouse
- Terminal
- Terminal
- Modem (14k4)
- HP Laserjet III

Mitsumi CD-ROM controller

- FX001D drive

Longshine 1MBit Floppy controller

- IOMega Tape Insider 250
- 3,5" Floppy
- 5,25" Floppy

No Network card, because the 4 ISA slots are full, and I don't have a PCI card. I (now) use kernel 2.0.22 with APM enabled, and the hard drives power down and up properly without panics. The system is 24hrs up a day and still running. Kernel compilation takes between 5 and 7 minutes, depending on options.

8.2 California Graphics - Sunray II Pro

Guido Trentalancia (guido@gulliver.unian.it) reported the California Graphics - Sunray II Pro with Triton chipset to work well with Pentium100, Hd: Conner cfs420a, Conner cfs210a, crunching numbers at 147492 dhrystones/second.

8.3 Micronics P54i-90 (root@intellibase.gte.com)

TOC

Pentium with 90Mhz, 32M RAM and 512K L2-cache. Works extremely well (a kernel recompile takes 10 minutes :-).

The board includes:

- UART - two 16550A high speed UARTS
- ECP - one enhanced parallel port
- Onboard IDE controller
- Onboard floppy controller

Pros: Currently, I'm using it with an Adaptec 1542CF and a 1G Seagate drive, No problems. Graphics is ATI Graphics Pro Turbo (PCI). Very fast. The serial ports can keep up with a TeleBit T3000 modem (38400) without overruns. Caching above 16M does occur. There are 3 banks of SIMM slots (2 SIMM's per bank), with each bank capable of 64M each (2 32M 72-pin SIMM's). Each bank must be filled completely to be used (I'm only using bank 0 with 2 16Mx72-pin SIMM's). The CPU socket is a ZIF type socket. The BIOS is Phoenix, FLASH type.

Drawbacks: RAM is expandable to 192M, but the L2 cache is maxed at 512K. While the graphics are very fast, there is currently no XF86 server for the Mach64 (well, actually there is, but it doesn't use any of the accelerator features; it's just an SVGA server). I don't know if the onboard IDE hard drive controller works; I'm prejudiced against a standard that won't allow my peripherals to operate across platforms, so I didn't buy an IDE disk; instead, I got a Seagate 31200N and a NEC 3Xi.

Mitch

8.4 Angelo Haritsis (ah@doc.ic.ac.uk) about SA486P AIO-II:

The motherboard I eventually bought (in the UK) is one supporting 486 SX/DX/DX2/DX4 chips. It is called SA486P AIO-II. Features include:

- Intel Saturn v2 chipset
- Phoenix BIOS (flash eprom option)
- NCR scsi BIOS v 3.04.00
- 256K 15ns cache (max 512) write back and write through
- 4 72-pin SIMM slots in 2 banks
- 3 PCI slots, 4 ISA
- On-board NCR 53c810 scsi controller
- On-board IDE / floppy / 2 x 16550A uarts / enhanced parallel

I bought it from a company (UK) called ICS, (note I have no connections whatsoever with the company, just a happy customer). I use a 486/DX2-66 CPU.

Before I had a VLB 486 m/board with a buslogic BT-445S controller that I was borrowing. I have 2 scsi

TOC

devices: 1 barracuda 2.1GB ST12550N disk and a Wangtek 5525ES tape drive. I was expecting a lot of adventures by switching to the new motherboard, esp after hearing all these non-success stories on the n To my surprise everything worked flawlessly on the 1st boot! (1.1.50). And it has been doing so for about a month now. I did not even have to repartition the disk: apparently the disk geometry bios translation of the 2 controllers is the same. Linux has had no problems at all. SCSI is visibly much faster as well (sorry, I have no actual performance measurements).

The only problems (related to Drew's linux ncr53c7,810 scsi driver - thanks for the good work Drew!) are:

- no synchronous transfers are yet supported => performance hit
- disconnect/reconnect is disabled => disk scsi ops "hold" during certain slow scsi device opeartions (eg tape rewind)
- tagged queuing is not there (?) => performance hit

If you get Windows complainingg about 32-bit disk driver problems, just disable 32-bit disk access via Control Panel. This should not hurt performance. (What I did is remove the WDCTRL driver from my SYSTEM.INI).

All else is fine. I tried the serial ports with some dos/windows s/w and worked ok. The IDE/floppy work ok as well. I have not tried the parallel yet. The motherboard is quite fast and so far I am very pleased with the upgrade. I have not yet tried a PCI graphics board. I will later on. I am using an old ISA S3 which is fine at the moment.

PS: the NCR drivers in the 2.0.x kernels should have no problems of that kind anymore. please consult the SCSI-HOWTO for further and hopefully more uptodate information.

8.5 bill.foster@mccaw.com about his Micronics M5Pi

Micronics M5Pi motherboard with 60 MHz Pentium, PCI bus having the following components:

```
16Mb RAM/512k cache
onboard IDE, parallel, 16550A UARTS
2 X 340MB Maxtor IDE Hard Drives
Soundblaster 16 SCSI-II
Toshiba 3401B SCSI CD-ROM
Archive Viper 525MB SCSI Tape Drive
Viewsonic 17 monitor
Cardex Challenger PCI video card (ET4000/W32P)
A4-Tech Serial Mouse
```

Everything works great, Slackware installation was very easy, I can run Quicken 7 for DOS under DOSEMU. I run X at 1152x900 resolution at 67Hz.

8.6 Simon Karpen (karpens@ncssm-server.ncssm.edu) with Micronics M54pi

TOC

I have had no problems with the above board, the on-board PCI IDE (hopefully soon will also have SCSI) and an ATI Mach32 (GUP) with 2MB of VRAM.

8.7 Goerg von Below (gbelow@pmail.sams.ch) about DELL Poweredge

```
- Intel 486DX4/100
- 16 MB RAM
- DELL SCSI array (DSA) with Firmware A07, DSA-Manager 1.7
- 1 GB SCSI HD DIGITAL
- NEC SCSI CD-ROM
- 2 GB internal SCSI streamer
- 3-Com C579 EISA Ethernet card
- ATI 6800AX PCI VGA subsystem, 1024 MB RAM

CAVE! DELL SCSI Array controller (DSA) runs only with firmware Rev. A07 !
A06 is buggy, impossible to reboot !
To get it: ftp dell.com , file is /dellbbs/dsa/dsaman17.zip
```

Apart from this firmware-problem there where no problems for the last 2 months, running with linux 1.1.42 as primary nameserver, newsserver and www-server on internet.

8.8 zenon@resonex.com about Gateway2000 P-66

Gateway2000's P5-66 system with Intel's PCI motherboard, with 5 ISA slots and 3 PCI slots. The only PCI card I am using is the # 9 GXe level 12 PCI card (2 MB VRAM and 1 MB DRAM). This card was bought from Dell. Under Linux I am using the graphics in the 80x25 mode only (I am waiting for some XFree86 refinements before using it in 1280x1024 resolution), but under DOS/Windows I have used the card in 1280x1024x256 mode without problems. Etherlink 3C509 Ethernet card, Mitsumi bus-interface card, Adaptec 1542C SCSI interface card and additional serial/parallel ports card (which makes the total of serial ports 3).

I have total of 32 MB RAM (recognized and used by both Linux and DOS). There is also a bus mouse (Microsoft in the PS2 mode).

No problems so far.

8.9 James D. Levine (jdl@netcom.com) with Gateway2000

Gateway 2000 P5-60 with an Intel Mercury motherboard, AMI-Flash-BIOS, (1.00.03.AF1, (c)'92) 16M RAM, on-board IDE controller and an ATI AX0 (Mach32 Ultra XLR) PCI display adapter. He had absolutely no problems with the hardware so far but has not tried anything fancy, such as accelerated IDE drivers or SCSI support.

TOC

8.10 hi86@rz.uni-karlsruhe.de with SPACE

SPACE-board, 8MB RAM, S3 805 1MB DRAM PCI 260MB Seagate IDE-hard disk because of lack of NCR53c810-Driver, 0.99pl15d, does seem to work well.

8.11 grif@cs.ucr.edu with INTEL

17 machines running a 60Mhz-i586 on Intel-Premier-PCI-Board

8.12 Jermoe Meyers (jeromem@amiserv.xnet.com) with Intel Premiere

Motherboard - Intel Premiere Plato-babyAT 90mhz with Buslogic bt946c w/4.86 mcode w/4.22 autoSCSI firmware, (note, mine came with 4.80 mcode and 4.17 autoSCSI firmware. (interrupt pins A,B,C conform to respective PCI slots!) ATI Xpression (Mach64) - using driver from sunsite, (running AcerView 56L monitor).

The motherboard has 4 IDE drives, Linux (Slackware 2.0) sees the first two and everything on the Buslogic as it emulates an adaptec 1542. Uh, yes, Dos sees them all. Buslogic is VERY accomodating in regards to shipping upgraded chips (you will have to know how to change PLCC (plastic leaded chip carrier) chips, 3 of them. Though, don't let that scare you :-) it's not that tough. Get a low end PLCC removal tool, and your in business. You also might want to "flash upgrade your system bios from Intel's IPAN BBS, a trivial process. Whats even more interesting is I also have a Sound Blaster SCSI-2 running a scsi CDROM drive off it's adaptech 1522 onboard controller. So thats 4 IDE drives (2 under Linux) and 2 SCSI-2 controllers.

I hope this helps others who are struggling with PCI technology use Linux! Jerry (jeromem@xnet.com)

8.13 Timothy Demarest (demarest@rerf.or.jp) Intel Plato Premiere II

My system is configured as follows: 16Mb 60ns RAM, 3Com Etherlink-III 53C809 ethernet card (using 10base2), ATI Mach 64 2Mb VRAM, Toshiba 2x SCSI CDROM, NCR 53c810 PCI SCSI, Syquest 3270 270Mb Cartridge Drive, Viewsonic 17 monitor, Pentium-90 (FDIV Bug Free). Running Slackware 2.1.0, Kernel 1.2.0, with other misc patches/upgrades.

Everything is functioning flawlessly. I dont recommend the Syquest drives. I have used the 3105 and the 3270 and both a very, very fragile. Also, the cartridges are easily damaged and I have had frequent problems with them. I am in the process of looking for alternative removable storage (MO, Zip, Minidisc, etc).

Some information you might need:

TOC

Flash Bios upgrades

Flash Bios updates can be ftp'd from wuarchive.wustl.edu:/pub/MSDOS_UPLOADS/plato. The current version is 1.00.12.AX1. The BIOS upgrades *must* be done in order. 1.00.03.AZ1 to 1.00.06.AX1 to 1.00.08.AX1 to 1.00.10.AX1 to 1.00.12.AX1. The Flash BIOS updates can also be downloaded from the Intel BBS. I do not have that number right now.

NCR 53c810 BIOSless PCI SCSI

If you are using an NCR 53c810 BIOSless PCI SCSI card in the Plato, you may have trouble getting the card to be recognized. I had to change one of the jumpers on the NCR card: the jumper that controls whether there is 1 or 2 NCR SCSI cards in your system must be set to "2". I dont know why, but this is how I got it to work. The other jumper controls the INT setting (A,B,C,D). I left mine at A (the default).

apart from that - plug and play!

There are no settings in the motherboard BIOS for setting the NCR 53c810. Dont worry - once the card is jumpered correctly, it will be recognized! So much for PCI Plug-n-Play!

8.14 heinrich@zsv.gmd.de with ASUS

ASUS-PCI-Board (SP3) having:

- -- Asus PCI-Board with AMD 486/dx2-66 and 16M RAM
- -- Fujitsu 2196ESA 1G SCSI-II
- -- Future Domain 850MEX Controller (cheap-SCSI-Controller, almost a clone to Seagate's ST01... want's to use ncr53c810 as soon as the driver comes out
- -- ATI Graphics Ultra (the older one with Mach-8 Chip, ISA-Bus)
- -- Slackware 1.1.1

He just exchanged the boards, plugged his cards in, connected the cables, and it worked perfect. He does not use any PCI-Cards yet, though.

8.15 CARSTEN@AWORLD.aworld.de with ASUS

ASUS-PCI-Board with 486DX66/2, miro-crystal 8s PCI driven by the S3-drivers of XFree86-2.0, using the onboard SCSI-Chip. No problems with compatibility at all.

8.16 Lars Heinemann (lars@uni-paderborn.de) with ASUS

ASUS PCI/I-486SP3 Motherboard w/ 486DX2/66 and 16M RAM (2x8), miroChrystal 8S/PCI (1MB) S3, Soundblaster PRO, Adaptec 1542b (3.20 ROM) SCSI host adapter with two hard disks (Fujitsu M2694ESA u. Quantum LPS52) and a QIC-150 Streamer attached. No problems at all!

8.17 Ruediger.Funck@Physik.TU-Muenchen.DE with ASUS

TOC

ASUS PCI/I-486SP3 / i486DX2-66 / 8 MB PS/2 70 ns BIOS: Award v 4.50 CPU TO DRAM write buffer: enabled CPU TO PCI write buffer: enabled PCI TO DRAM write buffer: disabled, unchangeable CPU TO PCI burst write: enabled Miro Crystal 8s PCI - S3 P86C805 - 1MB DRAM

Quantum LPS 540S SCSI-Harddisk on NCR53c810-controller.

8.18 robert logan (rl@de-montfort.ac.uk with GW/2000)

Gateway 2000 4DX2-66P 16 Megs RAM, PCI ATI AX0 2MB DRAM (ATI GUP). WD 2540 Hard Disk (528 Megs) CrystalScan 1776LE 17inch. (Runs up to 1280x1024) Slackware 1.1.2 (0.99pl15f)

It is giving no problems. He uses SLIP for networking and an Orchid-Soundwave-32 for niceties, awaiting the NCR-Driver. The only problem he has is that the IDE-Drive could be much faster on the PCI-IDE. It is one of the new Western Digital fast drives and in DOS/WfW it absolutely screams - on Linux it is just as slow as a good IDE-Drive.

8.19 archie@CS.Berkeley.EDU and his friend use ASUS

Archie and his friend have rather similar configurations:

- ASUS PCI-SP3 board (4 ISA, 3 PCI)
- Intel 486DX2/66
- Genoa Phantom 8900PCI card (friend: Tseng 3000/W32i chipset)
- Maxtor 345 MB IDE hard drive
- Supra 14.4 internal modem
- ViewSonic 6e monitor (Archie)
- NEC Multisync 4fge (friend)
- Slackware 1.2.0

The onboard-SCSI is disabled. First there were problems with the IDE-drive: ``on the board there's a jumper which selects whether IRQ14 comes from the ISA bus or the PCI bus. The manual has an example where they show connecting it to PCI INT-A. Well, we did that just like the example... but then later our IDE drive would not work (the IDE controller is on board). Had to take it back. The guys at NCA were puzzled, then traced it back to this jumper. I guess the IDE controller uses IRQ14 or something? That's not documented anywhere in the manual. Other than that, seems to be kicking ass nicely now. Running X, modeming, etc. (for the Supra you have to explicitly tell the kernel that the COM port has a 16550A using setserial (in Slackware /etc/rc.d/rc.serial))".

8.20 Michael Will with ASUS-SP3 486 (the old one)

TOC

used the following:

- ASUS PCI-SP3-Board with 486dx2/66 and 16M RAM
- NCR53c810-SCSI-II chip driving a 1GB-Seagate-SCSI-II disk and a Wangtec-tape
- ATI-GUP PCI Mach32 Graphics card with 2M VRAM running perfectly with XFree86(tm)-3.1 8bpp and 16bpp
- Linux kernel 1.1.69

It runs perfectly and I am content with the speed, the ATI-GUP-PCI (Mach32) does not give as good benchmarks as expected, though. Since I got the money by now, I got me an ASUS-SP4 with P90 which gives me better throughput on Mach32-PCI... If I had even more money I'd get me another 16M of RAM and a Mach64-PCI with 4M RAM, though... I still keep on dreaming :-)

8.21 Mike Frisch (mfrisch@saturn.tlug.org) Giga-Byte 486IM

- Motherboard: Giga-Byte 486IM
- Configuration: 4 ISA slots (2 double as VLB) and 4 PCI slots
- CPU: Intel 486DX/33
- BIOS: Award 4.50G
- PCI EIDE Disk Controller: Giga-Byte GA-107 (CMD 640x PCI Multi-I/O)
- PCI Video card: ATI Graphics eXpression PCI 2MB DRAM
- Linux Kernel: 1.2.9
- Linux Dist'n: Highly modified Slackware 2.2.0

I have been running this board 24 hours a day for the past 5-6 months. It has worked flawlessly for me under DOS/Windows, OS/2 Warp, and Linux (with Linux being run usually 24 hours a day).

8.22 Karl Keyte (kkeyte@esoc.bitnet) Gigabyte GA586 Pentium

- PCI/EISA Board Gigabyte GA586-ID 90MHz Pentium (dual processor, one fitted)
- 32M RAM
- SCSI - no scsi-NCR-chip on-board, using Adaptec 1542C,
- PCI ATI GUP 2M VRAM
- Adaptec 1742 EISA SCSI controller
- Soundblaster 16
- usual I/O

Everything under DOS AND Linux works perfectly. No problem whatsoever. A VERY fast machine! BYTE

Unix benchmarks place it about the same as a Sun SuperSPARC-20 running Solaris 2.3. The PC is faste integer arithmetic and process stuff (including context switching). The SPARC is faster for floating poi one of the disk benchmarks.

TOC

8.23 kenf@clark.net with G/W 2000

He uses a Gateway 2000 with no problems, except the soundcard (which one?). He is trading it in for a genuine soundblaster in hopes that will help.

8.24 Joerg Wedeck (jw@peanuts.informatik.uni-tuebingen.de) / ESCOM

originaly buyed a 486 DX2/66 from ESCOM (which board?) with onboard IDE and without (!) onboard NCR-SCSI-chip. ISA-adaptec 1542cf scsi-controller instead spea v7 mercury lite (s3, PCI, 1MB), ISA-Soundblaster-16, mitsumi-cdrom (the slower one). Everything except the archive-streamer works with no problems. The spea-v7 works perfectly since XFree86-2.1

He abandoned the Intel-board in favour of an ASUS-SP3-g and has some problems with PCI-to-Memory burstmode which is crashing only on Linux, "looking like a deadlock in the swapper". If you have any information on this, please eMail the maintainer of the PCI-HOWTO.

After turning off the PCI-to-Memory posting feature it just works perfect.

Rather than sending him mail please read his http-homepage at "http://wsiserv.informatik.uni-tuebingen.de/ jw" where he keeps information about his PCI-system, too.

8.25 Ulrich Teichert / ASUS

ASUS-PCI board with AMD486dx40 (but actually running at 33Mhz?!) His ISA-ET3000 Optima 1024A ISA works nice. No problems with Quantum540S SCSI Harddisk attached to the onboard NCR53c810.

---

9. Reports of problems

9.1 Compaq PCI systems, especially Presarios

Patrick Yaner (p_yaner@eos.ncsu.edu) reported a Compaq-speciality to me. It seems they are mapping the PCI BIOS data area to an obscure area of memory, one that Linux (or OS2) cannot access. It can usually find it, but it can't get in, and gives a message on startup (something like "pcibios_init: entry in high memory area, unable to access"). Although this is alright with the display (which is on the PCI bus) and the IDE controller (also PCI), it means any other PCI devices -- such as an Ethernet card -- cannot be detected by Linux.

Compaq offers a driver for DOS at ftp://ftp.compaq.com/pub/softpaq/Drivers/SP1116.ZIP

but using this with linux would mean using the program that boots linux from DOS, instead of LILO. Note that Compaq occasionally updates the software in this archive, so the file ftp://ftp.compaq.com/pub/softpaq/allfiles.html (also available as allfiles.txt) might be handy in checking to see that they haven't upgraded.

Oddly, this information can also be found in the SCSI HOWTO, although the Pressarios come with IDE built in.

9.2 VLSI Wildcat PCI chipset like in Zeos P120 box

Paul Bame (bame@sde.hp.com) reported:

The Wildcat PCI chipset works fine in late 1.3 and all 2.0 kernels.

9.3 dmarples@comms.eee.strathclyde.ac.uk G/W 2000

Gateway 2000 G/W 2000 4DX2/66 PCI ATI-Graphics-Ultra-Pro IDE of indeterminate make

It works well - only the IDE-Card runs in ISA-compatibility-mode, and works a lot faster when switched into PCI-Mode by a DOS-program... thus it's not that fast in Linux, and a patch would be nice.

9.4 cip574@wpax01.physik.uni-wuerzburg.de (Frank Hofmann) / ASUS

He uses the ASUS-board with 16MB-RAM, ISA-based S3/928, and the onboard-IDE-controller with a Seagate ST4550A harddisk. He's had no trouble with the newer Linux-kernels.

TOC

His problem:

```
using X, my mouse is not responding the way I was used to
before. It's sometimes behind movement and makes jumps if
moved quickly. I think this was discussed In a Linux
newsgroup before (I don't know which one) and is due to the
use of 16550 serial chips for the onboard serial interfaces.
After two weeks, I got used to it :-)
```

Reducing the threshold of the 16550 should help. There should be a patch to setserial available somewhere, but I do not know where.

9.5 axel@avalanche.cs.tu-berlin.de (Axel Mahler) / ASUS

ASUS PCI/I-486SP3 Motherboard (Award BIOS 4.50), 16 MB RAM the on-Board NCR Chip is disabled, he had the Genoa Phantom/W32 2MB for PCI and a Adaptec AHA-1542CF (BIOS v2.01) connected to:

- an IBM 1.05 GB Harddisk
- a Toshiba CD-ROM (XM4101-B)
- a HP DAT-Streamer (2GB)

when creating the filesystems, 'mke2fs' (0.4, v. 1.11.93) hung and installation was impossible. After replacing the Genoa Phantom/W32 2MB PCI with an ELSA Winner 1000 2MB PCI it worked perfectly. He tested it with an old Eizo VGA-ISA and it worked as well, so the problem was in the Genoa-PCI-card.

9.6 Frank Strauss (strauss@dagoba.escape.de) / ASUS

ASUS SP3 Board i486DX2/66 NCR53c810 disabled Adaptec 1542B in ISA Slot with 2 hard drives (200MB Maxtor, 420MB Fijutsu), SyQuest 88MB and Tandberg Streamer ELSA Winner 1000 PCI, 1MB-VRAM Soundblaster Pro in ISA Slot at IRQ 5 Onboard IDE disabled Onboard serial, parallel, FD enabled

After a reset, the machine sometimes 'hangs' (soft and hard-reset the same) - this is probably not related to the Adaptec and the Soundcard, because even without these the system sometimes fails to come up. But if it runs, (and the ELSA-WINNER-1000-PCI-message appears) it runs ok.

The two serial ports are detected as 16550 as they should, but at some mailbox-sessions there was heavy data-loss at V42bis... The problem seems to be in the hardware...

CPU>-PCI-Burst seems to work well with DOS/MS-Windows

CPU->PCI-Burst does not work properly with linux0.99p15, Messing up when switching the

TOC

virtual-consoles, crashing completely when calling big apps like ghostview, or xdvi, leaving the SCSI-LED on (!).

(I suspect these apps would be using a lot of CPU->PCI-burst because of the big heap of data to transmit to the PCI-Winner-1000)

After disabling CPU->PCI-Burst, it works well, the Winner-1000 at 1152x846 (not much font cache with 1MB) does 93k xstones. OpaqueMove with twm is more than just endureable :-)

He has got a SATURN.EXE which he loads under DOS before starting Linux, helping to turn on burst without hangs...

Someone stated that these problems might go away when turning off "sync negotiation" on the Adaptec - I do not know if this is possible with the adaptec1542B too? But I guess so.

With CPU->PCI-Burst it yielded 95k xstones, so he considers it as not too grave to do without. His only problem is that he would like to run his Winner-1000 at 1152x900 which fails because it seems to take any x-resolution higher than 1024pixels as a 1280pixel-resolution, thus wasting a lot end resulting in a y-resolution of 816pixels... but this is probably no PCI-related problem. It should have gone away with XFree86-2.1

9.7 egooch@mc.com / ASUS

- BOARD ASUS PCI/I-486 SP3 RAM: 16MB (4x4M-SIMM)
- CPU 486DX33 CPU
- BIOS Ver. 4.50 (12/30/93)
- Floppy Two floppy drives (1.2 and 1.44), using ASUS on-board floppy controller
- SCSI tried both WD7000 SCSI controller and Adaptec 1542CF and worked.
- Two SCSI 320M hard drives
- SCSI NEC84 CDROM drive
- SCSI QIC150 Archive tape drive
- Video - Tseng ET4000 ISA graphics card
- Sound PAS16 sound card
- Printer attached to on-board ASUS parallel port

He has nothing in the PCI-Slots yet, but wants to buy a PCI-Video-Card, currently uses WD7000 SCSI controller but will switch to the NCR-Chip onboard as soon as the driver is out.

Everything works perfectly - the first serial port which has a 14.4K-Modem attached does hang occasionally when reconnecting with the modem after having used it previously. He says that would not be unique to ASUS but rather a bug in the SMC-LSI device with its 16550UART. The logitech-serial-mouse on the second port works fine. Setting down the threshold of the 16550 for the mouseport would definitely help, one does seem to need a special patched setserial for that? I have not got the information yet, please contact me if you know more!

9.8 Stefan.Dalibor@informatik.uni-erlangen.de / GigaByte

TOC

- Board - GA-486iS from Gigabyte w/ 256Kb 2L-Cache, i486-DX2
- Bios - AMI, 93/8
- SCSI - no scsi-NCR-chip on-board, using Adaptec 1542C,
- Video - ELSA Winner 1000
- Linux 0.99pl14 + SCSI-Clustering-Patches / Slackware 1.1.1

All seems to go well, but he has not tried neither networking, printing or a streamer yet. Before applying the clustering- patches he had some problems with hangs triggered by "find", but this no longer is the case - perhaps it was an older kernel-bug.

The ELSA-Winner-1000 sometimes hangs, with very strange patterns on the screen resolved only by rebooting... The dealer has told him it was a bug in the ELSA-Card, but the manufacturer claims it had solved the problem. The bug is not reproducible so he does not plan to take any action at the moment.

All in all the machine seems to work very well under heavy text processing (emacs, LaTeX, xfig, ghostview) usage. Interaction is surprisingly responsive, little difference between it and the 3-4X as expensive Sun he works on...

CPU->PCI-Burst is still disabled because the bios does not support the PCI-things well?

A problem with his new modem (v32 terbo) arose: it looses characters. Especially when using SLIP it complains a lot about RX and TX errors. As soon as he runs X it gets unusable. He said he activated FIFO and RTS/CTS with stty, but to no avail...

9.9 Steve Durst (sdurst@burns.rl.af.mil) with UMC 8500 mainboard

Running Linux 1.2.12 on the UMC8500-100Mhz motherboard with the dreaded CMD PCIO640B (E)IDE controller, when booting the screen wiggles a few seconds, as if the Diamond Stealth64-DRAM (S3 864) has to warm up first, but he can live with that.

9.10 Tom Drabenstott (tldraben@Teleport.Com) with Comtrade / PCI48IX

PCI48IX Motherboard Rev. 1.0. Made by ??? documentation copyrighted by "exrc". The BIOS says not very much about PCI.

His E-315E Super IDE UMC (863+865) ISA-Controller-card does have problems. (It is a multifunction controller-card). It seems to work well under DOS/OS2 but not under Linux.

---

10. General tips for PCI-Motherboard + Linux NCR PCI SCSI

This was compiled by Angelo Haritsis (ah@doc.ic.ac.uk) from various people's postings:

10.1 DON'Ts:

Do *NOT* go for combination VLB/PCI motherboards. They usually have a lot of problems. Get a plain PCI version (with ISA slots as well of course). A lot of bad things have been heard about OPTI chipset PCI motherboards. Someone hints: "Avoid the OPTi (82C596/82C597/82C822) chipset based motherboards like the TMC PCI54PV".

(I know of at least one person having no problems with his TMC PCI54PV motherboard. He just had to put the NCR53c810 addonboard into slot-A which is the only slot capable of busmastering as it seems.)

Rumours say that Intel chipset PCI motherboards will have problems with more than one bus-mastering PCI board. I have not tried this one yet on mine and have nothing to suggest. I also heard that the Saturn II chipset is problematic, but this is the one I use and it is perfectly ok! Advice: Try to negotiate a 1-2 week money back agreement with your supplier, in case the motherboard you get has problems with the use you plan for it.

10.2 SIMM slots

Go for 72-pin only SIMMs for speed: Some (all?) of the mainboards which take 30 pin SIMMs use a 32 bit main memory interface, and will be significantly slower than the Intel based boards which all use a 64 bit or permantly interleaved memory interface. You might want to keep that in mind.

10.3 Praised PCI Pentium motherboard

The P90 Intel motherboard with the Intel Premiere II chipset (aka Plato). Get the latest BIOS which has concatenated NCR scsi BIOS 3.04.00. Otherwise DOS won't see your scsi disk(s) if you use a BIOS-less 53c810 based controller. NCR SCSI BIOS exists in the AMI BIOS of the plato after version 1.00.08 (or maybe verion 1.00.06). This BIOS is FLASH upgradeable so you should be able to get the upgrade on a floppy from your supplier. The current version is 1.00.10 and has all early problems fixed.

(Bios files should be available at ftp.demon.co.uk:/pub/ibmpc/intel, but I did not check that myself. the Autor.)

10.4 irq-lines

The value in the interrupt line PCI configuration register is usually set manually (for compatability with legacy ISA boards) in the extended CMOS setup screens on a per-slot or per-device basis. Older PCI mainboards also force you to set jumpers for each PCI slot/device which select how PCI INTA and perhaps INTB, INTC, and INTD are mapped to an 8259 IRQ line, Obviously, if these jumpers exist on your board, they must match the settings in the extended CMOS setup. Also note that some boards (notably Viglens) have silkscreens and instruction manuals which disagree with the wiring, and some experimentation may be in order.

10.5 Info about the different NCR 8xx family scsi chips

All NCR 8XX Chips are dircet connect PCI bus mastering devices, that have no preformance difference wether on motherboard or add in option card. All devices comply with PCI 2.0 Specification, and can burst 32 bit data at the full 33 MHz (133Mbytes/Sec)

53C810

53C810 = 8 bit Fast SCSI-2 (10 MB/Sec) Single ended only Requires Integrated Mother board BIOS 100 pin Quad Flat Pack (PQFP) Worlds first PCI SCSI Chip, Volumes make it the most inexpensive.

53C815

53C815 = 8 bit Fast SCSI-2 (10 MB/Sec) Single Ended only Support ROM BIOS interface, which makes it ideal for add-in card Designs. 128 Pin QFP

53C825

53C825 = 8 bit Fast SCSI-2, Single ended or Differential 16 bit Fast SCSI-2 (20 MB/Sec), Single ended or Differetial Also has support for external Rom, making it a good candidate for add in cards. 160 pin QFP Not supported by linux yet. (See section below on news about the 825). Must have devices with wide or differential scsi to use these features.

10.6 future of 53c8xx

There are 4 new devices planned for announcement late this year and into early next year. Footprint compitible with 810 and 825 with some new features.

All the Chips require a BIOS in DOS/Intel applications. The 810 is the only chip that needs it resident on the motherboard. Latest NCR SCSI BIOS version: 3.04.00 The bios supports disks >1GB, indeed up to 8G under MS-LOSS.

10.7 Performance of the 53c810

C't magazine's DOS benchmarks showed that it was significantly faster than the Buslogic BT-946, one user noted a 10-15% performance increase versus an Adaptec 2940, and with a very fast disk it may be 2.5X as fast as an Adaptec 1540.

10.8 News about NCR53c825 support

works. period.

10.9 Frederic POTTER (Frederic.Potter@masi.ibp.fr) about Pentium+NCR+Strap_bug

On some Intel Plato board, the NCR bios doesn't recognize the board, because it needs to see the board as a "secondary SCSI controller", and because on most SCSI board the jumper to select between primary/secondary has

been ironed to primary (to spare 1 cent, presumably).

Solution:

```
near the NCR chip, they are 3 via ( kind of holes ) with a strap like
that
                O--O  O

        this mean primary is selected as default setting. For the Plato Intel
        Mainboard, it should be like that

                O  O--O

        The best solution is to get rid of the strap and to put a 2 position
        jumper instead.
```

10.10 PCIprobe in the latest Linux Kernels by Frederic Potter

Frederic Potter has added a PCI-Probe into the latest kernels. If you do a "cat /proc/pci" it should list all your cards. If you own cards which are not properly recogniced, please contact him via mail as "Frederic.Potter@masi.ibp.fr".

See arch/i386/kernel/bios32.c and include/linux/pci.h in the kernel source for more information on PCI-Probe-Stuff.

10.11 Other PCI Devices

What other PCI-cards are supported? Apart from various graphicscards, I would like to know about other cards like ethernet, framegrabber, or the TSET boards Cyclades is about to beta-test at the moment:

Cyclades: a 16-port PCI RISC-based multiport card.

The product is called Cyclom-Ye, and has the following characteristics:

- PCI host card based on the PLX chip-set. This host card supports 8 to 32 serial ports, utilizing 8 or 16-port external boxes.
- SCSI II cable.
- 8 or 16-port external boxes with RJ45 or DB25 connectors (your choice). You can start with 8 ports and expand to 32, by just adding more boxes. Each external box contains 2 or 4 CD-1400 RISC Serial controllers (each CD-1400 controls 4 serial ports).
- Up to 4 Host cards can be installed in the PC system, allowing a maximum of 128 serial ports per system.

The product is being in the beta-test phase at July the 26th, 1995, and should be available by Octobre or something. eMail them at sales@cyclades.com.

11. Conclusion

If you have some moneny to put into your machine, you'd be well off with a Pentium90, ASUS-SP4, which is what I use at the moment. If you can afford 32M RAM that would be much better than 16M RAM.

Real soon now the upcoming standard will be the Triton Chipset with support for special SIMMS called EDODRAM, and SRAM. Both will be more expensive than PS2-RAM, and at the time of writing (28-June-1995) SRAM is not available. While EDO-DRAM is more expensive, this is not because of the production costs, they are said to be the same.

For a highperformance system I would still choose an ASUS-TP4/XE with EDO-DRAM, but if you do not need to use it at the moment, I d rather wait some more.

For Graphic-boards I'd say the best cheap board fitting perfectly on a good Multisync-15 like the Samsung SyncMaster 15Gli, is the SPEA V7 Mirage P64 with Trio64 Chipset and 2M DRAM. For more sophisticated Display like the Iiyama-IDEK 8617A-T I think the PCI Mach64 ATI-GUP-Turbo (not the cheaper GUP-Turbo-Windows) would be a good choice, with 4M RAM you can have truecolor in higher resolutions. It is well supported in the XFree86(tm)-3.1.1, and there are commercial X-Servers available of which I'd recommend Accelerated/X by Roell, which supports the Mach64 very well and fast.

For SCSI I'd take the DPT rather than the (much cheaper and very fast) NCR53c810 in case you plan to use SCSI-Tapes a lot. The NCR53c810 driver on Linux does lack disconnect/reconnect support, thus blocking the SCSIbus on operations like "mt rewind", "mt fsf" etc. It bears a performance penalty on tar-operations - but check out Drews new alpha drivers before making a decision, perhaps it does solve all the problems.

For building servers, the DPT would be the controller of choice anyway because of all the nifty hardware cache (with elevator sorting on accesses, so cache it is not a silly thing even in a Linux enviroment where the OS does the caching) and RAID-Support up to raid level 5.

If you do not want to spend that much money on computer equipment (e.g.: you are having a life) you might go for an ASUS-SP3-SiS with AMD-DX2/66 or DX4/100. The SPEA V7 Mirage P64 PCI with 2M DRAM would be a good choice, since it uses the Trio64 S3 Chip, which is well supported by XFree86(tm)-3.1.1, quite cheap to buy and fast, too.

Another fine card since XFree86(tm)-3.1 is the fast and cheap et4000/w32-PCI-card.

12. Thanks

I want to thank the following people for supporting this document:

- David Lesher (wb8foz@netcom.com) for extensive help with the english language
- Nathanael MAKAREVITCH (nat@nataa.frmug.fr.net) for translating into french
- Jun Morimoto (morimoto@lab.imagica.co.jp) for translating into japanese
- Marco Melgazzi (marco@vcldec1.polito.it) for translating into italian
- Donald Becker (becker@cesdis.gsfc.nasa.gov) for ethernet-informations
- Drew Eckhardt (drew@kinglear.cs.Colorado.EDU) for SCSI-informations
- Zhahai Stewart (zhahai@hisys.com) for help with the intro section

and many more peole adding information mostly by mail and by posts, some of them will be named here:

```
CARSTEN@AWORLD.aworld.de,
dmarples@comms.eee.strathclyde.ac.uk,
drew@kinglear.cs.Colorado.EDU (Working at the PCI-NCR53c810-Driver),
duncan@spd.eee.strathclyde.ac.uk,
fm3@irz.inf.tu-dresden.de,
grif@ucrengr.ucr.edu,
heinrich@zsv.gmd.de,
hm@ix.de (iX-Magazine),
hm@seneca.ix.de,
kebsch.pad@sni.de,
kenf@clark.net,
matthias@penthouse.boerde.de,
ortloff@omega.informatik.uni-dortmund.de,
preberle@cip.informatik.uni-erlangen.de,
rob@me62.lbl.gov,
rsi@netcom.com,
sk001sp@unidui.uni-duisburg.de,
strauss@dagoba.escape.de,
strauss@dagoba.priconet.de,
hi86@rz.uni-karlsruhe.de,
Ulrich Teichert, krypton@netzservice.de,
Stefan.Dalibor@informatik.uni-erlangen.de,
tldraben@teleport.com
mundkur@eagle.ece.uci.edu,
ooch@jericho.mc.com,
Gert Doering (gert@greenie.muc.de),
James D. Levine (jdl@netcom.com),
```

Georg von Below (gbelow@pmail.sams.ch),
Jerome Meyers (jeromem@quake.xnet.com),
Angelo Haritsis (ah@doc.ic.ac.uk),
archie@CS.Berkeley.EDU and his friend kenf@clark.net.

13. copyright/legalese

(c)opyright 1993,94 by Michael Will - the GPL (Gnu Public License) applies. See last section about this.

If you sell this HOWTO on a CD or in a book I would be happy to have a copy for reference.

(Michael.Will@student.uni-tuebingen.de)

Contact me, either via eMail or call +49-7071-969063.

Trademarks are owned by their owners. There is no warranty on the information in this document.

For german users I am offering tested, preinstalled / preconfigured and supported Linux-PCI-machines. Call me at 07071-969063.

14. GPL - Gnu Public License

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                          675 Mass Ave, Cambridge, MA 02139, USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must show them these terms so they know their
rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

  Also, for each author's protection and ours, we want to make certain
```

that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your

cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any

TOC

later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

```
    Copyright (C) 19yy  (name of author)

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
    Gnomovision version 69, Copyright (C) 19yy name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  (signature of Ty Coon), 1 April 1989
  Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Linux PCMCIA HOWTO

David Hinds, dhinds@hyper.stanford.edu.

v2.32, 15 May 1999

This document describes how to install and use PCMCIA Card Services for Linux, and answers some frequently asked questions. The latest version of this document can always be found at ftp://hyper.stanford.edu/pub/pcmcia/doc. An HTML version is at http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html.

1. General information and hardware requirements

- 1.1 Introduction
- 1.2 Copyright notice and disclaimer
- 1.3 What is the latest version, and where can I get it?
- 1.4 What systems are supported?
- 1.5 What cards are supported?
- 1.6 When will my favorite (unsupported) card become supported?
- 1.7 Mailing lists and other information sources
- 1.8 Why don't you distribute binaries?
- 1.9 Why is the package so darned big?

2. Compilation and installation

- 2.1 Prerequisites and kernel setup
- 2.2 Installation
- 2.3 Startup options
- 2.4 System resource settings
- 2.5 Notes about specific Linux distributions

TOC

3. Resolving installation and configuration problems

- 3.1 Base PCMCIA kernel modules do not load
- 3.2 Some client driver modules do not load
- 3.3 Interrupt scan failures
- 3.4 IO port scan failures
- 3.5 Memory probe failures
- 3.6 Failure to detect card insertions and removals
- 3.7 System resource starvation
- 3.8 Resource conflict between two cards
- 3.9 Device configuration does not complete

4. Usage and features

- 4.1 Tools for configuring and monitoring PCMCIA devices
- 4.2 Overview of the PCMCIA configuration scripts
- 4.3 PCMCIA network adapters
- 4.4 PCMCIA serial and modem devices
- 4.5 PCMCIA SCSI adapters
- 4.6 PCMCIA memory cards
- 4.7 PCMCIA ATA/IDE card drives
- 4.8 Multifunction cards

5. Advanced topics

- 5.1 Resource allocation for PCMCIA devices
- 5.2 How can I have separate device setups for home and work?
- 5.3 Booting from a PCMCIA device

6. Dealing with unsupported cards

- 6.1 Configuring unrecognized cards
- 6.2 Adding support for an NE2000-compatible ethernet card

- 6.3 PCMCIA floppy interface cards
- 6.4 What's up with support for Xircom cards?

TOC

7. Debugging tips and programming information

- 7.1 Submitting useful bug reports
- 7.2 Interpreting kernel trap reports
- 7.3 Low level PCMCIA debugging aids
- 7.4 /proc/bus/pccard
- 7.5 Writing Card Services drivers for new cards
- 7.6 Guidelines for PCMCIA client driver authors
- 7.7 Guidelines for Linux distribution maintainers

---

1. General information and hardware requirements

1.1 Introduction

Card Services for Linux is a complete PCMCIA or ``PC Card'' support package. It includes a set of loadable kernel modules that implement a version of the Card Services applications program interface, a set of client drivers for specific cards, and a card manager daemon that can respond to card insertion and removal events, loading and unloading drivers on demand. It supports ``hot swapping'' of most card types, so cards can be safely inserted and ejected at any time.

This software is continually under development. It probably contains bugs, and should be used with caution. I'll do my best to fix problems that are reported to me, but if you don't tell me, I may never know. If you use this code, I hope you will send me your experiences, good or bad!

If you have any suggestions for how this document could be improved, please let me know (`dhinds@hyper.stanford.edu`).

1.2 Copyright notice and disclaimer

Copyright (c) 1998 David A. Hinds

This document may be reproduced or distributed in any form without my prior permission. Modified versions of this document, including translations into other languages, may be freely distributed, provided that they are clearly identified as such, and this copyright is included intact.

This document may be included in commercial distributions without my prior consent. While it is not required, I would like to be informed of such usage. If you intend to incorporate this document in a published work, please contact me to make sure you have the latest available version.

This document is provided ``AS IS'', with no express or implied warranties. Use the information in this document at your own risk.

1.3 What is the latest version, and where can I get it?

The current major release of Card Services is version 3.0, and minor updates or bug fixes are numbered 3.0.1, 3.0.2, and so on.

Source code for the latest version is available from `hyper.stanford.edu` in the `/pub/pcmcia` directory, as `pcmcia-cs-3.0.?.tar.gz`. There will usually be several versions here. I generally only keep the latest minor release for a given major release. New major releases may contain relatively

TOC

untested code, so I also keep the latest version of the previous major release as a relatively stable fallback; the current fallback is 2.9.12. It is up to you to decide which version is more appropriate, but the `CHANGES` file will summarize the most important differences.

`hyper.stanford.edu` is mirrored at `sunsite.unc.edu` (and all sunsite mirror sites) in `/pub/Linux/kernel/pcmcia`.

If you do not feel up to compiling the drivers from scratch, pre-compiled drivers are included with current releases of most of the major Linux distributions, including Slackware, Debian, Red Hat, Caldera, SuSE, and Yggdrasil, among others.

1.4 What systems are supported?

This package should run on almost Intel-based Linux-capable laptop. It also runs on Alpha-based platforms (i.e., the DEC Multia). Work is being done to make the package fully dual-endian, so that it will also support PowerPC-based platforms (i.e., Apple Powerbooks). Most common socket controllers are supported. Card docks for desktop systems should work as long as they use a supported controller, and are plugged directly into the ISA or PCI bus, as opposed to SCSI-to-PCMCIA or IDE-to-PCMCIA adapters. The following controllers are recognized by the supplied socket drivers:

- Cirrus Logic PD6710, PD6720, PD6722, PD6729, PD6730, PD6732, PD6832
- Intel i82365sl B, C, and DF steps, 82092AA
- O2Micro OZ6729, OZ6730, OZ6832, OZ6833, OZ6836, OZ6860
- Omega Micro 82C092G
- Ricoh RF5C296, RF5C396, RL5C465, RL5C466, RL5C475, RL5C476, RL5C478
- SMC 34C90
- Texas Instruments PCI1130, PCI1131, PCI1210, PCI1220, PCI1221, PCI1250A, PCI1251A, PCI1251B, PCI1450
- Toshiba ToPIC95, ToPIC97 (experimental)
- Vadem VG465, VG468, VG469
- VLSI Technologies 82C146, VCF94365
- VIA VT83C469
- Databook DB86082, DB86082A, DB86084, DB86084A, DB86072, DB86082B

Other controllers that are register compatible with the Intel i82365sl will generally work, as well.

Support for 32-bit CardBus cards is still somewhat experimental. Drivers prior to version 3.0 only support 16-bit cards in CardBus sockets. Due to the rapid pace of technological change for laptop hardware, new controllers appear frequently, and there may be delays between when a new model appears on the market, and when driver support becomes available.

Toshiba has made available specifications for their ToPIC95 and ToPIC97 chipsets, however the information is somewhat sparse. Work has been hindered by the fact that much of the documentation apparently exists only in Japanese. Full support for these chipsets is still experimental.

The Motorola 6AHC05GA controller used in some Hyundai laptops is not supported. The custom hc controller in the HP Omnibook 600 is also unsupported.

1.5 What cards are supported?

The current release includes drivers for a variety of ethernet cards, a driver for modem and serial port cards, several SCSI adapter drivers, a driver for ATA/IDE drive cards, and memory card drivers that should support most SRAM cards and some flash cards. The `SUPPORTED.CARDS` file included with each release of Card Services lists all cards that are known to work in at least one actual system.

The likelihood that a card not on the supported list will work depends on the type of card. Essentially all modems should work with the supplied driver. Some network cards may work if they are OEM versions of supported cards. Other types of IO cards (frame buffers, sound cards, etc) will not work until someone writes the appropriate drivers.

1.6 When will my favorite (unsupported) card become supported?

Unfortunately, they usually don't pay me to write device drivers, so if you would like to have a driver for your favorite card, you are probably going to have to do at least some of the work. Ideally, I'd like to work towards a model like the Linux kernel, where I would be responsible mainly for the ``core'' driver code and other authors would contribute and maintain client drivers for specific cards. The `SUPPORTED.CARDS` file mentions some cards for which driver work is currently in progress. I will try to help where I can, but be warned that debugging kernel device drivers by email is not particularly effective.

Manufacturers interested in helping provide Linux support for their products can contact me about consulting arrangements.

1.7 Mailing lists and other information sources

I used to maintain a database and mailing list of Linux PCMCIA users. More recently, I've turned my web page for Linux PCMCIA information into a ``HyperNews'' site, with a set of message lists for Linux PCMCIA issues. There are lists for installation and configuration issues, for different types of cards, and for programming and debugging issues. The Linux PCMCIA information page is at http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html. Users can request email notification of new responses to particular questions, or notification for all new messages in a given category. I hope that this will become a useful repository of information, for questions that go beyond the scope of the HOWTO.

There is a Linux mailing list devoted to laptop issues, the ``linux-laptop'' list. For more information, send a message containing the word ``help'' to `majordomo@vger.rutgers.edu`. To subscribe, send a message containing ``subscribe linux-laptop'' to the same address. This mailing list might be a good forum for discussion of Linux PCMCIA issues.

The Linux Laptop Home Page at http://www.cs.utexas.edu/users/kharker/linux-laptop has links to m sites that have information about configuring specific types of laptops for Linux. There is also a searchable database of system configuration information.

TOC

1.8 Why don't you distribute binaries?

For me, distributing binaries would be a significant hassle. It is complicated because some features can only be selected at compile time, and because the modules are somewhat dependent on having the ``right'' kernel configuration. So, I would probably need to distribute precompiled modules along with matching kernels. Beyond this, the greatest need for precompiled modules is when installing Linux on a clean system. This typically requires setting up drivers so they can be used in the installation process for a particular Linux distribution. Each Linux distribution has its own ideosyncracies, and it is not feasible for me to provide boot and root disks for even just the common combinations of drivers and distributions.

PCMCIA is now a part of many of the major Linux distributions, including Red Hat, Caldera, Slackware, Yggdrasil, Craftworks, and Nascent Technology.

1.9 Why is the package so darned big?

Well, first of all, it isn't actually that large. All the driver modules together take up about 500K of disk space. The utility programs add up to about 70K, and the scripts in `/etc/pcmcia` are about 50K. The core driver modules take up about 55K of system memory. The `cardmgr` daemon will generally be swapped out except when cards are inserted or removed. The total package size is comparable to DOS/Windows Card Services implementations.

Compared to DOS ``point enablers'', this may still seem like a lot of overhead, especially for people that don't plan on using many of the features of PCMCIA, such as power management or hot swapping. Point enablers can be tiny because they generally support only one or a small set of cards, and also generally support a restricted set of host controllers. If someone were to write a genuinely ``generic'' modem enabler, it would end up incorporating much of the functionality of Card Services, to handle cards from different vendors and the full range of host controller variants.

---

2. Compilation and installation

2.1 Prerequisites and kernel setup

Before starting, you should think about whether you really need to compile the PCMCIA package yourself. All common Linux distributions come with pre-compiled driver packages. Generally, you only need to install the drivers from scratch if you need a new feature of the current drivers, or if you've updated and/or reconfigured your kernel in a way that is incompatible with the drivers included with your Linux distribution. While compiling the package is not technically difficult, it does require some general Linux familiarity.

The following things should be installed on your system before you begin:

- A 2.0.*, 2.1.*, or 2.2.* series kernel source tree.
- An appropriate set of module utilities.
- (Optional) the ``XForms'' X11 user interface toolkit.

You need to have a complete linux source tree for your kernel, not just an up-to-date kernel image. The driver modules contain some references to kernel source files. While you may want to build a new kernel to remove unnecessary drivers, installing PCMCIA does not require you to do so.

Current ``stable'' kernel sources and patches are available from ftp://sunsite.unc.edu/pub/Linux/kernel/v2.0, or from ftp://tsx-11.mit.edu/pub/linux/sources/system/v2.0. Development kernels can be found in the corresponding `v2.1` subdirectories. Current module utilities can be found in the same locations.

In the Linux kernel source tree, the `Documentation/Changes` file describes the versions of all sorts of other system components that are required for that kernel release. You may want to check through this and verify that your system is up to date, especially if you have updated your kernel. If you are using a development kernel, be sure that you are using the right combination of shared libraries and module tools.

When configuring your kernel, if you plan on using a PCMCIA ethernet card, you should turn on networking support but turn off the normal Linux network card drivers, including the ``pocket and portable adapters''. The PCMCIA network card drivers are all implemented as loadable modules. Any drivers compiled into your kernel will only waste space.

If you want to use SLIP, PPP, or PLIP, you do need to either configure your kernel with these enabled, or use the loadable module versions of these drivers. There is an unfortunate deficiency in the kernel config process in 1.2.X kernels, in that it is not possible to set configuration options (like SLIP compression) for a loadable module, so it is probably better to just link SLIP into the kernel if you need it.

In order to use a PCMCIA token ring adapter, your kernel should be configured with ``Token Ring driver support'' (`CONFIG_TR`) enabled, though you should leave `CONFIG_IBMTR` off.

TOC

If you want to use a PCMCIA IDE adapter, your kernel should be configured with `CONFIG_BLK_DEV_IDE_PCMCIA` enabled, for 2.0.* through 2.1.7 kernels. Older kernels do not support removeable IDE devices; newer kernels do not require a special configuration setting.

If you will be using a PCMCIA SCSI adapter, then enable `CONFIG_SCSI` when configuring your kernel. Also, enable any top level drivers (SCSI disk, tape, cdrom, generic) that you expect to use. All low-level drivers for particular host adapters should be disabled, as they will just take up space.

If you want to modularize a driver that is needed for a PCMCIA device, you must modify `/etc/pcmcia/config` to specify what modules need to be loaded for what card types. For example, if the serial driver is modularized, then the serial device definition should be:

```
device "serial_cs"
  class "serial" module "misc/serial", "serial_cs"
```

This package includes an X-based card status utility called `cardinfo`. This utility is based on a freely distributed user interface toolkit called the XForms Library. This library is available as a separate package with most Linux distributions. If you would like to build `cardinfo`, you should install XForms and all the normal X header files and libraries before configuring the PCMCIA package.

2.2 Installation

Here is a synopsis of the installation process:

- Unpack pcmcia-cs-3.0.?.tar.gz in /usr/src.
- Run ``make config'' in the new `pcmcia-cs-3.0.?` directory.
- Run ``make all'', then ``make install''.
- Customize the startup script and the option files in `/etc/pcmcia` for your site.

If you plan to install any contributed client drivers not included in the core PCMCIA distribution, unpack each of them in the top-level directory of the PCMCIA source tree. Then follow the normal build instructions. The extra drivers will be compiled and installed automatically.

Running ``make config'' prompts for a few configuration options, and checks out your system to verify that it satisfies all prerequisites for installing PCMCIA support. In most cases, you'll be able to just accept all the default configuration options. Be sure to carefully check the output of this command in case there are problems.

If you are compiling the package for installation on another machine, specify an alternate target directory when prompted by the configure script. This should be an absolute path. All filesf will be installed relative to this directory. You will then be able to `tar` this directory tree and copy to your target machine, and unpack relative to its root directory to install everything in the proper places.

If you are cross compiling on another machine, you may want to specify alternate names for the compiler and linker. This may also be helpful on mixed a.out and ELF systems. The script will also prompt for additional compiler flags for debugging.

Some of the support utilities (`cardctl` and `cardinfo`) can be compiled either in ``safe'' or ``trust'' forms. The ``safe'' forms prevent non-root users from modifying card configurations. The ``trusting'' forms permit ordinary users to issue commands to suspend and resume cards, reset cards, and change the current configuration scheme. The configuration script will ask if you want the utilities compiled as safe or trusting: the default is to be safe.

There are a few kernel configuration options that affect the PCMCIA tools. The configuration script can deduce these from the running kernel (the most common case). Alternatively, if you are compiling for installation on another machine, it can read the configuration from a kernel source tree, or each option can be set interactively.

Running ```make all`'' followed by ```make install`'' will build and then install the kernel modules and utility programs. Kernel modules are installed under `/lib/modules/<version>/pcmcia`. The `cardmgr` and `cardctl` programs are installed in `/sbin`. If `cardinfo` is built, it is installed in `/usr/bin/X11`.

Configuration files will be installed in the `/etc/pcmcia` directory. If you are installing over an older version, your old config scripts will be backed up before being replaced. The saved scripts will be given extensions like `*.~1~`, `*.~2~`, and so on.

If you don't know what kind of host controller your system uses, you can use the `probe` utility in the `cardmgr/` subdirectory to determine this. There are two major types: the Databook TCIC-2 type and the Intel i82365SL-compatible type.

In a few cases, the `probe` command will be unable to determine your controller type automatically. If you have a Halikan NBD 486 system, it has a TCIC-2 controller at an unusual location: you'll need to edit `rc.pcmcia` to load the `tcic` module, and also set the `PCIC_OPTS` parameter to ``tcic_base=0x02c0''.

On some systems using Cirrus controllers, including the NEC Versa M, the BIOS puts the controller in a special suspended state at system startup time. On these systems, the `probe` command will fail to find any known host controller. If this happens, edit `rc.pcmcia` and set `PCIC` to `i82365`, and `PCIC_OPTS` to ``wakeup=1''.

2.3 Startup options

The PCMCIA startup script recognizes several groups of startup options, set via environment variables. Multiple options should be separated by spaces and enclosed in quotes. Placement of startup options depends on the Linux distribution used. They may be placed directly in the startup script, or they may be kept in a separate option file. See the Notes about specific Linux distributions for specifics. The following variables can be set:

PCMCIA

This variable specifies whether PCMCIA support should be started up, or not. If it is set to anything other than ``yes'', then the startup script will be disabled.

PCIC

TOC

This identifies the PC Card Interface Controller driver module. There are two options: ``tcic'' and ``i82365''. Virtually all current controllers are in the ``i82365'' group. This is the only mandatory option setting.

PCIC_OPTS

This specifies options for the PCIC module. Some host controllers have optional features that may or may not be implemented in a particular system. In some cases, it is impossible for the socket driver to detect if these features are implemented. See the corresponding man page for a complete description of the available options.

CORE_OPTS

This specifies options for the `pcmcia_core` module, which implements the core PC Card driver services. See ``man pcmcia_core'' for more information.

CARDMGR_OPTS

This specifies options to be passed to the `cardmgr` daemon. See ``man cardmgr'' for more information.

SCHEME

If set, then the PC Card configuration scheme will be initialized to this at driver startup time. See the Overview of the PCMCIA configuration scripts for a discussion of schemes.

The low level socket drivers, `tcic` and `i82365`, have various bus timing parameters that may need to be adjusted for certain systems with unusual bus clocking. Symptoms of timing problems can include card recognition problems, lock-ups under heavy loads, high error rates, or poor device performance. Only certain host bridges have adjustable timing parameters: check the corresponding man page to see what options are available for your controller. Here is a brief summary:

- Cirrus controllers have numerous configurable timing parameters. The most important seems to be the `cmd_time` flag, which determines the length of PCMCIA bus cycles. Fast 486 systems (i.e., DX4-100) seem to often benefit from increasing this from 6 (the default) to 12 or 16.
- The Cirrus PD6729 PCI controller has the `fast_pci` flag, which should be set if the PCI bus speed is greater than 25 MHz.
- For Vadem VG-468 controllers and Databook TCIC-2 controllers, the `async_clock` flag changes the relative clocking of PCMCIA bus and host bus cycles. Setting this flag adds extra wait states to some operations. However, I have yet to hear of a laptop that needs this.
- The `pcmcia_core` module has the `cis_speed` parameter for changing the memory speed used for accessing a card's Card Information Structure (CIS). On some systems with fast bus clocks, increasing this parameter (i.e., slowing down card accesses) may be beneficial for card recognition problems.
- This is not a timing issue, but if you have more than one ISA-to-PCMCIA controller in your system or extra sockets in a docking station, the `i82365` module should be loaded with the `extra_sockets` parameter set to 1. This should not be necessary for detection of PCI-to-PCMCIA or PCI-to-CardBus bridges.

Here are some timing settings for specific systems:

TOC

- On the ARM Pentium-90 or Midwest Micro Soundbook Plus, use ``freq_bypass=1 cmd_time=8''.
- On a Midwest Micro Soundbook Elite, use ``cmd_time=12''.
- On a Gateway Liberty, try ``cmd_time=16''.
- On a Samsung SENS 810, use ``fast_pci=1''.

2.4 System resource settings

Card Services should automatically avoid allocating IO ports and interrupts already in use by other standard devices. It will also attempt to detect conflicts with unknown devices, but this is not completely reliable. In some cases, you may need to explicitly exclude resources for a device in `/etc/pcmcia/config.opts`.

Here are some resource settings for specific laptop types. View this list with suspicion: it may give useful hints for solving problems, but it is inevitably out of date and certainly contains mistakes. Corrections and additions are welcome.

- On the AMS SoundPro, exclude irq 10.
- On some AMS TravelPro 5300 models, use memory 0xc8000-0xcffff.
- On the BMX 486DX2-66, exclude irq 5, irq 9.
- On the Chicony NB5, use memory 0xda000-0xdffff.
- On the Compaq Presario 1020, exclude port 0x2f8-0x2ff, irq 3, irq 5.
- On the HP Omnibook 4000C, exclude port 0x300-0x30f.
- On the IBM ThinkPad 380, and maybe the 385 and 600 series, exclude port 0x230-0x233, and irq 5.
- On the IBM ThinkPad 600, exclude port 0x2f8-0x2ff.
- On the Micron Millenia Transport, exclude irq 5, irq 9.
- On the NEC Versa M, exclude irq 9, port 0x2e0-2ff.
- On the NEC Versa P/75, exclude irq 5, irq 9.
- On the NEC Versa S, exclude irq 9, irq 12.
- On the NEC Versa 6000 series, exclude port 0x2f8-0x33f, irq 9, irq 10.
- On the ProStar 9200, Altima Virage, and Acquiline Hurricane DX4-100, exclude irq 5, port 0x330-0x35f. Maybe use memory 0xd8000-0xdffff.
- On the Siemens Nixdorf SIMATIC PG 720C, use memory 0xc0000-0xcffff, port 0x300-0x3bf.
- On the TI TravelMate 5000, use memory 0xd4000-0xdffff.
- On the Toshiba T4900 CT, exclude irq 5, port 0x2e0-0x2e8, port 0x330-0x338.
- On the Twinhead 5100, HP 4000, Sharp PC-8700 and PC-8900, exclude irq 9 (sound), irq 12.
- On an MPC 800 Series, exclude irq 5, port 0x300-0x30f for the CD-ROM.

2.5 Notes about specific Linux distributions

TOC

This section is incomplete. Corrections and additions are welcome.

Debian

Debian uses a System V boot script arrangement. The PCMCIA startup script is installed as `/etc/init.d/pcmcia`, and startup options are specified in `/etc/pcmcia.conf`. Debian's syslog configuration will place kernel messages in `/var/log/messages` and `cardmgr` messages in `/var/log/daemon.log`.

Debian distributes the PCMCIA system in two packages: the ``pcmcia-cs'' package contains `cardmgr` and other tools, man pages, and configuration scripts; and the ``pcmcia-modules'' package contains the kernel driver modules.

Red Hat, and Caldera Open Desktop

These distributions use a System V boot script organization. The PCMCIA startup script is installed as `/etc/rc.d/init.d/pcmcia`, and boot options are kept in `/etc/sysconfig/pcmcia`. Beware that installing the Red Hat package may install a default boot option file that has PCMCIA disabled. To enable PCMCIA, the ``PCMCIA'' variable should be set to ``yes''. Red Hat's default syslogd configuration will record all interesting messages in `/var/log/messages`.

Red Hat's PCMCIA package contains a replacement for the network setup script, `/etc/pcmcia/network`, which meshes with the Red Hat network control panel. This is convenient for the case where just one network adapter is used, with one set of network parameters, but does not have the full flexibility of the regular PCMCIA network script. Compiling and installing a clean PCMCIA source distribution will overwrite the network script, breaking the link to the network control panel. If you prefer the Red Hat script, either use only Red Hat RPM's, or create `/etc/pcmcia/network.opts` with the following contents:

```
if [ -f /etc/sysconfig/network-scripts/ifcfg-eth0 ] ; then
    start_fn () {
        /sbin/ifup $1
    }
    stop_fn () {
        /sbin/ifdown $1
    }
fi
```

Red Hat bundles their slightly modified PCMCIA source distribution in their kernel SRPM, rather than as a separate package.

TOC

Slackware

Slackware uses a BSD boot script arrangement. The PCMCIA startup script is installed as `/etc/rc.d/rc.pcmcia`, and boot options are specified in `rc.pcmcia` itself. The PCMCIA startup script is invoked from `/etc/rc.d/rc.S`.

SuSE

SuSE uses a System V init script arrangement, with init scripts stored under `/sbin/init.d`. The PCMCIA startup script is installed as `/sbin/init.d/pcmcia`, and startup options are kept in `/etc/rc.config`.

---

3. Resolving installation and configuration problems

This section describes some of the most common failure modes for the PCMCIA subsystem. Try to match your symptoms against the examples. This section only describes general failures that are not specific to a particular client driver or type of card.

Before trying to diagnose a problem, you have to know where your system log is kept (see Notes about specific Linux distributions). You should also be familiar with basic diagnostic tools like `dmesg` and `lsmod`. Also, be aware that most driver components (including all the kernel modules) have their own individual man pages.

It is nearly impossible to debug driver problems encountered when attempting to install Linux via a PCMCIA device. Even if you can identify the problem based on its symptoms, installation disks are difficult to modify, especially without access to a running Linux system. Customization of installation disks is completely dependent on the choice of Linux distribution, and is beyond the scope of this document. In general, the best course of action is to install Linux using some other means, obtain the latest drivers, and then debug the problem if it persists.

3.1 Base PCMCIA kernel modules do not load

Symptoms:

- Kernel version mismatch errors are reported when the PCMCIA startup script runs.
- After startup, `lsmod` does not show any PCMCIA modules.
- `cardmgr` reports ``no pcmcia driver in /proc/devices'' in the system log.

Kernel modules contain version information that is checked against the current kernel when a module is loaded. The type of checking depends on the setting of the `CONFIG_MODVERSIONS` kernel option. If this is false, then the kernel version number is compiled into each module, and `insmod` checks this for a match with the running kernel. If `CONFIG_MODVERSIONS` is true, then each symbol exported by the kernel is given a sort of checksum. These codes are all compared against the corresponding codes compiled into a module. The intent was for this to make modules less version-dependent, because the checksums would only change if a kernel interface changed, and would generally stay the same across minor kernel updates. In practice, the checksums have turned out to be even more restrictive, because many kernel interfaces depend on compile-time kernel option settings. Also, the checksums turned out to be an excessively pessimistic judge of compatibility.

The practical upshot of this is that kernel modules are closely tied to both the kernel version, and the setting of many kernel configuration options. Generally, a set of modules compiled for one 2.0.31 kernel will not load against some other 2.0.31 kernel unless special care is taken to ensure that the two were built with similar configurations. This makes distribution of precompiled kernel modules a tricky

business.

You have several options:

- If you obtained precompiled drivers as part of a Linux distribution, verify that you are using an unmodified kernel as supplied with that distribution. If you intend to use precompiled modules, you generally must stick with the corresponding kernel.
- If you have reconfigured or upgraded your kernel, you will probably need to compile and install the PCMCIA package from scratch. This is easily done if you already have the kernel source tree installed. See Compilation and installation for detailed instructions.
- In some cases, incompatibilities in other system components can prevent correct loading of kernel modules. If you have upgraded your own kernel, pay attention to the ``minimal requirements'' for module utilities and binutils listed in the `Documentation/Changes` file in the kernel source code tree.

3.2 Some client driver modules do not load

Symptoms:

- The base modules (`pcmcia_core`, `ds`, `i82365`) load correctly.
- Inserting a card gives a high beep + low beep pattern.
- `cardmgr` reports version mismatch errors in the system log.

Some of the driver modules require kernel services that may or may not be present, depending on kernel configuration. For instance, the SCSI card drivers require that the kernel be configured with SCSI support, and the network drivers require a networking kernel. If a kernel lacks a necessary feature, `insmod` may report undefined symbols and refuse to load a particular module. Note that `insmod` error messages do not distinguish between version mismatch errors and missing symbol errors.

Specifically:

- The serial client driver `serial_cs` requires the kernel serial driver to be enabled with `CONFIG_SERIAL`. This driver may be built as a module.
- Support for multiport serial cards or multifunction cards that include serial or modem devices requires `CONFIG_SERIAL_SHARE_IRQ` to be enabled.
- The SCSI client drivers require that `CONFIG_SCSI` be enabled, along with the appropriate top level driver options (`CONFIG_BLK_DEV_SD`, `CONFIG_BLK_DEV_SR`, etc for 2.1 kernels). These may be built as modules.
- The network client drivers require that `CONFIG_INET` is enabled. Kernel networking support cannot be compiled as a module.
- The token-ring client requires that the kernel be compiled with `CONFIG_TR` enabled.

There are two ways to proceed:

- Rebuild your kernel with the necessary features enabled.
- If the features have been compiled as modules, then modify `/etc/pcmcia/config` to preload these modules.

TOC

The `/etc/pcmcia/config` file can specify that additional modules need to be loaded for a part client. For example, for the serial driver, one would use:

```
device "serial_cs"
  class "serial" module "misc/serial", "serial_cs"
```

Module paths are specified relative to the top-level module directory for the current kernel version; if no relative path is given, then the path defaults to the `pcmcia` subdirectory.

3.3 Interrupt scan failures

Symptoms:

- The system locks up when the PCMCIA drivers are loaded, even with no cards present.
- The system log shows a successful host controller probe just before the lock-up, but does not show interrupt probe results.

After identifying the host controller type, the socket driver probes for free interrupts. The probe involves programming the controller for each apparently free interrupt, then generating a ``soft'' interrupt, to see if the interrupt can be detected correctly. In some cases, probing a particular interrupt can interfere with another system device.

The reason for the probe is to identify interrupts which appear to be free (i.e., are not reserved by any other Linux device driver), yet are either not physically wired to the host controller, or are connected to another device that does not have a driver.

In the system log, a successful probe might look like:

```
Intel PCIC probe:
  TI 1130 CardBus at mem 0x10211000, 2 sockets
  ...
  ISA irqs (scanned) = 5,7,9,10 status change on irq 10
```

There are two ways to proceed:

- The interrupt probe can be restricted to a list of interrupts using the `irq_list` parameter for the socket drivers. For example, ```irq_list=5,9,10`'' would limit the scan to three interrupts. All PCMCIA devices will be restricted to using these interrupts (assuming they pass the probe). You may need to use trial and error to find out which interrupts can be safely probed.
- The interrupt probe can be disabled entirely by loading the socket driver with the ``do_scan=0'' option. In this case, a default interrupt list will be used, which avoids interrupts already allocated for other devices.

In either case, the probe options can be specified using the `PCIC_OPTS` definition in the PCMCIA startup script, for example:

```
PCIC_OPTS="irq_list=5,9,10"
```

TOC

It should be noted that `/proc/interrupts` is completely useless when it comes to diagnosing interrupt probe problems. The probe is sensible enough to never attempt to use an interrupt that is already in use by another Linux driver. So, the PCMCIA drivers are already using all the information in `/proc/interrupts`. Depending on system design, an inactive device can still occupy an interrupt and cause trouble if it is probed for PCMCIA.

3.4 IO port scan failures

Symptoms:

- The system locks up when `cardmgr` is first started, even with no cards present.
- The system log shows a successful host controller probe, including interrupt probe results, but does not show IO probe results.
- In some cases, the IO probe will succeed, but report large numbers of random exclusions.

When `cardmgr` processes IO port ranges listed in `/etc/pcmcia/config.opts`, the kernel probes these ranges to detect latent devices that occupy IO space but are not associated with a Linux driver. The probe is read-only, but in rare cases, reading from a device may interfere with an important system function, resulting in a lock-up.

Your system user's guide may include a map of system devices, showing their IO and memory ranges. These can be explicitly excluded in `config.opts`.

Alternatively, if the probe is unreliable on your system, it can be disabled by setting `CORE_OPTS` to ``probe_io=0''. In this case, you should be very careful to specify only genuinely available ranges of ports in `config.opts`, instead of using the default settings.

3.5 Memory probe failures

Symptoms:

- The core drivers load correctly when no cards are present, with no errors in the system log.
- The system freezes and/or reboots as soon as any card is inserted, before any beeps are heard.

Or alternately:

- All card insertions generate a high beep followed by a low beep.
- All cards are identified as ``anonymous memory cards''.
- The system log reports that various memory ranges have been excluded.

The core modules perform a memory scan at the time of first 16-bit card insertion. This scan can potentially interfere with other memory mapped devices. Also, pre-3.0.0 driver packages perform a more aggressive scan than more recent drivers. The memory window is defined in `/etc/pcmcia/config.opts`. The default window is large, so it may help to restrict the scan to a narrower range. Reasonable ranges to try include 0xd0000-0xdffff, 0xc0000-0xcffff, 0xc8000-0xcffff, or 0xd8000-0xdffff.

TOC

If you have DOS or Windows PCMCIA drivers, you may be able to deduce what memory region the drivers use. Note that DOS memory addresses are often specified in ``segment'' form, which leaves out the final hex digit (so an absolute address of 0xd0000 might be given as 0xd000). Be sure to add the extra digit back when making changes to `config.opts`.

In unusual cases, a memory probe failure can indicate a timing register setup problem with the host controller. See the Startup options section for information about dealing with common timing problems.

- `cs: warning: no high memory space available!`

CardBus bridges can allocate memory windows outside of the 640KB-1MB ``memory hole'' in the ISA bus architecture. It is generally a good idea to configure CardBus bridges to use high memory windows, because these are unlikely to conflict with other devices. Also, CardBus cards may require large memory windows, which may be difficult or impossible to fit into low memory. Card Services will preferentially allocate windows in high memory for CardBus bridges, if both low and high memory windows are defined in `config.opts`. The default `config.opts` now includes a high memory window of 0xa0000000-0xa0ffffff. If you have a CardBus bridge and have upgraded from an older PCMCIA driver release, add this memory window if it is not already defined.

In some cases, the default high memory window is not usable. On some IBM Thinkpad models, a window of 0x60000000-0x60ffffff will work in place of the default window.

3.6 Failure to detect card insertions and removals

Symptoms:

- Cards are detected and configured properly if present at boot time.
- The drivers do not respond to insertion and removal events, either by recording events in the system log, or by beeping.

In most cases, the socket driver (`i82365` or `tcic`) will automatically probe and select an appropriate interrupt to signal card status changes. The automatic interrupt probe doesn't work on some Intel-compatible controllers, including Cirrus chips and the chips used in some IBM ThinkPads. If a device is inactive at probe time, its interrupt may also appear to be available. In these cases, the socket driver may pick an interrupt that is used by another device.

With the `i82365` and `tcic` drivers, the `irq_list` option can be used to limit the interrupts that will be tested. This list limits the set of interrupts that can be used by PCMCIA cards as well as for monitoring card status changes. The `cs_irq` option can also be used to explicitly set the interrupt to be used for monitoring card status changes.

If you can't find an interrupt number that works, there is also a polled status mode: both `i82365` and `tcic` will accept a `poll_interval=100` option, to poll for card status changes once per second. This option should also be used if your system has a shortage of interrupts available for use by PCMCIA cards. Especially for systems with more than one host controller, there is little point in dedicating interrupts for monitoring card status changes.

All these options should be set in the `PCIC_OPTS=` line in either `/etc/rc.d/rc.pcmcia` or `/etc/sysconfig/pcmcia`, depending on your site setup.

TOC

3.7 System resource starvation

Symptoms:

- When a card is inserted, it is identified correctly but cannot be configured (high/low beep pattern).
- One of the following messages will appear in the system log:

```
RequestIO: Resource in use
RequestIRQ: Resource in use
RequestWindow: Resource in use
GetNextTuple: No more items
could not allocate nn IO ports for CardBus socket n
could not allocate nnK memory for CardBus socket n
could not allocate interrupt for CardBus socket n
```

Interrupt starvation often indicates a problem with the interrupt probe (see Interrupt probe failures). In some cases, the probe will seem to work, but only report one or two available interrupts. Check your system log to see if the scan results look sensible. Disabling the probe and selecting interrupts manually should help.

If the interrupt probe is not working properly, the socket driver may allocate an interrupt for monitoring card insertions, even when interrupts are too scarce for this to be a good idea. In that case, you can switch the controller to polled mode by setting `PCIC_OPTS` to ``poll_interval=100'. Or, if you have a CardBus controller, try ``pci_csc=1", which selects a PCI interrupt (if available) for card status changes.

IO port starvation is fairly uncommon, but sometimes happens with cards that require large, contiguous, aligned regions of IO port space, or that only recognize a few specific IO port positions. The default IO port ranges in `/etc/pcmcia/config.opts` are normally sufficient, but may be extended. In rare cases, starvation may indicate that the IO port probe failed (see IO port scan failures).

Memory starvation is also uncommon with the default memory window settings in `config.opts`. CardBus cards may require larger memory regions than typical 16-bit cards. Since CardBus memory windows can be mapped anywhere in the host's PCI address space (rather than just in the 640K-1MB ``hole" in PC systems), it is helpful to specify large memory windows in high memory, such as 0xa0000000-0xa0ffffff.

3.8 Resource conflict between two cards

Symptoms:

- Two cards each work fine when used separately.
- When both cards are inserted, only one works.

This usually indicates a resource conflict with a system device that Linux does not know about. PCMCIA devices are dynamically configured, so, for example, interrupts are allocated as needed, rather than

specifically assigned to particular cards or sockets. Given a list of resources that appear to be availab cards are assigned resources in the order they are configured. In this case, the card configured last is being assigned a resource that in fact is not free.

Check the system log to see what resources are used by the non-working card. Exclude these in `/etc/pcmcia/config.opts`, and restart the `cardmgr` daemon to reload the resource database.

3.9 Device configuration does not complete

Symptoms:

- When a card is inserted, exactly one high beep is heard.
- Subsequent card insertions and removals may be ignored.

This indicates that the card was identified successfully, however, `cardmgr` has been unable to complete the configuration process for some reason. The most likely reason is that a step in the card setup script has blocked. A good example would be the network script blocking if a network card is inserted with no actual network hookup present.

To pinpoint the problem, you can manually run a setup script to see where it is blocking. The scripts are in the `/etc/pcmcia` directory. They take two parameters: a device name, and an action. The `cardmgr` daemon records the configuration commands in the system log. For example, if the system log shows that the command ``./network start eth0" was the last command executed by `cardmgr`, the following command would trace the script:

```
sh -x /etc/pcmcia/network start eth0
```

4. Usage and features

4.1 Tools for configuring and monitoring PCMCIA devices

If the modules are all loaded correctly, the output of the `lsmod` command should look like the following, when no cards are inserted:

```
Module                  Size  Used by
ds                      5640   2
i82365                 15452   2
pcmcia_core            30012   3  [ds i82365]
```

The system log should also include output from the socket driver describing the host controller(s) found and the number of sockets detected.

The cardmgr configuration daemon

The `cardmgr` daemon is responsible for monitoring PCMCIA sockets, loading client drivers when needed, and running user-level scripts in response to card insertions and removals. It records its actions in the system log, but also uses beeps to signal card status changes. The tones of the beeps indicate success or failure of particular configuration steps. Two high beeps indicate that a card was identified and configured successfully. A high beep followed by a low beep indicates that a card was identified, but could not be configured for some reason. One low beep indicates that a card could not be identified.

`Cardmgr` records device information for each socket in `/var/run/stab`. Here is a sample `/var/run/stab` listing:

```
Socket 0: Adaptec APA-1460 SlimSCSI
0       scsi    aha152x_cs      0       sda     8       0
0       scsi    aha152x_cs      1       scd0    11      0
Socket 1: Serial or Modem Card
1       serial  serial_cs       0       ttyS1   5       65
```

For the lines describing devices, the first field is the socket, the second is the device class, the third is the driver name, the fourth is used to number multiple devices associated with the same driver, the fifth is the device name, and the final two fields are the major and minor device numbers for this device (if applicable).

The `cardmgr` daemon configures cards based on a database of known card types kept in `/etc/pcmcia/config`. This file describes the various client drivers, then describes how to identify

TOC

various cards, and which driver(s) belong with which cards. The format of this file is described in the `pcmcia(5)` man page.

The cardctl and cardinfo utilities

The `cardctl` command can be used to check the status of a socket, or to see how it is configured. It can also be used to alter the configuration status of a card. Here is an example of the output of the ``cardctl config'' command:

```
Socket 0:
  not configured
Socket 1:
  Vcc = 5.0, Vpp1 = 0.0, Vpp2 = 0.0
  Card type is memory and I/O
  IRQ 3 is dynamic shared, level mode, enabled
  Speaker output is enabled
  Function 0:
    Config register base = 0x0800
      Option = 0x63, status = 0x08
    I/O window 1: 0x0280 to 0x02bf, auto sized
    I/O window 2: 0x02f8 to 0x02ff, 8 bit
```

Or ``cardctl ident'', to get card identification information:

```
Socket 0:
  no product info available
Socket 1:
  product info: "LINKSYS", "PCMLM336", "A", "0040052D6400"
  manfid: 0x0143, 0xc0ab
  function: 0 (multifunction)
```

The ``cardctl suspend'' and ``cardctl resume'' commands can be used to shut down a card without unloading its associated drivers. The ``cardctl reset'' command attempts to reset and reconfigure a card. ``cardctl insert'' and ``cardctl eject'' mimic the actions performed when a card is physically inserted or ejected, including loading or unloading drivers, and configuring or shutting down devices.

If you are running X, the `cardinfo` utility produces a graphical display showing the current status of all PCMCIA sockets, similar in content to ``cardctl config''. It also provides a graphical interface to most other `cardctl` functions.

Inserting and ejecting cards

In theory, you can insert and remove PCMCIA cards at any time. However, it is a good idea not to eject a card that is currently being used by an application program. Kernels older than 1.1.77 would often lock up when serial/modem cards were ejected, but this should be fixed now.

TOC

Card Services and Advanced Power Management

Card Services can be compiled with support for APM (Advanced Power Management) if you've configured your kernel with APM support. The APM kernel driver is maintained by Stephen Rothwell (Stephen.Rothwell@canb.auug.org.au). The `apmd` daemon is maintained by Avery Pennarun (apenwarr@worldvisions.ca), with more information available at http://www.worldvisions.ca/~apenwarr/apmd/. The PCMCIA modules will automatically be configured for APM if a compatible version is detected on your system.

Whether or not APM is configured, you can use ``cardctl suspend'' before suspending your laptop, and ``cardctl resume'' after resuming, to cleanly shut down and restart your PCMCIA cards. This will not work with a modem that is in use, because the serial driver isn't able to save and restore the modem operating parameters.

APM seems to be unstable on some systems. If you experience trouble with APM and PCMCIA on your system, try to narrow down the problem to one package or the other before reporting a bug.

Some drivers, notably the PCMCIA SCSI drivers, cannot recover from a suspend/resume cycle. When using a PCMCIA SCSI card, always use ``cardctl eject'' prior to suspending the system.

Shutting down the PCMCIA system

To unload the entire PCMCIA package, invoke `rc.pcmcia` with:

```
/etc/rc.d/rc.pcmcia stop
```

This script will take several seconds to run, to give all client drivers time to shut down gracefully. If a device is currently in use, the shutdown will be incomplete, and some kernel modules may not be unloaded. To avoid this, use ``cardctl eject'' to shut down all sockets before invoking `rc.pcmcia`. The exit status of the `cardctl` command will indicate if any sockets could not be shut down.

4.2 Overview of the PCMCIA configuration scripts

Each PCMCIA device has an associated ``class'' that describes how it should be configured and managed. Classes are associated with device drivers in `/etc/pcmcia/config`. There are currently five IO device classes (network, SCSI, cdrom, fixed disk, and serial) and two memory device classes (memory and FTL). For each class, there are two scripts in `/etc/pcmcia`: a main configuration script (i.e., `/etc/pcmcia/scsi` for SCSI devices), and an options script (i.e., `/etc/pcmcia/scsi.opts`). The main script for a device will be invoked to configure that device when a card is inserted, and to shut down the device when the card is removed. For cards with several associated devices, the script will be invoked for each device.

The config scripts start by extracting some information about a device from `/var/run/stab`. Each script constructs a ``device address'', that uniquely describes the device it has been asked to configure, in the `ADDRESS` shell variable. This is passed to the `*.opts` script, which should return information about

how a device at this address should be configured. For some devices, the device address is just the sc number. For others, it includes extra information that may be useful in deciding how to configure the device. For example, network devices pass their hardware ethernet address as part of the device address, so the `network.opts` script could use this to select from several different configurations.

The first part of all device addresses is the current PCMCIA ``scheme''. This parameter is used to support multiple sets of device configurations based on a single external user-specified variable. One use of schemes would be to have a ``home'' scheme, and a ``work'' scheme, which would include different sets of network configuration parameters. The current scheme is selected using the ```cardctl scheme`'' command. The default if no scheme is set is ``default''.

As a general rule, when configuring Linux for a laptop, PCMCIA devices should only be configured from the PCMCIA device scripts. Do not try to configure a PCMCIA device the same way you would configure a permanently attached device. However, some Linux distributions provide PCMCIA packages that are hooked into those distributions' own device configuration tools. In that case, some of the following sections may not apply; ideally, this will be documented by the distribution maintainers.

4.3 PCMCIA network adapters

Linux ethernet-type network interfaces normally have names like `eth0`, `eth1`, and so on. Token-ring adapters are handled similarly, however they are named `tr0`, `tr1`, and so on. The `ifconfig` command is used to view or modify the state of a network interface. A peculiarity of Linux is that network interfaces do not have corresponding device files under `/dev`, so do not be surprised when you do not find them.

When an ethernet card is detected, it will be assigned the first free interface name, which will normally be `eth0`. `Cardmgr` will run the `/etc/pcmcia/network` script to configure the interface, which normally reads network settings from `/etc/pcmcia/network.opts`. The `network` and `network.opts` scripts will be executed only when your ethernet card is actually present. If your system has an automatic network configuration facility, it may or may not be PCMCIA-aware. Consult the documentation of your Linux distribution and the Notes about specific Linux distributions to determine if PCMCIA network devices should be configured with the automatic tools, or by editing `network.opts`.

The device address passed to `network.opts` consists of four comma-separated fields: the scheme, the socket number, the device instance, and the card's hardware ethernet address. The device instance is used to number devices for cards that have several network interfaces, so it will usually be 0. If you have several network cards used for different purposes, one option would be to configure the cards based on socket position, as in:

```
case "$ADDRESS" in
*,0,*,*)
    # definitions for network card in socket 0
    ;;
*,1,*,*)
    # definitions for network card in socket 1
```

```
    ;;
esac
```

Alternatively, they could be configured using their hardware addresses, as in:

```
case "$ADDRESS" in
*,*,*,00:80:C8:76:00:B1)
    # definitions for a D-Link card
    ;;
*,*,*,08:00:5A:44:80:01)
    # definitions for an IBM card
esac
```

Network device parameters

The following parameters can be defined in `network.opts`:

`IF_PORT`

Specifies the ethernet transceiver type, for cards that do not autodetect. See ``man ifport'' for transceiver names.

`BOOTP`

A boolean (y/n) value: indicates if the host's IP address and routing information should be obtained using the BOOTP protocol.

`DHCP`

A boolean (y/n) value: indicates if the host's IP address and routing information should be obtained from a DHCP server.

`IPADDR`

The IP address for this interface.

`NETMASK`, `BROADCAST`, `NETWORK`

Basic network parameters: see the networking HOWTO for more information.

`GATEWAY`

The IP address of a gateway for this host's subnet. Packets with destinations outside this subnet will be routed to this gateway.

`DOMAIN`

The local network domain name for this host, to be used in creating `/etc/resolv.conf`.

`SEARCH`

A search list for host name lookup, to be added to `/etc/resolv.conf`. `DOMAIN` and `SEARCH` are mutually exclusive: see ``man resolver'' for more information.

`DNS_1`, `DNS_2`, `DNS_3`

TOC

Host names or IP addresses for nameservers for this interface, to be added to `/etc/resolv.conf`

MOUNTS

A list of NFS mount points to be mounted for this interface.

IPX_FRAME, IPX_NETNUM

For IPX networks: the frame type and network number, passed to the `ipx_interface` command.

For example:

```
case "$ADDRESS" in
*,*,*,*)
    IF_PORT="10base2"
    BOOTP="n"
    IPADDR="10.0.0.1"
    NETMASK="255.255.255.0"
    NETWORK="10.0.0.0"
    BROADCAST="10.0.0.255"
    GATEWAY="10.0.0.1"
    DOMAIN="domain.org"
    DNS_1="dns1.domain.org"
    ;;
esac
```

To automatically mount and unmount NFS filesystems, first add all these filesystems to `/etc/fstab`, but include `noauto` in the mount options. In `network.opts`, list the filesystem mount points in the `MOUNTS` variable. It is especially important to use either `cardctl` or `cardinfo` to shut down a network card when NFS mounts are active. It is not possible to cleanly unmount NFS filesystems if a network card is simply ejected without warning.

In addition to the usual network configuration parameters, the `network.opts` script can specify extra actions to be taken after an interface is configured, or before an interface is shut down. If `network.opts` defines a shell function called `start_fn`, it will be invoked by the network script after the interface is configured, and the interface name will be passed to the function as its first (and only) argument. Similarly, if it is defined, `stop_fn` will be invoked before shutting down an interface.

The transceiver type can be selected using the `IF_PORT` setting. This can either be a numeric value as in previous PCMCIA releases, or a keyword identifying the transceiver type. All the network drivers default to either autodetect the interface if possible, or 10baseT otherwise. The `ifport` command can be used to check or set the current transceiver type. For example:

```
# ifport eth0 10base2
#
# ifport eth0
eth0    2 (10base2)
```

The current (3.0.10 or later) 3c589 driver should quickly autodetect transceiver changes at any time. Earlier releases of the 3c589 driver had a somewhat slow and flaky transceiver autodetection algorithm. For these releases, the appropriate network cable should be connected to the card when the card is configured, or you can force autodetection with:

```
ifconfig eth0 down up
```

Comments about specific cards

- With IBM CCAE and Socket EA cards, the transceiver type (10base2, 10baseT, AUI) needs to be set when the network device is configured. Make sure that the transceiver type reported in the system log matches your connection.
- The drivers for SMC, Megahertz, Ositech, and 3Com cards should autodetect the attached network type (10base2 or 10baseT). Setting the transceiver type when the driver is loaded serves to define the card's ``first guess''.
- The Farallon EtherWave is actually based on the 3Com 3c589, with a special transceiver. Though the EtherWave uses 10baseT-style connections, its transceiver requires that the 3c589 be configured in 10base2 mode.
- If you have trouble with an IBM CCAE, NE4100, Thomas Conrad, or Kingston adapter, try increasing the memory access time with the `mem_speed=#` option to the `pcnet_cs` module. An example of how to do this is given in the standard `config.opts` file. Try speeds of up to 1000 (in nanoseconds).
- For the New Media Ethernet adapter, on some systems, it may be necessary to increase the IO port access time with the `io_speed=#` option when the `pcmcia_core` module is loaded. Edit `CORE_OPTS` in the startup script to set this option.
- The multicast support in the New Media Ethernet driver is incomplete. The latest driver will function with multicast kernels, but will ignore multicast packets. Promiscuous mode should work properly.
- The driver used by the IBM and 3Com token ring adapters seems to behave very badly if the cards are not connected to a ring when they get initialized. Always connect these cards to the net before they are powered up. If `ifconfig` reports the hardware address as all 0's, this is likely to be due to a memory window configuration problem.
- Some Linksys, D-Link, and IC-Card 10baseT/10base2 cards have a unique way of selecting the transceiver type that isn't handled by the Linux drivers. One workaround is to boot DOS and use the vendor-supplied utility to select the transceiver, then warm boot Linux. Alternatively, a Linux utility to perform this function is available at ftp://hyper.stanford.edu/pub/pcmcia/extras/dlport.c.
- For WaveLAN wireless network adapters, Jean Tourrilhes (`jt@hplb.hpl.hp.com`) has put together a wireless HOWTO at http://www-uk.hpl.hp.com/people/jt/Linux/Wavelan.html.

Diagnosing problems with network adapters

- Is your card recognized as an ethernet card? Check the system log and make sure that `cardmgr` identifies the card correctly and starts up one of the network drivers. If it doesn't, your card might still be usable if it is compatible with a supported card. This will be most easily done if the card claims to be ``NE2000 compatible''.
- Is the card configured properly? If you are using a supported card, and it was recognized by `cardmgr`, but still doesn't work, there might be an interrupt or port conflict with another device. Find out what resources the card is using (from the system log), and try excluding these in `/etc/pcmcia/config.opts` to force the card to use something different.
- If your card seems to be configured properly, but sometimes locks up, particularly under high load, you may need to try changing your socket driver timing parameters. See the Startup options section for more information.
- If you get ``network unreachable'' messages when you try to access the network, then the routing information specified in `/etc/pcmcia/network.opts` is incorrect. This message is an absolutely foolproof indication of a routing error. On the other hand, mis-configured cards will usually fail silently.
- To diagnose problems in `/etc/pcmcia/network.opts`, start by trying to ping other systems on the same subnet using their IP addresses. Then try to ping your gateway, and then machines on other subnets. Ping machines by name only after trying these simpler tests.
- Make sure your problem is really a PCMCIA one. It may help to see see if the card works under DOS with the vendor's drivers. Double check your modifications to the `/etc/pcmcia/network.opts` script. Make sure your drop cable, ``T'' jack, terminator, etc are working.

4.4 PCMCIA serial and modem devices

Linux serial devices are accessed via the `/dev/cua*` and `/dev/ttyS*` special device files. The `ttyS*` devices are for incoming connections, such as directly connected terminals. The `cua*` devices are for outgoing connections, such as modems. Each physical serial port has both a `ttyS` and a `cua` device file: it is up to you to pick the appropriate device for your application. The configuration of a serial device can be examined and modified with the `setserial` command.

When a serial or modem card is detected, it will be assigned to the first available serial device slot. This will usually be `/dev/ttyS1` (`cua1`) or `/dev/ttyS2` (`cua2`), depending on the number of built-in serial ports. The `ttyS*` device is the one reported in `/var/run/stab`. The default serial device option script, `/etc/pcmcia/serial.opts`, will link the corresponding `cua*` device file to `/dev/modem` as a convenience.

Do not try to use `/etc/rc.d/rc.serial` to configure a PCMCIA modem. This script should only be used to configure non-removable devices. Modify `/etc/pcmcia/serial.opts` if you want to do anything special to set up your modem. Also, do not try to change the IO port and interrupt settings of a serial device using `setserial`. This would tell the serial driver to look for the device in a different place, but would not change how the card's hardware is actually configured. The serial configuration

TOC

script allows you to specify other `setserial` options, as well as whether a line should be added to `/etc/inittab` for this port.

The device address passed to `serial.opts` has three comma-separated fields: the first is the scheme, the second is the socket number, and the third is the device instance. The device instance may take several values for cards that support multiple serial ports, but for single-port cards, it will always be 0. If you commonly use more than one modem, you may want to specify different settings based on socket position, as in:

```
case "$ADDRESS" in
*,0,*)
    # Options for modem in socket 0
    LINK=/dev/modem0
    ;;
*,1,*)
    # Options for modem in socket 1
    LINK=/dev/modem1
    ;;
esac
```

If a PCMCIA modem is already configured when Linux boots, it may be incorrectly identified as an ordinary built-in serial port. This is harmless, however, when the PCMCIA drivers take control of the modem, it will be assigned a different device slot. It is best to either parse `/var/run/stab` or use `/dev/modem`, rather than expecting a PCMCIA modem to always have the same device assignment.

If you configure your kernel to load the basic Linux serial port driver as a module, you must edit `/etc/pcmcia/config` to indicate that this module must be loaded. Edit the serial device entry to read:

```
device "serial_cs"
  class "serial" module "misc/serial", "serial_cs"
```

Serial device parameters

The following parameters can be defined in `serial.opts`:

`LINK`

Specifies a path for a symbolic link to be created to the ``dialout'' or `/dev/cua*` device.

`SERIAL_OPTS`

Specifies options to be passed to the `setserial` command.

`INITTAB`

If specified, this will be used to construct an `inittab` entry for the device.

For example:

```
case "$ADDRESS" in
*,*,*,*)
    LINK="/dev/modem"
    SERIAL_OPTS=""
    INITTAB="/sbin/getty"
```

Diagnosing problems with serial devices

- Is your card recognized as a modem? Check the system log and make sure that `cardmgr` identifies the card correctly and starts up the `serial_cs` driver. If it doesn't, you may need to add a new entry to your `/etc/pcmcia/config` file so that it will be identified properly. See the Configuring unrecognized cards section for details.
- Is the modem configured successfully by serial_cs? Again, check the system log and look for messages from the serial_cs driver. If you see ``register_serial() failed'', you may have an I/O port conflict with another device. Another tip-off of a conflict is if the device is reported to be an 8250; most modern modems should be identified as 16550A UART's. If you think you're seeing a port conflict, edit `/etc/pcmcia/config.opts` and exclude the port range that was allocated for the modem.
- Is there an interrupt conflict? If the system log looks good, but the modem just doesn't seem to work, try using `setserial` to change the irq to 0, and see if the modem works. This causes the serial driver to use a slower polled mode instead of using interrupts. If this seems to fix the problem, it is likely that some other device in your system is using the interrupt selected by serial_cs. You should add a line to `/etc/pcmcia/config.opts` to exclude this interrupt.
- If the modem seems to work only very, very slowly, this is an almost certain indicator of an interrupt conflict.
- Make sure your problem is really a PCMCIA one. It may help to see if the card works under DOS with the vendor's drivers. Also, don't test the card with something complex like SLIP or PPP until you are sure you can make simple connections. If simple things work but SLIP does not, your problem is most likely with SLIP, not with PCMCIA.
- If you get kernel messages indicating that the serial_cs module cannot be loaded, it means that your kernel does not have serial device support. If you have compiled the serial driver as a module, you must modify `/etc/pcmcia/config` to indicate that the `serial` module should be loaded before `serial_cs`.

4.5 PCMCIA SCSI adapters

All the currently supported PCMCIA SCSI cards are work-alikes of one of the following ISA bus cards: the Qlogic, the Adaptec AHA-152X, or the Future Domain TMC-16x0. The PCMCIA drivers are built by linking some PCMCIA-specific code (in `qlogic_cs.c`, `toaster_cs.c`, or `fdomain_cs.c`) with the normal Linux SCSI driver. Due to limitations in the Linux SCSI driver model, only one removable card per driver is supported.

When a new SCSI host adapter is detected, the SCSI drivers will probe for devices. Check the system log to make sure your devices are detected properly. New SCSI devices will be assigned to the first available

TOC

SCSI device files. The first SCSI disk will be `/dev/sda`, the first SCSI tape will be `/dev/st0`, a the first CD-ROM will be `/dev/scd0`.

A list of SCSI devices connected to this host adapter will be shown in `/var/run/stab`, and the SCSI configuration script, `/etc/pcmcia/scsi`, will be called once for each attached device, to either configure or shut down that device. The default script does not take any actions to configure SCSI devices, but will properly unmount filesystems on SCSI devices when a card is removed.

The device addresses passed to `scsi.opts` are complicated, because of the variety of things that can be attached to a SCSI adapter. Addresses consist of either six or seven comma-separated fields: the current scheme, the device type, the socket number, the SCSI channel, ID, and logical unit number, and optionally, the partition number. The device type will be ``sd'' for disks, ``st'' for tapes, ``sr'' for CD-ROM devices, and ``sg'' for generic SCSI devices. For most setups, the SCSI channel and logical unit number will be 0. For disk devices with several partitions, `scsi.opts` will first be called for the whole device, with a five-field address. The script should set the `PARTS` variable to a list of partitions. Then, `scsi.opts` will be called for each partition, with the longer seven-field addresses.

If your kernel does not have a top-level driver (disk, tape, etc) for a particular SCSI device, then the device will not be configured by the PCMCIA drivers. As a side effect, the device's name in `/var/run/stab` will be something like ``sd#nnnn'' where ``nnnn'' is a four-digit hex number. This happens when `cardmgr` is unable to translate a SCSI device ID into a corresponding Linux device name.

It is possible to modularize the top-level SCSI drivers so that they are loaded on demand. To do so, you need to edit `/etc/pcmcia/config` to tell `cardmgr` which extra modules need to be loaded when your adapter is configured. For example:

```
device "aha152x_cs"
  class "scsi" module "scsi/scsi_mod", "scsi/sd_mod", "aha152x_cs"
```

would say to load the core SCSI module and the top-level disk driver module before loading the regular PCMCIA driver module. The PCMCIA Configure script will not automatically detect modularized SCSI modules, so you will need use the manual configure option to enable SCSI support.

Always turn on SCSI devices before powering up your laptop, or before inserting the adapter card, so that the SCSI bus is properly terminated when the adapter is configured. Also be very careful about ejecting a SCSI adapter. Be sure that all associated SCSI devices are unmounted and closed before ejecting the card. The best way to ensure this is to use either `cardctl` or `cardinfo` to request card removal before physically ejecting the card. For now, all SCSI devices should be powered up before plugging in a SCSI adapter, and should stay connected until after you unplug the adapter and/or power down your laptop.

There is a potential complication when using these cards that does not arise with ordinary ISA bus adapters. The SCSI bus carries a ``termination power'' signal that is necessary for proper operation of ordinary passive SCSI terminators. PCMCIA SCSI adapters do not supply termination power, so if it is required, an external device must supply it. Some external SCSI devices may be configured to supply termination power. Others, such as the Zip Drive and the Syquest EZ-Drive, use active terminators that do not depend on it. In some cases, it may be necessary to use a special terminator block such as the APS

SCSI Sentry 2, which has an external power supply. When configuring your SCSI device chain, be a of whether or not any of your devices require or can provide termination power.

SCSI device parameters

The following parameters can be defined in `scsi.opts`:

`DO_FSTAB`

A boolean (y/n) setting: specifies if an entry should be added to `/etc/fstab` for this device.

`DO_FSCK`

A boolean (y/n) setting: specifies if the filesystem should be checked before being mounted, with ``fsck -Ta''.

`DO_MOUNT`

A boolean (y/n) setting: specifies if this device should be automatically mounted at card insertion time.

`FSTYPE`, `OPTS`, `MOUNTPT`

The filesystem type, mount options, and mount point to be used for the fstab entry and/or mounting the device.

For example, here is a script for configuring a disk device at SCSI ID 3, with two partitions, and a CD-ROM at SCSI ID 6:

```
case "$ADDRESS" in
*,sd,*,0,3,0)
    # This device has two partitions...
    PARTS="1 2"
    ;;
*,sd,*,0,3,0,1)
    # Options for partition 1:
    #  update /etc/fstab, and mount an ext2 fs on /usr1
    DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="ext2"
    OPTS=""
    MOUNTPT="/usr1"
    ;;
*,sd,*,0,3,0,2)
    # Options for partition 2:
    #  update /etc/fstab, and mount an MS-DOS fs on /usr2
    DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="msdos"
    OPTS=""
    MOUNTPT="/usr2"
    ;;
```

```
*,sr,*,0,6,0)
    # Options for CD-ROM at SCSI ID 6
    PARTS=""
    DO_FSTAB="y" ; DO_FSCK="n" ; DO_MOUNT="y"
    FSTYPE="iso9660"
    OPTS="ro"
    MOUNTPT="/cdrom"
    ;;
esac
```

Comments about specific cards

- The Adaptec APA-1480 CardBus card needs a large IO port window (256 contiguous ports aligned on a 256-port boundary). It may be necessary to expand the IO port regions in `/etc/pcmcia/config.opts` to guarantee that such a large window can be found.
- The Adaptec APA-460 SlimSCSI adapter is not supported. This card was originally sold under the Trantor name, and when Adaptec merged with Trantor, they continued to sell the Trantor card with an Adaptec label. The APA-460 is not compatible with any existing Linux driver.
- I have had one report of a bad interaction between the New Media Bus Toaster and a UMAX Astra 1200s scanner. Due to the complexity of the SCSI protocol, when diagnosing problems with SCSI devices, it is worth considering that incompatible combinations like this may exist and may not be documented.

Diagnosing problems with SCSI adapters

- With the `aha152x_cs` driver (used by Adaptec, New Media, and a few others), it seems that SCSI disconnect/reconnect support is a frequent source of trouble with tape drives. To disable this ``feature," add the following to `/etc/pcmcia/config.opts`:

  ```
  module "aha152x_cs" opts "reconnect=0"
  ```

- Also with the `aha152x_cs` driver, certain devices seem to require a longer startup delay, controlled via the `reset_delay` module parameter. The Yamaha 4416S CDR drive is one such device. The result is the device is identified successfully, then hangs the system. In such cases, try:

  ```
  module "aha152x_cs" opts "reset_delay=500"
  ```

- If you have compiled SCSI support as modules (`CONFIG_SCSI` is ``m"), when configuring PCMCIA, you must explicitly specify that you want the SCSI drivers to be built. You must also modify `/etc/pcmcia/config` to load the SCSI modules before the appropriate `*_cs` driver is loaded.
- If you get ``aborting command due to timeout" messages when the SCSI bus is probed, you almost certainly have an interrupt conflict.
- If the host driver reports ``no SCSI devices found", verify that your kernel was compiled with the appropriate top-level SCSI drivers for your devices (i.e., disk, tape, CD-ROM, and/or generic). If a

top-level driver is missing, devices of that type will be ignored.

4.6 PCMCIA memory cards

The `memory_cs` driver handles all types of memory cards, as well as providing direct access to the PCMCIA memory address space for cards that have other functions. When loaded, it creates a combination of character and block devices. See the man page for the module for a complete description of the device naming scheme. Block devices are used for disk-like access (creating and mounting filesystems, etc). The character devices are for "raw" unbuffered reads and writes at arbitrary locations.

The device address passed to `memory.opts` consists of two fields: the scheme, and the socket number. The options are applied to the first common memory partition on the corresponding memory card.

Some older memory cards, and most simple static RAM cards, lack a ``Card Information Structure" (CIS), which is the scheme PCMCIA cards use to identify themselves. Normally, `cardmgr` will assume that any card that lacks a CIS is a simple memory card, and load the `memory_cs` driver. Thus, a common side effect of a general card identification problem is that other types of cards may be misdetected as memory cards.

The `memory_cs` driver uses a heuristic to guess the capacity of these cards. The heuristic does not work for write protected cards, and may make mistakes in some other cases as well. If a card is misdetected, its size should then be explicitly specified when using commands such as `dd` or `mkfs`.

Memory device parameters

The following parameters can be specified in `memory.opts`:

`DO_FSTAB`

A boolean (y/n) setting: specifies if an entry should be added to `/etc/fstab` for this device.

`DO_FSCK`

A boolean (y/n) setting: specifies if the filesystem should be checked before being mounted, with ``fsck -Ta".

`DO_MOUNT`

A boolean (y/n) setting: specifies if this device should be automatically mounted at card insertion time.

`FSTYPE, OPTS, MOUNTPT`

The filesystem type, mount options, and mount point to be used for the fstab entry and/or mounting the device.

Here is an example of a script that will automatically mount memory cards based on which socket they are inserted into:

```
case "$ADDRESS" in
*,0,0)
```

```
        # Mount filesystem, but don't update /etc/fstab
        DO_FSTAB="n" ; DO_FSCK="y" ; DO_MOUNT="y"
        FSTYPE="ext2" ; OPTS=""
        MOUNTPT="/mem0"
        ;;
    *,1,0)
        # Mount filesystem, but don't update /etc/fstab
        DO_FSTAB="n" ; DO_FSCK="y" ; DO_MOUNT="y"
        FSTYPE="ext2" ; OPTS=""
        MOUNTPT="/mem1"
        ;;
    esac
```

Using flash memory cards

The device address passed to `ftl.opts` consists of three or four fields: the scheme, the socket number, the region number, and optionally, the partition number. Most flash cards have just one flash memory region, so the region number will generally always be zero.

To use a flash memory card as an ordinary disk-like block device, first create an FTL, or ``flash translation layer'', partition on the device with the `ftl_format` command. This layer hides the device-specific details of flash memory programming and make the card look like a simple block device. For example:

```
ftl_format -i /dev/mem0c0c
```

Note that this command accesses the card through the ``raw'' memory card interface. Once formatted, the card can be accessed as an ordinary block device via the `ftl_cs` driver. For example:

```
mke2fs /dev/ftl0c0
mount -t ext2 /dev/ftl0c0 /mnt
```

Device naming for FTL devices is tricky. Minor device numbers have three parts: the card number, the region number on that card, and optionally, the partition within that region. A region can either be treated as a single block device with no partition table (like a floppy), or it can be partitioned like a hard disk device. The ``ftl0c0'' device is card 0, common memory region 0, the entire region. The ``ftl0c0p1'' through ``ftl0c0p4'' devices are primary partitions 1 through 4 if the region has been partitioned.

There are two major formats for flash memory cards: the FTL style, and the Microsoft Flash File System. The FTL format is generally more flexible because it allows any ordinary high-level filesystem (ext2, ms-dos, etc) to be used on a flash card as if it were an ordinary disk device. The FFS is a completely different filesystem type. Linux cannot currently handle cards formated with FFS.

Intel Series 100 flash cards use the first 128K flash block to store the cards' configuration information. To prevent accidental erasure of this information, `ftl_format` will automatically detect this and skip the first block when creating an FTL partition.

4.7 PCMCIA ATA/IDE card drives

TOC

ATA/IDE drive support is based on the regular kernel IDE driver. The PCMCIA-specific part of the driver is `ide_cs`. Be sure to use `cardctl` or `cardinfo` to shut down an ATA/IDE card before ejecting it, as the driver has not been made ``hot-swap-proof''.

The device addresses passed to `ide.opts` consist of either three or four fields: the current scheme, the socket number, the drive's serial number, and an optional partition number. The `ide_info` command can be used to obtain an IDE device's serial number. As with SCSI devices, `ide.opts` is first called for the entire device. If `ide.opts` returns a list of partitions in the `PARTS` variable, the script will then be called for each partition.

ATA/IDE fixed-disk device parameters

The following parameters can be specified in `ide.opts`:

`DO_FSTAB`

A boolean (y/n) setting: specifies if an entry should be added to `/etc/fstab` for this device.

`DO_FSCK`

A boolean (y/n) setting: specifies if the filesystem should be checked before being mounted, with ``fsck -Ta''.

`DO_MOUNT`

A boolean (y/n) setting: specifies if this device should be automatically mounted at card insertion time.

`FSTYPE`, `OPTS`, `MOUNTPT`

The filesystem type, mount options, and mount point to be used for the fstab entry and/or mounting the device.

Here is an example `ide.opts` file to mount the first partition of any ATA/IDE card on `/mnt`.

```
case "$ADDRESS" in
*,*,*,1)
    DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="msdos"
    OPTS=""
    MOUNTPT="/mnt"
    ;;
*,*,*)
    PARTS="1"
    ;;
esac
```

Diagnosing problems with ATA/IDE adapters

- Some IDE drives violate the PCMCIA specification by requiring a longer time to spin up than the maximum allowed card setup time. Starting with release 3.0.6, the ide_cs driver will automatically retry the device probe to give these drives time to spin up. With older drivers, you may need to load the `pcmcia_core` module with:

```
CORE_OPTS="unreset_delay=400"
```

- To use an ATA/IDE CD-ROM device, your kernel must be compiled with `CONFIG_BLK_DEV_IDECD` enabled. This will normally be the case for standard kernels, however it is something to be aware of if you compile a custom kernel.

4.8 Multifunction cards

A single interrupt can be shared by several drivers, such as the serial driver and an ethernet driver: in fact, the PCMCIA specification requires all card functions to share the same interrupt. Normally, all card functions are available without having to swap drivers.

Simultaneous use of two card functions is ``tricky'' and various hardware vendors have implemented interrupt sharing in their own incompatible (and sometimes proprietary) ways. The drivers for some cards (Ositech Jack of Diamonds, 3Com 3c562, Linksys) properly support simultaneous access, but others (Megahertz in particular) do not.

Older kernels do not support interrupt sharing between different device drivers, so it is not possible for the PCMCIA drivers to configure this card for simultaneous ethernet and modem access. The ethernet and serial drivers are both loaded automatically. However, the ethernet driver ``owns'' the card interrupt by default. To use the modem, you can unload the ethernet driver and reconfigure the serial port by doing something like:

```
ifconfig eth0 down
rmmod 3c589_cs
setserial /dev/modem autoconfig auto_irq
setserial /dev/modem
```

The second `setserial` should verify that the port has been configured to use the interrupt previously used by the ethernet driver.

5. Advanced topics

5.1 Resource allocation for PCMCIA devices

In theory, it should not really matter which interrupt is allocated to which device, as long as two devices are not configured to use the same interrupt. In `/etc/pcmcia/config.opts` you'll find a place for excluding interrupts that are used by non-PCMCIA devices.

Similarly, there is no way to directly specify the I/O addresses for a card to use. The `/etc/pcmcia/config.opts` file allows you to specify ranges of ports available for use by any card, or to exclude ranges that conflict with other devices.

After modifying `/etc/pcmcia/config.opts`, you can reinitialize `cardmgr` with ``kill -HUP''.

The interrupt used to monitor card status changes is chosen by the low-level socket driver module (`i82365` or `tcic`) before `cardmgr` parses `/etc/pcmcia/config`, so it is not affected by changes to this file. To set this interrupt, use the `cs_irq=` option when the socket driver is loaded, by setting the `PCIC_OPTS` variable in `/etc/rc.d/rc.pcmcia`.

All the client card drivers have a parameter called `irq_list` for specifying which interrupts they may try to allocate. These driver options should be set in your `/etc/pcmcia/config` file. For example:

```
device "serial_cs"
  module "serial_cs" opts "irq_list=8,12"
  ...
```

would specify that the serial driver should only use irq 8 or irq 12. Regardless of `irq_list` settings, Card Services will never allocate an interrupt that is already in use by another device, or an interrupt that is excluded in the config file.

5.2 How can I have separate device setups for home and work?

This is fairly easy using ``scheme'' support. Use two configuration schemes, called ``home'' and ``work''. Here is an example of a `network.opts` script with scheme-specific settings:

```
case "$ADDRESS" in
work,*,*,*)
    # definitions for network card in work scheme
    ...
    ;;
```

```
    home,*,*,*|default,*,*,*)
        # definitions for network card in home scheme
        ...
        ;;
    esac
```

The first part of a device address is always the configuration scheme. In this example, the second ``case'' clause will select for both the ``home'' and ``default'' schemes. So, if the scheme is unset for any reason, it will default to the ``home'' setup.

Now, to select between the two sets of settings, run either:

```
cardctl scheme home
```

or

```
cardctl scheme work
```

The `cardctl` command does the equivalent of shutting down all your cards and restarting them. The command can be safely executed whether or not the PCMCIA system is loaded, but the command may fail if you are using other PCMCIA devices at the time (even if their configurations are not explicitly dependant on the scheme setting).

To find out the current scheme setting, run:

```
cardctl scheme
```

By default, the scheme setting is persistent across boots. This can have undesirable effects if networking is initialized for the wrong environment. Optionally, you can set the initial scheme value with the `SCHEME` startup option (see Startup options for details). It is also possible to set the scheme from the `lilo` boot prompt. Since `lilo` passes unrecognized options to `init` as environment variables, a value for `SCHEME` (or any other PCMCIA startup option) at the boot prompt will be propagated into the PCMCIA startup script.

To save even more keystrokes, schemes can be specified in `lilo`'s configuration file. For instance, you could have:

```
root = /dev/hda1
read-only
image = /boot/vmlinuz
  label  = home
  append = "SCHEME=home"
image = /boot/vmlinuz
  label  = work
  append = "SCHEME=work"
```

Typing ``home'' or ``work'' at the boot prompt would then boot into the appropriate scheme.

5.3 Booting from a PCMCIA device

TOC

Having the root filesystem on a PCMCIA device is tricky because the Linux PCMCIA system is not designed to be linked into the kernel. Its core components, the loadable kernel modules and the user mode cardmgr daemon, depend on an already running system. The kernel's ``initrd'' facility works around this requirement by allowing Linux to boot using a temporary ram disk as a minimal root image, load drivers, and then re-mount a different root filesystem. The temporary root can configure PCMCIA devices and then re-mount a PCMCIA device as root.

The initrd image absolutely must reside on a bootable device: this generally cannot be put on a PCMCIA device. This is a BIOS limitation, not a kernel limitation. It is useful here to distinguish between ``boot-able'' devices (i.e., devices that can be booted), and ``root-able'' devices (i.e., devices that can be mounted as root). ``Boot-able'' devices are determined by the BIOS, and are generally limited to internal floppy and hard disk drives. ``Root-able'' devices are any block devices that the kernel supports once it has been loaded. The initrd facility makes more devices ``root-able'', not ``boot-able''.

Some Linux distributions will allow installation to a device connected to a PCMCIA SCSI adapter, as an unintended side-effect of their support for installs from PCMCIA SCSI CD-ROM devices. However, at present, no Linux installation tools support configuring an appropriate ``initrd'' to boot Linux with a PCMCIA root filesystem. Setting up a system with a PCMCIA root thus requires that you use another Linux system to create the ``initrd'' image. If another Linux system is not available, another option would be to temporarily install a minimal Linux setup on a non-PCMCIA drive, create an initrd image, and then reinstall to the PCMCIA target.

The Linux Bootdisk-HOWTO has some general information about setting up boot disks but nothing specific to initrd. The main initrd document is included with recent kernel source code distributions, in `linux/Documentation/initrd.txt`. Before beginning, you should read this document. A familiarity with `lilo` is also helpful. Using initrd also requires that you have a kernel compiled with `CONFIG_BLK_DEV_RAM` and `CONFIG_BLK_DEV_INITRD` enabled.

This is an advanced configuration technique, and requires a high level of familiarity with Linux and the PCMCIA system. Be sure to read all the relevant documentation before starting. The following cookbook instructions should work, but deviations from the examples will quickly put you in uncharted and ``unsupported'' territory, and you will be on your own.

This method absolutely requires that you use a PCMCIA driver release of 2.9.5 or later. Older PCMCIA packages or individual components will not work in the initrd context. Do not mix components from different releases.

The pcinitrd helper script

The `pcinitrd` script creates a basic initrd image for booting with a PCMCIA root partition. The image includes a minimal directory heirarchy, a handful of device files, a few binaries, shared libraries, and a set of PCMCIA driver modules. When invoking `pcinitrd`, you specify the driver modules that you want to be included in the image. The core PCMCIA components, `pcmcia_core` and `ds`, are automatically included.

TOC

As an example, say that your laptop uses an i82365-compatible host controller, and you want to boot Linux with the root filesystem on a hard drive attached to an Adaptec SlimSCSI adapter. You could create an appropriate initrd image with:

```
pcinitrd -v initrd pcmcia/i82365.o pcmcia/aha152x_cs.o
```

To customize the initrd startup sequence, you could mount the image using the ``loopback'' device with a command like:

```
mount -o loop -t ext2 initrd /mnt
```

and then edit the `linuxrc` script. The configuration files will be installed under `/etc` in the image, and can also be customized. See the man page for `pcinitrd` for more information.

Creating an initrd boot floppy

After creating an image with `pcinitrd`, you can create a boot floppy by copying the kernel, the compressed initrd image, and a few support files for `lilo` to a clean floppy. In the following example, we assume that the desired PCMCIA root device is `/dev/sda1`:

```
mke2fs /dev/fd0
mount /dev/fd0 /mnt
mkdir /mnt/etc /mnt/boot /mnt/dev
cp -a /dev/fd0 /dev/sda1 /mnt/dev
cp [kernel-image] /mnt/vmlinuz
cp /boot/boot.b /mnt/boot/boot.b
gzip < [initrd-image] > /mnt/initrd
```

Create `/mnt/etc/lilo.conf` with the contents:

```
boot=/dev/fd0
compact
image=/vmlinuz
    label=linux
    initrd=/initrd
    read-only
    root=/dev/sda1
```

Finally, invoke lilo with:

```
lilo -r /mnt
```

When `lilo` is invoked with `-r`, it performs all actions relative to the specified alternate root directory. The reason for creating the device files under `/mnt/dev` was that `lilo` will not be able to use the files in `/dev` when it is running in this alternate-root mode.

Installing an initrd image on a non-Linux drive

TOC

One common use of the initrd facility would be on systems where the internal hard drive is dedicated to another operating system. The Linux kernel and initrd image can be placed in a non-Linux partition, and `lilo` or `LOADLIN` can be set up to boot Linux from these images.

Assuming that you have a kernel has been configured for the appropriate root device, and an initrd image created on another system, the easiest way to get started is to boot Linux using `LOADLIN`, as:

```
LOADLIN <kernel> initrd=<initrd-image>
```

Once you can boot Linux on your target machine, you could then install `lilo` to allow booting Linux directly. For example, say that `/dev/hda1` is the non-Linux target partition and `/mnt` can be used as a mount point. First, create a subdirectory on the target for the Linux files:

```
mount /dev/hda1 /mnt
mkdir /mnt/linux
cp [kernel-image] /mnt/linux/vmlinuz
cp [initrd-image] /mnt/linux/initrd
```

In this example, say that `/dev/sda1` is the desired Linux root partition, a SCSI hard drive mounted via a PCMCIA SCSI adapter. To install `lilo`, create a `lilo.conf` file with the contents:

```
boot=/dev/hda
map=/mnt/linux/map
compact
image=/mnt/linux/vmlinuz
        label=linux
        root=/dev/sda1
        initrd=/mnt/linux/initrd
        read-only
other=/dev/hda1
        table=/dev/hda
        label=windows
```

The `boot=` line says to install the boot loader in the master boot record of the specified device. The `root=` line identifies the desired root filesystem to be used after loading the initrd image, and may be unnecessary if the kernel image is already configured this way. The `other=` section is used to describe the other operating system installed on `/dev/hda1`.

To install `lilo` in this case, use:

```
lilo -C lilo.conf
```

Note that in this case, the `lilo.conf` file uses absolute paths that include `/mnt`. I did this in the example because the target filesystem may not support the creation of Linux device files for the `boot=` and `root=` options.

6. Dealing with unsupported cards

6.1 Configuring unrecognized cards

Assuming that your card is supported by an existing driver, all that needs to be done is to add an entry to `/etc/pcmcia/config` to tell `cardmgr` how to identify the card, and which driver(s) need to be linked up to this card. Check the man page for `pcmcia` for more information about the config file format. If you insert an unknown card, `cardmgr` will normally record some identification information in the system log that can be used to construct the config entry. This information can also be displayed with the ``cardctl ident'' command.

Here is an example of how cardmgr will report an unsupported card in `/usr/adm/messages`.

```
cardmgr[460]: unsupported card in socket 1
cardmgr[460]: product info: "MEGAHERTZ", "XJ2288", "V.34 PCMCIA MODEM"
cardmgr[460]: manfid: 0x0101, 0x1234  function: 2 (serial)
```

The corresponding entry in `/etc/pcmcia/config` would be:

```
card "Megahertz XJ2288 V.34 Fax Modem"
  version "MEGAHERTZ", "XJ2288", "V.34 PCMCIA MODEM"
  bind "serial_cs"
```

or using the more compact product ID codes:

```
card "Megahertz XJ2288 V.34 Fax Modem"
  manfid 0x0101, 0x1234
  bind "serial_cs"
```

You can use ``*'' to match strings that don't need to match exactly, like version numbers. When making new config entries, be careful to copy the strings exactly, preserving case and blank spaces. Also be sure that the config entry has the same number of strings as are reported in the log file.

Beware that you can specify just about any driver for a card, but if you're just shooting in the dark, there is not much reason to expect this to be productive. You may get lucky and find that your card is supported by an existing driver. However, the most likely outcome is that the driver won't work, and may have unfortunate side effects like locking up your system. Unlike most ordinary device drivers, which probe for an appropriate card, the probe for a PCMCIA device is done by `cardmgr`, and the driver itself may not do much validation before attempting to communicate with the device.

After editing `/etc/pcmcia/config`, you can signal `cardmgr` to reload the file with:

```
kill -HUP `cat /var/run/cardmgr.pid`
```

If you do set up an entry for a new card, please send me a copy so that I can include it in the standard config

file.

6.2 Adding support for an NE2000-compatible ethernet card

Before you begin: this procedure will only work for simple ethernet cards. Multifunction cards (i.e., ethernet/modem combo cards) have an extra layer of complexity regarding how the two functions are integrated, and generally cannot be supported without obtaining some configuration information from the card vendor. Using the following procedure for a multifunction card will not be productive.

First, see if the card is already recognized by `cardmgr`. Some cards not listed in `SUPPORTED.CARDS` are actually OEM versions of cards that are supported. If you find a card like this, let me know so I can add it to the list.

If your card is not recognized, follow the instructions in the Configuring unrecognized cards section to create a config entry for your card, and bind the card to the `pcnet_cs` driver. Restart `cardmgr` to use the updated config file.

If the `pcnet_cs` driver says that it is unable to determine your card's hardware ethernet address, then edit your new config entry to bind the card to the memory card driver, `memory_cs`. Restart `cardmgr` to use the new updated config file. You will need to know your card's hardware ethernet address. This address is a series of six two-digit hex numbers, often printed on the card itself. If it is not printed on the card, you may be able to use a DOS driver to display the address. In any case, once you know it, run:

```
dd if=/dev/mem0a count=20 | od -Ax -t x1
```

and search the output for your address. Only the even bytes are defined, so ignore the odd bytes in the dump. Record the hex offset of the first byte of the address. Now, edit `clients/pcnet_cs.c` and find the `hw_info` structure. You'll need to create a new entry for your card. The first field is the memory offset. The next three fields are the first three bytes of the hardware address. The final field contains some flags for specific card features; to start, try setting it to 0.

After editing `pcnet_cs.c`, compile and install the new module. Edit `/etc/pcmcia/config` again, and change the card binding from `memory_cs` to `pcnet_cs`. Follow the instructions for reloading the config file, and you should be all set. Please send me copies of your new `hw_info` and config entries.

If you can't find your card's hardware address in the hex dump, as a method of last resort, it is possible to ``hard-wire'' the address when the `pcnet_cs` module is initialized. Edit `/etc/pcmcia/config.opts` and add a `hw_addr=` option, like so:

```
module "pcnet_cs" opts "hw_addr=0x00,0x80,0xc8,0x01,0x02,0x03"
```

Substitute your own card's hardware address in the appropriate spot, of course. Beware that if you've gotten this far, it is very unlikely that your card is genuinely NE2000 compatible. In fact, I'm not sure if there are *any* cards that are not handled by one of the first two methods.

TOC

6.3 PCMCIA floppy interface cards

The PCMCIA floppy interface used in the Compaq Aero and a few other laptops is not yet supported by this package. The snag in supporting the Aero floppy is that the Aero seems to use a customized PCMCIA controller to support DMA to the floppy. Without knowing exactly how this is done, there isn't any way to implement support under Linux.

If the floppy adapter card is present when an Aero is booted, the Aero BIOS will configure the card, and Linux will identify it as a normal floppy drive. When the Linux PCMCIA drivers are loaded, they will notice that the card is already configured and attached to a Linux driver, and this socket will be left alone. So, the drive can be used if it is present at boot time, but the card is not hot swappable.

6.4 What's up with support for Xircom cards?

A driver for Xircom ethernet and ethernet/modem cards is included in the current PCMCIA package, thanks to the work of Werner Koch. I've set up a HyperNews forum specifically for discussion of Xircom driver development, at http://hyper.stanford.edu/HyperNews/get/pcmcia/xircom.html.

For a long time, Xircom cards were not supported because Xircom had a company policy of not disclosing technical information about their cards. However, they have relaxed their rules, and now, they do distribute driver information.

---

7. Debugging tips and programming information

7.1 Submitting useful bug reports

The best way to submit bug reports is to use the HyperNews message lists on the Linux PCMCIA information site. That way, other people can see current problems (and fixes or workarounds, if available). Here are some things that should be included in all bug reports:

- Your system type, and the output of the `probe` command.
- What PCMCIA cards you are using.
- Your Linux kernel version, and PCMCIA driver version.
- Any changes you have made to the startup files in `/etc/pcmcia`, or to the PCMCIA startup script.
- All PCMCIA-related messages in your system log file.

All the PCMCIA modules and the `cardmgr` daemon send status messages to the system log. This will usually be something like `/var/log/messages` or `/usr/adm/messages`. This file should be the first place to look when tracking down a problem. When submitting a bug report, always include the contents of this file. If you are having trouble finding your system messages, check `/etc/syslog.conf` to see how different classes of messages are handled.

Before submitting a bug report, please check to make sure that you are using an up-to-date copy of the driver package. While it is somewhat gratifying to read bug reports for things I've already fixed, it isn't a particularly constructive use of my time.

If you do not have web access, bug reports can be sent to me at `dhinds@hyper.stanford.edu`. However, I prefer that bug reports be posted to my web site, so that they can be seen by others.

7.2 Interpreting kernel trap reports

If your problem involves a kernel fault, the register dump from the fault is only useful if you can translate the fault address, EIP, to something meaningful. Recent versions of `klogd` attempt to translate fault addresses based on the current kernel symbol map, but this may not work if the fault is in a module, or if the problem is severe enough that `klogd` cannot finish writing the fault information to the system log.

If a fault is in the main kernel, the fault address can be looked up in the `System.map` file. This may be installed in `/System.map` or `/boot/System.map`. If a fault is in a module, the `nm` command gives the same information, however, the fault address has to be adjusted based on the module's load address. Let's say that you have the following kernel fault:

```
Unable to handle kernel NULL pointer dereference
```

```
current->tss.cr3 = 014c9000, %cr3 = 014c9000
*pde = 00000000
Oops: 0002
CPU:    0
EIP:    0010:[<c2026081>]
EFLAGS: 00010282
```

The fault address is 0xc2026081. Looking at `System.map`, we see that this is past the end of the kernel, i.e., is in a kernel module. To determine which module, check the output of ``ksyms -m | sort'':

```
Address   Symbol                                Defined by
c200d000  (35k)                                 [pcmcia_core]
c200d10c  register_ss_entry                     [pcmcia_core]
c200d230  unregister_ss_entry                   [pcmcia_core]
          ...
c2026000  (9k)                                  [3c574_cs]
c202a000  (4k)                                  [serial_cs]
```

So, 0xc2026081 is in the `3c574_cs` module, and is at an offset of 0x0081 from the start of the module. We cannot look up this offset in `3c574_cs.o` yet: when the kernel loads a module, it inserts a header at the module load address, so the real start of the module is offset from the address shown in `ksyms`. The size of the header varies with kernel version: to find out the size for your kernel, check a module that exports symbols (like `pcmcia_core` above), and compare a symbol address with `nm` output for that symbol. In this example, `register_ss_entry` is loaded at an offset of 0xc200d10c - 0xc200d000 = 0x010c, while ``nm pcmcia_core.o'' shows the offset as 0x00c0, so the header size is 0x010c - 0x00c0 = 0x004c bytes.

Back to `3c574_cs`, our fault offset is 0x0081, and subtracting the 0x004c header, the real module offset is 0x0035. Now looking at ``nm 3c574_cs.o | sort'', we see:

```
0000002c d if_names
0000002c t tc574_attach
00000040 d mii_preamble_required
00000041 d dev_info
```

So, the fault is located in `tc574_attach()`.

In this example, the fault did not cause a total system lockup, so `ksyms` could be executed after the fault happened. In other cases, you may have to infer the module load addresses indirectly. The same sequence of events will normally load modules in the same order and at the same addresses. If a fault happens when a certain card is inserted, get the `ksyms` output before inserting the card, or with a different card inserted. You can also manually load the card's driver modules with `insmod` and run `ksyms` before inserting the card.

For more background, see ``man insmod'', ``man ksyms'', and ``man klogd''. In the kernel source tree, `Documentation/oops-tracing.txt` is also relevant. Here are a few more kernel debugging hints:

- Depending on the fault, it may also be useful to translate addresses in the ``Call Trace'', using the same procedure as for the main fault address.
- To diagnose a silent lock-up, try to provoke the problem with X disabled, since kernel messages sent to the text console will not be visible under X.
- If you kill `klogd`, most kernel messages will be echoed directly on the text console, which may be helpful if the problem prevents `klogd` from writing to the system log.
- To cause all kernel messages to be sent to the console, for 2.1 kernels, if `/proc/sys/kernel/printk` exists, do:

```
echo 8 > /proc/sys/kernel/printk
```

- The key combination <RightAlt><ScrLk> prints a register dump on the text console. This may work even if the system is otherwise completely unresponsive, and the EIP address can be interpreted as for a kernel fault.
- For 2.1 kernels configured with `CONFIG_MAGIC_SYSRQ` enabled, various emergency functions are available via special <Alt><SysRq> key combinations, documented in `Documentation/sysrq.txt` in the kernel source tree.

7.3 Low level PCMCIA debugging aids

The PCMCIA modules contain a lot of conditionally-compiled debugging code. Most of this code is under control of the `PCMCIA_DEBUG` preprocessor define. If this is undefined, debugging code will not be compiled. If set to 0, the code is compiled but inactive. Larger numbers specify increasing levels of verbosity. Each module built with `PCMCIA_DEBUG` defined will have an integer parameter, `pc_debug`, that controls the verbosity of its output. This can be adjusted when the module is loaded, so output can be controlled on a per-module basis without recompiling.

Your default configuration for `syslogd` may discard kernel debugging messages. To ensure that they are recorded, edit `/etc/syslog.conf` to ensure that ```kern.debug`'' messages are recorded somewhere. See ```man syslog.conf`'' for details.

There are a few debugging tools in the `debug_tools/` subdirectory of the PCMCIA distribution. The `dump_tcic` and `dump_i365` utilities generate complete register dumps of the PCMCIA controllers, and decode a lot of the register information. They are most useful if you have access to a datasheet for the corresponding controller chip. The `dump_cis` utility (`dump_tuples` in pre-3.0.2 distributions) lists the contents of a card's CIS (Card Information Structure), and decodes some of the important bits. And the `dump_cisreg` utility displays a card's local configuration registers.

The `memory_cs` memory card driver is also sometimes useful for debugging problems with 16-bit PC Cards. It can be bound to any card, and does not interfere with other drivers. It can be used to directly access any card's attribute memory or common memory. Similarly for CardBus cards, the `memory_cb` driver can be bound to any 32-bit card, to give direct access to that card's address spaces. See the man pages for more information.

TOC

7.4 /proc/bus/pccard

Starting with 2.1.103 kernels, the PCMCIA package will create a tree of status information under `/proc/bus/pccard`. The `memory` entry shows memory allocations for PC Card devices in a format similar to `/proc/ioports`. Each socket also has its own subdirectory of status entries. The `info` entry identifies the host controller and describes its capabilities. The `exca` entry is a dump of the ``ExCA'' Intel i82365sl-compatible register set for that socket. For CardBus bridges, the `pci` entry is a dump of the bridge's PCI configuration space, and the `cardbus` entry is a dump of the CardBus configuration registers.

7.5 Writing Card Services drivers for new cards

The Linux PCMCIA Programmer's Guide is the best documentation for the client driver interface. The latest version is always available from `hyper.stanford.edu` in `/pub/pcmcia/doc`, or on the web at http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html.

For devices that are close relatives of normal ISA devices, you will probably be able to use parts of existing Linux drivers. In some cases, the biggest stumbling block will be modifying an existing driver so that it can handle adding and removing devices after boot time. Of the current drivers, the memory card driver is the only ``self-contained'' driver that does not depend on other parts of the Linux kernel to do most of the dirty work.

In many cases, the largest barrier to supporting a new card type is obtaining technical information from the manufacturer. It may be difficult to figure out who to ask, or to explain exactly what information is needed. However, with a few exceptions, it is very difficult if not impossible to implement a driver for a card without technical information from the manufacturer.

I have written a dummy driver with lots of comments that explains a lot of how a driver communicates with Card Services; you will find this in the PCMCIA source distribution in `clients/dummy_cs.c`.

7.6 Guidelines for PCMCIA client driver authors

I have decided that it is not really feasible for me to distribute all PCMCIA client drivers as part of the PCMCIA package. Each new driver makes the main package incrementally harder to maintain, and including a driver inevitably transfers some of the maintenance work from the driver author to me. Instead, I will decide on a case by case basis whether or not to include contributed drivers, based on user demand as well as maintainability. For drivers not included in the core package, I suggest that driver authors adopt the following scheme for packaging their drivers for distribution.

Driver files should be arranged in the same directory scheme used in the PCMCIA source distribution, so that the driver can be unpacked on top of a complete PCMCIA source tree. A driver should include source files (in `./modules/`), a man page (in `./man/`), and configuration files (in `./etc/`). The top level directory should also include a README file.

The top-level directory should include a makefile, set up so that ```make -f ...` all'' and ```make -f`

`... install`" compile the driver and install all appropriate files. If this makefile is given an exten of `.mk`, then it will automatically be invoked by the top-level `Makefile` for the `all` and `instal` targets. Here is an example of how such a makefile could be constructed:

```
# Sample Makefile for contributed client driver
FILES = sample_cs.mk README.sample_cs \
        modules/sample_cs.c modules/sample_cs.h \
        etc/sample etc/sample.opts man/sample_cs.4
all:
        $(MAKE) -C modules MODULES=sample_cs.o
install:
        $(MAKE) -C modules install-modules MODULES=sample_cs.o
        $(MAKE) -C etc install-clients CLIENTS=sample
        $(MAKE) -C man install-man4 MAN4=sample_cs.4
dist:
        tar czvf sample_cs.tar.gz $(FILES)
```

This makefile uses install targets defined in 2.9.10 and later versions of the PCMCIA package. This makefile also includes a ``dist'' target for the convenience of the driver author. You would probably want to add a version number to the final package filename (for example, `sample_cs-1.5.tar.gz`). A complete distribution could look like:

```
sample_cs.mk
README.sample_cs
modules/sample_cs.c
modules/sample_cs.h
etc/sample
etc/sample.opts
man/sample_cs.4
```

With this arrangement, when the contributed driver is unpacked, it becomes essentially part of the PCMCIA source tree. It can make use of the PCMCIA header files, as well as the machinery for checking the user's system configuration, and automatic dependency checking, just like a ``normal'' client driver.

I will accept client drivers prepared according to this specification and place them in the `/pub/pcmcia/contrib` directory on my FTP server, `hyper.stanford.edu`. The README in this directory will describe how to unpack a contributed driver.

The client driver interface has not changed much over time, and has almost always preserved backwards compatibility. A client driver will not normally need to be updated for minor revisions in the main package. I will try to notify authors of contributed drivers of changes that require updates to their drivers.

7.7 Guidelines for Linux distribution maintainers

TOC

If your distribution has system configuration tools that you would like to be PCMCIA-aware, please use the `*.opts` files in `/etc/pcmcia` for your ``hooks.'' These files will not be modified if a user compiles and installs a new release of the PCMCIA package. If you modify the main configuration scripts, then a fresh install will silently overwrite your custom scripts and break the connection with your configuration tools. Contact me if you are not sure how to write an appropriate option script.

It is helpful for users (and for me) if you can document how your distribution deviates from the PCMCIA package as described in this document. In particular, please document changes to the startup script and configuration scripts. If you send me the appropriate information, I will include it in the Notes about specific Linux distributions.

When building PCMCIA for distribution, consider including contributed drivers that are not part of the main PCMCIA package. For reasons of maintainability, I am trying to limit the core package size, by only adding new drivers if I think they are of particularly broad interest. Other drivers will be distributed separately, as described in the previous section. The split between integral and separate drivers is somewhat arbitrary and partly historical, and should not imply a difference in quality.

Palm OS Desktop HOWTO

by David H. Silber palm@orbits.com

v1.0.0, 20 September 1998

This HOWTO document explains how to use your Palm OS device with a Linux system. Although HOWTO documents are targeted towards use with the Linux operating system, this one is not dependent on the version of unix used.

1. Introduction

- 1.1 This Document
- 1.2 See Also
- 1.3 Mailing List
- 1.4 Mailing List Archives
- 1.5 FTP Site

2. General Information

- 2.1 Hardware Installation

3. Sharing Palm OS device Data with your Linux System

- 3.1 The pilot-link software
- 3.2 MakeDoc

Appendix

TOC

4. People

5. A History of Palm OS devices.

---

1. Introduction

First there was the Pilot, then the PalmPilot, and now we have the Palm III from Palm Computing Devices. IBM sells a rebadged version of the PalmPilot Professional, which they call the WorkPad. What are these gadgets? They are Personal Digital Assistants (PDAs), sometimes known as palm-top computers. Sold as an electronic combination of address book, telephone directory and memo pad, they are actually general-purpose computers capable of running almost any kind of application. The trick is to find Palm OS applications to fit your needs and to be able to share your data with your Linux system. Palm OS devices incorporate software to synchronize their data with programs running on another computer. Such software for use on a Windows 95 system is generally bundled with the device. A MacOS version is also available. If the foriegn computer is running some other operating system, you need to find third party software to work with your Palm OS device. This document describes such software that is available for use on a Linux or unix system.

1.1 This Document

The latest version of this document can be read at `http://www.orbits.com/Palm/PalmOS-HOWTO.html`, and is part of the Linux Documentation Project (LDP). This document superceedes the previous version, which was called the `Pilot HOWTO`. See `http://sunsite.unc.edu/LDP/` for further information about the LDP and other HOWTO documents.

If you find anything in this document which needs to be corrected or better explained, please send me e-mail at the address above and specify which version of this document you are referring to.

This document is Copyright © 1997 & 1998 by David H. Silber. It is released under the copyright terms in the LDP HOWTO-INDEX document.

1.2 See Also

The `http://www.orbits.com/Palm/PalmDevelompent-HOWTO.html` document describes how to use the gcc-based development system to write Palm OS applications.

The `http://www.orbits.com/Palm/PalmConduit-HOWTO.html` document describes how to use the pilot-link libraries to write Linux-resident conduits for use with Palm OS devices.

1.3 Mailing List

The pilot-unix mailing list is maintained by Matthew Cravit pilot-unix-owner@lists.best.com. Its mandate is:

```
The pilot-unix mailing list is for discussion and idea-sharing for those
interested in using the US Robotics Pilot PDAs with UNIX systems. This
includes people who are interested in helping to develop tools to allow the
Pilot to operate with UNIX, and possibly to develop an SDK for the Pilot
for Unix.
```

For more information, including how to subscribe to the list, send mail containing the word ``INFO" to `pilot-unix-request@lists.best.com`. The subject line does not matter.

TOC

1.4 Mailing List Archives

An archive of the pilot-unix mailing list can be found at `http:///www.acm.rpi.edu/~albert/pilot/`. It is maintained by Chris Stevens albert@acm.rpi.edu.

1.5 FTP Site

An FTP site containing an archive of Palm OS tools for use on unix systems is located at `ftp://ryeham.ee.ryerson.ca/pub/PalmOS/`. It is maintained by Jeff Dionne jeff@ryeham.ee.ryerson.ca.

---

2. General Information

2.1 Hardware Installation

Palm OS devices come with a ``cradle'' for exchanging data with the desktop computer. This device is actually a serial cable with a custom holder for the Palm OS devices and an integrated `HotSync' button. Plug your cradle into a spare serial port on your computer. When you run each of the stand-alone programs, you will need to place your Palm OS device in the cradle and push the `HotSync' button so the Palm OS device knows that it has to communicate. If the Palm OS device happens to be off when the button is pushed, it will turn itself on.

For convenience, create a device, `/dev/pilot` which will be an alternate name for the serial port to which your Palm OS device cradle is connected. As the root user, enter the following at the shell prompt:

```
ln /dev/cua0 /dev/pilot
```

Replace `cua0` with the name of the port to which you connected your Palm OS device's cradle.

3. Sharing Palm OS device Data with your Linux System

3.1 The pilot-link software

The pilot-link suite of software tools allows you to download programs onto your Palm OS device, and transfer data for the Palm OS device's various built-in programs between the Linux system and the Palm OS device. While these programs are not quite as seamless as the desktop software that comes with the Palm OS device, they do allow you to copy your data in both directions. In general, each separate program in the `pilot-link` suite manages one type of data.

Installing the pilot-link software

The prepackaged versions will inevitably lag slightly behind the master distribution, but will be easier to install and not require configuration. The master distribution might be a better choice in those rare occasions when you have been waiting for a particular feature or bug fix.

You can get the *Debian Linux* (i386) port of *pilot-link* version 0.8.7 from `ftp://ftp.debian.org/pub/debian/dists/stable/main/binary-i386/otherosfs/pilot-link_0.8.7-2.deb`. `ftp://ftp.debian.org/pub/debian/dists/stable/main/binary-i386/otherosfs/pilot-link_0.8.7-2.deb`. If this file does not exist, you will likely find a newer version located in the same directory. If all else fails, or if you want a more recent version than has been packaged for Debian, you can install the generic distribution, as detailed below. Install this file in the normal manner and skip to *Using the pilot-link software*.

You can get the *RedHat Linux* (i386) port of *pilot-link* version 0.8.9 from `ftp://ftp.redhat.com/pub/contrib/i386/pilot-link-0.8.9-1.i386.rpm`. If this file does not exist, you will likely find a newer version located in the same directory. If all else fails, or if you want a more recent version than has been packaged for RedHat, you can install the generic distribution, as detailed below. Install this file in the normal manner and skip to *Using the pilot-link software*.

For other versions of Linux or unix, or if you need the absolute latest version, download the version 0.9.0 of pilot-link from `ftp://ryeham.ee.ryerson.ca/pub/PalmOS/pilot-link.0.9.0.tar.gz`. The version number is likely to change, but new versions should end up in the same location with a similar name.

Once you have the software distribution, unpack it with:

```
tar -xvzf pilot-link.0.9.0.tar.gz
```

If your version of tar does not support the `-z` option, decompress the archive with gunzip, as follows:

```
gunzip pilot-link.0.9.0.tar.gz | tar -xvf -
```

This will create a directory (`pilot-link.0.9.0`) containing the source.

Change your working directory to the source (`pilot-link.0.9.0`) directory.

Run `./configure`. This will search through your system for information needed to compile the software. `configure` will set things up to be installed in `/usr/local` by default. If you want to change it, run `./configure --prefix=DIR`, where `DIR` is replaced with the name of the directory to which the software will be installed.

Run `make`. This will compile the software. The software will not be installed until later, so that you have a chance to try it out first. (If you are replacing an older version with a newer release, you may wish to check and make sure that no functionality that you need has been broken. Generally, this is not a problem.)

As the root user, run `make install`. This will copy the software into directories under `/usr/local` (or wherever you specified with the `--prefix` option). If you can not log in as root, you can install the software to some directory where you have write access.

Don't forget to add any new directories of executables to your search path.

TOC

Using the pilot-link software

Most of the programs in the `pilot-link` suite are *conduits*, that is they transfer data into or out of your Palm OS device.

Each time you use of one of these programs, press the HotSync button on your Palm OS device's cradle. This will initiate the Palm OS device side of the data transfer. Note that not all of these programs prompt you to press the `HotSync' button, so you may have to remember to do it yourself.

For more details, and other options to these programs, view the corresponding manual page. There is an overview manual page under the name `pilot-link`. For the `pilot-xfer` program, for example, type `man pilot-xfer` at your unix shell prompt.

pilot-xfer

Possibly the most useful program in the pilot-link suite, `pilot-xfer` allows you to install programs on your Palm OS device, make a backup, and restore that backup.

To install a program:

```
pilot-xfer /dev/pilot -i program.prc
```

To backup your Palm OS device:

```
pilot-xfer /dev/pilot -b backup-directory
```

This will copy all of the databases on your Palm OS device, (including programs?) to a directory called ``backup-directory'', creating it if it does not already exist.

To restore data to Palm OS device:

```
pilot-xfer /dev/pilot -r backup-directory
```

Generally, you will only need to do this if your Palm OS device loses power or if you have to do a hard reset.

To list the programs on your Palm OS device:

```
pilot-xfer /dev/pilot -l
```

install-memo

Install a Linux file onto the Palm OS device as a memo.

To install a memo into your (already existing) *project* category:

```
install-memo /dev/pilot -c project project.memo
```

The name of the file will be inserted into the memo as its first line and will appear in the directory of memos on your Palm OS device.

memos

This program grabs each memo from the Palm OS device and prints it out in standard mailbox format.

To view your memos:

```
memos /dev/pilot
```

pilot-addresses

`pilot-addresses` Transfer the address database to or from the Palm OS device.

To write your address data to a Linux file from your Palm OS device:

```
pilot-addresses /dev/pilot -w storage.file
```

To read your address data from a Linux file onto your Palm OS device:

```
pilot-addresses /dev/pilot -r storage.file
```

3.2 MakeDoc

TOC

One short-coming of the Palm OS device's built-in memo program is that it does not deal well with large documents. To compensate for this, Rick Bram wrote *Doc*, a document reader for the Palm OS device. (See `http://www.concentric.net/~rbram/doc.shtml`). Documents can be converted to the Doc format with `MakeDoc`, by Pat Beirne.

Installing MakeDoc

`MakeDoc` can be downloaded from `http://www.concentric.net/~rbram/makedoc7.cpp`. Compile it with your C++ compiler and install the resulting executable as ``makedoc'' in a directory in your search path. There seems to be a small bug in `makedoc` (version 0.7a) in that it does not output a newline as the last character displayed to the user. This does not seem to affect the resulting document file, but it is annoying.

There is a new version out, but it requires Java. Take a look at Pat Beirne's MakeDoc web page at `http://cpu563.adsl.sympatico.ca/MakeDocJ.htm`.

Using MakeDoc

Use `MakeDoc` as follows:

```
makedoc data.txt data.prc "Data to display with Doc"
```

This will create a file `data.prc`, which can be installed on your Palm OS device with pilot-xfer. The text *"Data to display with Doc"* will be displayed in the directory of documents that *Doc* manages.

The syntax for `MakeDoc` is as follows:

```
makedoc [-n] [-b] <text-file> <prc-file> <story-name>
or
makedoc -d [-b] <prc-file> <text-file>
```

<text-file>

The file that you wish to convert.

<prc-file>

The name of the resulting file. (End the name with ``.prc''.)

<story-name>

The name you want displayed in the `Doc` or `Jdoc` directory of documents.

There are also options to decode the resulting `.prc` file and manage various compression options.

---

4. People

Kenneth Albanowski Maintains the pilot-link suite of tools.

Donnie Barnes Packaged pilot-link suite as Red Hat RPM files.

Rick Bram Author of Doc.

Matthew Cravit List owner for the *pilot-unix* mailing list.

Jeff Dionne Original author of the pilot-link suite of tools. Manages the FTP area for the UNIX Palm OS/Pilot development project.

Mark W. Eichin Ported pilot-link suite to Debian Linux.

David H. Silber Author of this document.

Chris Stevens Maintains the pilot-unix mailing list archives.

5. A History of Palm OS devices.

There are two versions of the Pilot. They are the *1000* and the *5000* and have 128k and 512k of RAM, respectively. They come with Palm OS 1.0 in ROM.

Then came two devices with the name *PalmPilot*. They are the *PalmPilot Personal* and the *PalmPilot Professional* and have 512k and 1 Meg of RAM, respectively. They also have a backlighting feature for the LCD panel and version 2.0 of the operating system. The *Professional* also comes with a TCP/IP stack and a few extra programs built-in.

(IBM has released a rebadged version of the PalmPilot Professional, which they call a *WorkPad*.)

Now we have the Palm III, which has a redesigned case and an infrared port, not to mention the 2 Meg of RAM and Palm OS 3.0.

It is possible to upgrade any Pilot by swapping out the memory card, which includes both RAM & ROM. Of course, this doesn't get you backlighting for the original Pilot series.

Linux Parallel Processing HOWTO

Hank Dietz, pplinux@ecn.purdue.edu

v980105, 5 January 1998

Parallel Processing *refers to the concept of speeding-up the execution of a program by dividing the program into multiple fragments that can execute simultaneously, each on its own processor. A program being executed across N processors might execute N times faster than it would using a single processor. This document discusses the four basic approaches to parallel processing that are available to Linux users: SMP Linux systems, clusters of networked Linux systems, parallel execution using multimedia instructions (i.e., MMX), and attached (parallel) processors hosted by a Linux system.*

1. Introduction

- 1.1 Is Parallel Processing What I Want?
- 1.2 Terminology
- 1.3 Example Algorithm
- 1.4 Organization Of This Document

2. SMP Linux

- 2.1 SMP Hardware
- 2.2 Introduction To Shared Memory Programming
- 2.3 bb_threads
- 2.4 LinuxThreads
- 2.5 System V Shared Memory
- 2.6 Memory Map Call

3. Clusters Of Linux Systems

- 3.1 Why A Cluster?
- 3.2 Network Hardware

- 3.3 Network Software Interface
- 3.4 PVM (Parallel Virtual Machine)
- 3.5 MPI (Message Passing Interface)
- 3.6 AFAPI (Aggregate Function API)
- 3.7 Other Cluster Support Libraries
- 3.8 General Cluster References

4. SIMD Within A Register (e.g., using MMX)

- 4.1 SWAR: What Is It Good For?
- 4.2 Introduction To SWAR Programming
- 4.3 MMX SWAR Under Linux

5. Linux-Hosted Attached Processors

- 5.1 A Linux PC Is A Good Host
- 5.2 Did You DSP That?
- 5.3 FPGAs And Reconfigurable Logic Computing

6. Of General Interest

- 6.1 Programming Languages And Compilers
- 6.2 Performance Issues
- 6.3 Conclusion - It's Out There

1. Introduction

Parallel Processing refers to the concept of speeding-up the execution of a program by dividing the program into multiple fragments that can execute simultaneously, each on its own processor. A program being executed across *n* processors might execute *n* times faster than it would using a single processor.

Traditionally, multiple processors were provided within a specially designed "parallel computer"; along these lines, Linux now supports **SMP** systems (often sold as "servers") in which multiple processors share a single memory and bus interface within a single computer. It is also possible for a group of computers (for example, a group of PCs each running Linux) to be interconnected by a network to form a parallel-processing **cluster**. The third alternative for parallel computing using Linux is to use the **multimedia instruction extensions** (i.e., MMX) to operate in parallel on vectors of integer data. Finally, it is also possible to use a Linux system as a "host" for a specialized **attached** parallel processing compute engine. All these approaches are discussed in detail in this document.

1.1 Is Parallel Processing What I Want?

Although use of multiple processors can speed-up many operations, most applications cannot yet benefit from parallel processing. Basically, parallel processing is appropriate only if:

- Your application has enough parallelism to make good use of multiple processors. In part, this is a matter of identifying portions of the program that can execute independently and simultaneously on separate processors, but you will also find that some things that *could* execute in parallel might actually slow execution if executed in parallel using a particular system. For example, a program that takes four seconds to execute within a single machine might be able to execute in only one second of processor time on each of four machines, but no speedup would be achieved if it took three seconds or more for these machines to coordinate their actions.
- Either the particular application program you are interested in already has been **parallelized** (rewritten to take advantage of parallel processing) or you are willing to do at least some new coding to take advantage of parallel processing.
- You are interested in researching, or at least becoming familiar with, issues involving parallel processing. Parallel processing using Linux systems isn't necessarily difficult, but it is not familiar to most computer users, and there isn't any book called "Parallel Processing for Dummies"... at least not yet. This HOWTO is a good starting point, not all you need to know.

The good news is that if all the above are true, you'll find that parallel processing using Linux can yield supercomputer performance for some programs that perform complex computations or operate on large data sets. What's more, it can do that using cheap hardware... which you might already own. As an added bonus, it is also easy to use a parallel Linux system for other things when it is not busy executing a parallel job.

If parallel processing is *not* what you want, but you would like to achieve at least a modest improvement

in performance, there are still things you can do. For example, you can improve performance of sequential programs by moving to a faster processor, adding memory, replacing an IDE disk with fa wide SCSI, etc. If that's all you are interested in, jump to section 6.2; otherwise, read on.

TOC

1.2 Terminology

Although parallel processing has been used for many years in many systems, it is still somewhat unfamiliar to most computer users. Thus, before discussing the various alternatives, it is important to become familiar with a few commonly used terms.

SIMD:

SIMD (Single Instruction stream, Multiple Data stream) refers to a parallel execution model in which all processors execute the same operation at the same time, but each processor is allowed to operate upon its own data. This model naturally fits the concept of performing the same operation on every element of an array, and is thus often associated with vector or array manipulation. Because all operations are inherently synchronized, interactions among SIMD processors tend to be easily and efficiently implemented.

MIMD:

MIMD (Multiple Instruction stream, Multiple Data stream) refers to a parallel execution model in which each processor is essentially acting independently. This model most naturally fits the concept of decomposing a program for parallel execution on a functional basis; for example, one processor might update a database file while another processor generates a graphic display of the new entry. This is a more flexible model than SIMD execution, but it is achieved at the risk of debugging nightmares called **race conditions**, in which a program may intermittently fail due to timing variations reordering the operations of one processor relative to those of another.

SPMD:

SPMD (Single Program, Multiple Data) is a restricted version of MIMD in which all processors are running the same program. Unlike SIMD, each processor executing SPMD code may take a different control flow path through the program.

Communication Bandwidth:

The bandwidth of a communication system is the maximum amount of data that can be transmitted in a unit of time... once data transmission has begun. Bandwidth for serial connections is often measured in **baud** or **bits/second (b/s)**, which generally correspond to 1/10 to 1/8 that many **Bytes/second (B/s)**. For example, a 1,200 baud modem transfers about 120 B/s, whereas a 155 Mb/s ATM network connection is nearly 130,000 times faster, transferring about about 17 MB/s. High bandwidth allows large blocks of data to be transferred efficiently between processors.

Communication Latency:

The latency of a communication system is the minimum time taken to transmit one object, including any send and receive software overhead. Latency is very important in parallel processing because it determines the minimum useful **grain size**, the minimum run time for a segment of code to yield speed-up through parallel execution. Basically, if a segment of code runs for less time than it takes to transmit its result value (i.e., latency), executing that code segment serially on the

processor that needed the result value would be faster than parallel execution; serial execution would avoid the communication overhead.

Message Passing:

Message passing is a model for interactions between processors within a parallel system. In general, a message is constructed by software on one processor and is sent through an interconnection network to another processor, which then must accept and act upon the message contents. Although the overhead in handling each message (latency) may be high, there are typically few restrictions on how much information each message may contain. Thus, message passing can yield high bandwidth making it a very effective way to transmit a large block of data from one processor to another. However, to minimize the need for expensive message passing operations, data structures within a parallel program must be spread across the processors so that most data referenced by each processor is in its local memory... this task is known as **data layout**.

Shared Memory:

Shared memory is a model for interactions between processors within a parallel system. Systems like the multi-processor Pentium machines running Linux **physically** share a single memory among their processors, so that a value written to shared memory by one processor can be directly accessed by any processor. Alternatively, **logically** shared memory can be implemented for systems in which each processor has it own memory by converting each non-local memory reference into an appropriate inter-processor communication. Either implementation of shared memory is generally considered easier to use than message passing. Physically shared memory can have both high bandwidth and low latency, but only when multiple processors do not try to access the bus simultaneously; thus, data layout still can seriously impact performance, and cache effects, etc., can make it difficult to determine what the best layout is.

Aggregate Functions:

In both the message passing and shared memory models, a communication is initiated by a single processor; in contrast, aggregate function communication is an inherently parallel communication model in which an entire group of processors act together. The simplest such action is a **barrier synchronization**, in which each individual processor waits until every processor in the group has arrived at the barrier. By having each processor output a datum as a side-effect of reaching a barrier, it is possible to have the communication hardware return a value to each processor which is an arbitrary function of the values collected from all processors. For example, the return value might be the answer to the question "did any processor find a solution?" or it might be the sum of one value from each processor. Latency can be very low, but bandwidth per processor also tends to be low. Traditionally, this model is used primarily to control parallel execution rather than to distribute data values.

Collective Communication:

This is another name for aggregate functions, most often used when referring to aggregate functions that are constructed using multiple message-passing operations.

SMP:

SMP (Symmetric Multi-Processor) refers to the operating system concept of a group of processors working together as peers, so that any piece of work could be done equally well by any processor.

TOC

Typically, SMP implies the combination of MIMD and shared memory. In the IA32 world, SMP generally means compliant with MPS (the Intel MultiProcessor Specification); in the future, it may mean "Slot 2"....

SWAR:

SWAR (SIMD Within A Register) is a generic term for the concept of partitioning a register into multiple integer fields and using register-width operations to perform SIMD-parallel computations across those fields. Given a machine with *k*-bit registers, data paths, and function units, it has long been known that ordinary register operations can function as SIMD parallel operations on as many as *n*, *k*/*n*-bit, field values. Although this type of parallelism can be implemented using ordinary integer registers and instructions, many high-end microprocessors have recently added specialized instructions to enhance the performance of this technique for multimedia-oriented tasks. In addition to the Intel/AMD/Cyrix **MMX** (MultiMedia eXtensions), there are: Digital Alpha **MAX** (MultimediA eXtensions), Hewlett-Packard PA-RISC **MAX** (Multimedia Acceleration eXtensions), MIPS **MDMX** (Digital Media eXtension, pronounced "Mad Max"), and Sun SPARC V9 **VIS** (Visual Instruction Set). Aside from the three vendors who have agreed on MMX, all of these instruction set extensions are roughly comparable, but mutually incompatible.

Attached Processors:

Attached processors are essentially special-purpose computers that are connected to a **host** system to accelerate specific types of computation. For example, many video and audio cards for PCs contain attached processors designed, respectively, to accelerate common graphics operations and audio **DSP** (Digital Signal Processing). There is also a wide range of attached **array processors**, so called because they are designed to accelerate arithmetic operations on arrays. In fact, many commercial supercomputers are really attached processors with workstation hosts.

RAID:

RAID (Redundant Array of Inexpensive Disks) is a simple technology for increasing both the bandwidth and reliability of disk I/O. Although there are many different variations, all have two key concepts in common. First, each data block is **striped** across a group of $n+k$ disk drives such that each drive only has to read or write $1/n$ of the data... yielding n times the bandwidth of one drive. Second, redundant data is written so that data can be recovered if a disk drive fails; this is important because otherwise if any one of the $n+k$ drives were to fail, the entire file system could be lost. A good overview of RAID in general is given at http://www.dpt.com/uraiddoc.html, and information about RAID options for Linux systems is at http://linas.org/linux/raid.html. Aside from specialized RAID hardware support, Linux also supports software RAID 0, 1, 4, and 5 across multiple disks hosted by a single Linux system; see the Software RAID mini-HOWTO and the Multi-Disk System Tuning mini-HOWTO for details. RAID across disk drives *on multiple machines in a cluster* is not directly supported.

IA32:

IA32 (Intel Architecture, 32-bit) really has nothing to do with parallel processing, but rather refers to the class of processors whose instruction sets are generally compatible with that of the Intel 386. Basically, any Intel x86 processor after the 286 is compatible with the 32-bit flat memory model that characterizes IA32. AMD and Cyrix also make a multitude of IA32-compatible processors. Because Linux evolved primarily on IA32 processors and that is where the commodity market is

centered, it is convenient to use IA32 to distinguish any of these processors from the PowerPC, Alpha, PA-RISC, MIPS, SPARC, etc. The upcoming IA64 (64-bit with EPIC, Explicitly Parallel Instruction Computing) will certainly complicate matters, but Merced, the first IA64 processor, is not scheduled for production until 1999.

COTS:

Since the demise of many parallel supercomputer companies, COTS (Commercial Off-The-Shelf) is commonly discussed as a requirement for parallel computing systems. Being fanatically pure, the only COTS parallel processing techniques using PCs are things like SMP Windows NT servers and various MMX Windows applications; it really doesn't pay to be that fanatical. The underlying concept of COTS is really minimization of development time and cost. Thus, a more useful, more common, meaning of COTS is that at least most subsystems benefit from commodity marketing, but other technologies are used where they are effective. Most often, COTS parallel processing refers to a cluster in which the nodes are commodity PCs, but the network interface and software are somewhat customized... typically running Linux and applications codes that are freely available (e.g., copyleft or public domain), but not literally COTS.

1.3 Example Algorithm

In order to better understand the use of the various parallel programming approaches outlined in this HOWTO, it is useful to have an example problem. Although just about any simple parallel algorithm would do, by selecting an algorithm that has been used to demonstrate various other parallel programming systems, it becomes a bit easier to compare and contrast approaches. M. J. Quinn's book, *Parallel Computing Theory And Practice*, second edition, McGraw Hill, New York, 1994, uses a parallel algorithm that computes the value of Pi to demonstrate a variety of different parallel supercomputer programming environments (e.g., nCUBE message passing, Sequent shared memory). In this HOWTO, we use the same basic algorithm.

The algorithm computes the approximate value of Pi by summing the area under *x* squared. As a purely sequential C program, the algorithm looks like:

```
#include <stdlib.h>;
#include <stdio.h>;

main(int argc, char **argv)
{
  register double width, sum;
  register int intervals, i;

  /* get the number of intervals */
  intervals = atoi(argv[1]);
  width = 1.0 / intervals;

  /* do the computation */
```

```
  sum = 0;
  for (i=0; i<intervals; ++i) {
    register double x = (i + 0.5) * width;
    sum += 4.0 / (1.0 + x * x);
  }
  sum *= width;

  printf("Estimation of pi is %f\n", sum);

  return(0);
}
```

However, this sequential algorithm easily yields an "embarrassingly parallel" implementation. The area is subdivided into intervals, and any number of processors can each independently sum the intervals assigned to it, with no need for interaction between processors. Once the local sums have been computed, they are added together to create a global sum; this step requires some level of coordination and communication between processors. Finally, this global sum is printed by one processor as the approximate value of Pi.

In this HOWTO, the various parallel implementations of this algorithm appear where each of the different programming methods is discussed.

1.4 Organization Of This Document

The remainder of this document is divided into five parts. Sections 2, 3, 4, and 5 correspond to the three different types of hardware configurations supporting parallel processing using Linux:

- Section 2 discusses SMP Linux systems. These directly support MIMD execution using shared memory, although message passing also is implemented easily. Although Linux supports SMP configurations up to 16 processors, most SMP PC systems have either two or four identical processors.
- Section 3 discusses clusters of networked machines, each running Linux. A cluster can be used as a parallel processing system that directly supports MIMD execution and message passing, perhaps also providing logically shared memory. Simulated SIMD execution and aggregate function communication also can be supported, depending on the networking method used. The number of processors in a cluster can range from two to thousands, primarily limited by the physical wiring constraints of the network. In some cases, various types of machines can be mixed within a cluster; for example, a network combining DEC Alpha and Pentium Linux systems would be a **heterogeneous cluster**.
- Section 4 discusses SWAR, SIMD Within A Register. This is a very restrictive type of parallel execution model, but on the other hand, it is a built-in capability of ordinary processors. Recently, MMX (and other) instruction set extensions to modern processors have made this approach even more effective.
- Section 5 discusses the use of Linux PCs as hosts for simple parallel computing systems. Either as an add-in card or as an external box, attached processors can provide a Linux system with

formidable processing power for specific types of applications. For example, inexpensive ISA cards are available that provide multiple DSP processors offering hundreds of MFLOPS for compute-bound problems. However, these add-in boards are *just* processors; they generally do not run an OS, have disk or console I/O capability, etc. To make such systems useful, the Linux "host" must provide these functions.

The final section of this document covers aspects that are of general interest for parallel processing using Linux, not specific to a particular one of the approaches listed above.

As you read this document, keep in mind that we haven't tested everything, and a lot of stuff reported here "still has a research character" (a nice way to say "doesn't quite work like it should" ;-). However, parallel processing using Linux is useful now, and an increasingly large group is working to make it better.

The author of this HOWTO is Hank Dietz, Ph.D., currently Associate Professor of Electrical and Computer Engineering at Purdue University, in West Lafayette, IN, 47907-1285. Dietz retains rights to this document as per the Linux Documentation Project guidelines. Although an effort has been made to ensure the correctness and fairness of this presentation, neither Dietz nor Purdue University can be held responsible for any problems or errors, and Purdue University does not endorse any of the work/products discussed.

---

2. SMP Linux

This document gives a brief overview of how to use SMP Linux systems for parallel processing. The most up-to-date information on SMP Linux is probably available via the SMP Linux project mailing list; send email to majordomo@vger.rutgers.edu with the text `subscribe linux-smp` to join the list.

Does SMP Linux really work? In June 1996, I purchased a brand new (well, new off-brand ;-) two-processor 100MHz Pentium system. The fully assembled system, including both processors, Asus motherboard, 256K cache, 32M RAM, 1.6G disk, 6X CDROM, Stealth 64, and 15" Acer monitor, cost a total of $1,800. This was just a few hundred dollars more than a comparable uniprocessor system. Getting SMP Linux running was simply a matter of installing the "stock" uniprocessor Linux, recompiling the kernel with the `SMP=1` line in the makefile uncommented (although I find setting `SMP` to `1` a bit ironic ;-), and informing `lilo` about the new kernel. This system performs well enough, and has been stable enough, to serve as my primary workstation ever since. In summary, SMP Linux really does work.

The next question is how much high-level support is available for writing and executing shared memory parallel programs under SMP Linux. Through early 1996, there wasn't much. Things have changed. For example, there is now a very complete POSIX threads library.

Although performance may be lower than for native shared-memory mechanisms, an SMP Linux system also can use most parallel processing software that was originally developed for a workstation cluster using socket communication. Sockets (see section 3.3) work within an SMP Linux system, and even for multiple SMPs networked as a cluster. However, sockets imply a lot of unnecessary overhead for an SMP. Much of that overhead is within the kernel or interrupt handlers; this worsens the problem because SMP Linux generally allows only one processor to be in the kernel at a time and the interrupt controller is set so that only the boot processor can process interrupts. Despite this, typical SMP communication hardware is so much better than most cluster networks that cluster software will often run better on an SMP than on the cluster for which it was designed.

The remainder of this section discusses SMP hardware, reviews the basic Linux mechanisms for sharing memory across the processes of a parallel program, makes a few observations about atomicity, volatility, locks, and cache lines, and finally gives some pointers to other shared memory parallel processing resources.

2.1 SMP Hardware

Although SMP systems have been around for many years, until very recently, each such machine tended to implement basic functions differently enough so that operating system support was not portable. The thing that has changed this situation is Intel's Multiprocessor Specification, often referred to as simply **MPS**. The MPS 1.4 specification is currently available as a PDF file at http://www.intel.com/design/pro/datashts/242016.htm, and there is a brief overview of MPS 1.1 at

TOC

http://support.intel.com/oem_developer/ial/support/9300.HTM, but be aware that Intel does re-arrange their WWW site often. A wide range of vendors are building MPS-compliant systems supporting up to four processors, but MPS theoretically allows many more processors.

The only non-MPS, non-IA32, systems supported by SMP Linux are Sun4m multiprocessor SPARC machines. SMP Linux supports most Intel MPS version 1.1 or 1.4 compliant machines with up to sixteen 486DX, Pentium, Pentium MMX, Pentium Pro, or Pentium II processors. Unsupported IA32 processors include the Intel 386, Intel 486SX/SLC processors (the lack of floating point hardware interferes with the SMP mechanisms), and AMD & Cyrix processors (they require different SMP support chips that do not seem to be available at this writing).

It is important to understand that the performance of MPS-compliant systems can vary widely. As expected, one cause for performance differences is processor speed: faster clock speeds tend to yield faster systems, and a Pentium Pro processor is faster than a Pentium. However, MPS does not really specify how hardware implements shared memory, but only how that implementation must function from a software point of view; this means that performance is also a function of how the shared memory implementation interacts with the characteristics of SMP Linux and your particular programs.

The primary way in which systems that comply with MPS differ is in how they implement access to physically shared memory.

Does each processor have its own L2 cache?

Some MPS Pentium systems, and all MPS Pentium Pro and Pentium II systems, have independent L2 caches. (The L2 cache is packaged within the Pentium Pro or Pentium II modules.) Separate L2 caches are generally viewed as maximizing compute performance, but things are not quite so obvious under Linux. The primary complication is that the current SMP Linux scheduler does not attempt to keep each process on the same processor, a concept known as **processor affinity**. This may change soon; there has recently been some discussion about this in the SMP Linux development community under the title "processor binding." Without processor affinity, having separate L2 caches may introduce significant overhead when a process is given a timeslice on a processor other than the one that was executing it last.

Many relatively inexpensive systems are organized so that two Pentium processors share a single L2 cache. The bad news is that this causes contention for the cache, seriously degrading performance when running multiple independent sequential programs. The good news is that many parallel programs might actually benefit from the shared cache because if both processors will want to access the same line from shared memory, only one had to fetch it into cache and contention for the bus is averted. The lack of processor affinity also causes less damage with a shared L2 cache. Thus, for parallel programs, it isn't really clear that sharing L2 cache is as harmful as one might expect.

Experience with our dual Pentium shared 256K cache system shows quite a wide range of performance depending on the level of kernel activity required. At worst, we see only about 1.2x speedup. However, we also have seen up to 2.1x speedup, which suggests that compute-intensive SPMD-style code really does profit from the "shared fetch" effect.

TOC

Bus configuration?

The first thing to say is that most modern systems connect the processors to one or more PCI buses that in turn are "bridged" to one or more ISA/EISA buses. These bridges add latency, and both EISA and ISA generally offer lower bandwidth than PCI (ISA being the lowest), so disk drives, video cards, and other high-performance devices generally should be connected via a PCI bus interface.

Although an MPS system can achieve good speed-up for many compute-intensive parallel programs even if there is only one PCI bus, I/O operations occur at no better than uniprocessor performance... and probably a little worse due to bus contention from the processors. Thus, if you are looking to speed-up I/O, make sure that you get an MPS system with multiple independent PCI busses and I/O controllers (e.g., multiple SCSI chains). You will need to be careful to make sure SMP Linux supports what you get. Also keep in mind that the current SMP Linux essentially allows only one processor in the kernel at any time, so you should choose your I/O controllers carefully to pick ones that minimize the kernel time required for each I/O operation. For really high performance, you might even consider doing raw device I/O directly from user processes, without a system call... this isn't necessarily as hard as it sounds, and need not compromise security (see section 3.3 for a description of the basic techniques).

It is important to note that the relationship between bus speed and processor clock rate has become very fuzzy over the past few years. Although most systems now use the same PCI clock rate, it is not uncommon to find a faster processor clock paired with a slower bus clock. The classic example of this was that the Pentium 133 generally used a faster bus than a Pentium 150, with appropriately strange-looking performance on various benchmarks. These effects are amplified in SMP systems; it is even more important to have a faster bus clock.

Memory interleaving and DRAM technologies?

Memory interleaving actually has nothing whatsoever to do with MPS, but you will often see it mentioned for MPS systems because these systems are typically more demanding of memory bandwidth. Basically, two-way or four-way interleaving organizes RAM so that a block access is accomplished using multiple banks of RAM rather than just one. This provides higher memory access bandwidth, particularly for cache line loads and stores.

The waters are a bit muddied about this, however, because EDO DRAM and various other memory technologies tend to improve similar kinds of operations. An excellent overview of DRAM technologies is given in http://www.pcguide.com/ref/ram/tech.htm.

So, for example, is it better to have 2-way interleaved EDO DRAM or non-interleaved SDRAM? That is a very good question with no simple answer, because both interleaving and exotic DRAM technologies tend to be expensive. The same dollar investment in more ordinary memory configurations generally will give you a significantly larger main memory. Even the slowest DRAM is still a heck of a lot faster than using disk-based virtual memory....

2.2 Introduction To Shared Memory Programming

TOC

Ok, so you have decided that parallel processing on an SMP is a great thing to do... how do you get started? Well, the first step is to learn a little bit about how shared memory communication really works.

It sounds like you simply have one processor store a value into memory and another processor load it; unfortunately, it isn't quite that simple. For example, the relationship between processes and processors is very blurry; however, if we have no more active processes than there are processors, the terms are roughly interchangeable. The remainder of this section briefly summarizes the key issues that could cause serious problems, if you were not aware of them: the two different models used to determine what is shared, atomicity issues, the concept of volatility, hardware lock instructions, cache line effects, and Linux scheduler issues.

Shared Everything Vs. Shared Something

There are two fundamentally different models commonly used for shared memory programming: **shared everything** and **shared something**. Both of these models allow processors to communicate by loads and stores from/into shared memory; the distinction comes in the fact that shared everything places all data structures in shared memory, while shared something requires the user to explicitly indicate which data structures are potentially shared and which are **private** to a single processor.

Which shared memory model should you use? That is mostly a question of religion. A lot of people like the shared everything model because they do not really need to identify which data structures should be shared at the time they are declared... you simply put locks around potentially-conflicting accesses to shared objects to ensure that only one process(or) has access at any moment. Then again, that really isn't all that simple... so many people prefer the relative safety of shared something.

Shared Everything

The nice thing about sharing everything is that you can easily take an existing sequential program and incrementally convert it into a shared everything parallel program. You do not have to first determine which data need to be accessible by other processors.

Put simply, the primary problem with sharing everything is that any action taken by one processor could affect the other processors. This problem surfaces in two ways:

- Many libraries use data structures that simply are not sharable. For example, the UNIX convention is that most functions can return an error code in a variable called `errno`; if two shared everything processes perform various calls, they would interfere with each other because they share the same `errno`. Although there is now a library version that fixes the `errno` problem, similar problems still exist in most libraries. For example, unless special precautions are taken, the X library will not work if calls are made from multiple shared everything processes.
- Normally, the worst-case behavior for a program with a bad pointer or array subscript is that the process that contains the offending code dies. It might even generate a `core` file that clues you in to what happened. In shared everything parallel processing, it is very likely that the stray accesses will bring the demise of *a process other than the one at fault*, making it nearly impossible to

localize and correct the error.

Neither of these types of problems is common when shared something is used, because only the explicitly-marked data structures are shared. It also is fairly obvious that shared everything only works if all processors are executing the exact same memory image; you cannot use shared everything across multiple different code images (i.e., can use only SPMD, not general MIMD).

The most common type of shared everything programming support is a **threads library**. Threads are essentially "light-weight" processes that might not be scheduled in the same way as regular UNIX processes and, most importantly, share access to a single memory map. The POSIX Pthreads package has been the focus of a number of porting efforts; the big question is whether any of these ports actually run the threads of a program in parallel under SMP Linux (ideally, with a processor for each thread). The POSIX API doesn't require it, and versions like http://www.aa.net/~mtp/PCthreads.html apparently do not implement parallel thread execution - all the threads of a program are kept within a single Linux process.

The first threads library that supported SMP Linux parallelism was the now somewhat obsolete bb_threads library, ftp://caliban.physics.utoronto.ca/pub/linux/, a very small library that used the Linux `clone()` call to fork new, independently scheduled, Linux processes all sharing a single address space. SMP Linux machines can run multiple of these "threads" in parallel because each "thread" is a full Linux process; the trade-off is that you do not get the same "light-weight" scheduling control provided by some thread libraries under other operating systems. The library used a bit of C-wrapped assembly code to install a new chunk of memory as each thread's stack and to provide atomic access functions for an array of locks (mutex objects). Documentation consisted of a `README` and a short sample program.

More recently, a version of POSIX threads using `clone()` has been developed. This library, LinuxThreads, is clearly the preferred shared everything library for use under SMP Linux. POSIX threads are well documented, and the LinuxThreads README and LinuxThreads FAQ are very well done. The primary problem now is simply that POSIX threads have a lot of details to get right and LinuxThreads is still a work in progress. There is also the problem that the POSIX thread standard has evolved through the standardization process, so you need to be a bit careful not to program for obsolete early versions of the standard.

Shared Something

Shared something is really "only share what needs to be shared." This approach can work for general MIMD (not just SPMD) provided that care is taken for the shared objects to be allocated at the same places in each processor's memory map. More importantly, shared something makes it easier to predict and tune performance, debug code, etc. The only problems are:

- It can be hard to know beforehand what really needs to be shared.
- The actual allocation of objects in shared memory may be awkward, especially for what would have been stack-allocated objects. For example, it may be necessary to explicitly allocate shared objects in a separate memory segment, requiring separate memory allocation routines and introducing extra pointer indirections in each reference.

Currently, there are two very similar mechanisms that allow groups of Linux processes to have

independent memory spaces, all sharing only a relatively small memory segment. Assuming that you didn't foolishly exclude "System V IPC" when you configured your Linux system, Linux supports a very portable mechanism that has generally become known as "System V Shared Memory." The other alternative is a memory mapping facility whose implementation varies widely across different UNIX systems: the `mmap()` system call. You can, and should, learn about these calls from the manual pages... but a brief overview of each is given in sections 2.5 and 2.6 to help get you started.

Atomicity And Ordering

No matter which of the above two models you use, the result is pretty much the same: you get a pointer to a chunk of read/write memory that is accessible by all processes within your parallel program. Does that mean I can just have my parallel program access shared memory objects as though they were in ordinary local memory? Well, not quite....

Atomicity refers to the concept that an operation on an object is accomplished as an indivisible, uninterruptible, sequence. Unfortunately, sharing memory access does not imply that all operations on data in shared memory occur atomically. Unless special precautions are taken, only simple load or store operations that occur within a single bus transaction (i.e., aligned 8, 16, or 32-bit operations, but not misaligned nor 64-bit operations) are atomic. Worse still, "smart" compilers like GCC will often perform optimizations that could eliminate the memory operations needed to ensure that other processors can see what this processor has done. Fortunately, both these problems can be remedied... leaving only the relationship between access efficiency and cache line size for us to worry about.

However, before discussing these issues, it is useful to point-out that all of this assumes that memory references for each processor happen in the order in which they were coded. The Pentium does this, but also notes that future Intel processors might not. So, for future processors, keep in mind that it may be necessary to surround some shared memory accesses with instructions that cause all pending memory accesses to complete, thus providing memory access ordering. The `CPUID` instruction apparently is reserved to have this side-effect.

Volatility

To prevent GCC's optimizer from buffering values of shared memory objects in registers, all objects in shared memory should be declared as having types with the `volatile` attribute. If this is done, all shared object reads and writes that require just one word access will occur atomically. For example, suppose that *p* is a pointer to an integer, where both the pointer and the integer it will point at are in shared memory; the ANSI C declaration might be:

```
volatile int * volatile p;
```

In this code, the first `volatile` refers to the `int` that `p` will eventually point at; the second `volatile` refers to the pointer itself. Yes, it is annoying, but it is the price one pays for enabling GCC to perform some very powerful optimizations. At least in theory, the `-traditional` option to GCC might suffice to produce correct code at the expense of some optimization, because pre-ANSI K&R C

essentially claimed that all variables were volatile unless explicitly declared as `register`. Still, if y typical GCC compile looks like `cc -O6 ...`, you really will want to explicitly mark things as vo only where necessary.

There has been a rumor to the effect that using assembly-language locks that are marked as modifying all processor registers will cause GCC to appropriately flush all variables, thus avoiding the "inefficient" compiled code associated with things declared as `volatile`. This hack appears to work for statically allocated global variables using version 2.7.0 of GCC... however, that behavior is *not* required by the ANSI C standard. Still worse, other processes that are making only read accesses can buffer the values in registers forever, thus *never* noticing that the shared memory value has actually changed. In summary, do what you want, but only variables accessed through `volatile` are *guaranteed* to work correctly.

Note that you can cause a volatile access to an ordinary variable by using a type cast that imposes the `volatile` attribute. For example, the ordinary `int i;` can be referenced as a volatile by `*((volatile int *) &i)`; thus, you can explicitly invoke the "overhead" of volatility only where it is critical.

Locks

If you thought that `++i;` would always work to add one to a variable `i` in shared memory, you've got a nasty little surprise coming: even if coded as a single instruction, the load and store of the result are separate memory transactions, and other processors could access `i` between these two transactions. For example, having two processes both perform `++i;` might only increment `i` by one, rather than by two. According to the Intel Pentium "Architecture and Programming Manual," the `LOCK` prefix can be used to ensure that any of the following instructions is atomic relative to the data memory location it accesses:

```
BTS, BTR, BTC                        mem, reg/imm
XCHG                                 reg, mem
XCHG                                 mem, reg
ADD, OR, ADC, SBB, AND, SUB, XOR     mem, reg/imm
NOT, NEG, INC, DEC                   mem
CMPXCHG, XADD
```

However, it probably is not a good idea to use all these operations. For example, `XADD` did not even exist for the 386, so coding it may cause portability problems.

The `XCHG` instruction *always* asserts a lock, even without the `LOCK` prefix, and thus is clearly the preferred atomic operation from which to build higher-level atomic constructs such as semaphores and shared queues. Of course, you can't get GCC to generate this instruction just by writing C code... instead, you must use a bit of in-line assembly code. Given a word-size volatile object *obj* and a word-size register value *reg*, the GCC in-line assembly code is:

```
__asm__ __volatile__ ("xchgl %1,%0"
```

```
:"=r" (reg), "=m" (obj)
:"r" (reg), "m" (obj));
```


Examples of GCC in-line assembly code using bit operations for locking are given in the source code for the bb_threads library.

It is important to remember, however, that there is a cost associated with making memory transactions atomic. A locking operation carries a fair amount of overhead and may delay memory activity from other processors, whereas ordinary references may use local cache. The best performance results when locking operations are used as infrequently as possible. Further, these IA32 atomic instructions obviously are not portable to other systems.

There are many alternative approaches that allow ordinary instructions to be used to implement various synchronizations, including **mutual exclusion** - ensuring that at most one processor is updating a given shared object at any moment. Most OS textbooks discuss at least one of these techniques. There is a fairly good discussion in the Fourth Edition of *Operating System Concepts*, by Abraham Silberschatz and Peter B. Galvin, ISBN 0-201-50480-4.

Cache Line Size

One more fundamental atomicity concern can have a dramatic impact on SMP performance: cache line size. Although the MPS standard requires references to be coherent no matter what caching is used, the fact is that when one processor writes to a particular line of memory, every cached copy of the old line must be invalidated or updated. This implies that if two or more processors are both writing data to different portions of the same line a lot of cache and bus traffic may result, effectively to pass the line from cache to cache. This problem is known as **false sharing**. The solution is simply to try to *organize data so that what is accessed in parallel tends to come from a different cache line for each process*.

You might be thinking that false sharing is not a problem using a system with a shared L2 cache, but remember that there are still separate L1 caches. Cache organization and number of separate levels can both vary, but the Pentium L1 cache line size is 32 bytes and typical external cache line sizes are around 256 bytes. Suppose that the addresses (physical or virtual) of two items are *a* and *b* and that the largest per-processor cache line size is *c*, which we assume to be a power of two. To be very precise, if `((int) a) & ~(c - 1)` is equal to `((int) b) & ~(c - 1)`, then both references are in the same cache line. A simpler rule is that if shared objects being referenced in parallel are at least *c* bytes apart, they should map to different cache lines.

Linux Scheduler Issues

Although the whole point of using shared memory for parallel processing is to avoid OS overhead, OS overhead can come from things other than communication per se. We have already said that the number of processes that should be constructed is less than or equal to the number of processors in the machine. But how do you decide exactly how many processes to make?

For best performance, *the number of processes in your parallel program should be equal to the expected number of your program's processes that simultaneously can be running on different processors*. For

example, if a four-processor SMP typically has one process actively running for some other purpose (e.g., a WWW server), then your parallel program should use only three processes. You can get a rough idea of how many other processes are active on your system by looking at the "load average" quoted by the `uptime` command.

Alternatively, you could boost the priority of the processes in your parallel program using, for example, the `renice` command or `nice()` system call. You must be privileged to increase priority. The idea is simply to force the other processes out of processors so that your program can run simultaneously across all processors. This can be accomplished somewhat more explicitly using the prototype version of SMP Linux at http://luz.cs.nmt.edu/~rtlinux/, which offers real-time schedulers.

If you are not the only user treating your SMP system as a parallel machine, you may also have conflicts between the two or more parallel programs trying to execute simultaneously. This standard solution is **gang scheduling** - i.e., manipulating scheduling priority so that at any given moment, only the processes of a single parallel program are running. It is useful to recall, however, that using more parallelism tends to have diminishing returns and scheduler activity adds overhead. Thus, for example, it is probably better for a four-processor machine to run two programs with two processes each rather than gang scheduling between two programs with four processes each.

There is one more twist to this. Suppose that you are developing a program on a machine that is heavily used all day, but will be fully available for parallel execution at night. You need to write and test your code for correctness with the full number of processes, even though you know that your daytime test runs will be slow. Well, they will be *very* slow if you have processes **busy waiting** for shared memory values to be changed by other processes that are not currently running (on other processors). The same problem occurs if you develop and test your code on a single-processor system.

The solution is to embed calls in your code, wherever it may loop awaiting an action from another processor, so that Linux will give another process a chance to run. I use a C macro, call it `IDLE_ME`, to do this: for a test run, compile with `cc -DIDLE_ME=usleep(1); ...`; for a "production" run, compile with `cc -DIDLE_ME={} ...`. The `usleep(1)` call requests a 1 microsecond sleep, which has the effect of allowing the Linux scheduler to select a different process to run on that processor. If the number of processes is more than twice the number of processors available, it is not unusual for codes to run ten times faster with `usleep(1)` calls than without them.

2.3 bb_threads

The bb_threads ("Bare Bones" threads) library, ftp://caliban.physics.utoronto.ca/pub/linux/, is a remarkably simple library that demonstrates use of the Linux `clone()` call. The `gzip tar` file is only 7K bytes! Although this library is essentially made obsolete by the LinuxThreads library discussed in section 2.4, bb_threads is still usable, and it is small and simple enough to serve well as an introduction to use of Linux thread support. Certainly, it is far less daunting to read this source code than to browse the source code for LinuxThreads. In summary, the bb_threads library is a good starting point, but is not really suitable for coding large projects.

The basic program structure for using the bb_threads library is:

1. Start the program running as a single process.

2. You will need to estimate the maximum stack space that will be required for each thread. Gue large is relatively harmless (that is what virtual memory is for ;-), but remember that *all* the st are coming from a single virtual address space, so guessing huge is not a great idea. The demo suggests 64K. This size is set to *b* bytes by `bb_threads_stacksize(b)`.
3. The next step is to initialize any locks that you will need. The lock mechanism built-into this library numbers locks from 0 to `MAX_MUTEXES`, and initializes lock *i* by `bb_threads_mutexcreate(i)`.
4. Spawning a new thread is done by calling a library routine that takes arguments specifying what function the new thread should execute and what arguments should be transmitted to it. To start a new thread executing the `void`-returning function *f* with the single argument *arg*, you do something like `bb_threads_newthread(f, &arg)`, where *f* should be declared something like `void f(void *arg, size_t dummy)`. If you need to pass more than one argument, pass a pointer to a structure initialized to hold the argument values.
5. Run parallel code, being careful to use `bb_threads_lock(n)` and `bb_threads_unlock(n)` where *n* is an integer identifying which lock to use. Note that the lock and unlock operations in this library are very basic spin locks using atomic bus-locking instructions, which can cause excessive memory-reference interference and do not make any attempt to ensure fairness. The demo program packaged with bb_threads did not correctly use locks to prevent `printf()` from being executed simultaneously from within the functions `fnn` and `main`... and because of this, the demo does not always work. I'm not saying this to knock the demo, but rather to emphasize that this stuff is *very tricky*; also, it is only slightly easier using LinuxThreads.
6. When a thread executes a `return`, it actually destroys the process... but the local stack memory is not automatically deallocated. To be precise, Linux doesn't support deallocation, but the memory space is not automatically added back to the `malloc()` free list. Thus, the parent process should reclaim the space for each dead child by `bb_threads_cleanup(wait(NULL))`.

The following C program uses the algorithm discussed in section 1.3 to compute the approximate value of Pi using two bb_threads threads.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include "bb_threads.h"

volatile double pi = 0.0;
volatile int intervals;
volatile int pids[2];      /* Unix PIDs of threads */

void
do_pi(void *data, size_t len)
```

```
{
  register double width, localsum;
  register int i;
  register int iproc = (getpid() != pids[0]);

  /* set width */
  width = 1.0 / intervals;

  /* do the local computations */
  localsum = 0;
  for (i=iproc; i<intervals; i+=2) {
    register double x = (i + 0.5) * width;
    localsum += 4.0 / (1.0 + x * x);
  }
  localsum *= width;

  /* get permission, update pi, and unlock */
  bb_threads_lock(0);
  pi += localsum;
  bb_threads_unlock(0);
}

int
main(int argc, char **argv)
{
  /* get the number of intervals */
  intervals = atoi(argv[1]);

  /* set stack size and create lock... */
  bb_threads_stacksize(65536);
  bb_threads_mutexcreate(0);

  /* make two threads... */
  pids[0] = bb_threads_newthread(do_pi, NULL);
  pids[1] = bb_threads_newthread(do_pi, NULL);

  /* cleanup after two threads (really a barrier sync) */
  bb_threads_cleanup(wait(NULL));
  bb_threads_cleanup(wait(NULL));

  /* print the result */
  printf("Estimation of pi is %f\n", pi);

  /* check-out */
  exit(0);
}
```

TOC

2.4 LinuxThreads

LinuxThreads http://pauillac.inria.fr/~xleroy/linuxthreads/ is a fairly complete and solid implementation of "shared everything" as per the POSIX 1003.1c threads standard. Unlike other POSIX threads ports, LinuxThreads uses the same Linux kernel threads facility (`clone()`) that is used by bb_threads. POSIX compatibility means that it is relatively easy to port quite a few threaded applications from other systems and various tutorial materials are available. In short, this is definitely the threads package to use under Linux for developing large-scale threaded programs.

The basic program structure for using the LinuxThreads library is:

1. Start the program running as a single process.
2. The next step is to initialize any locks that you will need. Unlike bb_threads locks, which are identified by numbers, POSIX locks are declared as variables of type `pthread_mutex_t lock`. Use `pthread_mutex_init(&lock,val)` to initialize each one you will need to use.
3. As with bb_threads, spawning a new thread is done by calling a library routine that takes arguments specifying what function the new thread should execute and what arguments should be transmitted to it. However, POSIX requires the user to declare a variable of type `pthread_t` to identify each thread. To create a thread `pthread_t thread` running `f()`, one calls `pthread_create(&thread,NULL,f,&arg)`.
4. Run parallel code, being careful to use `pthread_mutex_lock(&lock)` and `pthread_mutex_unlock(&lock)` as appropriate.
5. Use `pthread_join(thread,&retval)` to clean-up after each thread.
6. Use `-D_REENTRANT` when compiling your C code.

An example parallel computation of Pi using LinuxThreads follows. The algorithm of section 1.3 is used and, as for the bb_threads example, two threads execute in parallel.

```
#include <stdio.h>
#include <stdlib.h>
#include "pthread.h"

volatile double pi = 0.0;  /* Approximation to pi (shared) */
pthread_mutex_t pi_lock;   /* Lock for above */
volatile double intervals; /* How many intervals? */

void *
process(void *arg)
{
  register double width, localsum;
  register int i;
  register int iproc = (*((char *) arg) - '0');
```

```
  /* Set width */
  width = 1.0 / intervals;

  /* Do the local computations */
  localsum = 0;
  for (i=iproc; i<intervals; i+=2) {
    register double x = (i + 0.5) * width;
    localsum += 4.0 / (1.0 + x * x);
  }
  localsum *= width;

  /* Lock pi for update, update it, and unlock */
  pthread_mutex_lock(&pi_lock);
  pi += localsum;
  pthread_mutex_unlock(&pi_lock);

  return(NULL);
}

int
main(int argc, char **argv)
{
  pthread_t thread0, thread1;
  void * retval;

  /* Get the number of intervals */
  intervals = atoi(argv[1]);

  /* Initialize the lock on pi */
  pthread_mutex_init(&pi_lock, NULL);

  /* Make the two threads */
  if (pthread_create(&thread0, NULL, process, "0") ||
      pthread_create(&thread1, NULL, process, "1")) {
    fprintf(stderr, "%s: cannot make thread\n", argv[0]);
    exit(1);
  }

  /* Join (collapse) the two threads */
  if (pthread_join(thread0, &retval) ||
      pthread_join(thread1, &retval)) {
    fprintf(stderr, "%s: thread join failed\n", argv[0]);
    exit(1);
  }
```

```
  /* Print the result */
  printf("Estimation of pi is %f\n", pi);

  /* Check-out */
  exit(0);
}
```

2.5 System V Shared Memory

The System V IPC (Inter-Process Communication) support consists of a number of system calls providing message queues, semaphores, and a shared memory mechanism. Of course, these mechanisms were originally intended to be used for multiple processes to communicate within a uniprocessor system. However, that implies that it also should work to communicate between processes under SMP Linux, no matter which processors they run on.

Before going into how these calls are used, it is important to understand that although System V IPC calls exist for things like semaphores and message transmission, you probably should not use them. Why not? These functions are generally slow and serialized under SMP Linux. Enough said.

The basic procedure for creating a group of processes sharing access to a shared memory segment is:

1. Start the program running as a single process.
2. Typically, you will want each run of a parallel program to have its own shared memory segment, so you will need to call `shmget()` to create a new segment of the desired size. Alternatively, this call can be used to get the ID of a pre-existing shared memory segment. In either case, the return value is either the shared memory segment ID or -1 for error. For example, to create a shared memory segment of *b* bytes, the call might be `shmid = shmget(IPC_PRIVATE, b, (IPC_CREAT | 0666))`.
3. The next step is to attach this shared memory segment to this process, literally adding it to the virtual memory map of this process. Although the `shmat()` call allows the programmer to specify the virtual address at which the segment should appear, the address selected must be aligned on a page boundary (i.e., be a multiple of the page size returned by `getpagesize()`, which is usually 4096 bytes), and will override the mapping of any memory formerly at that address. Thus, we instead prefer to let the system pick the address. In either case, the return value is a pointer to the base virtual address of the segment just mapped. The code is `shmptr = shmat(shmid, 0, 0)`. Notice that you can allocate all your static shared variables into this shared memory segment by simply declaring all shared variables as members of a `struct` type, and declaring *shmptr* to be a pointer to that type. Using this technique, shared variable *x* would be accessed as *shmptr*`->`*x*.
4. Since this shared memory segment should be destroyed when the last process with access to it terminates or detaches from it, we need to call `shmctl()` to set-up this default action. The code is something like `shmctl(shmid, IPC_RMID, 0)`.
5. Use the standard Linux `fork()` call to make the desired number of processes... each will inherit the shared memory segment.

TOC

6. When a process is done using a shared memory segment, it really should detach from that shar memory segment. This is done by `shmdt(shmptr)`.

Although the above set-up does require a few system calls, once the shared memory segment has been established, any change made by one processor to a value in that memory will automatically be visible to all processes. Most importantly, each communication operation will occur without the overhead of a system call.

An example C program using System V shared memory segments follows. It computes Pi, using the same algorithm given in section 1.3.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/ipc.h>
#include <sys/shm.h>

volatile struct shared { double pi; int lock; } *shared;

inline extern int xchg(register int reg,
volatile int * volatile obj)
{
  /* Atomic exchange instruction */
__asm__ __volatile__ ("xchgl %1,%0"
                      :"=r" (reg), "=m" (*obj)
                      :"r" (reg), "m" (*obj));
  return(reg);
}

main(int argc, char **argv)
{
  register double width, localsum;
  register int intervals, i;
  register int shmid;
  register int iproc = 0;;

  /* Allocate System V shared memory */
  shmid = shmget(IPC_PRIVATE,
                 sizeof(struct shared),
                 (IPC_CREAT | 0600));
  shared = ((volatile struct shared *) shmat(shmid, 0, 0));
  shmctl(shmid, IPC_RMID, 0);
```

```
  /* Initialize... */
  shared->pi = 0.0;
  shared->lock = 0;

  /* Fork a child */
  if (!fork()) ++iproc;

  /* get the number of intervals */
  intervals = atoi(argv[1]);
  width = 1.0 / intervals;

  /* do the local computations */
  localsum = 0;
  for (i=iproc; i<intervals; i+=2) {
    register double x = (i + 0.5) * width;
    localsum += 4.0 / (1.0 + x * x);
  }
  localsum *= width;

  /* Atomic spin lock, add, unlock... */
  while (xchg((iproc + 1), &(shared->lock))) ;
  shared->pi += localsum;
  shared->lock = 0;

  /* Terminate child (barrier sync) */
  if (iproc == 0) {
    wait(NULL);
    printf("Estimation of pi is %f\n", shared->pi);
  }

  /* Check out */
  return(0);
}
```

In this example, I have used the IA32 atomic exchange instruction to implement locking. For better performance and portability, substitute a synchronization technique that avoids atomic bus-locking instructions (discussed in section 2.2).

When debugging your code, it is useful to remember that the `ipcs` command will report the status of the System V IPC facilities currently in use.

2.6 Memory Map Call

TOC

Using system calls for file I/O can be very expensive; in fact, that is why there is a user-buffered file I/O library (`getchar()`, `fwrite()`, etc.). But user buffers don't work if multiple processes are accessing the same writeable file, and the user buffer management overhead is significant. The BSD UNIX fix for this was the addition of a system call that allows a portion of a file to be mapped into user memory, essentially using virtual memory paging mechanisms to cause updates. This same mechanism also has been used in systems from Sequent for many years as the basis for their shared memory parallel processing support. Despite some very negative comments in the (quite old) man page, Linux seems to correctly perform at least some of the basic functions, and it supports the degenerate use of this system call to map an anonymous segment of memory that can be shared across multiple processes.

In essence, the Linux implementation of `mmap()` is a plug-in replacement for steps 2, 3, and 4 in the System V shared memory scheme outlined in section 2.5. To create an anonymous shared memory segment:

```
shmptr =
    mmap(0,                          /* system assigns address */
         b,                          /* size of shared memory segment */
         (PROT_READ | PROT_WRITE),   /* access rights, can be rwx */
         (MAP_ANON | MAP_SHARED),    /* anonymous, shared */
         0,                          /* file descriptor (not used) */
         0);                         /* file offset (not used) */
```

The equivalent to the System V shared memory `shmdt()` call is `munmap()`:

```
munmap(shmptr, b);
```

In my opinion, there is no real benefit in using `mmap()` instead of the System V shared memory support.

3. Clusters Of Linux Systems

This section attempts to give an overview of cluster parallel processing using Linux. Clusters are currently both the most popular and the most varied approach, ranging from a conventional network of workstations (**NOW**) to essentially custom parallel machines that just happen to use Linux PCs as processor nodes. There is also quite a lot of software support for parallel processing using clusters of Linux machines.

3.1 Why A Cluster?

Cluster parallel processing offers several important advantages:

- Each of the machines in a cluster can be a complete system, usable for a wide range of other computing applications. This leads many people to suggest that cluster parallel computing can simply claim all the "wasted cycles" of workstations sitting idle on people's desks. It is not really so easy to salvage those cycles, and it will probably slow your co-worker's screen saver, but it can be done.
- The current explosion in networked systems means that most of the hardware for building a cluster is being sold in high volume, with correspondingly low "commodity" prices as the result. Further savings come from the fact that only one video card, monitor, and keyboard are needed for each cluster (although you may need to swap these into each machine to perform the initial installation of Linux, once running, a typical Linux PC does not need a "console"). In comparison, SMP and attached processors are much smaller markets, tending toward somewhat higher price per unit performance.
- Cluster computing can *scale to very large systems*. While it is currently hard to find a Linux-compatible SMP with many more than four processors, most commonly available network hardware easily builds a cluster with up to 16 machines. With a little work, hundreds or even thousands of machines can be networked. In fact, the entire Internet can be viewed as one truly huge cluster.
- The fact that replacing a "bad machine" within a cluster is trivial compared to fixing a partly faulty SMP yields much higher availability for carefully designed cluster configurations. This becomes important not only for particular applications that cannot tolerate significant service interruptions, but also for general use of systems containing enough processors so that single-machine failures are fairly common. (For example, even though the average time to failure of a PC might be two years, in a cluster with 32 machines, the probability that at least one will fail within 6 months is quite high.)

OK, so clusters are free or cheap and can be very large and highly available... why doesn't everyone use a cluster? Well, there are problems too:

- With a few exceptions, network hardware is not designed for parallel processing. Typically latency is very high and bandwidth relatively low compared to SMP and attached processors. For example,

TOC

SMP latency is generally no more than a few microseconds, but is commonly hundreds or thousands of microseconds for a cluster. SMP communication bandwidth is often more than 10 MBytes/second; although the fastest network hardware (e.g., "Gigabit Ethernet") offers comparable speed, the most commonly used networks are between 10 and 1000 times slower. The performance of network hardware is poor enough as an *isolated cluster network*. If the network is not isolated from other traffic, as is often the case using "machines that happen to be networked" rather than a system designed as a cluster, performance can be substantially worse.

- There is very little software support for treating a cluster as a single system. For example, the `ps` command only reports the processes running on one Linux system, not all processes running across a cluster of Linux systems.

Thus, the basic story is that clusters offer great potential, but that potential may be very difficult to achieve for most applications. The good news is that there is quite a lot of software support that will help you achieve good performance for programs that are well suited to this environment, and there are also networks designed specifically to widen the range of programs that can achieve good performance.

3.2 Network Hardware

Computer networking is an exploding field... but you already knew that. An ever-increasing range of networking technologies and products are being developed, and most are available in forms that could be applied to make a parallel-processing cluster out of a group of machines (i.e., PCs each running Linux).

Unfortunately, no one network technology solves all problems best; in fact, the range of approach, cost, and performance is at first hard to believe. For example, using standard commercially-available hardware, the cost per machine networked ranges from less than $5 to over $4,000. The delivered bandwidth and latency each also vary over four orders of magnitude.

Before trying to learn about specific networks, it is important to recognize that these things change like the wind (see http://www.uk.linux.org/NetNews.html for Linux networking news), and it is very difficult to get accurate data about some networks.

Where I was particularly uncertain, I've placed a *?*. I have spent a lot of time researching this topic, but I'm sure my summary is full of errors and has omitted many important things. If you have any corrections or additions, please send email to pplinux@ecn.purdue.edu.

Summaries like the LAN Technology Scorecard at http://web.syr.edu/~jmwobus/comfaqs/lan-technology.html give some characteristics of many different types of networks and LAN standards. However, the summary in this HOWTO centers on the network properties that are most relevant to construction of Linux clusters. The section discussing each network begins with a short list of characteristics. The following defines what these entries mean.

Linux support:

If the answer is *no*, the meaning is pretty clear. Other answers try to describe the basic program interface that is used to access the network. Most network hardware is interfaced via a kernel driver, typically supporting TCP/UDP communication. Some other networks use more direct (e.g., library) interfaces to reduce latency by bypassing the kernel.

TOC

Years ago, it used to be considered perfectly acceptable to access a floating point unit via an OS call, but that is now clearly ludicrous; in my opinion, it is just as awkward for each communication between processors executing a parallel program to require an OS call. The problem is that computers haven't yet integrated these communication mechanisms, so non-kernel approaches tend to have portability problems. You are going to hear a lot more about this in the near future, mostly in the form of the new **Virtual Interface (VI) Architecture**, http://www.viarch.org/, which is a standardized method for most network interface operations to bypass the usual OS call layers. The VI standard is backed by Compaq, Intel, and Microsoft, and is sure to have a strong impact on SAN (System Area Network) designs over the next few years.

Maximum bandwidth:

This is the number everybody cares about. I have generally used the theoretical best case numbers; your mileage *will* vary.

Minimum latency:

In my opinion, this is the number everybody should care about even more than bandwidth. Again, I have used the unrealistic best-case numbers, but at least these numbers do include *all* sources of latency, both hardware and software. In most cases, the network latency is just a few microseconds; the much larger numbers reflect layers of inefficient hardware and software interfaces.

Available as:

Simply put, this describes how you get this type of network hardware. Commodity stuff is widely available from many vendors, with price as the primary distinguishing factor. Multiple-vendor things are available from more than one competing vendor, but there are significant differences and potential interoperability problems. Single-vendor networks leave you at the mercy of that supplier (however benevolent it may be). Public domain designs mean that even if you cannot find somebody to sell you one, you or anybody else can buy parts and make one. Research prototypes are just that; they are generally neither ready for external users nor available to them.

Interface port/bus used:

How does one hook-up this network? The highest performance and most common now is a PCI bus interface card. There are also EISA, VESA local bus (VL bus), and ISA bus cards. ISA was there first, and is still commonly used for low-performance cards. EISA is still around as the second bus in a lot of PCI machines, so there are a few cards. These days, you don't see much VL stuff (although http://www.vesa.org/ would beg to differ).

Of course, any interface that you can use without having to open your PC's case has more than a little appeal. IrDA and USB interfaces are appearing with increasing frequency. The Standard Parallel Port (SPP) used to be what your printer was plugged into, but it has seen a lot of use lately as an external extension of the ISA bus; this new functionality is enhanced by the IEEE 1284 standard, which specifies EPP and ECP improvements. There is also the old, reliable, slow RS232 serial port. I don't know of anybody connecting machines using VGA video connectors, keyboard, mouse, or game ports... so that's about it.

Network structure:

A bus is a wire, set of wires, or fiber. A hub is a little box that knows how to connect different wires/fibers plugged into it; switched hubs allow multiple connections to be actively transmitting data simultaneously.

Cost per machine connected:

Here's how to use these numbers. Suppose that, not counting the network connection, it costs $2,000 to purchase a PC for use as a node in your cluster. Adding a Fast Ethernet brings the per node cost to about $2,400; adding a Myrinet instead brings the cost to about $3,800. If you have about $20,000 to spend, that means you could have either 8 machines connected by Fast Ethernet or 5 machines connected by Myrinet. It also can be very reasonable to have multiple networks; e.g., $20,000 could buy 8 machines connected by both Fast Ethernet and TTL_PAPERS. Pick the network, or set of networks, that is most likely to yield a cluster that will run your application fastest.

By the time you read this, these numbers will be wrong... heck, they're probably wrong already. There may also be quantity discounts, special deals, etc. Still, the prices quoted here aren't likely to be wrong enough to lead you to a totally inappropriate choice. It doesn't take a PhD (although I do have one ;-) to see that expensive networks only make sense if your application needs their special properties or if the PCs being clustered are relatively expensive.

Now that you have the disclaimers, on with the show....

ArcNet

- Linux support: *kernel drivers*
- Maximum bandwidth: *2.5 Mb/s*
- Minimum latency: *1,000 microseconds?*
- Available as: *multiple-vendor hardware*
- Interface port/bus used: *ISA*
- Network structure: *unswitched hub or bus (logical ring)*
- Cost per machine connected: *$200*

ARCNET is a local area network that is primarily intended for use in embedded real-time control systems. Like Ethernet, the network is physically organized either as taps on a bus or one or more hubs, however, unlike Ethernet, it uses a token-based protocol logically structuring the network as a ring. Packet headers are small (3 or 4 bytes) and messages can carry as little as a single byte of data. Thus, ARCNET yields more consistent performance than Ethernet, with bounded delays, etc. Unfortunately, it is slower than Ethernet and less popular, making it more expensive. More information is available from the ARCNET Trade Association at http://www.arcnet.com/.

ATM

- Linux support: *kernel driver, AAL* library*
- Maximum bandwidth: *155 Mb/s* (soon, *1,200 Mb/s*)
- Minimum latency: *120 microseconds*

- Available as: *multiple-vendor hardware*
- Interface port/bus used: *PCI*
- Network structure: *switched hubs*
- Cost per machine connected: *$3,000*

Unless you've been in a coma for the past few years, you have probably heard a lot about how ATM (Asynchronous Transfer Mode) *is* the future... well, sort-of. ATM is cheaper than HiPPI and faster than Fast Ethernet, and it can be used over the very long distances that the phone companies care about. The ATM network protocol is also designed to provide a lower-overhead software interface and to more efficiently manage small messages and real-time communications (e.g., digital audio and video). It is also one of the highest-bandwidth networks that Linux currently supports. The bad news is that ATM isn't cheap, and there are still some compatibility problems across vendors. An overview of Linux ATM development is available at http://lrcwww.epfl.ch/linux-atm/.

CAPERS

- Linux support: *AFAPI library*
- Maximum bandwidth: *1.2 Mb/s*
- Minimum latency: *3 microseconds*
- Available as: *commodity hardware*
- Interface port/bus used: *SPP*
- Network structure: *cable between 2 machines*
- Cost per machine connected: *$2*

CAPERS (Cable Adapter for Parallel Execution and Rapid Synchronization) is a spin-off of the PAPERS project, http://garage.ecn.purdue.edu/~papers/, at the Purdue University School of Electrical and Computer Engineering. In essence, it defines a software protocol for using an ordinary "LapLink" SPP-to-SPP cable to implement the PAPERS library for two Linux PCs. The idea doesn't scale, but you can't beat the price. As with TTL_PAPERS, to improve system security, there is a minor kernel patch recommended, but not required: http://garage.ecn.purdue.edu/~papers/giveioperm.html.

Ethernet

- Linux support: *kernel drivers*
- Maximum bandwidth: *10 Mb/s*
- Minimum latency: *100 microseconds*
- Available as: *commodity hardware*
- Interface port/bus used: *PCI*
- Network structure: *switched or unswitched hubs, or hubless bus*
- Cost per machine connected: *$100* (hubless, *$50*)

For some years now, 10 Mbits/s Ethernet has been the standard network technology. Good Ethernet interface cards can be purchased for well under $50, and a fair number of PCs now have an Ethernet

TOC

controller built-into the motherboard. For lightly-used networks, Ethernet connections can be organized as a multi-tap bus without a hub; such configurations can serve up to 200 machines with minimal cost but are not appropriate for parallel processing. Adding an unswitched hub does not really help performance. However, switched hubs that can provide full bandwidth to simultaneous connections cost only about $100 per port. Linux supports an amazing range of Ethernet interfaces, but it is important to keep in mind that variations in the interface hardware can yield significant performance differences. See the Hardware Compatibility HOWTO for comments on which are supported and how well they work; also see http://cesdis1.gsfc.nasa.gov/linux/drivers/.

An interesting way to improve performance is offered by the 16-machine Linux cluster work done in the Beowulf project, http://cesdis.gsfc.nasa.gov/linux/beowulf/beowulf.html, at NASA CESDIS. There, Donald Becker, who is the author of many Ethernet card drivers, has developed support for load sharing across multiple Ethernet networks that shadow each other (i.e., share the same network addresses). This load sharing is built-into the standard Linux distribution, and is done invisibly below the socket operation level. Because hub cost is significant, having each machine connected to two or more hubless or unswitched hub Ethernet networks can be a very cost-effective way to improve performance. In fact, in situations where one machine is the network performance bottleneck, load sharing using shadow networks works much better than using a single switched hub network.

Ethernet (Fast Ethernet)

- Linux support: *kernel drivers*
- Maximum bandwidth: *100 Mb/s*
- Minimum latency: *80 microseconds*
- Available as: *commodity hardware*
- Interface port/bus used: *PCI*
- Network structure: *switched or unswitched hubs*
- Cost per machine connected: *$400?*

Although there are really quite a few different technologies calling themselves "Fast Ethernet," this term most often refers to a hub-based 100 Mbits/s Ethernet that is somewhat compatible with older "10 BaseT" 10 Mbits/s devices and cables. As might be expected, anything called Ethernet is generally priced for a volume market, and these interfaces are generally a small fraction of the price of 155 Mbits/s ATM cards. The catch is that having a bunch of machines dividing the bandwidth of a single 100 Mbits/s "bus" (using an unswitched hub) yields performance that might not even be as good on average as using 10 Mbits/s Ethernet with a switched hub that can give each machine's connection a full 10 Mbits/s.

Switched hubs that can provide 100 Mbits/s for each machine simultaneously are expensive, but prices are dropping every day, and these switches do yield much higher total network bandwidth than unswitched hubs. The thing that makes ATM switches so expensive is that they must switch for each (relatively short) ATM cell; some Fast Ethernet switches take advantage of the expected lower switching frequency by using techniques that may have low latency through the switch, but take multiple milliseconds to change the switch path... if your routing pattern changes frequently, avoid those switches. See http://cesdis1.gsfc.nasa.gov/linux/drivers/ for information about the various cards and drivers.

TOC

Also note that, as described for Ethernet, the Beowulf project, http://cesdis.gsfc.nasa.gov/linux/beowulf/beowulf.html, at NASA has been developing support that c improved performance by load sharing across multiple Fast Ethernets.

Ethernet (Gigabit Ethernet)

- Linux support: *kernel drivers*
- Maximum bandwidth: *1,000 Mb/s*
- Minimum latency: *300 microseconds?*
- Available as: *multiple-vendor hardware*
- Interface port/bus used: *PCI*
- Network structure: *switched hubs or FDRs*
- Cost per machine connected: *$2,500?*

I'm not sure that Gigabit Ethernet, http://www.gigabit-ethernet.org/, has a good technological reason to be called Ethernet... but the name does accurately reflect the fact that this is intended to be a cheap, mass-market, computer network technology with native support for IP. However, current pricing reflects the fact that Gb/s hardware is still a tricky thing to build.

Unlike other Ethernet technologies, Gigabit Ethernet provides for a level of flow control that should make it a more reliable network. FDRs, or Full-Duplex Repeaters, simply multiplex lines, using buffering and localized flow control to improve performance. Most switched hubs are being built as new interface modules for existing gigabit-capable switch fabrics. Switch/FDR products have been shipped or announced by at least http://www.acacianet.com/, http://www.baynetworks.com/, http://www.cabletron.com/, http://www.networks.digital.com/, http://www.extremenetworks.com/, http://www.foundrynet.com/, http://www.gigalabs.com/, http://www.packetengines.com/. http://www.plaintree.com/, http://www.prominet.com/, http://www.sun.com/, and http://www.xlnt.com/.

There is a Linux driver, http://cesdis.gsfc.nasa.gov/linux/drivers/yellowfin.html, for the Packet Engines "Yellowfin" G-NIC, http://www.packetengines.com/. Early tests under Linux achieved about 2.5x higher bandwidth than could be achieved with the best 100 Mb/s Fast Ethernet; with gigabit networks, careful tuning of PCI bus use is a critical factor. There is little doubt that driver improvements, and Linux drivers for other NICs, will follow.

FC (Fibre Channel)

- Linux support: *no*
- Maximum bandwidth: *1,062 Mb/s*
- Minimum latency: *?*
- Available as: *multiple-vendor hardware*
- Interface port/bus used: *PCI?*
- Network structure: *?*
- Cost per machine connected: *?*

TOC

The goal of FC (Fibre Channel) is to provide high-performance block I/O (an FC frame carries a 2,0 byte data payload), particularly for sharing disks and other storage devices that can be directly conne to the FC rather than connected through a computer. Bandwidth-wise, FC is specified to be relatively fast, running anywhere between 133 and 1,062 Mbits/s. If FC becomes popular as a high-end SCSI replacement, it may quickly become a cheap technology; for now, it is not cheap and is not supported by Linux. A good collection of FC references is maintained by the Fibre Channel Association at http://www.amdahl.com/ext/CARP/FCA/FCA.html

FireWire (IEEE 1394)

- Linux support: *no*
- Maximum bandwidth: *196.608 Mb/s* (soon, *393.216 Mb/s*)
- Minimum latency: *?*
- Available as: *multiple-vendor hardware*
- Interface port/bus used: *PCI*
- Network structure: *random without cycles (self-configuring)*
- Cost per machine connected: *$600*

FireWire, http://www.firewire.org/, the IEEE 1394-1995 standard, is destined to be the low-cost high-speed digital network for consumer electronics. The showcase application is connecting DV digital video camcorders to computers, but FireWire is intended to be used for applications ranging from being a SCSI replacement to interconnecting the components of your home theater. It allows up to 64K devices to be connected in any topology using busses and bridges that does not create a cycle, and automatically detects the configuration when components are added or removed. Short (four-byte "quadlet") low-latency messages are supported as well as ATM-like isochronous transmission (used to keep multimedia messages synchronized). Adaptec has FireWire products that allow up to 63 devices to be connected to a single PCI interface card, and also has good general FireWire information at http://www.adaptec.com/serialio/.

Although FireWire will not be the highest bandwidth network available, the consumer-level market (which should drive prices very low) and low latency support might make this one of the best Linux PC cluster message-passing network technologies within the next year or so.

HiPPI And Serial HiPPI

- Linux support: *no*
- Maximum bandwidth: *1,600 Mb/s* (serial is *1,200 Mb/s*)
- Minimum latency: *?*
- Available as: *multiple-vendor hardware*
- Interface port/bus used: *EISA, PCI*
- Network structure: *switched hubs*
- Cost per machine connected: *$3,500* (serial is *$4,500*)

HiPPI (High Performance Parallel Interface) was originally intended to provide very high bandwidth transfer of huge data sets between a supercomputer and another machine (a supercomputer, frame buffer, disk array, etc.), and has become the dominant standard for supercomputers. Although it is an oxymoron, **Serial HiPPI** is also becoming popular, typically using a fiber optic cable instead of the 32-bit wide standard (parallel) HiPPI cables. Over the past few years, HiPPI crossbar switches have become common and prices have dropped sharply; unfortunately, serial HiPPI is still pricey, and that is what PCI bus interface cards generally support. Worse still, Linux doesn't yet support HiPPI. A good overview of HiPPI is maintained by CERN at http://www.cern.ch/HSI/hippi/; they also maintain a rather long list of HiPPI vendors at http://www.cern.ch/HSI/hippi/procintf/manufact.htm.

IrDA (Infrared Data Association)

- Linux support: *no?*
- Maximum bandwidth: *1.15 Mb/s* and *4 Mb/s*
- Minimum latency: *?*
- Available as: *multiple-vendor hardware*
- Interface port/bus used: *IrDA*
- Network structure: *thin air* ;-)
- Cost per machine connected: *$0*

IrDA (Infrared Data Association, http://www.irda.org/) is that little infrared device on the side of a lot of laptop PCs. It is inherently difficult to connect more than two machines using this interface, so it is unlikely to be used for clustering. Don Becker did some preliminary work with IrDA.

Myrinet

- Linux support: *library*
- Maximum bandwidth: *1,280 Mb/s*
- Minimum latency: *9 microseconds*
- Available as: *single-vendor hardware*
- Interface port/bus used: *PCI*
- Network structure: *switched hubs*
- Cost per machine connected: *$1,800*

Myrinet http://www.myri.com/ is a local area network (LAN) designed to also serve as a "system area network" (SAN), i.e., the network within a cabinet full of machines connected as a parallel system. The LAN and SAN versions use different physical media and have somewhat different characteristics; generally, the SAN version would be used within a cluster.

Myrinet is fairly conventional in structure, but has a reputation for being particularly well-implemented. The drivers for Linux are said to perform very well, although shockingly large performance variations have been reported with different PCI bus implementations for the host computers.

Currently, Myrinet is clearly the favorite network of cluster groups that are not too severely "budgetarily

challenged." If your idea of a Linux PC is a high-end Pentium Pro or Pentium II with at least 256 MB RAM and a SCSI RAID, the cost of Myrinet is quite reasonable. However, using more ordinary PC configurations, you may find that your choice is between *N* machines linked by Myrinet or *2N* linked by multiple Fast Ethernets and TTL_PAPERS. It really depends on what your budget is and what types of computations you care about most.

Parastation

- Linux support: *HAL or socket library*
- Maximum bandwidth: *125 Mb/s*
- Minimum latency: *2 microseconds*
- Available as: *single-vendor hardware*
- Interface port/bus used: *PCI*
- Network structure: *hubless mesh*
- Cost per machine connected: *> $1,000*

The ParaStation project http://wwwipd.ira.uka.de/parastation at University of Karlsruhe Department of Informatics is building a PVM-compatible custom low-latency network. They first constructed a two-processor ParaPC prototype using a custom EISA card interface and PCs running BSD UNIX, and then built larger clusters using DEC Alphas. Since January 1997, ParaStation has been available for Linux. The PCI cards are being made in cooperation with a company called Hitex (see http://www.hitex.com:80/parastation/). Parastation hardware implements both fast, reliable, message transmission and simple barrier synchronization.

PLIP

- Linux support: *kernel driver*
- Maximum bandwidth: *1.2 Mb/s*
- Minimum latency: *1,000 microseconds?*
- Available as: *commodity hardware*
- Interface port/bus used: *SPP*
- Network structure: *cable between 2 machines*
- Cost per machine connected: *$2*

For just the cost of a "LapLink" cable, PLIP (Parallel Line Interface Protocol) allows two Linux machines to communicate through standard parallel ports using standard socket-based software. In terms of bandwidth, latency, and scalability, this is not a very serious network technology; however, the near-zero cost and the software compatibility are useful. The driver is part of the standard Linux kernel distributions.

TOC

SCI

- Linux support: *no*
- Maximum bandwidth: *4,000 Mb/s*
- Minimum latency: *2.7 microseconds*
- Available as: *multiple-vendor hardware*
- Interface port/bus used: *PCI, proprietary*
- Network structure: *?*
- Cost per machine connected: > *$1,000*

The goal of SCI (Scalable Coherent Interconnect, ANSI/IEEE 1596-1992) is essentially to provide a high performance mechanism that can support coherent shared memory access across large numbers of machines, as well various types of block message transfers. It is fairly safe to say that the designed bandwidth and latency of SCI are both "awesome" in comparison to most other network technologies. The catch is that SCI is not widely available as cheap production units, and there isn't any Linux support.

SCI primarily is used in various proprietary designs for logically-shared physically-distributed memory machines, such as the HP/Convex Exemplar SPP and the Sequent NUMA-Q 2000 (see http://www.sequent.com/). However, SCI is available as a PCI interface card and 4-way switches (up to 16 machines can be connected by cascading four 4-way switches) from Dolphin, http://www.dolphinics.com/, as their CluStar product line. A good set of links overviewing SCI is maintained by CERN at http://www.cern.ch/HSI/sci/sci.html.

SCSI

- Linux support: *kernel drivers*
- Maximum bandwidth: *5 Mb/s* to over *20 Mb/s*
- Minimum latency: *?*
- Available as: *multiple-vendor hardware*
- Interface port/bus used: *PCI, EISA, ISA card*
- Network structure: *inter-machine bus sharing SCSI devices*
- Cost per machine connected: *?*

SCSI (Small Computer Systems Interconnect) is essentially an I/O bus that is used for disk drives, CD ROMS, image scanners, etc. There are three separate standards SCSI-1, SCSI-2, and SCSI-3; Fast and Ultra speeds; and data path widths of 8, 16, or 32 bits (with FireWire compatibility also mentioned in SCSI-3). It is all pretty confusing, but we all know a good SCSI is somewhat faster than EIDE and can handle more devices more efficiently.

What many people do not realize is that it is fairly simple for two computers to share a single SCSI bus. This type of configuration is very useful for sharing disk drives between machines and implementing **fail-over** - having one machine take over database requests when the other machine fails. Currently, this is the only mechanism supported by Microsoft's PC cluster product, WolfPack. However, the inability to scale to larger systems renders shared SCSI uninteresting for parallel processing in general.

TOC

ServerNet

- Linux support: *no*
- Maximum bandwidth: *400 Mb/s*
- Minimum latency: *3 microseconds*
- Available as: *single-vendor hardware*
- Interface port/bus used: *PCI*
- Network structure: *hexagonal tree/tetrahedral lattice of hubs*
- Cost per machine connected: *?*

ServerNet is the high-performance network hardware from Tandem, http://www.tandem.com. Especially in the online transation processing (OLTP) world, Tandem is well known as a leading producer of high-reliability systems, so it is not surprising that their network claims not just high performance, but also "high data integrity and reliability." Another interesting aspect of ServerNet is that it claims to be able to transfer data from any device directly to any device; not just between processors, but also disk drives, etc., in a one-sided style similar to that suggested by the MPI remote memory access mechanisms described in section 3.5. One last comment about ServerNet: although there is just a single vendor, that vendor is powerful enough to potentially establish ServerNet as a major standard... Tandem is owned by Compaq.

SHRIMP

- Linux support: *user-level memory mapped interface*
- Maximum bandwidth: *180 Mb/s*
- Minimum latency: *5 microseconds*
- Available as: *research prototype*
- Interface port/bus used: *EISA*
- Network structure: *mesh backplane (as in Intel Paragon)*
- Cost per machine connected: *?*

The SHRIMP project, http://www.CS.Princeton.EDU/shrimp/, at the Princeton University Computer Science Department is building a parallel computer using PCs running Linux as the processing elements. The first SHRIMP (Scalable, High-Performance, Really Inexpensive Multi-Processor) was a simple two-processor prototype using a dual-ported RAM on a custom EISA card interface. There is now a prototype that will scale to larger configurations using a custom interface card to connect to a "hub" that is essentially the same mesh routing network used in the Intel Paragon (see http://www.ssd.intel.com/paragon.html). Considerable effort has gone into developing low-overhead "virtual memory mapped communication" hardware and support software.

TOC

SLIP

- Linux support: *kernel drivers*
- Maximum bandwidth: *0.1 Mb/s*
- Minimum latency: *1,000 microseconds?*
- Available as: *commodity hardware*
- Interface port/bus used: *RS232C*
- Network structure: *cable between 2 machines*
- Cost per machine connected: *$2*

Although SLIP (Serial Line Interface Protocol) is firmly planted at the low end of the performance spectrum, SLIP (or CSLIP or PPP) allows two machines to perform socket communication via ordinary RS232 serial ports. The RS232 ports can be connected using a null-modem RS232 serial cable, or they can even be connected via dial-up through a modem. In any case, latency is high and bandwidth is low, so SLIP should be used only when no other alternatives are available. It is worth noting, however, that most PCs have two RS232 ports, so it would be possible to network a group of machines simply by connecting the machines as a linear array or as a ring. There is even load sharing software called EQL.

TTL_PAPERS

- Linux support: *AFAPI library*
- Maximum bandwidth: *1.6 Mb/s*
- Minimum latency: *3 microseconds*
- Available as: *public-domain design, single-vendor hardware*
- Interface port/bus used: *SPP*
- Network structure: *tree of hubs*
- Cost per machine connected: *$100*

The PAPERS (Purdue's Adapter for Parallel Execution and Rapid Synchronization) project, http://garage.ecn.purdue.edu/~papers/, at the Purdue University School of Electrical and Computer Engineering is building scalable, low-latency, aggregate function communication hardware and software that allows a parallel supercomputer to be built using unmodified PCs/workstations as nodes.

There have been over a dozen different types of PAPERS hardware built that connect to PCs/workstations via the SPP (Standard Parallel Port), roughly following two development lines. The versions called "PAPERS" target higher performance, using whatever technologies are appropriate; current work uses FPGAs, and high bandwidth PCI bus interface designs are also under development. In contrast, the versions called "TTL_PAPERS" are designed to be easily reproduced outside Purdue, and are remarkably simple public domain designs that can be built using ordinary TTL logic. One such design is produced commercially, http://chelsea.ios.com:80/~hgdietz/sbm4.html.

Unlike the custom hardware designs from other universities, TTL_PAPERS clusters have been assembled at many universities from the USA to South Korea. Bandwidth is severely limited by the SPP connections, but PAPERS implements very low latency aggregate function communications; even the

fastest message-oriented systems cannot provide comparable performance on those aggregate functi
Thus, PAPERS is particularly good for synchronizing the displays of a video wall (to be discussed further in the upcoming Video Wall HOWTO), scheduling accesses to a high-bandwidth network, evaluating global fitness in genetic searches, etc. Although PAPERS clusters have been built using IBM PowerPC AIX, DEC Alpha OSF/1, and HP PA-RISC HP-UX machines, Linux-based PCs are the platforms best supported.

User programs using TTL_PAPERS AFAPI directly access the SPP hardware port registers under Linux, without an OS call for each access. To do this, AFAPI first gets port permission using either `iopl()` or `ioperm()`. The problem with these calls is that both require the user program to be privileged, yielding a potential security hole. The solution is an optional kernel patch, http://garage.ecn.purdue.edu/~papers/giveioperm.html, that allows a privileged process to control port permission for any process.

USB (Universal Serial Bus)

- Linux support: *kernel driver*
- Maximum bandwidth: *12 Mb/s*
- Minimum latency: *?*
- Available as: *commodity hardware*
- Interface port/bus used: *USB*
- Network structure: *bus*
- Cost per machine connected: *$5?*

USB (Universal Serial Bus, http://www.usb.org/) is a hot-pluggable conventional-Ethernet-speed, bus for up to 127 peripherals ranging from keyboards to video conferencing cameras. It isn't really clear how multiple computers get connected to each other using USB. In any case, USB ports are quickly becoming as standard on PC motherboards as RS232 and SPP, so don't be surprised if one or two USB ports are lurking on the back of the next PC you buy. Development of a Linux driver is discussed at http://peloncho.fis.ucm.es/~inaky/USB.html.

In some ways, USB is almost the low-performance, zero-cost, version of FireWire that you can purchase today.

WAPERS

- Linux support: *AFAPI library*
- Maximum bandwidth: *0.4 Mb/s*
- Minimum latency: *3 microseconds*
- Available as: *public-domain design*
- Interface port/bus used: *SPP*
- Network structure: *wiring pattern between 2-64 machines*
- Cost per machine connected: *$5*

WAPERS (Wired-AND Adapter for Parallel Execution and Rapid Synchronization) is a spin-off of t PAPERS project, http://garage.ecn.purdue.edu/~papers/, at the Purdue University School of Electric and Computer Engineering. If implemented properly, the SPP has four bits of open-collector output that can be wired together across machines to implement a 4-bit wide wired AND. This wired-AND is electrically touchy, and the maximum number of machines that can be connected in this way critically depends on the analog properties of the ports (maximum sink current and pull-up resistor value); typically, up to 7 or 8 machines can be networked by WAPERS. Although cost and latency are very low, so is bandwidth; WAPERS is much better as a second network for aggregate operations than as the only network in a cluster. As with TTL_PAPERS, to improve system security, there is a minor kernel patch recommended, but not required: http://garage.ecn.purdue.edu/~papers/giveioperm.html.

3.3 Network Software Interface

Before moving on to discuss the software support for parallel applications, it is useful to first briefly cover the basics of low-level software interface to the network hardware. There are really only three basic choices: sockets, device drivers, and user-level libraries.

Sockets

By far the most common low-level network interface is a socket interface. Sockets have been a part of unix for over a decade, and most standard network hardware is designed to support at least two types of socket protocols: UDP and TCP. Both types of socket allow you to send arbitrary size blocks of data from one machine to another, but there are several important differences. Typically, both yield a minimum latency of around 1,000 microseconds, although performance can be far worse depending on network traffic.

These socket types are the basic network software interface for most of the portable, higher-level, parallel processing software; for example, PVM uses a combination of UDP and TCP, so knowing the difference will help you tune performance. For even better performance, you can also use these mechanisms directly in your program. The following is just a simple overview of UDP and TCP; see the manual pages and a good network programming book for details.

UDP Protocol (SOCK_DGRAM)

UDP is the User Datagram Protocol, but you more easily can remember the properties of UDP as Unreliable Datagram Processing. In other words, UDP allows each block to be sent as an individual message, but a message might be lost in transmission. In fact, depending on network traffic, UDP messages can be lost, can arrive multiple times, or can arrive in an order different from that in which they were sent. The sender of a UDP message does not automatically get an acknowledgment, so it is up to user-written code to detect and compensate for these problems. Fortunately, UDP does ensure that if a message arrives, the message contents are intact (i.e., you never get just part of a UDP message).

The nice thing about UDP is that it tends to be the fastest socket protocol. Further, UDP is "connectionless," which means that each message is essentially independent of all others. A good analogy is that each message is like a letter to be mailed; you might send multiple letters to the same

TOC

address, but each one is independent of the others and there is no limit on how many people you can letters to.

TCP Protocol (SOCK_STREAM)

Unlike UDP, **TCP** is a reliable, connection-based, protocol. Each block sent is not seen as a message, but as a block of data within an apparently continuous stream of bytes being transmitted through a connection between sender and receiver. This is very different from UDP messaging because each block is simply part of the byte stream and it is up to the user code to figure-out how to extract each block from the byte stream; there are no markings separating messages. Further, the connections are more fragile with respect to network problems, and only a limited number of connections can exist simultaneously for each process. Because it is reliable, TCP generally implies significantly more overhead than UDP.

There are, however, a few pleasant surprises about TCP. One is that, if multiple messages are sent through a connection, TCP is able to pack them together in a buffer to better match network hardware packet sizes, potentially yielding better-than-UDP performance for groups of short or oddly-sized messages. The other bonus is that networks constructed using reliable direct physical links between machines can easily and efficiently simulate TCP connections. For example, this was done for the ParaStation's "Socket Library" interface software, which provides TCP semantics using user-level calls that differ from the standard TCP OS calls only by the addition of the prefix `PSS` to each function name.

Device Drivers

When it comes to actually pushing data onto the network or pulling data off the network, the standard unix software interface is a part of the unix kernel called a device driver. UDP and TCP don't just transport data, they also imply a fair amount of overhead for socket management. For example, something has to manage the fact that multiple TCP connections can share a single physical network interface. In contrast, a device driver for a dedicated network interface only needs to implement a few simple data transport functions. These device driver functions can then be invoked by user programs by using `open()` to identify the proper device and then using system calls like `read()` and `write()` on the open "file." Thus, each such operation could transport a block of data with little more than the overhead of a system call, which might be as fast as tens of microseconds.

Writing a device driver to be used with Linux is not hard... if you know *precisely* how the device hardware works. If you are not sure how it works, don't guess. Debugging device drivers isn't fun and mistakes can fry hardware. However, if that hasn't scared you off, it may be possible to write a device driver to, for example, use dedicated Ethernet cards as dumb but fast direct machine-to-machine connections without the usual Ethernet protocol overhead. In fact, that's pretty much what some early Intel supercomputers did.... Look at the Device Driver HOWTO for more information.

User-Level Libraries

If you've taken an OS course, user-level access to hardware device registers is exactly what you have been taught never to do, because one of the primary purposes of an OS is to control device access. However, an OS call is at least tens of microseconds of overhead. For custom network hardware like TTL_PAPERS, which can perform a basic network operation in just 3 microseconds, such OS call

TOC

overhead is intolerable. The only way to avoid that overhead is to have user-level code - a user-level library - directly access hardware device registers. Thus, the question becomes one of how a user-level library can access hardware directly, yet not compromise the OS control of device access rights.

On a typical system, the only way for a user-level library to directly access hardware device registers is to:

1. At user program start-up, use an OS call to map the page of memory address space containing the device registers into the user process virtual memory map. For some systems, the `mmap()` call (first mentioned in section 2.6) can be used to map a special file which represents the physical memory page addresses of the I/O devices. Alternatively, it is relatively simple to write a device driver to perform this function. Further, this device driver can control access by only mapping the page(s) containing the specific device registers needed, thereby maintaining OS access control.
2. Access device registers without an OS call by simply loading or storing to the mapped addresses. For example, `*((char *) 0x1234) = 5;` would store the byte value 5 into memory location 1234 (hexadecimal).

Fortunately, it happens that Linux for the Intel 386 (and compatible processors) offers an even better solution:

1. Using the `ioperm()` OS call from a privileged process, get permission to access the precise I/O port addresses that correspond to the device registers. Alternatively, permission can be managed by an independent privileged user process (i.e., a "meta OS") using the giveioperm() OS call patch for Linux.
2. Access device registers without an OS call by using 386 port I/O instructions.

This second solution is preferable because it is common that multiple I/O devices have their registers within a single page, in which case the first technique would not provide protection against accessing other device registers that happened to reside in the same page as the ones intended. Of course, the down side is that 386 port I/O instructions cannot be coded in C - instead, you will need to use a bit of assembly code. The GCC-wrapped (usable in C programs) inline assembly code function for a port input of a byte value is:

```
extern inline unsigned char
inb(unsigned short port)
{
    unsigned char _v;
__asm__ __volatile__ ("inb %w1,%b0"
                      :"=a" (_v)
                      :"d" (port), "0" (0));
    return _v;
}
```

Similarly, the GCC-wrapped code for a byte port output is:

```
extern inline void
outb(unsigned char value,
unsigned short port)
{
__asm__ __volatile__ ("outb %b0,%w1"
                      :/* no outputs */
                      :"a" (value), "d" (port));
}
```

3.4 PVM (Parallel Virtual Machine)

PVM (Parallel Virtual Machine) is a freely-available, portable, message-passing library generally implemented on top of sockets. It is clearly established as the de-facto standard for message-passing cluster parallel computing.

PVM supports single-processor and SMP Linux machines, as well as clusters of Linux machines linked by socket-capable networks (e.g., SLIP, PLIP, Ethernet, ATM). In fact, PVM will even work across groups of machines in which a variety of different types of processors, configurations, and physical networks are used - **Heterogeneous Clusters** - even to the scale of treating machines linked by the Internet as a parallel cluster. PVM also provides facilities for parallel job control across a cluster. Best of all, PVM has long been freely available (currently from http://www.epm.ornl.gov/pvm/pvm_home.html), which has led to many programming language compilers, application libraries, programming and debugging tools, etc., using it as their "portable message-passing target library." There is also a network newsgroup, comp.parallel.pvm.

It is important to note, however, that PVM message-passing calls generally add significant overhead to standard socket operations, which already had high latency. Further, the message handling calls themselves do not constitute a particularly "friendly" programming model.

Using the same Pi computation example first described in section 1.3, the version using C with PVM library calls is:

```
#include <stdlib.h>
#include <stdio.h>
#include <pvm3.h>

#define NPROC   4

main(int argc, char **argv)
{
  register double lsum, width;
  double sum;
```

TOC

```
  register int intervals, i;
  int mytid, iproc, msgtag = 4;
  int tids[NPROC];  /* array of task ids */

  /* enroll in pvm */
  mytid = pvm_mytid();

  /* Join a group and, if I am the first instance,
     iproc=0, spawn more copies of myself
  */
  iproc = pvm_joingroup("pi");

  if (iproc == 0) {
    tids[0] = pvm_mytid();
    pvm_spawn("pvm_pi", &argv[1], 0, NULL, NPROC-1, &tids[1]);
  }
  /* make sure all processes are here */
  pvm_barrier("pi", NPROC);

  /* get the number of intervals */
  intervals = atoi(argv[1]);
  width = 1.0 / intervals;

  lsum = 0.0;
  for (i = iproc; i<intervals; i+=NPROC) {
    register double x = (i + 0.5) * width;
    lsum += 4.0 / (1.0 + x * x);
  }

  /* sum across the local results & scale by width */
  sum = lsum * width;
  pvm_reduce(PvmSum, &sum, 1, PVM_DOUBLE, msgtag, "pi", 0);

  /* have only the console PE print the result */
  if (iproc == 0) {
    printf("Estimation of pi is %f\n", sum);
  }

  /* Check program finished, leave group, exit pvm */
  pvm_barrier("pi", NPROC);
  pvm_lvgroup("pi");
  pvm_exit();
  return(0);
}
```

3.5 MPI (Message Passing Interface)

TOC

Although PVM is the de-facto standard message-passing library, MPI (Message Passing Interface) is the relatively new official standard. The home page for the MPI standard is http://www.mcs.anl.gov:80/mpi/ and the newsgroup is comp.parallel.mpi.

However, before discussing MPI, I feel compelled to say a little bit about the PVM vs. MPI religious war that has been going on for the past few years. I'm not really on either side. Here's my attempt at a relatively unbiased summary of the differences:

Execution control environment.

Put simply, PVM has one and MPI doesn't specify how/if one is implemented. Thus, things like starting a PVM program executing are done identically everywhere, while for MPI it depends on which implementation is being used.

Support for heterogeneous clusters.

PVM grew-up in the workstation cycle-scavenging world, and thus directly manages heterogeneous mixes of machines and operating systems. In contrast, MPI largely assumes that the target is an MPP (Massively Parallel Processor) or a dedicated cluster of nearly identical workstations.

Kitchen sink syndrome.

PVM evidences a unity of purpose that MPI 2.0 doesn't. The new MPI 2.0 standard includes a lot of features that go way beyond the basic message passing model - things like RMA (Remote Memory Access) and parallel file I/O. Are these things useful? Of course they are... but learning MPI 2.0 is a lot like learning a complete new programming language.

User interface design.

MPI was designed after PVM, and clearly learned from it. MPI offers simpler, more efficient, buffer handling and higher-level abstractions allowing user-defined data structures to be transmitted in messages.

The force of law.

By my count, there are still significantly more things designed to use PVM than there are to use MPI; however, porting them to MPI is easy, and the fact that MPI is backed by a widely-supported formal standard means that using MPI is, for many institutions, a matter of policy.

Conclusion? Well, there are at least three independently developed, freely available, versions of MPI that can run on clusters of Linux systems (and I wrote one of them):

- LAM (Local Area Multicomputer) is a full implementation of the MPI 1.1 standard. It allows MPI programs to be executed within an individual Linux system or across a cluster of Linux systems using UDP/TCP socket communication. The system includes simple execution control facilities, as well as a variety of program development and debugging aids. It is freely available from http://www.osc.edu/lam.html.
- MPICH (MPI CHameleon) is designed as a highly portable full implementation of the MPI 1.1

standard. Like LAM, it allows MPI programs to be executed within an individual Linux system across a cluster of Linux systems using UDP/TCP socket communication. However, the emphasis is definitely on promoting MPI by providing an efficient, easily retargetable, implementation. To port this MPI implementation, one implements either the five functions of the "channel interface" or, for better performance, the full MPICH ADI (Abstract Device Interface). MPICH, and lots of information about it and porting, are available from http://www.mcs.anl.gov/mpi/mpich/.

- AFMPI (Aggregate Function MPI) is a subset implementation of the MPI 2.0 standard. This is the one that I wrote. Built on top of the AFAPI, it is designed to showcase low-latency collective communication functions and RMAs, and thus provides only minimal support for MPI data types, communicators, etc. It allows C programs using MPI to run on an individual Linux system or across a cluster connected by AFAPI-capable network hardware. It is freely available from http://garage.ecn.purdue.edu/~papers/.

No matter which of these (or other) MPI implementations one uses, it is fairly simple to perform the most common types of communications.

However, MPI 2.0 incorporates several communication paradigms that are fundamentally different enough so that a programmer using one of them might not even recognize the other coding styles as MPI. Thus, rather than giving a single example program, it is useful to have an example of each of the fundamentally different communication paradigms that MPI supports. All three programs implement the same basic algorithm (from section 1.3) that is used throughout this HOWTO to compute the value of Pi.

The first MPI program uses basic MPI message-passing calls for each processor to send its partial sum to processor 0, which sums and prints the result:

```
#include <stdlib.h>
#include <stdio.h>
#include <mpi.h>

main(int argc, char **argv)
{
  register double width;
  double sum, lsum;
  register int intervals, i;
  int nproc, iproc;
  MPI_Status status;

  if (MPI_Init(&argc, &argv) != MPI_SUCCESS) exit(1);
  MPI_Comm_size(MPI_COMM_WORLD, &nproc);
  MPI_Comm_rank(MPI_COMM_WORLD, &iproc);
  intervals = atoi(argv[1]);
  width = 1.0 / intervals;
  lsum = 0;
  for (i=iproc; i<intervals; i+=nproc) {
    register double x = (i + 0.5) * width;
```

```
    lsum += 4.0 / (1.0 + x * x);
  }
  lsum *= width;
  if (iproc != 0) {
    MPI_Send(&lbuf, 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD);
  } else {
    sum = lsum;
    for (i=1; i<nproc; ++i) {
      MPI_Recv(&lbuf, 1, MPI_DOUBLE, MPI_ANY_SOURCE,
               MPI_ANY_TAG, MPI_COMM_WORLD, &status);
      sum += lsum;
    }
    printf("Estimation of pi is %f\n", sum);
  }
  MPI_Finalize();
  return(0);
}
```

The second MPI version uses collective communication (which, for this particular application, is clearly the most appropriate):

```
#include <stdlib.h>
#include <stdio.h>
#include <mpi.h>

main(int argc, char **argv)
{
  register double width;
  double sum, lsum;
  register int intervals, i;
  int nproc, iproc;

  if (MPI_Init(&argc, &argv) != MPI_SUCCESS) exit(1);
  MPI_Comm_size(MPI_COMM_WORLD, &nproc);
  MPI_Comm_rank(MPI_COMM_WORLD, &iproc);
  intervals = atoi(argv[1]);
  width = 1.0 / intervals;
  lsum = 0;
  for (i=iproc; i<intervals; i+=nproc) {
    register double x = (i + 0.5) * width;
    lsum += 4.0 / (1.0 + x * x);
  }
  lsum *= width;
  MPI_Reduce(&lsum, &sum, 1, MPI_DOUBLE,
```

```
            MPI_SUM, 0, MPI_COMM_WORLD);
  if (iproc == 0) {
    printf("Estimation of pi is %f\n", sum);
  }
  MPI_Finalize();
  return(0);
}
```

The third MPI version uses the MPI 2.0 RMA mechanism for each processor to add its local `lsum` into `sum` on processor 0:

```
#include <stdlib.h>
#include <stdio.h>
#include <mpi.h>

main(int argc, char **argv)
{
  register double width;
  double sum = 0, lsum;
  register int intervals, i;
  int nproc, iproc;
  MPI_Win sum_win;

  if (MPI_Init(&argc, &argv) != MPI_SUCCESS) exit(1);
  MPI_Comm_size(MPI_COMM_WORLD, &nproc);
  MPI_Comm_rank(MPI_COMM_WORLD, &iproc);
  MPI_Win_create(&sum, sizeof(sum), sizeof(sum),
                 0, MPI_COMM_WORLD, &sum_win);
  MPI_Win_fence(0, sum_win);
  intervals = atoi(argv[1]);
  width = 1.0 / intervals;
  lsum = 0;
  for (i=iproc; i<intervals; i+=nproc) {
    register double x = (i + 0.5) * width;
    lsum += 4.0 / (1.0 + x * x);
  }
  lsum *= width;
  MPI_Accumulate(&lsum, 1, MPI_DOUBLE, 0, 0,
                 1, MPI_DOUBLE, MPI_SUM, sum_win);
  MPI_Win_fence(0, sum_win);
  if (iproc == 0) {
    printf("Estimation of pi is %f\n", sum);
  }
```

```
  MPI_Finalize();
  return(0);
}
```


It is useful to note that the MPI 2.0 RMA mechanism very neatly overcomes any potential problems with the corresponding data structure on various processors residing at different memory locations. This is done by referencing a "window" that implies the base address, protection against out-of-bound accesses, and even address scaling. Efficient implementation is aided by the fact that RMA processing may be delayed until the next `MPI_Win_fence`. In summary, the RMA mechanism may be a strange cross between distributed shared memory and message passing, but it is a very clean interface that potentially generates very efficient communication.

3.6 AFAPI (Aggregate Function API)

Unlike PVM, MPI, etc., the AFAPI (Aggregate Function Application Program Interface) did not start life as an attempt to build a portable abstract interface layered on top of existing network hardware and software. Rather, AFAPI began as the very hardware-specific low-level support library for PAPERS (Purdue's Adapter for Parallel Execution and Rapid Synchronization; see http://garage.ecn.purdue.edu/~papers/).

PAPERS was discussed briefly in section 3.2; it is a public domain design custom aggregate function network that delivers latencies as low as a few microseconds. However, the important thing about PAPERS is that it was developed as an attempt to build a supercomputer that would be a better target for compiler technology than existing supercomputers. This is qualitatively different from most Linux cluster efforts and PVM/MPI, which generally focus on trying to use standard networks for the relatively few sufficiently coarse-grain parallel applications. The fact that Linux PCs are used as components of PAPERS systems is simply an artifact of implementing prototypes in the most cost-effective way possible.

The need for a common low-level software interface across more than a dozen different prototype implementations was what made the PAPERS library become standardized as AFAPI. However, the model used by AFAPI is inherently simpler and better suited for the finer-grain interactions typical of code compiled by parallelizing compilers or written for SIMD architectures. The simplicity of the model not only makes PAPERS hardware easy to build, but also yields surprisingly efficient AFAPI ports for a variety of other hardware systems, such as SMPs.

AFAPI currently runs on Linux clusters connected using TTL_PAPERS, CAPERS, or WAPERS. It also runs (without OS calls or even bus-lock instructions, see section 2.2) on SMP systems using a System V Shared Memory library called SHMAPERS. A version that runs across Linux clusters using UDP broadcasts on conventional networks (e.g., Ethernet) is under development. All released versions are available from http://garage.ecn.purdue.edu/~papers/. All versions of the AFAPI are designed to be called from C or C++.

The following example program is the AFAPI version of the Pi computation described in section 1.3.

```
#include <stdlib.h>
#include <stdio.h>
#include "afapi.h"

main(int argc, char **argv)
{
  register double width, sum;
  register int intervals, i;

  if (p_init()) exit(1);

  intervals = atoi(argv[1]);
  width = 1.0 / intervals;

  sum = 0;
  for (i=IPROC; i<intervals; i+=NPROC) {
    register double x = (i + 0.5) * width;
    sum += 4.0 / (1.0 + x * x);
  }

  sum = p_reduceAdd64f(sum) * width;

  if (IPROC == CPROC) {
    printf("Estimation of pi is %f\n", sum);
  }

  p_exit();
  return(0);
}
```

3.7 Other Cluster Support Libraries

In addition to PVM, MPI, and AFAPI, the following libraries offer features that may be useful in parallel computing using a cluster of Linux systems. These systems are given a lighter treatment in this document simply because, unlike PVM, MPI, and AFAPI, I have little or no direct experience with the use of these systems on Linux clusters. If you find any of these or other libraries to be especially useful, please send email to me at pplinux@ecn.purdue.edu describing what you've found, and I will consider adding an expanded section on that library.

TOC

Condor (process migration support)

Condor is a distributed resource management system that can manage large heterogeneous clusters of workstations. Its design has been motivated by the needs of users who would like to use the unutilized capacity of such clusters for their long-running, computation-intensive jobs. Condor preserves a large measure of the originating machine's environment on the execution machine, even if the originating and execution machines do not share a common file system and/or password mechanisms. Condor jobs that consist of a single process are automatically checkpointed and migrated between workstations as needed to ensure eventual completion.

Condor is available at http://www.cs.wisc.edu/condor/. A Linux port exists; more information is available at http://www.cs.wisc.edu/condor/linux/linux.html. Contact condor-admin@cs.wisc.edu for details.

DFN-RPC (German Research Network - Remote Procedure Call)

The DFN-RPC, a (German Research Network Remote Procedure Call) tool, was developed to distribute and parallelize scientific-technical application programs between a workstation and a compute server or a cluster. The interface is optimized for applications written in fortran, but the DFN-RPC can also be used in a C environment. It has been ported to Linux. More information is at ftp://ftp.uni-stuttgart.de/pub/rus/dfn_rpc/README_dfnrpc.html.

DQS (Distributed Queueing System)

Not exactly a library, DQS 3.0 (Distributed Queueing System) is a job queueing system that has been developed and tested under Linux. It is designed to allow both use and administration of a heterogeneous cluster as a single entity. It is available from http://www.scri.fsu.edu/~pasko/dqs.html.

There is also a commercial version called CODINE 4.1.1 (COmputing in DIstributed Network Environments). Information on it is available from http://www.genias.de/genias_welcome.html.

3.8 General Cluster References

Because clusters can be constructed and used in so many different ways, there are quite a few groups that have made interesting contributions. The following are references to various cluster-related projects that may be of general interest. This includes a mix of Linux-specific and generic cluster references. The list is given in alphabetical order.

Beowulf

The Beowulf project, http://cesdis1.gsfc.nasa.gov/beowulf/, centers on production of software for using off-the-shelf clustered workstations based on commodity PC-class hardware, a high-bandwidth cluster-internal network, and the Linux operating system.

Thomas Sterling has been the driving force behind Beowulf, and continues to be an eloquent and

TOC

outspoken proponent of Linux clustering for scientific computing in general. In fact, many groups no refer to their clusters as "Beowulf class" systems - even if the cluster isn't really all that similar to the official Beowulf design.

Don Becker, working in support of the Beowulf project, has produced many of the network drivers used by Linux in general. Many of these drivers have even been adapted for use in BSD. Don also is responsible for many of these Linux network drivers allowing load-sharing across multiple parallel connections to achieve higher bandwidth without expensive switched hubs. This type of load sharing was the original distinguishing feature of the Beowulf cluster.

Linux/AP+

The Linux/AP+ project, http://cap.anu.edu.au/cap/projects/linux/, is not exactly about Linux clustering, but centers on porting Linux to the Fujitsu AP1000+ and adding appropriate parallel processing enhancements. The AP1000+ is a commercially available SPARC-based parallel machine that uses a custom network with a torus topology, 25 MB/s bandwidth, and 10 microsecond latency... in short, it looks a lot like a SPARC Linux cluster.

Locust

The Locust project, http://www.ecsl.cs.sunysb.edu/~manish/locust/, is building a distributed virtual shared memory system that uses compile-time information to hide message-latency and to reduce network traffic at run time. Pupa is the underlying communication subsystem of Locust, and is implemented using Ethernet to connect 486 PCs under FreeBSD. Linux?

Midway DSM (Distributed Shared Memory)

Midway, http://www.cs.cmu.edu/afs/cs.cmu.edu/project/midway/WWW/HomePage.html, is a software-based DSM (Distributed Shared Memory) system, not unlike TreadMarks. The good news is that it uses compile-time aids rather than relatively slow page-fault mechanisms, and it is free. The bad news is that it doesn't run on Linux clusters.

Mosix

MOSIX modifies the BSDI BSD/OS to provide dynamic load balancing and preemptive process migration across a networked group of PCs. This is nice stuff not just for parallel processing, but for generally using a cluster much like a scalable SMP. Will there be a Linux version? Look at http://www.cs.huji.ac.il/mosix/ for more information.

NOW (Network Of Workstations)

The Berkeley NOW (Network Of Workstations) project, http://now.cs.berkeley.edu/, has led much of the push toward parallel computing using networks of workstations. There is a lot work going on here, all aimed toward "demonstrating a practical 100 processor system in the next few years." Alas, they don't use Linux.

Parallel Processing Using Linux

The parallel processing using Linux WWW site, http://yara.ecn.purdue.edu/~pplinux/, is the home of this HOWTO and many related documents including online slides for a full-day tutorial. Aside from the work on the PAPERS project, the Purdue University School of Electrical and Computer Engineering generally has been a leader in parallel processing; this site was established to help others apply Linux PCs for parallel processing.

Since Purdue's first cluster of Linux PCs was assembled in February 1994, there have been many Linux PC clusters assembled at Purdue, including several with video walls. Although these clusters used 386, 486, and Pentium systems (no Pentium Pro systems), Intel recently awarded Purdue a donation which will allow it to construct multiple large clusters of Pentium II systems (with as many as 165 machines planned for a single cluster). Although these clusters all have/will have PAPERS networks, most also have conventional networks.

Pentium Pro Cluster Workshop

In Des Moines, Iowa, April 10-11, 1997, AMES Laboratory held the Pentium Pro Cluster Workshop. The WWW site from this workshop, http://www.scl.ameslab.gov/workshops/PPCworkshop.html, contains a wealth of PC cluster information gathered from all the attendees.

TreadMarks DSM (Distributed Shared Memory)

DSM (Distributed Shared Memory) is a technique whereby a message-passing system can appear to behave as an SMP. There are quite a few such systems, most of which use the OS page-fault mechanism to trigger message transmissions. TreadMarks, http://www.cs.rice.edu/~willy/TreadMarks/overview.html, is one of the more efficient of such systems and does run on Linux clusters. The bad news is "TreadMarks is being distributed at a small cost to universities and nonprofit institutions." For more information about the software, contact treadmarks@ece.rice.edu.

U-Net (User-level NETwork interface architecture)

The U-Net (User-level NETwork interface architecture) project at Cornell, http://www2.cs.cornell.edu/U-Net/Default.html, attempts to provide low-latency and high-bandwidth using commodity network hardware by by virtualizing the network interface so that applications can send and receive messages without operating system intervention. U-Net runs on Linux PCs using a DECchip DC21140 based Fast Ethernet card or a Fore Systems PCA-200 (not PCA-200E) ATM card.

WWT (Wisconsin Wind Tunnel)

There is really quite a lot of cluster-related work at Wisconsin. The WWT (Wisconsin Wind Tunnel) project, http://www.cs.wisc.edu/~wwt/, is doing all sorts of work toward developing a "standard" interface between compilers and the underlying parallel hardware. There is the Wisconsin COW (Cluster Of Workstations), Cooperative Shared Memory and Tempest, the Paradyn Parallel Performance Tools,

etc. Unfortunately, there is not much about Linux.

4. SIMD Within A Register (e.g., using MMX)

SIMD (Single Instruction stream, Multiple Data stream) Within A Register (SWAR) isn't a new idea. Given a machine with *k*-bit registers, data paths, and function units, it has long been known that ordinary register operations can function as SIMD parallel operations on *n*, *k*/*n*-bit, integer field values. However, it is only with the recent push for multimedia that the 2x to 8x speedup offered by SWAR techniques has become a concern for mainstream computing. The 1997 versions of most microprocessors incorporate hardware support for SWAR:

- AMD K6 MMX (MultiMedia eXtensions)
- Cyrix M2 MMX (MultiMedia eXtensions)
- Digital Alpha MAX (MultimediA eXtensions)
- Hewlett-Packard PA-RISC MAX (Multimedia Acceleration eXtensions)
- Intel Pentium II & Pentium with MMX (MultiMedia eXtensions)
- Microunity Mediaprocessor SIGD (Single Instruction on Groups of Data)
- MIPS Digital Media eXtension (MDMX, pronounced Mad Max)
- Sun SPARC V9 VIS (Visual Instruction Set)

There are a few holes in the hardware support provided by the new microprocessors, quirks like only supporting some operations for some field sizes. It is important to remember, however, that you don't need any hardware support for many SWAR operations to be efficient. For example, bitwise operations are not affected by the logical partitioning of a register.

4.1 SWAR: What Is It Good For?

Although *every* modern processor is capable of executing with at least some SWAR parallelism, the sad fact is that even the best SWAR-enhanced instruction sets do not support very general-purpose parallelism. In fact, many people have noticed that the performance difference between Pentium and "Pentium with MMX technology" is often due to things like the larger L1 cache that coincided with appearance of MMX. So, realistically, what is SWAR (or MMX) good for?

- Integers only, the smaller the better. Two 32-bit values fit in a 64-bit MMX register, but so do eight one-byte characters or even an entire chess board worth of one-bit values. Note: there *will be a floating-point version of MMX*, although very little has been said about it at this writing. Cyrix has posted a set of slides, ftp://ftp.cyrix.com/developr/mpf97rm.pdf, that includes a few comments about **MMFP**. Apparently, MMFP will support two 32-bit floating-point numbers to be packed into a 64-bit MMX register; combining this with two MMFP pipelines will yield four single-precision FLOPs per clock.
- SIMD or vector-style parallelism. The same operation is applied to all fields simultaneously. There are ways to nullify the effects on selected fields (i.e., equivalent to SIMD enable masking), but they complicate coding and hurt performance.
- Localized, regular (preferably packed), memory reference patterns. SWAR in general, and MMX in

TOC

particular, are terrible at randomly-ordered accesses; gathering a vector `x[y]` (where `y` is an index array) is prohibitively expensive.

These are serious restrictions, but this type of parallelism occurs in many parallel algorithms - not just multimedia applications. For the right type of algorithm, SWAR is more effective than SMP or cluster parallelism... and it doesn't cost anything to use it.

4.2 Introduction To SWAR Programming

The basic concept of SWAR, SIMD Within A Register, is that operations on word-length registers can be used to speed-up computations by performing SIMD parallel operations on *n k*/*n*-bit field values. However, making use of SWAR technology can be awkward, and some SWAR operations are actually more expensive than the corresponding sequences of serial operations because they require additional instructions to enforce the field partitioning.

To illustrate this point, let's consider a greatly simplified SWAR mechanism that manages four 8-bit fields within each 32-bit register. The values in two registers might be represented as:

```
          PE3      PE2      PE1      PE0
       +-------+-------+-------+-------+
Reg0   | D 7:0 | C 7:0 | B 7:0 | A 7:0 |
       +-------+-------+-------+-------+
Reg1   | H 7:0 | G 7:0 | F 7:0 | E 7:0 |
       +-------+-------+-------+-------+
```

This simply indicates that each register is viewed as essentially a vector of four independent 8-bit integer values. Alternatively, think of `A` and `E` as values in Reg0 and Reg1 of processing element 0 (PE0), `B` and `F` as values in PE1's registers, and so forth.

The remainder of this document briefly reviews the basic classes of SIMD parallel operations on these integer vectors and how these functions can be implemented.

Polymorphic Operations

Some SWAR operations can be performed trivially using ordinary 32-bit integer operations, without concern for the fact that the operation is really intended to operate independently in parallel on these 8-bit fields. We call any such SWAR operation *polymorphic*, since the function is unaffected by the field types (sizes).

Testing if any field is non-zero is polymorphic, as are all bitwise logic operations. For example, an ordinary bitwise-and operation (C's `&` operator) performs a bitwise and no matter what the field sizes are. A simple bitwise and of the above registers yields:

```
           PE3        PE2        PE1        PE0
       +---------+---------+---------+---------+
Reg2   | D&H 7:0 | C&G 7:0 | B&F 7:0 | A&E 7:0 |
```

```
+---------+---------+---------+---------+
```


Because the bitwise and operation always has the value of result bit *k* affected only by the values of the operand bit *k* values, all field sizes are supported using the same single instruction.

Partitioned Operations

Unfortunately, lots of important SWAR operations are not polymorphic. Arithmetic operations such as add, subtract, multiply, and divide are all subject to carry/borrow interactions between fields. We call such SWAR operations *partitioned*, because each such operation must effectively partition the operands and result to prevent interactions between fields. However, there are actually three different methods that can be used to achieve this effect.

Partitioned Instructions

Perhaps the most obvious approach to implementing partitioned operations is to provide hardware support for "partitioned parallel instructions" that cut the carry/borrow logic between fields. This approach can yield the highest performance, but it requires a change to the processor's instruction set and generally places many restrictions on field size (e.g., 8-bit fields might be supported, but not 12-bit fields).

The AMD/Cyrix/Intel MMX, Digital MAX, HP MAX, and Sun VIS all implement restricted versions of partitioned instructions. Unfortunately, these different instruction set extensions have significantly different restrictions, making algorithms somewhat non-portable between them. For example, consider the following sampling of partitioned operations:

Instruction	AMD/Cyrix/Intel MMX	DEC MAX	HP MAX	Sun VIS
Absolute Difference		8		8
Merge Maximum		8, 16		
Compare	8, 16, 32			16, 32
Multiply	16			8x16
Add	8, 16, 32		16	16, 32

In the table, the numbers indicate the field sizes, in bits, for which each operation is supported. Even though the table omits many instructions including all the more exotic ones, it is clear that there are many differences. The direct result is that high-level languages (HLLs) really are not very effective as programming models, and portability is generally poor.

Unpartitioned Operations With Correction Code

TOC

Implementing partitioned operations using partitioned instructions can certainly be efficient, but what do you do if the partitioned operation you need is not supported by the hardware? The answer is that you use a series of ordinary instructions to perform the operation with carry/borrow across fields, and then correct for the undesired field interactions.

This is a purely software approach, and the corrections do introduce overhead, but it works with fully general field partitioning. This approach is also fully general in that it can be used either to fill gaps in the hardware support for partitioned instructions, or it can be used to provide full functionality for target machines that have no hardware support at all. In fact, by expressing the code sequences in a language like C, this approach allows SWAR programs to be fully portable.

The question immediately arises: precisely how inefficient is it to simulate SWAR partitioned operations using unpartitioned operations with correction code? Well, that is certainly the $64k question... but many operations are not as difficult as one might expect.

Consider implementing a four-element 8-bit integer vector add of two source vectors, `x+y`, using ordinary 32-bit operations.

An ordinary 32-bit add might actually yield the correct result, but not if any 8-bit field carries into the next field. Thus, our goal is simply to ensure that such a carry does not occur. Because adding two *k*-bit fields generates an at most *k*+1 bit result, we can ensure that no carry occurs by simply "masking out" the most significant bit of each field. This is done by bitwise anding each operand with `0x7f7f7f7f` and then performing an ordinary 32-bit add.

```
t = ((x & 0x7f7f7f7f) + (y & 0x7f7f7f7f));
```

That result is correct... except for the most significant bit within each field. Computing the correct value for each field is simply a matter of doing two 1-bit partitioned adds of the most significant bits from `x` and `y` to the 7-bit carry result which was computed for `t`. Fortunately, a 1-bit partitioned add is implemented by an ordinary exclusive or operation. Thus, the result is simply:

```
(t ^ ((x ^ y) & 0x80808080))
```

Ok, well, maybe that isn't so simple. After all, it is six operations to do just four adds. However, notice that the number of operations is not a function of how many fields there are... so, with more fields, we get speedup. In fact, we may get speedup anyway simply because the fields were loaded and stored in a single (integer vector) operation, register availability may be improved, and there are fewer dynamic code scheduling dependencies (because partial word references are avoided).

TOC

Controlling Field Values

While the other two approaches to partitioned operation implementation both center on getting the maximum space utilization for the registers, it can be computationally more efficient to instead control the field values so that inter-field carry/borrow events should never occur. For example, if we know that all the field values being added are such that no field overflow will occur, a partitioned add operation can be implemented using an ordinary add instruction; in fact, given this constraint, an ordinary add instruction appears polymorphic, and is usable for any field sizes without correction code. The question thus becomes how to ensure that field values will not cause carry/borrow events.

One way to ensure this property is to implement partitioned instructions that can restrict the range of field values. The Digital MAX vector minimum and maximum instructions can be viewed as hardware support for clipping field values to avoid inter-field carry/borrow.

However, suppose that we do not have partitioned instructions that can efficiently restrict the range of field values... is there a sufficient condition that can be cheaply imposed to ensure carry/borrow events will not interfere with adjacent fields? The answer lies in analysis of the arithmetic properties. Adding two *k*-bit numbers generates a result with at most *k*+1 bits; thus, a field of *k*+1 bits can safely contain such an operation despite using ordinary instructions.

Thus, suppose that the 8-bit fields in our earlier example are now 7-bit fields with 1-bit "carry/borrow spacers":

```
              PE3          PE2          PE1          PE0
      +----+-------+----+-------+----+-------+----+-------+
Reg0  | D' | D 6:0 | C' | C 6:0 | B' | B 6:0 | A' | A 6:0 |
      +----+-------+----+-------+----+-------+----+-------+
```

A vector of 7-bit adds is performed as follows. Let us assume that, prior to the start of any partitioned operation, all the carry spacer bits (`A'`, `B'`, `C'`, and `D'`) have the value 0. By simply executing an ordinary add operation, all the fields obtain the correct 7-bit values; however, some spacer bit values might now be 1. We can correct this by just one more conventional operation, masking-out the spacer bits. Our 7-bit integer vector add, `x+y`, is thus:

```
((x + y) & 0x7f7f7f7f)
```

This is just two instructions for four adds, clearly yielding good speedup.

The sharp reader may have noticed that setting the spacer bits to 0 does not work for subtract operations. The correction is, however, remarkably simple. To compute `x-y`, we simply ensure the initial condition that the spacers in `x` are all 1, while the spacers in `y` are all 0. In the worst case, we would thus get:

```
(((x | 0x80808080) - y) & 0x7f7f7f7f)
```

TOC

However, the additional bitwise or operation can often be optimized out by ensuring that the operation generating the value for `x` used `| 0x80808080` rather than `& 0x7f7f7f7f` as the last step.

Which method should be used for SWAR partitioned operations? The answer is simply "whichever yields the best speedup." Interestingly, the ideal method to use may be different for different field sizes within the same program running on the same machine.

Communication & Type Conversion Operations

Although some parallel computations, including many operations on image pixels, have the property that the *i*th value in a vector is a function only of values that appear in the *i*th position of the operand vectors, this is generally not the case. For example, even pixel operations such as smoothing require values from adjacent pixels as operands, and transformations like FFTs require more complex (less localized) communication patterns.

It is not difficult to efficiently implement 1-dimensional nearest neighbor communication for SWAR using unpartitioned shift operations. For example, to move a value from PE*i* to PE(*i*+1), a simple shift operation suffices. If the fields are 8-bits in length, we would use:

```
(x << 8)
```

Still, it isn't always quite that simple. For example, to move a value from PE*i* to PE(*i*-1), a simple shift operation might suffice... but the C language does not specify if shifts right preserve the sign bit, and some machines only provide signed shift right. Thus, in the general case, we must explicitly zero the potentially replicated sign bits:

```
((x >> 8) & 0x00ffffff)
```

Adding "wrap-around connections" is also reasonably efficient using unpartitioned shifts. For example, to move a value from PE*i* to PE(*i*+1) with wraparound:

```
((x << 8) | ((x >> 24) & 0x000000ff))
```

The real problem comes when more general communication patterns must be implemented. Only the HP MAX instruction set supports arbitrary rearrangement of fields with a single instruction, which is called `Permute`. This `Permute` instruction is really misnamed; not only can it perform an arbitrary permutation of the fields, but it also allows repetition. In short, it implements an arbitrary `x[y]` operation.

Unfortunately, `x[y]` is very difficult to implement without such an instruction. The code sequence is generally both long and inefficient; in fact, it is sequential code. This is very disappointing. The relatively high speed of `x[y]` operations in the MasPar MP1/MP2 and Thinking Machines CM1/CM2/CM200 SIMD supercomputers was one of the key reasons these machines performed well. However, `x[y]` has always been

slower than nearest neighbor communication, even on those supercomputers, so many algorithms have be designed to minimize the need for `x[y]` operations. In short, without hardware support, it is probably be develop SWAR algorithms as though `x[y]` wasn't legal... or at least isn't cheap.

TOC

Recurrence Operations (Reductions, Scans, etc.)

A recurrence is a computation in which there is an apparently sequential relationship between values being computed. However, if these recurrences involve associative operations, it may be possible to recode the computation using a tree-structured parallel algorithm.

The most common type of parallelizable recurrence is probably the class known as associative reductions. For example, to compute the sum of a vector's values, one commonly writes purely sequential C code like:

```
t = 0;
for (i=0; i<MAX; ++i) t += x[i];
```

However, the order of the additions is rarely important. Floating point and saturation math can yield different answers if the order of additions is changed, but ordinary wrap-around integer additions will yield the same results independent of addition order. Thus, we can re-write this sequence into a tree-structured parallel summation in which we first add pairs of values, then pairs of those partial sums, and so forth, until a single final sum results. For a vector of four 8-bit values, just two addition steps are needed; the first step does two 8-bit adds, yielding two 16-bit result fields (each containing a 9-bit result):

```
t = ((x & 0x00ff00ff) + ((x >> 8) & 0x00ff00ff));
```

The second step adds these two 9-bit values in 16-bit fields to produce a single 10-bit result:

```
((t + (t >> 16)) & 0x000003ff)
```

Actually, the second step performs two 16-bit field adds... but the top 16-bit add is meaningless, which is why the result is masked to a single 10-bit result value.

Scans, also known as "parallel prefix" operations, are somewhat harder to implement efficiently. This is because, unlike reductions, scans produce partitioned results. For this reason, scans can be implemented using a fairly obvious sequence of partitioned operations.

4.3 MMX SWAR Under Linux

For Linux, IA32 processors are our primary concern. The good news is that AMD, Cyrix, and Intel all implement the same MMX instructions. However, MMX performance varies; for example, the K6 has only one MMX pipeline - the Pentium with MMX has two. The only really bad news is that Intel is still running those stupid MMX commercials.... ;-)

There are really three approaches to using MMX for SWAR:

1. Use routines from an MMX library. In particular, Intel has developed several "performance libraries," http://developer.intel.com/drg/tools/ad.htm, that offer a variety of hand-optimized routines for common multimedia tasks. With a little effort, many non-multimedia algorithms can be reworked to enable some of the most compute-intensive portions to be implemented using one or more of these library routines. These libraries are not currently available for Linux, but could be ported.
2. Use MMX instructions directly. This is somewhat complicated by two facts. The first problem is that MMX might not be available on the processor, so an alternative implementation must also be provided. The second problem is that the IA32 assembler generally used under Linux does not currently recognize MMX instructions.
3. Use a high-level language or module compiler that can directly generate appropriate MMX instructions. Such tools are currently under development, but none is yet fully functional under Linux. For example, at Purdue University (http://dynamo.ecn.purdue.edu/~hankd/SWAR/) we are currently developing a compiler that will take functions written in an explicitly parallel C dialect and will generate SWAR modules that are callable as C functions, yet make use of whatever SWAR support is available, including MMX. The first prototype module compilers were built in Fall 1996, however, bringing this technology to a usable state is taking much longer than was originally expected.

In summary, MMX SWAR is still awkward to use. However, with a little extra effort, the second approach given above can be used now. Here are the basics:

1. You cannot use MMX if your processor does not support it. The following GCC code can be used to test if MMX is supported on your processor. It returns 0 if not, non-zero if it is supported.

```
inline extern
int mmx_init(void)
{
        int mmx_available;

        __asm__ __volatile__ (
                /* Get CPU version information */
                "movl $1, %%eax\n\t"
                "cpuid\n\t"
                "andl $0x800000, %%edx\n\t"
                "movl %%edx, %0"
                : "=q" (mmx_available)
                : /* no input */
        );
        return mmx_available;
}
```

2. An MMX register essentially holds one of what GCC would call an `unsigned long long`. Thus, memory-based variables of this type become the communication mechanism between your MMX modules and the C programs that call them. Alternatively, you can declare your MMX data as any 64-bit aligned data structure (it is convenient to ensure 64-bit alignment by declaring your data type as a `union` with an `unsigned long long` field).

3. If MMX is available, you can write your MMX code using the `.byte` assembler directive to encode each instruction. This is painful stuff to do by hand, but not difficult for a compiler to generate. For example, the MMX instruction `PADDB MM0,MM1` could be encoded as the GCC in-line assembly code:

```
__asm__ __volatile__ (".byte 0x0f, 0xfc, 0xc1\n\t");
```

   Remember that MMX uses some of the same hardware that is used for floating point operations, so code intermixed with MMX code must not invoke any floating point operations. The floating point stack also should be empty before executing any MMX code; the floating point stack is normally empty at the beginning of a C function that does not use floating point.

4. Exit your MMX code by executing the `EMMS` instruction, which can be encoded as:

```
__asm__ __volatile__ (".byte 0x0f, 0x77\n\t");
```

If the above looks very awkward and crude, it is. However, MMX is still quite young.... future versions of this document will offer better ways to program MMX SWAR.

5. Linux-Hosted Attached Processors

Although this approach has recently fallen out of favor, it is virtually impossible for other parallel processing methods to achieve the low cost and high performance possible by using a Linux system to host an attached parallel computing system. The problem is that very little software support is available; you are pretty much on your own.

5.1 A Linux PC Is A Good Host

In general, attached parallel processors tend to be specialized to perform specific types of functions.

Before becoming discouraged by the fact that you are somewhat on your own, it is useful to understand that, although it may be difficult to get a Linux PC to appropriately host a particular system, a Linux PC is one of the few platforms well suited to this type of use.

PCs make a good host for two primary reasons. The first is the cheap and easy expansion capability; resources such as more memory, disks, networks, etc., are trivially added to a PC. The second is the ease of interfacing. Not only are ISA and PCI bus prototyping cards widely available, but the parallel port offers reasonable performance in a completely non-invasive interface. The IA32 separate I/O space also facilitates interfacing by providing hardware I/O address protection at the level of individual I/O port addresses.

Linux also makes a good host OS. The free availability of full source code, and extensive "hacking" guides, obviously are a tremendous help. However, Linux also provides good near-real-time scheduling, and there is even a true real-time version of Linux at http://luz.cs.nmt.edu/~rtlinux/. Perhaps even more important is the fact that while providing a full UNIX environment, Linux can support development tools that were written to run under Microsoft DOS and/or Windows. MSDOS programs can execute within a Linux process using `dosemu` to provide a protected virtual machine that can literally run MSDOS. Linux support for Windows 3.xx programs is even more direct: free software such as `wine`, http://www.linpro.no/wine/, simulates Windows 3.11 well enough for most programs to execute correctly and efficiently within a UNIX/X environment.

The following two sections give examples of attached parallel systems that I'd like to see supported under Linux....

5.2 Did You DSP That?

There is a thriving market for high-performance DSP (Digital Signal Processing) processors. Although these chips were generally designed to be embedded in application-specific systems, they also make great attached parallel computers. Why?

- Many of them, such as the Texas Instruments (http://www.ti.com/) TMS320 and the Analog

TOC

Devices (http://www.analog.com/) SHARC DSP families, are designed to construct parallel machines with little or no "glue" logic.

- They are cheap, especially per MIP or MFLOP. Including the cost of basic support logic, it is not unheard of for a DSP processor to be one tenth the cost of a PC processor with comparable performance.
- They do not use much power nor generate much heat. This means that it is possible to have a bunch of these chips powered by a conventional PC's power supply - and enclosing them in your PC's case will not turn it into an oven.
- There are strange-looking things in most DSP instruction sets that high-level (e.g., C) compilers are unlikely to use well - for example, "Bit Reverse Addressing." Using an attached parallel system, it is possible to straightforwardly compile and run most code on the host, while running the most time-consuming few algorithms on the DSPs as carefully hand-tuned code.
- These DSP processors are not really designed to run a UNIX-like OS, and generally are not very good as stand-alone general-purpose computer processors. For example, many do not have memory management hardware. In other words, they work best when hosted by a more general-purpose machine... such as a Linux PC.

Although some audio cards and modems include DSP processors that Linux drivers can access, the big payoff comes from using an attached parallel system that has four or more DSP processors.

Because the Texas Instruments TMS320 series, http://www.ti.com/sc/docs/dsps/dsphome.htm, has been very popular for a long time, and it is trivial to construct a TMS320-based parallel processor, there are quite a few such systems available. There are both integer-only and floating-point capable versions of the TMS320; older designs used a somewhat unusual single-precision floating-point format, but the new models support IEEE formats. The older TMS320C4x (aka, 'C4x) achieves up to 80 MFLOPS using the TI-specific single-precision floating-point format; in contrast, a single 'C67x will provide up to 1 GFLOPS single-precision or 420 MFLOPS double-precision for IEEE floating point calculations, using a VLIW-based chip architecture called VelociTI. Not only is it easy to configure a group of these chips as a multiprocessor, but in a single chip, the 'C8x multiprocessor will provide a 100 MFLOPS IEEE floating-point RISC master processor along with either two or four integer slave DSPs.

The other DSP processor family that has been used in more than a few attached parallel systems lately is the SHARC (aka, ADSP-2106x) from Analog Devices http://www.analog.com/. These chips can be configured as a 6-processor shared memory multiprocessor without external glue logic, and larger systems also can be configured using six 4-bit links/chip. Most of the larger systems seem targeted to military applications, and are a bit pricey. However, Integrated Computing Engines, Inc., http://www.iced.com/, makes an interesting little two-board PCI card set called GreenICE. This unit contains an array of 16 SHARC processors, and is capable of delivering a peak speed of about 1.9 GFLOPS using a single-precision IEEE format. GreenICE costs less than $5,000.

In my opinion, attached parallel DSPs really deserve a lot more attention from the Linux parallel processing community....

5.3 FPGAs And Reconfigurable Logic Computing

If parallel processing is all about getting the highest speedup, then why not build custom hardware? Well, we all know the answers; it costs too much, takes too long to develop, becomes useless when we change the algorithm even slightly, etc. However, recent advances in electrically reprogrammable FPGAs (Field Programmable Gate Arrays) have nullified most of those objections. Now, the gate density is high enough so that an entire simple processor can be built within a single FPGA, and the time to reconfigure (reprogram) an FPGA has also been dropping to a level where it is reasonable to reconfigure even when moving from one phase of an algorithm to the next.

This stuff is not for the weak of heart: you'll have to work with hardware description languages like VHDL for the FPGA configuration, as well as writing low-level code to interface to programs on the Linux host system. However, the cost of FPGAs is low, and especially for algorithms operating on low-precision integer data (actually, a small superset of the stuff SWAR is good at), FPGAs can perform complex operations just about as fast as you can feed them data. For example, simple FPGA-based systems have yielded better-than-supercomputer times for searching gene databases.

There are other companies making appropriate FPGA-based hardware, but the following two companies represent a good sample.

Virtual Computer Company offers a variety of products using dynamically reconfigurable SRAM-based Xilinx FPGAs. Their 8/16 bit "Virtual ISA Proto Board" http://www.vcc.com/products/isa.html is less than $2,000.

The Altera ARC-PCI (Altera Reconfigurable Computer, PCI bus), http://www.altera.com/html/new/pressrel/pr_arc-pci.html, is a similar type of card, but uses Altera FPGAs and a PCI bus interface rather than ISA.

Many of the design tools, hardware description languages, compilers, routers, mappers, etc., come as object code only that runs under Windows and/or DOS. You could simply keep a disk partition with DOS/Windows on your host PC and reboot whenever you need to use them, however, many of these software packages may work under Linux using `dosemu` or Windows emulators like `wine`.

---

6. Of General Interest

The material covered in this section applies to all four parallel processing models for Linux.

6.1 Programming Languages And Compilers

I am primarily known as a compiler researcher, so I'd like to be able to say that there are lots of really great compilers automatically generating efficient parallel code for Linux systems. Unfortunately, the truth is that it is hard to beat the performance obtained by expressing your parallel program using various explicit communication and other parallel operations within C code that is compiled by GCC.

The following language/compiler projects represent some of the best efforts toward producing reasonably efficient code from high-level languages. Generally, each is reasonably effective for the kinds of programming tasks it targets, but none is the powerful general-purpose language and compiler system that will make you forever stop writing C programs to compile with GCC... which is fine. Use these languages and compilers as they were intended, and you'll be rewarded with shorter development times, easier debugging and maintenance, etc.

There are plenty of languages and compilers beyond those listed here (in alphabetical order). A list of freely available compilers (most of which have nothing to do with Linux parallel processing) is at http://www.idiom.com/free-compilers/.

Fortran 66/77/PCF/90/HPF/95

At least in the scientific computing community, there will always be Fortran. Of course, now Fortran doesn't mean the same thing it did in the 1966 ANSI standard. Basically, Fortran 66 was pretty simple stuff. Fortran 77 added tons of features, the most noticeable of which were the improved support for character data and the change of `DO` loop semantics. PCF (Parallel Computing Forum) Fortran attempted to add a variety of parallel processing support features to 77. Fortran 90 is a fully-featured modern language, essentially adding C++-like object-oriented programming features and parallel array syntax to the 77 language. HPF (High-Performance Fortran, http://www.crpc.rice.edu/HPFF/home.html), which has itself gone through two versions (HPF-1 and HPF-2), is essentially the enhanced, standardized, version of what many of us used to know as CM Fortran, MasPar Fortran, or Fortran D; it extends Fortran 90 with a variety of parallel processing enhancements, largely focussed on specifying data layouts. Finally, Fortran 95 represents a relatively minor enhancement and refinement of 90.

What works with C generally can also work with `f2c`, `g77` (a nice Linux-specific overview is at http://linux.uni-regensburg.de/psi_linux/gcc/html_g77/g77_91.html), or the commercial Fortran 90/95 products from http://extweb.nag.co.uk/nagware/NCNJNKNM.html. This is because all of these compilers eventually come down to the same code-generation used in the back-end of GCC.

Commercial Fortran parallelizers that can generate code for SMPs are available from

TOC

http://www.kai.com/ and http://www.psrv.com/vast/vast_parallel.html. It is not clear if these compilers will work for SMP Linux, but it should be possible given that the standard POSIX threads (i.e., LinuxThreads) work under SMP Linux.

The Portland Group, http://www.pgroup.com/, has commercial parallelizing HPF Fortran (and C, C++) compilers that generate code for SMP Linux; they also have a version targeting clusters using MPI or PVM. FORGE/spf/xHPF products at http://www.apri.com/ might also be useful for SMPs or clusters.

Freely available parallelizing Fortrans that might be made to work with parallel Linux systems include:

- ADAPTOR (Automatic DAta Parallelism TranslaTOR, http://www.gmd.de/SCAI/lab/adaptor/adaptor_home.html), which can translate HPF into Fortran 77/90 code with MPI or PVM calls, but does not mention Linux.
- Fx http://www.cs.cmu.edu/~fx/Fx at Carnegie Mellon targets some workstation clusters, but Linux?
- HPFC (prototype HPF Compiler, http://www.cri.ensmp.fr/~coelho/hpfc.html) generates Fortran 77 code with PVM calls. Is it usable on a Linux cluster?
- Can PARADIGM (PARAllelizing compiler for DIstributed-memory General-purpose Multicomputers, http://www.crhc.uiuc.edu/Paradigm/) be used with Linux?
- The Polaris compiler, http://ece.www.ecn.purdue.edu/~eigenman/polaris/, generates Fortran code for shared memory multiprocessors, and may soon be retargeted to PAPERS Linux clusters.
- PREPARE, http://www.irisa.fr/EXTERNE/projet/pampa/PREPARE/prepare.html, targets MPI clusters... it is not clear if it can generate code to run on IA32 processors.
- Combining ADAPT and ADLIB, shpf (Subset High Performance Fortran compilation system, http://www.ccg.ecs.soton.ac.uk/Projects/shpf/shpf.html) is public domain and generates Fortran 90 with MPI calls... so, if you have a Fortran 90 compiler under Linux....
- SUIF (Stanford University Intermediate Form, see http://suif.stanford.edu/) has parallelizing compilers for both C and Fortran. This is also the focus of the National Compiler Infrastructure Project... so, is anybody targeting parallel Linux systems?

I'm sure that I have omitted many potentially useful compilers for various dialects of Fortran, but there are so many that it is difficult to keep track. In the future, I would prefer to list only those compilers known to work with Linux. Please email comments and/or corrections to pplinux@ecn.purdue.edu.

GLU (Granular Lucid)

GLU (Granular Lucid) is a very high-level programming system based on a hybrid programming model that combines intensional (Lucid) and imperative models. It supports both PVM and TCP sockets. Does it run under Linux? More information is available at http://www.csl.sri.com/GLU.html.

TOC

Jade And SAM

Jade is a parallel programming language that extends C to exploit coarse-grain concurrency in sequential, imperative programs. It assumes a distributed shared memory model, which is implemented by SAM for workstation clusters using PVM. More information is available at http://suif.stanford.edu/~scales/sam.html.

Mentat And Legion

Mentat is an object-oriented parallel processing system that works with workstation clusters and has been ported to Linux. Mentat Programming Language (MPL) is an object-oriented programming language based on C++. The Mentat run-time system uses something vaguely resembling non-blocking remote procedure calls. More information is available at http://www.cs.virginia.edu/~mentat/.

Legion http://www.cs.virginia.edu/~legion/ is built on top on Mentat, providing the appearance of a single virtual machine across wide-area networked machines.

MPL (MasPar Programming Language)

Not to be confussed with Mentat's MPL, this language was originally developed as the native parallel C dialect for the MasPar SIMD supercomputers. Well, MasPar isn't really in that business any more (they are now NeoVista Solutions, http://www.neovista.com, a data mining company), but their MPL compiler was built using GCC, so it is still freely available. In a joint effort between the University of Alabama at Huntsville and Purdue University, MasPar's MPL has been retargeted to generate C code with AFAPI calls (see section 3.6), and thus runs on both Linux SMPs and clusters. The compiler is, however, somewhat buggy... see http://www.math.luc.edu/~laufer/mspls/papers/cohen.ps.

PAMS (Parallel Application Management System)

Myrias is a company selling a software product called PAMS (Parallel Application Management System). PAMS provides very simple directives for virtual shared memory parallel processing. Networks of Linux machines are not yet supported. See http://www.myrias.com/ for more information.

Parallaxis-III

Parallaxis-III is a structured programming language that extends Modula-2 with "virtual processors and connections" for data parallelism (a SIMD model). The Parallaxis software comprises compilers for sequential and parallel computer systems, a debugger (extensions to the gdb and xgbd debugger), and a large variety of sample algorithms from different areas, especially image processing. This runs on sequential Linux systems... an old version supported various parallel targets, and the new version also will (e.g., targeting a PVM cluster). More information is available at http://www.informatik.uni-stuttgart.de/ipvr/bv/p3/p3.html.

TOC

pC++/Sage++

pC++/Sage++ is a language extension to C++ that permits data-parallel style operations using "collections of objects" from some base "element" class. It is a preprocessor generating C++ code that can run under PVM. Does it run under Linux? More information is available at http://www.extreme.indiana.edu/sage/.

SR (Synchronizing Resources)

SR (Synchronizing Resources) is a concurrent programming language in which resources encapsulate processes and the variables they share; operations provide the primary mechanism for process interaction. SR provides a novel integration of the mechanisms for invoking and servicing operations. Consequently, all of local and remote procedure call, rendezvous, message passing, dynamic process creation, multicast, and semaphores are supported. SR also supports shared global variables and operations.

It has been ported to Linux, but it isn't clear what parallelism it can execute with. More information is available at http://www.cs.arizona.edu/sr/www/index.html.

ZPL And IronMan

ZPL is an array-based programming language intended to support engineering and scientific applications. It generates calls to a simple message-passing interface called IronMan, and the few functions which constitute this interface can be easily implemented using nearly any message-passing system. However, it is primarily targeted to PVM and MPI on workstation clusters, and Linux is supported. More information is available at http://www.cs.washington.edu/research/projects/orca3/zpl/www/.

6.2 Performance Issues

There are a lot of people who spend a lot of time benchmarking particular motherboards, network cards, etc., trying to determine which is the best. The problem with that approach is that by the time you've been able to benchmark something, it is no longer the best available; it even may have been taken off the market and replaced by a revised model with entirely different properties.

Buying PC hardware is like buying orange juice. Usually, it is made with pretty good stuff no matter what company name is on the label. Few people know, or care, where the components (or orange juice concentrate) came from. That said, there are some hardware differences that you should pay attention to. My advice is simply that you be aware of what you can expect from the hardware under Linux, and then focus your attention on getting rapid delivery, a good price, and a reasonable policy for returns.

An excellent overview of the different PC processors is given in http://www.pcguide.com/ref/cpu/fam/; in fact, the whole WWW site http://www.pcguide.com/ is full of good technical overviews of PC hardware. It is also useful to know a bit about performance of specific hardware configurations, and the Linux Benchmarking HOWTO http://sunsite.unc.edu/LDP/HOWTO/Benchmarking-HOWTO.html is a good place to start.

The Intel IA32 processors have many special registers that can be used to measure the performance of a running system in exquisite detail. Intel VTune, http://developer.intel.com/design/perftool/vtune/, uses the performance registers extensively in a very complete code-tuning system... that unfortunately doesn't run under Linux. A loadable module device driver, and library routines, for accessing the Pentium performance registers is available from http://www.cs.umd.edu/users/akinlar/driver.html. Keep in mind that these performance registers are different on different IA32 processors; this code works only with Pentium, not with 486, Pentium Pro, Pentium II, K6, etc.

Another comment on performance is appropriate, especially for those of you who want to build big clusters and put them in small spaces. At least some modern processors incorporate thermal sensors and circuits that are used to slow the internal clock rate if operating temperature gets too high (an attempt to reduce heat output and improve reliability). I'm not suggesting that everyone should go buy a peltier device (heat pump) to cool each CPU, but you should be aware that high operating temperature does not just shorten component life - it also can directly reduce system performance. Do not arrange your computers in physical configurations that block airflow, trap heat within confined areas, etc.

Finally, performance isn't just speed, but also reliability and availability. High reliability means that your system almost never crashes, even when components fail... which generally requires special features like redundant power supplies and hot-swap motherboards. That usually isn't cheap. High availability refers to the concept that your system is available for use nearly all the time... the system may crash when components fail, but the system is quickly repaired and rebooted. There is a High-Availability HOWTO that discusses many of the basic issues. However, especially for clusters, high availablity can be achieved simply by having a few spares. I recommend at least one spare, and prefer to have at least one spare for every 16 machines in a large cluster. Discarding faulty hardware and replacing it with a spare can yield both higher availability and lower cost than a maintenance contract.

6.3 Conclusion - It's Out There

So, is anybody doing parallel processing using Linux? Yes!

It wasn't very long ago that a lot of people were wondering if the death of many parallel-processing supercomputer companies meant that parallel processing was on its way out. I didn't think it was dead then (see http://dynamo.ecn.purdue.edu/~hankd/Opinions/pardead.html for a fun overview of what I think really happened), and it seems quite clear now that parallel processing is again on the rise. Even Intel, which just recently stopped making parallel supercomputers, is proud of the parallel processing support in things like MMX and the upcoming IA64 EPIC (Explicitly Parallel Instruction Computer).

If you search for "Linux" and "parallel" with your favorite search engine, you'll find quite a few places are involved in parallel processing using Linux. In particular, Linux PC clusters seem to be popping-up everywhere. The appropriateness of Linux, combined with the low cost and high performance of PC hardware, have made parallel processing using Linux a popular approach to supercomputing for both small, budget-constrained, groups and large, well-funded, national research laboratories.

Various projects listed elsewhere in this document maintain lists of "kindred" research sites that have similar parallel Linux configurations. However, at http://yara.ecn.purdue.edu/~pplinux/Sites/, there is a hypertext document intended to provide photographs, descriptions, and contact information for all the

TOC

various sites using Linux systems for parallel processing. To have information about your site posted there:

- You must have a "permanent" parallel Linux site: an SMP, cluster of machines, SWAR system, or PC with attached processor, which is configured to allow users to *execute parallel programs under Linux*. A Linux-based software environment (e.g., PVM, MPI, AFAPI) that directly supports parallel processing must be installed on the system. However, the hardware need not be dedicated to parallel processing under Linux, and may be used for completely different purposes when parallel programs are not being run.
- Request that your site be listed. Send your site information to pplinux@ecn.purdue.edu. Please follow the format used in other entries for your site information. *No site will be listed without an explicit request from the contact person for that site.*

There are 14 clusters in the current listing, but we are aware of at least several dozen Linux clusters world-wide. Of course, listing does not imply any endorsement, etc.; our hope is simply to increase awareness, research, and collaboration involving parallel processing using Linux.

The Linux Plug-and-Play-HOWTO

David S.Lawyer mailto:bf347@lafn.org

v0.05, August 1999

Help with understanding and dealing with the complex Plug-and-Play issue. How to get your Linux system to support Plug-and-Play.

1. Introduction

- 1.1 Copyright, Trademarks, Disclaimer, & Credits
- 1.2 Future Plans; You Can Help
- 1.3 New Versions of this HOWTO

2. What PnP Should Do: Allocate "Resources"

- 2.1 What is Plug-and-Play (PnP)?
- 2.2 How a Computer Finds Devices (and conversely)
- 2.3 I/O Addresses, etc.
- 2.4 IRQs --Overview
- 2.5 DMA Channels
- 2.6 Memory Ranges
- 2.7 "Resources" to both Device and Driver
- 2.8 The Problem
- 2.9 PnP Finds Devices Plugged Into Serial Ports

3. The Plug-and-Play (PnP) Solution

- 3.1 Introduction to PnP
- 3.2 How It Works (simplified)
- 3.3 Starting Up the PC

- 3.4 Buses
- 3.5 Linux Needs to Cope Better with PnP

4. Configuring a PnP BIOS

- 4.1 Do you have a PnP operating system?
- 4.2 How are resources to be controlled?
- 4.3 Reset the configuration?

5. How to Deal with PnP Cards

- 5.1 Introduction to Dealing with PnP Cards
- 5.2 Disable PnP ?
- 5.3 BIOS Configures PnP
- 5.4 Isapnp (part of isapnptools)
- 5.5 PCI Utilities
- 5.6 Patch the Kernel to Make Linux PnP
- 5.7 Windows Configures
- 5.8 Device Driver Configures
- 5.9 PnP Software/Documents

6. Tell the Driver the Configuration

- 6.1 Introduction
- 6.2 Serial Port Driver: setserial
- 6.3 Sound Card Drivers

7. What Is My Current Configuration?

- 7.1 How Are My Device Drivers Configured?
- 7.2 How Are My Hardware Devices Configured?

8. Appendix

- 8.1 Addresses
- 8.2 Interrupts --Details
- 8.3 PCI Interrupts
- 8.4 Isolation

1. Introduction

1.1 Copyright, Trademarks, Disclaimer, & Credits

Copyright

Copyright (c) 1998, 1999 by David S. Lawyer. Please freely copy and distribute (sell or give away) this document. For corrections and minor changes contact the maintainer. Otherwise you may create derivative works and distribute them provided you:

1. Discuss it with the maintainer (if there is one). 2. Put the derivative work at the mirrored LDP Internet site (or the like) for free downloading. 3. License the work in the spirit of this license or use GPL. 4. Give due credit to previous authors and major contributors.

Trademarks

If certain words are trademarks, the context should make it clear to whom they belong. For example "MS Windows" (or just "Windows") implies that "Windows" belongs to Microsoft.

Disclaimer

Much of the info in this HOWTO was obtained from the Internet, implications in books that may be obsolete, etc. While I haven't intentionally tried to mislead you, there are likely a number of errors in this document. Please let me know about them. Since this is free documentation, it should be obvious that neither I nor previous authors can be held legally responsible for any errors.

1.2 Future Plans; You Can Help

Please let me know of any errors in facts, opinions, logic, spelling, grammar, clarity, links, etc. But first, if the date is over a month old, check to see that you have the latest version. Please send me any info that you think belongs in this document.

I haven't studied in detail neither isapnptools nor David Howells' patches to the kernel (but I plan to). Nor do I fully understand how PnP is configured by the BIOS (it depends on which BIOS) nor how Windows9x updates the ESCD. Thus this HOWTO is still incomplete and may be inaccurate (let me know where I'm wrong). In this HOWTO I've sometimes used ?? to indicate that I don't really know the answer.

1.3 New Versions of this HOWTO

TOC

New versions of the Plug-and-Play-HOWTO should appear every month or so and will be available to browse and/or download at LDP mirror sites. For a list of mirror sites see: http://metalab.unc.edu/LDP/mirrors.html. Various formats are available. If you only want to quickly check the date of the latest version look at: http://metalab.unc.edu/LDP/HOWTO/Plug-and-Play-HOWTO.html.

---

2. What PnP Should Do: Allocate "Resources"

2.1 What is Plug-and-Play (PnP)?

Oversimplified, Plug-and-Play automatically tells the software (device drivers) where to find various pieces of hardware (devices) such as modems, network cards, sound cards, etc. Plug-and-Play's task is to match up physical devices with the software (device drivers) that operates them and to establish channels of communication between each device and its driver. In order to achieve this, PnP allocates the following "resources" to both drivers and hardware: I/O addresses, IRQ's, DMA channels (ISA bus only), and memory regions. If you don't understand what these 4 items are read the following subsections. Once these resources have been assigned (and if the correct driver is installed), the names for such devices in the /dev directory are ready to use.

This PnP assignment of certain resources is sometimes called "configuring" but it is only a low level type of configuring. Even with PnP fully utilized, much configuring of devices is done by other than PnP. For example, for modem configuration an "init string" is sent to the modem over the I/0 address "channel". This "init string" has nothing to do with PnP although the "channel" used to send it to the modem was allocated by PnP. Setting the speed (and many other parameters) of a serial port is done by sending messages to the device driver from programs run by the user (often automatically boot-time). This configuring also has nothing to do with PnP. Thus when talking about PnP, "resources" means only a limited subset of resources and "configuring" means only a certain type of configuring.

2.2 How a Computer Finds Devices (and conversely)

A computer consists of a CPU/processor to do the computing and memory to store programs and data. In addition, there are a number of devices such as various kinds of disk-drives, a video card, a keyboard, network cards, modem cards, sound cards, serial and parallel ports, etc. There is also a power supply to provide electric energy, various buses on a motherboard to connect the devices to the CPU, and a case to put all this into.

In olden days most all devices had their own plug-in cards (printed circuit boards). Today, in addition to plug-in cards, many "devices" are small chips permanently mounted on the "motherboard". Cards which plug into the motherboard may contain more than one device. Memory chips are also sometimes considered to be devices but are not plug-and-play in the sense used in this HOWTO.

For the computer system to work right, each device must be under the control of its "device driver". This is software which is a part of the operating system (or a module) and runs on the CPU. Device drivers are associated with "special files" in the /dev directory although they are not really files. They have names such as hda1 (first partition on hard drive a), ttyS0 (the first serial port), eth1 (the second ethernet card), etc. To make matters more complicated, the particular device driver selected, say for eth1, will depend on

the type of ethernet card you have. Thus eth1 can't just be assigned to any ethernet driver. It must be assigned to a certain driver that will work for the type of ethernet card you have installed. To control a device, the CPU (under the control of the device driver) sends commands (and data) to and reads info from the various devices. In order to do this each device driver must know the address of the device it controls. Knowing such an address is equivalent to setting up a communication channel, even though the physical "channel" is actually the data bus inside the PC which is shared with almost everything else.

The communication channel is actually a little more complex than described above. An "address" is actually a range of addresses and there is a reverse part of the channel (known as interrupts) which allows devices to send an urgent "help" request to their device driver.

2.3 I/O Addresses, etc.

PC's have 3 address spaces: I/O, main memory, and configuration (only on the PCI bus). All of these 3 types of addresses share the same address bus inside the PC. But the voltage on certain dedicated wires tells all devices which "space" an address is in: I/O, main memory, or configuration. See Addresses for more details. Devices were normally located in I/O address space although today they may use space in main memory. An I/0 address is sometimes just called "I/O", "IO", "i/o" or "io". The term "I/O port" also used. There are two main steps to allocate the I/O addresses (or other resources such as interrupts):

1. Set the I/O address, etc. on the card (in one of its registers)
2. Let its device driver know what this I/O address, etc. is

The two step process above is something like the two part problem of finding someone's house number on a street. You must obtain (and write down) the house number and someone must install a number on the front of the house so that it may be found. In computers, the device driver must obtain the address and the device hardware must get the same address set in one of its registers. Both of these must be done, but some people make the mistake of doing only one of these and then wonder why the computer can't find the device. For example, they will use "setserial" to assign an address to a serial port without realizing that this only tells the driver the address. It doesn't set the address in the serial port hardware itself. If you told the driver the wrong address, you're in trouble.

Another obvious requirement is that the I/O address must be set on the card before the device driver tries to use this address. Since device drivers often start up soon after you start the computer, they sometimes try to access a card (to see if it's there, etc.) before the address has been set in the card by a PnP configuration program. Then you see an error message that they can't find the card even though it's there (but doesn't yet have an address).

What was said in the last 2 paragraphs regarding I/O addresses applies with equal force to other resources: IRQs --Overview, DMA Channels, and Memory Regions. What theses are will be explained in the next 3 sections.

TOC

2.4 IRQs --Overview

After reading this you may read Interrupts --Details for some more details. The following is intentionally oversimplified: Besides the address, there is also an interrupt number to deal with (such as IRQ 5). It's called an IRQ (Interrupt ReQuest) number. We already mentioned above that the device driver must know the address of a card in order to be able to communicate with it. But what about communication in the opposite direction? Suppose the the device needs to tell its device driver something immediately? For example, the device may have just received a lot of bytes destined for main memory and the device needs to call its driver to fetch these bytes at once and transfer them from the device's nearly full buffer into main memory.

How should the device call for help? It can't use the main data bus since it's likely already in use. Instead it puts a voltage on an dedicated interrupt wire (part of the bus) which is often reserved for that device alone. This signal is called an interrupt. There are the equivalent of 16 such wires in a PC and each wire leads (indirectly) to a certain device driver. Each wire has a unique IRQ (Interrupt ReQuest) number. The device must put its interrupt on the correct wire and the device driver must listen for the interrupt on the correct wire. Which wire it's put on is determined by the IRQ number stored in the device. This same IRQ number must be known to the device driver so that the device driver knows which IRQ line to listen to.

Once the device driver gets the interrupt (a call for help) it must find out why the interrupt was issued and take appropriate action to service the interrupt. On the ISA bus each device needs its own unique IRQ number. For the PCI bus and other special cases the sharing of IRQs is allowed.

2.5 DMA Channels

DMA channels are only for the ISA bus. DMA stands for "Direct Memory Access". This is where a device is allowed to take over the main computer bus from the CPU and transfer bytes directly to main memory. Normally the CPU would make such a transfer in a two step process: 1. reading from the I/O memory space of the device and putting these bytes into the CPU itself 2. writing these bytes from the CPU to main memory. With DMA it's usually a one step process of sending the bytes directly from the device to memory. The device must have such capabilities built into its hardware and thus not all devices can do DMA. While DMA is going on the CPU can't do too much since the main bus is being used by The DMA transfer.

The PCI bus doesn't really have any DMA but instead it has something even better: bus mastering. It works something like DMA and is sometimes called DMA (for example, hard disk drives that call themselves "UltraDMA"). It allows devices to temporarily become bus masters and to transfer bytes almost like the bus master was the CPU. It doesn't use any channel numbers since the organization of the PCI bus is such that the PCI hardware knows which device is currently the bus master and which device is requesting to become a bus master. Thus there is no allocation of DMA channels for the PCI bus.

When a device on the ISA bus wants to do DMA it issues a DMA-request using dedicated DMA request wires much like an interrupt request. DMA actually could have been handled by using interrupts but this would introduce some delays so it's faster to do it by having a special type of interrupt known as a DMA-request. Like interrupts, DMA-request are numbered so as to identify which device is making the

TOC

request. This number is called a DMA-channel. Since DMA transfers all use the main bus (and only can run at a time) they all actually use the same channel but the "DMA channel" number serves to identify who is using the "channel". Hardware registers exist on the motherboard which store the current status of each "channel". Thus in order to issue a DMA-request, the device must know its DMA-channel number which must be stored in a register on the physical device.

2.6 Memory Ranges

Some devices are assigned address space in main memory. It's often "shared memory" or "memory-mapped I/O". Sometimes it's ROM memory on the device. When discussing PnP resources it's often just called "memory". Such a device might also use I/O address space.

When you plug in such a card, you are in effect also plugging in a memory module for main memory. This memory can either be ROM (Read Only Memory) or shared memory. Shared memory is shared between the device and the CPU (running the device driver). This memory can serve as a means of direct data "transfer" between the device and main memory. It's not really a transfer since the device puts data into its own memory on its card which also happens to be in main memory. Both the card and the device driver need to know where it is. The memory address are likely to be very high so that they do not conflict with the lower addresses of the memory chips in your computer.

ROM is different. It is likely a program (perhaps a device driver) which will be used with the device. Hopefully, it may work with Linux and not just Windows ?? It may need to be shadowed which means that it is copied to your main memory chips in order to run faster. Once it's shadowed it's no longer "read only".

2.7 "Resources" to both Device and Driver

Thus device drivers must be "attached" in some way to the hardware they control. This is done by supplying "resources" (I/O, Memory, IRQ's, DMA's) to both the physical device and the device driver software. For example, a serial port uses only 2 (out of 4 possible) resources: an IRQ and an I/O address. Both of these values must be supplied to the device driver and the physical device. The driver (and its device) is also given a name in the /dev directory (such as ttyS1). The address and IRQ number is stored by the physical device in registers on the card (or in a chip on the motherboard). For the case of jumpers, this info is always stored in the device hardware (on the card, etc.). But for the case of PnP, the register data is usually lost when the PC is powered down (turned off) so that the resource data must be supplied to each device anew each time the PC is powered on.

2.8 The Problem

The architecture of the PC provides only a limited number of IRQ's, DMA channels, I/O address, and memory regions. If there were only several devices and they all had standardized resource (such as unique I/O addresses and IRQ numbers) there would be no problem of attaching device drivers to devices. Each device would have a fixed resources which would not conflict with any other device on your computer. No two devices would have the same addresses, there would be no IRQ conflicts, etc.

Each driver would be programmed with the unique addresses, IRQ, etc. hard-coded into the program. Life would be simple.

But it's not. Not only are there so many different devices today that conflicts are frequent, but one sometimes needs to have more than one of the same type of device. For example, one may want to have a few different disk-drives, a few serial ports, etc. For these reasons devices need to have some flexibility so that they can be set to whatever address, IRQ, etc. is needed to avoid conflicts. But some IRQ's and addresses are pretty standard such as the ones for the clock and keyboard. These don't need such flexibility.

Besides the problem of conflicting allocation of resources, there is a problem of making a mistake in telling the device driver what the resources are. For example, suppose that you enter IRQ 4 in a configuration file when the device is actually set at IRQ 5. This is another type of resource allocation error.

The allocation of resources, if done correctly, establishes channels of communication between physical hardware and their device drivers. For example, if a certain I/O address range (resource) is allocated to both a device driver and a piece of hardware, then this has established a one-way communication channel between them. The driver may send commands and info to the device. It's actually a little more than one-way since the driver may get information from the device by reading it's registers. But the device can't initiate any communication this way. To initiate the device needs an IRQ in order to create a two-way communication channel where both the driver and the device can initiate communication.

2.9 PnP Finds Devices Plugged Into Serial Ports

External devices that connect to the serial port via a cable (such as external modems) can also be called Plug-and-Play. Since only the serial port itself needs resources (an IRQ and I/O address) there are no resources to allocate to such plug-in devices. Thus PnP is not really needed for them. Even so, there is a PnP specification for such external serial devices.

A PnP operating system will find such an external device and read its model number, etc. Then it may be able to find a device driver for it so that you don't have to tell an application program that you have a certain device on say /dev/ttyS1. Since you should be able to manually inform your application program (via a configuration file, etc.) what serial port the device is on (and possibly what model number it is) you should not really need this "serial port" feature of PnP.

---

3. The Plug-and-Play (PnP) Solution

3.1 Introduction to PnP

The term Plug-and-Play (PnP) has various meanings. In the broad sense it is just auto-configuration where one just plugs in a device and it configures itself. In the sense used in this HOWTO, the configuration is only that of configuring PnP resources and letting the device drivers know about it. In a more narrow sense it just setting resources in the hardware devices. It may also mean the PnP specifications which (among other things) specify how PnP resource data is to be read and written to devices (often cards) on the ISA bus. The standard PCI (and not PnP) specifications do the same for the PCI bus.

PnP matches up devices with their device drivers and specifies their communication channels. On the ISA bus before Plug-and-Play the resources were set in hardware devices by jumpers. Software drivers were assigned resources by configuration files (or the like) or by probing the for the device at addresses where it's expected to reside. The PCI bus was PnP-like from the beginning so it was trivial to implement PnP for this bus. Since the PCI bus specifications don't use the term PnP it's not clear whether or not the PCI bus should be called PnP (but it supports in hardware what today is called PnP).

3.2 How It Works (simplified)

Here's an oversimplified view of how PnP works. The PnP configuration program (perhaps a program in the BIOS) finds all PnP devices and asks each what resources it needs. Then it checks what resources (IRQs, etc.) it has to give away. Of course if it has reserved resources used by non-PnP (legacy) devices (if it knows about them) it doesn't give these away. Then it uses some criteria (not specified by PnP specifications) to give out the resources so that there are no conflicts and so that all devices get what they need (if possible). It then tells each physical device what resources are assigned to it and the devices set themselves up to use only the assigned resources. Then the device drivers somehow find out what resources their devices use and are thus able to communicate effectively with the devices they control.

For example, suppose a card needs one interrupt (IRQ number) and 1 MB of shared memory. The PnP program reads this request from the card. It then assigns the card IRQ5 and 1 MB of memory addresses space, starting at address 0xe9000000. It's not always this simple as the card may specify that it can only use certain IRQ numbers (ISA only) or that the 1 MB of memory must lie within a certain range of addresses. The details are different for the PCI and ISA buses with more complexity on the ISA bus.

There are some shortcuts that PnP software may use. One is to keep track of how it assigned resources at the last configuration (when the computer was last used) and reuse this. Windows9x and PnP BIOSs do this but standard Linux doesn't. Windows9x stores this info in its "Registry" and a PnP BIOS stores it in non-volatile memory in your PC (known as ESCD; see The BIOS's ESCD Database).

Under Linux it's each device for itself and there is no centralized registry of resource assignments. Some

device drivers store the last configuration they used and use it next time the computer is powered on. They implicitly assume that the hardware will be somehow configured with the same resources.

If the device hardware remembered their previous configuration, then there wouldn't be any hardware to configure at the next boot-time, but they seem to forget their configuration when the power is turned off. Some devices contain a default configuration (but not necessarily the last one used). Thus a PnP configuration program needs to be run each time the PC is powered on. Also, if a new device has been added, then it too needs to be configured. Allocating resources to this new device might involve taking some resources away from an existing device and assigning the existing device alternative resources that it can use instead.

3.3 Starting Up the PC

When the PC is first turned on the BIOS chip runs its program to get the computer started (the first step is to check out the hardware). If the operating system is stored on the hard-drive (as it normally is) then the BIOS must know about the hard-drive. If the hard-drive is PnP then the BIOS may use PnP methods to find it. Also, in order to permit the user to manually configure the BIOS's CMOS and respond to error messages when the computer starts up, a screen (video card) and keyboard are also required. Thus the BIOS must PnP-configure these devices on its own.

Once the BIOS has identified the hard-drive, the video card, and the keyboard it is ready to start booting (loading the operating system into memory from the hard-disk). If you've told the BIOS that you a have a PnP operating system (PnP OS), it should start booting the PC as above and let the operating system finish the PnP configuring. Otherwise, a PnP-BIOS will (prior to booting) likely try to do the rest of the PnP configuring of devices (but not their drivers).

3.4 Buses

ISA is the old bus of the old IBM PC's while PCI is a newer and faster bus from Intel. The PCI bus was designed for what is today called PnP. It makes it easy (as compared to the ISA bus) to find out how PnP resources have been assigned to hardware devices. To see what has happened look at the /proc/pci "file" (/proc/bus/pnp/devices for kernel 2.2+), the boot-up messages on your display (use shift-PageUp to back up), or use PCI Utilities (for kernel 2.2+).

For the ISA bus there is a real problem with implementing PnP since no one had PnP in mind when the ISA bus was designed and there are almost no I/O addresses available for PnP to use for sending configuration info to physical device. As a result, the way PnP was shoehorned onto the ISA bus is very complicated. A whole book has been written about it. See PnP Book. Among other things, it requires that each PnP device be assigned a temporary "handle" by the PnP program so that one may address it for PnP configuring. Assigning these "handles" is call "isolation". See Isolation for the complex details.

Eventually, the ISA bus should become extinct. When it does, PnP will be easier since it will be easy to find out how the BIOS has configured the hardware. There will still be the need to match up device drivers with devices and also a need to configure devices that are added when the PC is up and running. These needs would be satisfied if Linux was a PnP operating system.

3.5 Linux Needs to Cope Better with PnP

TOC

PnP (for the ISA bus) was invented by Compaq, Intel, and Phoenix. Microsoft has been a leading promoter of it. Linux would have been better off if PnP had never been "invented". Eventually the ISA bus will have become extinct and the PnP-like PCI bus will prevail so that we will have in effect gotten an easy-to-implement PnP. But like it or not, most all new ISA hardware today is PnP and Linux has no choice but to deal effectively with PnP. But standard Linux (as of early 1999) makes dealing with PnP complicated (especially on the ISA bus) while the purpose of PnP was to make it simple.

In a sense, Linux is already somewhat PnP for the PCI bus. When the PC starts up you may note from the messages on the screen that some Linux device drivers often find their hardware devices (and the resources the BIOS has assigned them). But there are situations that a PnP operating system could handle better: 1. A shortage of resources 2. More than one driver for a physical device 3. An activated driver which can't find it's physical device 4. Hot installation of a device (docking, etc.)

Linux users should not need to delve into the details of PnP to configure ISA PnP devices as they now need to. One solution would be a standardized version of the Linux kernel that supports Plug-and-Play on the ISA, PCI, and other buses. A patch to the kernel has been written although most drivers don't support it. It's not part of standard Linux. See Patch Kernel.

---

4. Configuring a PnP BIOS

When the computer is first turned on, the BIOS runs before the operating system is loaded. Newer BIOSs are PnP and will configure some or all of the PnP devices. For most PnP BIOSs there is no way to disable PnP so you have to live with it. Here are some of the choices which may exist in your BIOS's CMOS menu:

- Do you have a PnP operating system?
- How are resources to be controlled?
- Reset the configuration?

4.1 Do you have a PnP operating system?

If you say yes, then the PnP BIOS will PnP-configure the hard-drive, video card, and keyboard to make the system bootable. But it will leave it up to the operating system to finish the configuration job. It may do an Isolation on the ISA bus leaving the devices disabled but ready to be configured by the operating system. For Linux you should probably tell it that you don't have a PnP operating system. If you don't do this, the BIOS might leave the ISA devices it hasn't configured in a disabled state ?? Also PCI devices might not get configured ??

If you tell the BIOS you don't have a PnP OS, then the BIOS will do the configuring itself. Unless you have added new PnP devices, it should use the configuration which it has stored in its non-volatile memory (ESCD). See The BIOS's ESCD Database If the last session on your computer was with Linux, then there should be no change in configuration. See BIOS Configures PnP. But if the last session was with Windows9x (which are PnP) then Windows could have set up the configuration differently and possibly saved some of it in the ESCD. See Using Windows to set ESCD. If you are using the isapnp or PCI Utilities program(s) to do configuring, they will run after the BIOS runs and change things the way you told them to.

Interoperability with Windows9x

If you are running Windows on the same PC, how do you answer the question: Do you have a PnP OS? Normally (and truthfully) you would say no for standard Linux and yes for Windows9x. But it's a lot of bother to have to set up the CMOS menu manually each time you want to switch OSs. One solution is set the CMOS for no PnP OS, including when you start Windows. One might expect that Windows would be able to handle this situation where it is presented hardware that has been fully configured by the BIOS. In addition, one might expect that even if Windows didn't realize that the hardware was already configured, it would do the configuration itself and then work OK. But it doesn't seem to work this way. It seems that Windows may just tell its device drivers what has been stored in the Windows' Registry. But the actual hardware configuration (done by the BIOS) is what was stored in the ESCD and may not be the same as the Registry => trouble.

TOC

One way (the only way??) to try to get the Registry and the ESCD the same is to install (or reinstall) Windows when the BIOS is set for "not a PnP OS". This should present Windows with system configured by the BIOS (except for the devices drivers). If this configuration is without conflicts, Windows will hopefully leave it alone and save it in it's Registry. If this works for you (and this is the latest version of this HOWTO), let me know as I only have one report of this working out OK.

Another thing you might try if there is only a problem with one device is to tell Windows you are removing the device (perhaps hide the device driver). Then restart the PC with "not a PnP OS" and install the device under Windows. I have no idea if this works or not. Let me know if it does and what you did (only if this is the latest version of this HOWTO).

4.2 How are resources to be controlled?

This may involve just deciding how to allocate IRQ and DMA resources. If set to "auto", the BIOS will do the allocation. If set to manual, you manually reserve some IRQ's for use on "legacy" (non-pnp) cards. The BIOS may or may not otherwise know about your legacy cards. The BIOS will only know about your legacy cards if you ran ICU (or the like) under Windows to tell the BIOS about them. If the BIOS knows about them, then try using "auto". If it doesn't know about them then manually reserve the IRQ's needed for the legacy ISA cards and let the rest be for the BIOS PnP to allocate.

4.3 Reset the configuration?

This will erase the BIOSs ESCD data-base of how your PnP devices should be configured as well as the list of how legacy (non-PnP) devices are configured. Never do this unless you are convinced that this data-base is wrong and needs to be remade. It was stated somewhere that you should do this only if you can't get your computer to boot. If the BIOS loses the data on legacy devices, then you'll need to run ICA again under DOS/Windows to reestablish this data.

---

5. How to Deal with PnP Cards

5.1 Introduction to Dealing with PnP Cards

Today most all new internal boards (cards) are Plug-and-Play (PnP). Although some software exists in Linux to handle PnP, it is not always easy to use. There are 6 different methods listed below to cope with PnP (but some may not be feasible in your situation). Which one(s) you should use depends on your goals. What may be most expedient to do now may not be the easiest and best in the long run. A seemingly simple way is to do nothing and just let a PnP-BIOS configure it but then you may need to do some exploring to to find out what the BIOS has done. A comparison of these methods needs to be written by someone who has tried them all. You may need to use more than one method to do the job.

- Disable PnP by jumpers or DOS/Windows software (but many cards can't do this)
- BIOS Configures PnP (For the PCI bus you only need a PCI BIOS, otherwise you need a PnP BIOS)
- Isapnp is a program you can always use to configure PnP devices on the ISA bus only
- PCI Utilities is for configuring the PCI bus
- Windows Configures and then you boot Linux from within Windows/DOS. Use as a last resort
- Patch Kernel to transform Linux into a PnP operating system
- Device Driver Configures but few do

Any of the above will set the resources in the hardware. But only the last two should tell device driver what it's done. Only the last one definitely tells the driver (since it is the driver). How the driver gets informed depends on the driver and you may need to do something to inform it. See Tell the Driver the Configuration

5.2 Disable PnP ?

Many devices are PnP only with no option for disabling PnP. But for some, you may be able to disable PnP by jumpers or by running a Windows program that comes with the device (jumperless configuration). This will avoid the often complicated task of configuring PnP. Don't forget to tell the BIOS that these resources are reserved. There are also some reasons why you might not want to disable PnP:

1. If you have MS Windows on the same machine, then you may want to allow PnP to configure devices differently under Windows from what it does under Linux.
2. The range of selection for IRQ numbers (or port addresses) etc. may be quite limited unless you use PnP.
3. You might have a Linux device driver that uses PnP methods to search for the device it controls.

4. If you need to change the configuration in the future, it may be easier to do this if it's PnP (no setting of jumpers or running a Dos/Windows program).
5. You may have (or will have) other PnP devices that need configuring so that you'll need to provide for (or learn about) PnP anyway.

Once configured as non-PnP devices, they can't be configured by PnP software or the BIOS (until you move jumpers and/or use the Dos/Windows configuration software again).

5.3 BIOS Configures PnP

Intro to Using the BIOS to Configure PnP

If you have a PnP BIOS, it can configure the hardware. This means that your BIOS reads the resource requirements of all devices and configures them (allocates resources to them). It is a substitute for a PnP OS except that the BIOS doesn't match up the devices with their drivers nor tell the drivers how it has done the configuring. It should give preference to using the configuration it has stored in its non-volatile memory (ESCD).

Your BIOS must support such configuring but there have been cases where it doesn't do it correctly or completely. An advantage of using the BIOS is that it's simple since in most cases there is nothing to set up (except to tell the BIOS's CMOS menu it's not a PnP OS). While some device drivers may be able to automatically detect what the BIOS has done, in some cases you'll need to determine it (not always easy). See What Is My Current Configuration? Another advantage is that the BIOS does its work before Linux starts so that all the resources are ready to be used (and found) by the device drivers that start up later.

According to MS it's only optional (not required) that a PnP BIOS be able to PnP-configure the devices (without help from MS Windows). But it seems that most of the ones made after 1996 ?? or so can do it. We should send them thank-you notes if they do it right. They configure both the PCI and ISA buses, but it has been claimed that some older BIOSs can only do the PCI. To try to find out more about your BIOS, look on the Web. Please don't ask me as I don't have data on this. The details of the BIOS that you would like to know about may be hard to find (or not available). Some BIOSs may have minimal PnP capabilities and try to turn over the difficult parts of the configuration task to Window utilities. If this happens you'll either have to find another method (such as isapnptools) or try to set up the ESCD database if the BIOS has one. See the next section.

The BIOS's ESCD Database

The BIOS's maintains a non-volatile database containing a PnP-configuration that it will try to use. It's called the ESCD (Extended System Configuration Data). Again, the provision of ESCD is optional but most PnP-BIOSs have it. The ESCD not only stores the resource-configuration of PnP devices but also stores configuration information of non-PnP devices (and marks them as such) so as to avoid conflicts. The ESCD data is usually saved on a chip and remains intact when the power is off, but sometimes it's put on a hard-drive??

The ESCD is intended to hold the last used configuration, but if you use a program such as Linux's isapnp or pci utilities (which doesn't update the ESCD) then the ESCD will not know about what isapnp

TOC

has set. A good PnP OS might update the ESCD so you can use it later on for a non-PnP OS (like standard Linux). Windows may do this in some cases. See Using Windows to set ESCD.

To use what's set in ESCD be sure you've set "Not a PnP OS" or the like in the BIOS. Then each time the BIOS starts up (before the Linux OS is loaded) it should configure things this way. If the BIOS detects a new PnP card which is not in the ESCD, then it must then allocate resources to the card and update the ESCD. It may even have to change the resources assigned to existing PnP cards and modify ESCD accordingly.

If devices saved their last configuration in their hardware, hardware configuring wouldn't be needed each time you start your PC. But it doesn't work this way. So all the ESCD data needs to be kept correct if you use the BIOS for PnP. There are some BIOSs that don't have an ESCD but do have some non-volatile memory to store info on which resources have been reserved for use by non-PnP cards. Many BIOSs have both.

Using Windows to set the ESCD

If the BIOS doesn't set up the ESCD the way you want it (or the way it should be) then it would be nice to have a Linux utility to set the ESCD. As of early 1999 there isn't any. Thus one may resort to attempting to use Windows (if you have it on the same PC) to do this.

There are two ways to use Windows to try to set/modify the ESCD. One way is to use the ICU utility designed for DOS or Windows 3.x. It should also work OK for Windows 9x/2k ?? Another way is to set up devices manually ("forced") under Windows 9x/2k so that Windows will put this info into the ESCD when Windows is shut down normally. If devices are configured automatically by Windows (without the user telling it to "change setting") the setting will probably not make it into the ESCD ?? Of course Windows may well decide on its own to configure the same as what is set in the ESCD so they could wind up being the same by coincidence.

Windows 9x are PnP operating systems and automatically PnP-configure devices. They maintain their own PnP-database deep down in the Registry (stored in binary Windows files). There is also a lot of other configuration stuff in the Registry besides PnP-resources. There is both a current PnP resource configuration in memory and another (perhaps about the same) stored on the hard disk. To look at (the one in memory?) in Windows98 you use the Device Manager.

In Windows98 There are 2 ways to get to the Device Manager: 1. Control Panel --> System Properties --> Device Manager. 2. My Computer --> Properties --> Device Manager. Then in Device Manager you select a device (sometimes a multi-step process if there are a few devices of the same class). Then click on Properties and then on Resources. To attempt to change the resource configuration manually, uncheck "Use automatic settings" and then click on Change Settings. It may not let you change it. Also, since Windows may assign IRQs differently than Linux, what is a conflict under Windows may not be a conflict under Linux and conversely. When you "force" a change of resources in Windows, it should put your change into the ESCD (provided you exit Windows normally).

TOC

Adding a New Device (under Linux or Windows)

If you add a new PnP device and have the BIOS set to "not a PnP OS", then the BIOS should automatically configure it and store the configuration in ESCD. If it's a non-PnP legacy device (or one made that way by jumpers, etc.) then there are a few options to handle it.

You may be able to tell the BIOS directly (via the CMOS setup menus) that certain resources it uses (such as IRQs) are reserved and are not to be allocated by PnP. This does not put this info into the ESCD. But there may be a BIOS menu selection as to whether or not to have these CMOS choices override what may be in the ESCD in case of conflict. Another method is to run ICU under DOS/Windows. Still another is to install it manually under Windows 9x/2k. Since this configuration is "forced" Windows should update the ESCD when you shut down the PC.

5.4 Isapnp (part of isapnptools)

Unfortunately, much of the documentation for isapnp is still difficult to understand unless you know the basics of PnP. This HOWTO should help you understand it as well the FAQ that comes with it. `isapnp` is only for PnP devices on the ISA bus (non-PCI). Running the Linux program "isapnp" at boot-time will configure such devices to the resource values you set in /etc/isapnp.conf. Its possible to create this configuration file automatically but you then must edit it manually to chose between various options. With isapnp, a device driver which is part of the kernel may run too early before isapnp has set the address, etc. in the hardware. This results in the device driver not being able to find the device. The driver trys the right address but the address hasn't been set yet in the hardware.

If your Linux distribution automatically installed isapnptools, isapnp may already be running at startup. In this case, all you need to do is to edit /etc/isapnp.conf per "`man isapnp.conf`". Note that this is like manually configuring PnP since you make the decisions as to how to configure as you edit the configuration file. You can use the program "pnpdump" to help create the configuration file. If you use "isapnp" like this and have a PnP BIOS, you should probably tell the BIOS (when you set it up) that you don't have a PnP OS since you still want the BIOS to configure the PCI devices. While the BIOS may also configure the ISA devices, isapnp will redo it.

5.5 PCI Utilities

The new package PCI Utilities (= pciutils, incorrectly called "pcitools"), should let you manually PnP-configure the PCI bus. "lspci" lists resources while "setpci" sets resource allocations in the hardware devices.

5.6 Patch the Kernel to Make Linux PnP

David Howells has created a patch to do this called "Linux Kernel Configuration/Resource Manager" (sometimes called Hardware Configuration Manager). The patch may not be against the most recent kernel. The resulting kernel is is claimed to be stable but bugs have been reported. It includes documentation such as serial.txt to show how to deal with the serial port. It provides "files" in the /proc

tree so that you can see what is going on and can echo commands into one of these files for custom configuration. One problem is that most device drivers don't know about it so that you still may have use the traditional configuration files, etc. for configuration. The webpage for it is http://www.astarte.free-online.co.uk

5.7 Windows Configures

If you have Windows9x (or 2k) on the same PC, then just start Windows and let it configure PnP. Then start Linux from Windows (or DOS). It been reported that Windows erased the IRQs from PCI devices registers. Then Linux complained that it found a zero IRQ. Thus you may not be able to use this method.

5.8 Device Driver Configures

A few device drivers will use PnP methods to set the resources in the hardware but only for the device that they control. Since the driver has done the configuring, it obviously knows the configuration and there is no need for you to tell it this info.

The problem with this is twofold. It's difficult to incorporate all of this into the driver, and the driver may grab resources that are needed by other devices. It does make it easy for the user but a PnP Linux kernel might be better. See Linux Needs to Cope Better with PnP

5.9 PnP Software/Documents

- Isapnptools homepage
- Patch to make the Linux kernel PnP
- PnP driver project
- PnP Specs. from Microsoft
- Book: PCI System Architecture, 3rd ed. by Tom Shanley +, MindShare 1995. Covers PnP-like features on the PCI bus.
- Book: Plug and Play System Architecture, by Tom Shanley, Mind Share 1995. Details of PnP on the ISA bus. Only a terse overview of PnP on the PCI bus.

---

6. Tell the Driver the Configuration

6.1 Introduction

Just how this is done depends upon the driver. Some drivers have more than one way to find out how their physical device is configured. At one extreme is the case where you must hard-code the resources into the kernel and recompile. At the other extreme, the driver does everything automatically and you have nothing to do. It may even set the resources in the hardware using PnP methods.

In the middle are cases where you run a program to give the resource info to the driver or put the info in a file. In some cases the driver may probe for the device at addresses where it suspects the device resides. It may then try to test various IRQs to see which one works. It may or may not automatically do this. In other cases the driver may use PnP methods to find the device and how the resources have been set, but will not actually set them. It may also look in some of the files in the /proc directory.

One may need to give the resources as a parameter to the kernel to to a loadable module. See /usr/lib/modules_help/descr.gz for a list of possible parameters. The module to load is listed in /etc/modules along with its parameters. In some other case the resources may be given as parameters to the kernel. These are put into the lilo.conf file as append="...". Then the lilo program must be run to save this in the kernel boot code.

While there is great non-uniformity about how drivers find out about resources, the end goal is the same. There are so many different hardware devices and drivers for them that you may need to look at documentation for your driver to find out how it finds out about resources and what you need to do to insure that it gets the info it needs. Some brief info on a few drivers is presented in the following section.

6.2 Serial Port Driver: setserial

For the standard serial port driver (not for multiport cards) you use setserial to configure the driver. It is often run from a start-up file. In newer versions there is a /etc/serial.conf file that you "edit" by simply using the setserial command in the normal way and what you set using `setserial` is saved in the `serial.conf` configuration file. The `serial.conf` file should be consulted when the `setserial` command runs from a start-up file. Your distribution may or may not set this up for you.

There are two different ways to use `setserial` depending on the options you give it. One way is used to manually tell the driver the configuration. The other way is to probe at a given address and report if a serial port exists there. It can also probe this address and try to detect what IRQ is used for this port. The driver runs something like `setserial` at start-up but it doesn't probe for IRQs, it just assigns the "standard" IRQ which may be wrong. It does probe for the existence of a port. See Serial-HOWTO for more details.

6.3 Sound Card Drivers

OSS-Lite

You must give the IO, IRQ, and DMA as parameters to a module or compile them into the kernel. But some PCI cards will get automatically detected (likely by looking at /proc/pci or the like). RedHat supplies a program "sndconfig" which detects ISA PnP cards and automatically sets up the modules for loading with the detected resources.

OSS (Open Sound System) and ALSA

These will detect the card by PnP methods and then select the appropriate driver and load it. It will also set the resources on an ISA-PnP card. You may need to manually intervene to avoid conflicts. For the ALSA driver, support for ISA-PnP is optional and you may use isapnp tools if you want to.

---

7. What Is My Current Configuration?

Here "configuration" means the assignment of PnP resources (addresses, IRQs, and DMAs). There are two parts to this question for each device. Each part should have the same answer.

1. What is the configuration of the device driver software? I.e.: What does the driver think the hardware configuration is?
2. What configuration (if any) is set in the device hardware?

Of course the configuration of the device hardware and its driver should be the same (and it normally is). But if things are not working right, there may be a difference. This means the the driver has incorrect information about the actually configuration of the hardware. This spells trouble. If the software you use doesn't adequately tell you what's wrong (or automatically configure it correctly) then you need to investigate how your hardware devices and their drivers are configured. While Linux device drivers should "tell all" in some cases it's not easy to determine what has been set in the hardware.

Another problem is that when you view configuration messages on the screen, it's sometimes not clear whether the reported configuration is that of the device driver, the device hardware, or both. If the device driver is assigned a configuration and then checks the hardware out to see if it's configured the same, then the configuration reported by the driver should be that of both the hardware and the driver.

But some drivers don't do this may accept a configuration that doesn't check out. For example, "setserial" will accept a configuration that doesn't check out (even if you've told it to probe for resources). Thus "setserial" may only be telling you the configuration of the driver and not the hardware.

Some info on configuration may be obtained from the messages from the BIOSs and Linux that appear on the screen when you first start the computer. After all the messages have flashed by, type shift-PageUp to scroll back to them. Typing "dmesg" at any time to the shell prompt will show only the Linux kernel messages and miss some of the most important ones (including ones from the BIOS). The messages from Linux may sometimes only show what the device driver thinks the configuration is, perhaps as told it via an incorrect configuration file. The BIOS messages will show the actual hardware configuration at that time, but a PnP OS, isapnp, or pci utilities, may have changed it since then.

7.1 How Are My Device Drivers Configured?

There may be a programs you can run from the command line (such as "setserial" for serial ports) to determine this. The /proc directory tree is useful. /proc/ioports shows the I/O addresses that the drivers use (or try if it's wrong). They might not be set this way in hardware.

/proc/interrupts shows only interrupts currently in use and many that have been allocated to drivers don't show at all since they're not currently being used. For example, even though your floppy drive has a floppy disk in it and is ready to use, the interrupt for it will not show unless its in use. Again, just because an interrupt shows up here doesn't mean that it exists in the hardware. A clue that it doesn't exist in

hardware will be if it shows that 0 interrupts have been issued by this interrupt. Even if it shows some interrupts have been issued, it may mean that this interrupt doesn't exist on that device does exist on some other device which is not in use, but which somehow has issued an interrupt or two. As of kernel 2.2 the /proc tree has changed.

7.2 How Are My Hardware Devices Configured?

It's easy to find out what resources have been assigned to devices on the PCI bus: Either use the "lspci" command or for kernel <: 2.2: look at `/proc/pci`; for kernel 2.2+: `/proc/bus/pci/devices`. For the ISA bus you may try running `pnpdump --dumpregs` but it's not a sure thing. The results may be difficult to decipher. Don't confuse the read-port address which `pnpdump` "trys" (and finds something there) with the I/O address of the found device. They are not the same.

Messages from the BIOS at boot-time tell you how the hardware configuration was then. If you rely on the BIOS for configuring, then it should still be the same. Messages from Linux may be from drivers that have checked to see that the hardware is there (and possibly checked the IRQ and DMA). Of course, if the device works fine, then it's likely configured the same as the driver.

---

8. Appendix

8.1 Addresses

There are three types of addresses: main memory addresses, I/O addresses and configuration addresses. On the PCI bus, configuration addresses constitute a separate address space just like I/O addresses do. Except for the complicated case of ISA configuration addresses, whether or not an address on the bus is a memory address, I/O address, or configuration address depends only on the voltage on other wires (traces) of the bus.

ISA Bus Configuration Address (Read-Port etc.)

For the ISA bus, there is technically no configuration address space, but there is a special way for the CPU to access PnP configuration registers on the PnP cards. For this purpose 3 @ I/O addresses are allocated. This is not 3 addresses for each card but 3 addresses shared by all cards.

These 3 addresses are named read-port, write-port, and address-port. Each port is just one byte in size. Each PnP card has many configuration registers so that just 3 addresses are not even sufficient for these registers on a single card. To communicate with a certain card, a specially-assigned card number (handle) is sent to all cards at the write-port address. After that only that the only card still listening is the card with this handle. Then the address of the configuration register (of that card) is sent to the address-port (of all cards --but only one is listening). Next communication takes place with one configuration register on that card by either doing a read on the read-port or a write on the write-port.

The write-port is always at A79 and the address-port is always at 279 (hex). But the read-port is not fixed but is set by the configuration software at some address that will supposedly not conflict with any other ISA card. If there is a conflict, it will change the address. All PnP cards get "programmed" with this address. Thus if you use say isapnp to set or check configuration data it must determine this read-port address.

Address ranges

The term "address" is sometimes used in this document to mean a contiguous range of addresses. Since addresses are given in bytes, a single address only contains one byte but I/O (and main memory) addresses need more than this. So a range of say 8 bytes is often used for I/O address while the range for main memory addresses allocated to a device is much larger. For a serial port (an I/O device) it's sufficient to give the starting I/O address of the device (such as 3F8) since it's well known that the range of addresses for serial port is only 8 bytes. The starting address is known as the "base address".

TOC

Address space

For ISA, to access both I/O and (main) memory address "spaces" the same address bus is used (the wires used for the address are shared). How does the device know whether or not an address which appears on the address bus is a memory address or I/O address? Well, there are 4 dedicated wires on the bus that convey this information and more. If a certain one of these 4 wires is asserted, it says that the CPU wants to read from an I/O address, and the main memory ignores the address on the bus. The other 3 wires serve similar purposes. In summary: Read and write wires exist for both main memory and I/O addresses (4 wires in all).

For the PCI bus it's the same basic idea also using 4 wires but it's done a little differently. Instead of only one or the four wires being asserted, a binary number is put on the wires (16 different possibilities). Thus more info may be conveyed. Four of these 16 numbers serve the I/O and memory spaces as in the above paragraph. In addition there is also configuration address space which uses up two more numbers. Ten extra numbers are left over for other purposes.

Range Check (ISA Testing for IO Address Conflicts)

On the ISA bus, there's a method built into each PnP card for checking that there are no other cards that use the same address. If two or more cards use the same IO address, neither card is likely to work right (if at all). Good PnP software should assign resources so as to avoid this conflict, but even in this case a legacy card might be lurking somewhere with the same address.

The test works by a card putting a test number it's own IO registers. Then the PnP software reads it and verifies that it reads the same test number. If not, something is wrong (such as another card with the same address. It repeats the same test with another test number. Since it actually checks the range of IO addresses assigned to the card, it's called a "range check". It could be better called an address-conflict test. If there is an address conflict you get an error message and need to resolve it yourself.

Communicating Directly via Memory

Traditionally, most I/O devices used only I/O memory to communicate with the CPU. For example, the serial port does this. The device driver, running on the CPU would read and write data to/from the I/O address space and main memory. A faster way would be for the device itself to put the data directly into main memory. One way to do this is by using DMA Channels or bus mastering. Another way is to allocate some space in main memory to the device. This way the device reads and writes directly to main memory without having to bother with DMA or bus mastering. Such a device may also use IO addresses.

8.2 Interrupts --Details

Interrupts convey a lot of information but only indirectly. The interrupt signal (a voltage on a wire) just tells a chip called the interrupt controller that a certain device needs attention. The interrupt controller then signals the CPU. The CPU find the driver for this device and runs a part of it known as an "interrupt service routine" (or "interrupt handler"). This "routine" tries to find out what has happened and then deals with the problem such a transferring bytes from (or to) the device. This program (routine) can easily find

TOC

out what has happened since the device has registers at addresses known to the the driver software (provided the IRQ number and the I/O address of the device has been set correctly). These registers contain status information about the device . The software reads the contents of these registers and by inspecting the contents, finds out what happened, and takes appropriate action..

Thus each device driver needs to know what interrupt number (IRQ) to listen to. On the PCI bus (and for the serial ports on the ISA bus starting with Kernel 2.2) it's possible for two (or more) devices to share the same IRQ number. When such an interrupt is issued, the CPU runs all interrupt service routines for all devices using that interrupt. The first thing the first service routine does is to check to see if an interrupt actually happened for its device. If there was no interrupt (false alarm) it likely will exit and the next service routine starts, etc.

8.3 PCI Interrupts

PCI interrupts are different but since they are normally mapped to IRQ's they behave in about the same way. A major difference is that PCI interrupts may be shared. For example IRQ5 may be shared between two PCI devices. This sharing ability is automatic: you don't need special hardware or software. There have been some reports of situations where such sharing didn't work, but it's likely due to a defect in the device driver software. All device drivers for PCI devices are supposed to provide for interrupt sharing. Note that you can't share the same interrupt between the PCI and ISA bus. However, illegal sharing will work provided the devices which are in conflict are not in use at the same time. "In use" here means that a program is running which "opened" the device in it's C programming code.

You may need to know some of the details of the PCI interrupt system in order to set up the BIOS's CMOS or to set jumpers on old PCI cards. Each PCI card has 4 possible interrupts: INTA#, INTB#, INTC#, INTD#. Thus for a 7-slot system there could be 7 x 4 = 28 different interrupt lines. But the specs permit a fewer number of interrupt lines. This is not too restrictive since interrupts may be shared. Many PCI buses seem to be made with only 4 interrupt lines. Call these lines (wires or traces) W, X, Y, Z. Suppose we designate the B interrupt from slot 3 as interrupt 3B. Then wire W could be used to share interrupts 1A, 2B, 3C, 4D, 5A, 6B, 7C. This is done by physically connecting wire W to wires 1A, 2B, etc. Likewise wire X could be connected to wires 1B, 2C, 3D, 4A, 5B, 6C, 7D. Then on startup, the BIOS maps the X, W, Y, Z to IRQ's. After that it writes the IRQ that each device is mapped to into a hardware register in each device. Then and anything interrogating the device can find out what IRQ it uses.

The above mentioned wires X, W, Y, Z are labeled per PCI specs as INTA#, INTB#, INTC# and INTD#. This official PCI notation is confusing since now INTA# has 2 possible meanings depending on whether we are talking about a slot or the PCI bus. For example, if 3C is mapped to X then we say that INTC# of slot 3 is cabled to INTA# (X) of the PCI bus. Confusing notation.

There's another requirement also. A PCI slot must use the lower interrupt letters first. Thus if a slot only uses one interrupt, it must be INTA#. If it uses 2 interrupts they must be INTA# and INTB#, etc. A card in a slot may have up to 8 devices on it but there are only 4 PCI interrupts for it. This is OK since interrupts may be shared so that each of the 8 devices (if they exist) can have an interrupt. The PCI interrupt letter of a device is often fixed and hardwired into the device.

The BIOS assigns IRQs (interrupts) so as to avoid conflicts with the IRQs it knows about on the ISA bus.

TOC

Sometimes in the CMOS BIOS menu one may assign IRQs to PCI cards (but it's not simple as expla above). There's a situation where Windows zeroed out all the IRQ numbers in the PCI cards after the IRQ mappings had been set. Then someone running Windows booted Linux from Windows with the result that Linux only found only incorrect IRQs of zero.

You might reason that since the PCI is using IRQ's (ISA bus) it might be slow, etc. Not really. The ISA Interrupt Controller Chip(s) has a direct interrupt wire going to the CPU so it can get immediate attention. While signals on the ISA address and data buses need to go thru the PCI bus to get to the CPU, the IRQ interrupt signals go there almost directly.

8.4 Isolation

This is only for the ISA bus. Isolation is a complex method of assigning a temporary handle (id number or Card Select Number = CSN) to each PnP device on the ISA bus. Since there are more efficient (but more complex) ways to do this, some might claim that it's a simple method. Only one write address is used for PnP writes to all PnP devices so that writing to this address goes to all PnP device that are listening. This write address is used to send (assign) a unique handle to each PnP device. To assign this handle requires that only one device be listening when the handle is sent (written) to this common address. All PnP devices have a unique serial number which they use for the process of isolation. Doing isolation is something like a game. It's done using the equivalent of just one common bus wire connecting all PnP devices and the isolation program.

For the first round of the "game" all PnP devices listen on this wire and send out simultaneously a sequence of bits to the wire. The allowed bits are either a 1 (positive voltage) or an "open 0" of no voltage (open circuit or tri-state). Each PnP device just starts to sequentially send out its serial number, bit-by-bit, starting with the high-order bit, on this wire. If any device sends a 1, a 1 will be heard on the wire by all other devices. If all devices send an "open 0" nothing will be heard on the wire. The object is to eliminate (by the end of this first round) all but highest serial number device. "Eliminate" means to cease to listen anymore to the write address that all devices still in the game are still listening to. This is also called "dropping out". (Note that all serial numbers are of the same length.)

First consider only the high order bit of the serial number which is put on the wire first by all devices which have no handle yet. If any PnP device sends out a 0 (open 0) but hears a 1, this means that some other PnP device has a higher serial number, so it temporarily drops out of this round and doesn't listen anymore until the round is finished (when a handle is assigned to the winner: the highest serial number). Now the devices still in the game all have the same leading digit (a 1) so we may strip off this digit and consider only the resulting "stripped serial number" for future participation in this round. Then go to the start of this paragraph and repeat until the entire serial number has been examined for each device (see below for the all-0 case).

Thus it's clear that the highest serial number will not be eliminated from the game. But what happens if the leading digits (of the possibly stripped serial numbers) are all 0? In this case an "open 0" is sent on the line and all participants stay in the game. If they all have a leading 0 then this is a tie and the 0's are stripped off just like the 1's were in the above paragraph. The game then continues as the next digit (of the serial number) is sent out.

At the end of the round (after the low-order bit of the serial number has been sent out by whatever

participants remain) only one PnP device with the highest serial number remains. It then gets assigne handle and drops out of the game permanently. Then all the dropouts from the last round (that don't l a handle yet) reenter the game and a new round begins with one less participant. Eventually, all PnP devices are assigned handles. It's easy to prove that this algorithm works.

Once all handles are assigned, they are used to address each PnP device and send it a configuration as well as to read configuration info from the PnP device. Note that these handles are only used for PnP configuration and are not used for normal communication with the PnP device. When the computer starts up, all of the handles are lost so that a PnP BIOS usually does the isolation process again each time you start your PC.

END OF Plug-and-Play-HOWTO

Polish HOWTO

Sergiusz Pawlowicz, ser@arch.pwr.wroc.pl

v2.1, 8 February 1999

Jak to zrobic w Linuxie? Pomoc zlozona z niektorych Linux-HOWTO oraz empirii, dostosowana do polskich warunkow i potrzeb. Wersja ta jest pozbawiona polskich liter ze wzgledu na standard LDP. Oryginal mozna znalezc w http://serek.arch.pwr.wroc.pl/linux/Polish-HOWTO/.

1. Wstep.

- 1.1 Informacje ogolne o Polish-HOWTO.
- 1.2 Jak pomoc w rozwinieciu podrecznika?
- 1.3 Jak czytac aktualna wersje Polish-HOWTO?

2. Zrodla Linuxa w Polsce.

- 2.1 Pliki binarne i zrodlowe Linuxa.
- 2.2 Dokumentacje po angielsku i polsku.

3. Jezyk polski i Linux.

- 3.1 Programy pocztowe, ktorym polskie litery nie przeszkadzaja.
- 3.2 Programy do czytania newsow, ktorym polskie litery nie przeszkadzaja.
- 3.3 Komunikaty programow po polsku.

4. Gdzie zapytac, gdy czegos nie wiemy.

- 4.1 Grupa pocztowa i UseNet news.

5. Zycie Linuxa w Polsce

- 5.1 PLUG - Polish Linux Users Group.
- 5.2 Licznik polskich uzytkownikow Linuxa.

1. Wstep.

1.1 Informacje ogolne o Polish-HOWTO.

Krotki podrecznik Linuxa po polsku? Mozliwe, ze dzieki pomocy wielu ludzi sie to uda. Chwilowo jednak mam zamiar opisac typowo polskie problemy: gdzie znalezc w Polsce zrodla i dystrybucje Linuxa, jak uzyskac polskie litery pod konsola Linuxa oraz w X11, jak prawidlowo przesylac i odczytywac poczte z polskimi literami, gdzie w Polsce uzyskac pomoc (grupy dyskusyjne usenet, pocztowe), gdzie znalezc dobra dokumentacje angielska... Uwaga: Autor nie bierze odpowiedzialnosci za ewentualne uszkodzenia mienia poprzez zastosowanie sie do rad zawartych w podreczniku :)).

1.2 Jak pomoc w rozwinieciu podrecznika?

Nie jestem w stanie opisac po polsku wszystkiego, co dotyczy Linuxa. Wszelkie materialy, ktore wydaja sie wam ciekawe, prosze przesylac na adres `ser@arch.pwr.wroc.pl`. Najlepiej w formacie SGML, ale czyste ascii lub tekst w kodzie ISO-8859-2 rowniez z checia przyjme.

Oto lista (kolejnosc chronologiczna) osob, ktore pomogly w rozny sposob przy powstawaniu Polish-HOWTO:

- Linus Torvalds :), tworca Linuxa, `linus@linux.helsinki.fi`
- Wojciech Myszka, prekursor internetu wroclawskiego, `myszka@immt.pwr.wroc.pl`
- Adam Tlal/ka, Politechnika Gdanska, `atlka@pg.gda.pl`
- Jaroslaw Strzalkowski, `js@arrow.uci.agh.edu.pl`, Piotr Piatkowski, `kompas@uci.agh.edu.pl`, za Polska Strone Ogonkowa.
- Rafal Maszkowski, PDI, `rzm@torun.pdi.net`.

A oto lista najbardziej potrzebnych dokumentacji do opracowania po polsku:

- Jak zainstalowac Linuxa? Odpowiednik Install-HOWTO, obejmujacy RedHat, Debian, Slackware, PLD.
- Jak zadbac o bezpieczenstwo systemu?

1.3 Jak czytac aktualna wersje Polish-HOWTO?

Wersja HTML (do przegladania na www) tego tekstu znajduje sie pod: http://serek.arch.pwr.wroc.pl/linux/Polish-HOWTO/. Tutaj zawsze jest najnowsza postac podrecznika, jest to wersja eksportowa :-), pisana do SunSite w USA na potrzeby Linux Documentation Project.

Poza tym, wersja ascii dokumentu jest umieszczana czasmi na newsach, w grupach news:pl.answers, news:pl.comp.os.linux.

2. Zrodla Linuxa w Polsce.

2.1 Pliki binarne i zrodlowe Linuxa.

Dystubucje Linuxa czyli cos, co latwo zainstalowac nawet laikowi.

Podstawowe dytrybucje Linuxa mozna znalezc w wiekszosci polskich sieci metropolitalnych, prosze wybierac zawsze miejsce najblizsze:

- **Wroclaw:** Politechnika Wroclawska
  - RedHat: ftp://ftp.pwr.wroc.pl/pub/linux/redhat/
  - SlackWare: ftp://ftp.pwr.wroc.pl/pub/linux/INSTALL/
  - PLD: ftp://ftp.arch.pwr.wroc.pl/pub/linux/PLD/
- **Szczecin:** Politechnika Szczecinska
  - PLD: ftp://ftp.ps.pl/pub/Linux/PLD/
- **Poznan:** Poznan Metropolitan Area Network
  - Slackware: ftp://ftp.man.poznan.pl/pub/linux/Slackware/
  - RedHat: ftp://ftp.man.poznan.pl/pub/linux/redhat/
- **Gdansk:** Politechnika Gdanska
  - RedHat: ftp://ftp.pg.gda.pl/pub/OS/Linux/redhat/
  - SlackWare: ftp://ftp.pg.gda.pl/pub/OS/Linux/slackware/
  - Debian: ftp://ftp.pg.gda.pl/pub/OS/Linux/debian/
- **Warszawa:** Interdyscyplinarne Centrum Modelowania
  - RedHat: ftp://sunsite.icm.edu.pl/pub/Linux/redhat/
  - SlackWare: ftp://sunsite.icm.edu.pl/pub/Linux/slackware/
  - Debian: ftp://sunsite.icm.edu.pl/pub/Linux/debian/
- **Krakow:** Cyfronet
  - RedHat: ftp://ftp.cyf-kr.edu.pl/pub/linux/redhat/

Piszac: podstawowe, mam na mysli:

- **RedHat Linux.**
- **Slackware.**
- **Debian.**
- **PLD - Polish(ed) Linux Distribution.**

TOC

Nowe wersje jadra systemu Linux.

Najnowsza wersje jadra, aktualizowane co godzine, znajdziecie w: ftp://serek.arch.pwr.wroc.pl/pub/CVSROOT/linux/.

Natomiast jadra oficjalne, spakowane, mirror `ftp.cs.helsinki.fi`, na: ftp://sunsite.icm.edu.pl/pub/Linux/kernel/.

X-Free

Wersje X-11 dla Linuxa mozna sciagnac z SunSite, ftp://sunsite.icm.edu.pl/pub/Linux/XFree86/

Archiwa programow na Linuxa.

Prosze wybierac zawsze miejsce najblizsze:

- **Wroclaw:** Politechnika Wroclawska
  - ftp://ftp.pwr.wroc.pl/pub/linux/
- **Szczecin:** Academic Centre of Informatics
  - ftp://ftp.man.szczecin.pl/pub/Linux/
- **Gdansk:** Politechnika Gdanska
  - ftp://ftp.pg.gda.pl/pub/OS/Linux/

Pewne miejsce do szukania zbiorow linuxowych.

Jezeli tych zrodel nie jestescie w stanie znalezc u siebie, znajdziecie je na pewno na **SunSite Polska** (Warszawa, Interdyscyplinarne Centrum Modelowania):

- ftp://sunsite.icm.edu.pl/pub/Linux/

Jest to **najwieksze polskie** archiwum anomimowego ftp.

2.2 Dokumentacje po angielsku i polsku.

Dokumentacje Linuxa sa prawie w calosci po angielsku. Dobry zbior dokumentacji zgromadzil Greg Hankins, `gregh@sunsite.unc.edu`: **Linux Documentation Project**. Pomoc, ktora czytasz, jest czescia tego projektu. Oficjalne (i aktualne) kopie dokumentacji mozna znalezc w:

- **SunSite** Polska, ICM Warszawa, http://sunsite.icm.edu.pl/pub/Linux/Documentation/

Polska dokumentacja stricte do Linuxa w sieci to przede wszystkim

- **Jak To Zrobic**, http://www.jtz.org.pl/,

opiekunem projektu jest Bartek Maruszewski, mailto:bart@jtz.org.pl.

Dokumentacja po polsku, dotyczaca **nie tylko Linuxa**, ale sieci w ogole, jest takze w:

- http://www.immt.pwr.wroc.pl/export_hp/tool/tool.html.

  Autor: Wojciech Myszka, `myszka@immt.pwr.wroc.pl`.

3. Jezyk polski i Linux.

Uzyskanie polskich liter pod Linuxem jest proste. Kiedys trudno bylo znalezc standard, ktory odpowiadalby wszystkim, teraz pewne jest, ze przyjal sie oficjalny standard ISO, ISO-8859-2. Dokladniejszych informacji o problemach z polskimi literami mozna szukac na **Polskiej Stronie Ogonkowej**: http://www.agh.edu.pl/ogonki/.

3.1 Programy pocztowe, ktorym polskie litery nie przeszkadzaja.

Calkowicie spolszczonym programem pocztowym, ktorego uzywam, jest Mutt. Oprocz tego, ze dobrze wyswietla polskie litery, ma mozliwosc podania strony kodowej, w naszej sytuacji jest to iso-8859-2. Jesli list napisany w moim Mutcie`ie dojdzie do uzytkownika uzywajacego innej strony kodowej, wyswietli mu sie informacja, ze moze nie zobaczyc poprawnie czesci liter. Jesli natomiast posluguje sie nasza strona kodowa, list bedzie wyswietlany bez uwag i z polskimi literami.

3.2 Programy do czytania newsow, ktorym polskie litery nie przeszkadzaja.

Do czytania i pisania news po polsku nadaje sie tin-beta-unoff, do sciagniecia z:

- ftp://ftp.akk.uni-karlsruhe.de/pub/tin/
- ftp://ftp.onet.pl/pub/unix/tin/

3.3 Komunikaty programow po polsku.

Od ponad dwoch lat dziala grupa tlumaczaca komunikaty programow GNU. Wiecej informacji o tym projekcie znajdziesz pod adresem http://www.ceti.com.pl/~kravietz/gnu/gnu_tp.html. Archiwum przetlumaczonego oprogramowania:

- ftp://ftp.arch.pwr.wroc.pl/pub/nls/

Opiekunem grupy tlumaczy jest Rafal Maszkowski, mailto:rzm@icm.edu.pl. Zapraszamy do tlumaczenia, na potrzeby tlumaczy dziala serwer CVS.

Aby uzywac programow z polskimi komunikatami nalezy nadac srodowisku powloki zmienna LANG=pl_PL. Jesli dystrybucja z pudelka nie ma przygotowanej wersji miedzynarodowej, trzeba samemu skompilowac wersje miedzynarodowa pakietu, umieszczajac w podkatalogu 'po' plik pl.po dla danego pakietu (chyba ze juz tam jest, ale w pakiecie moze byc starsza wersja). Pakiet mozna zainstalowac dla wybranych jezykow - p. instrukcje instalacji pakietow miedzynarodowych GNU. Jezeli

TOC

w danym pakiecie nie bylo jeszcze pl.po moze byc potrzebne dopisanie w zmiennej LINGUAS w skrypcie configure 'pl', pamietajac potem o odpowiednim ustawieniu zmiennych.

Chetni do wziecia udzialu w pracy nad tlumaczeniami musza wypelnic zobowiazanie przekazujace prawa autorskie do tlumaczen na Free Software Foundation. Oryginalne archiwum 'GNU translation project':

- ftp://alpha.gnu.ai.mit.edu/gnu/po/

Adres grupy tlumaczacej: `pl@li.org` (via majordomo).

---

4. Gdzie zapytac, gdy czegos nie wiemy.

4.1 Grupa pocztowa i UseNet news.

Wlasciwie lista pocztowa i UseNet`owa to jedno i to samo, gdyz listy wyslane na liste pocztowa, trafiaja na newsy i odwrotnie.

Aby zajrzec na UseNet news, nalezy dostac sie do grupy: news:pl.comp.os.linux

Aby zapisac sie na liste pocztowa, prosze wyslac list bez subjectu o tresci:

```
sub linux Imie Nazwisko
```

na adres `listserv@sprocket.ict.pwr.wroc.pl`. Lista utrzymywana jest na komputerze linuxowym znajdujacym sie w Instytucie Cybernetyki Technicznej Politechniki Wroclawskiej. Archiwum listy od kwietnia 1995 roku mozna obejrzec na http://www.ict.pwr.wroc.pl/listserv/linux/

5. Zycie Linuxa w Polsce

5.1 PLUG - Polish Linux Users Group.

Polska Grupa Uzytkownikow Linuxa, wiecej informacji mozna znalezc na: http://www.linux.org.pl/,

- oddzial krakowski: http://www.krakow.linux.org.pl/,
- odzial wroclawski: http://www.wroclaw.linux.org.pl/,
- oddzial slaski: http://www.silesia.linux.org.pl/,
- oddzial torunski: http://eris.phys.uni.torun.pl/Linux/,
- oddzial poznanski: http://www.poznan.linux.org.pl/.

Polish(ed) Linux Distribution - polska grupa majaca na celu przygotowanie bezpiecznej dystrybucji Linuxa, wyposazonej w nowinki, czyli IPv6 oraz kerberosa, glibc-2.1, ssh, nowe jadra Linuxa; wiecej informacji na http://www.pld.org.pl/.

5.2 Licznik polskich uzytkownikow Linuxa.

Uruchomiono licznik uzytkownikow Linuxa w Polsce.

Tak wiec, kazdy, kto uzywa na swoim PC (Digitalu, IBMie, SUNie, Amidze, Macintoshu etc.) systemu operacyjnego Linux, jest proszony o wyslanie listu na adres:

`licznik@arch.pwr.wroc.pl`

Wystarczy wiec napisac w swoim systemie nastepujaca komende:

```
mail licznik@arch.pwr.wroc.pl < /dev/null
```

Jezeli komus chcialoby sie wiecej napisac, nie ma sprawy, wszystkie zgloszenia beda starannie analizowane. Powstanie z tego mapa Linuxow w Polsce, mozna wiec w listach do "licznika" przesylac URLe serwerow www posadzonych na zglaszanych przez siebie Linuxach. Listy przyslane na powyzszy adres sa archiwizowane i jak tylko nabierze to sensu, udostepniane pod adresem: http://licznik.arch.pwr.wroc.pl/.

Dziekuje z gory za listy inicjator polish-howto - Sergiusz Pawlowicz, `ser@arch.pwr.wroc.pl`

Linux Portuguese-HOWTO

Configurações do Linux para a Língua Portuguesa

Carlos A. M. dos Santos < casantos@cpmet.ufpel.tche.br>

CPMet/UFPEL -- Pelotas, RS, Brasil

v3.2, 24 May 1999

*Este documento pretende ser um guia de referência de configuração do **Linux** e seus programas, teclados e fontes de caracteres, permitindo sua utilização mais confortável por pessoas que falem a Língua Portuguesa.*

1. Introdução

- 1.1 Finalidades deste HOWTO
- 1.2 Onde encontrar a versão mais atual
- 1.3 Página oficial na WWW
- 1.4 Como enviar colaborações

2. Informações gerais

- 2.1 Conjuntos de caracteres
- 2.2 Modo texto versus Sistema de Janelas X

3. Configuração do console (modo texto)

- 3.1 Mapas de teclado
- 3.2 Mapas de tradução de tela
- 3.3 Comandos do pacote KBD
- 3.4 Carregamento de uma fonte de caracteres

- 3.5 Carregando um mapa de teclado
- 3.6 Verificando erros

4. Biblioteca de funções libc e aplicativos GNU

- 4.1 Configurando o suporte internacional
- 4.2 Problemas com a libc 5

5. Configuração do X

- 5.1 Configuração do xinit
- 5.2 Configuração do XDM
- 5.3 Compose
- 5.4 Locale
- 5.5 Geração de mapas com o XKeyCaps
- 5.6 Contornando os limites do X

6. Configuração dos vários programas

- 6.1 Aplicativos
- 6.2 Rede local e Internet

7. Ficheiros necessários

8. Informações Adicionais

- 8.1 Fontes de informação sobre Linux em Português
- 8.2 Free Translation Project
- 8.3 LIE-BR
- 8.4 Versões de software testadas

9. Observações finais

- 9.1 Futuras adições a este documento
- 9.2 Nota de Direitos de Autor

- 9.3 Garantia (inexistência de) e nota de responsabilidade
- 9.4 Agradecimentos

10. Bibliografia comentada

---

1. Introdução

À semelhança de outras línguas faladas na Europa, a Língua Portuguesa tem características especiais, como a utilização de caracteres acentuados, que tornam o seu suporte difícil, pouco intuitivo ou até mesmo impossível, por parte do software utilizado nos nossos computadores.

Nossa língua assume particularidades em cada um dos países em que é falada e mesmo dentro de um único país como o Brasil, de dimensões continentais e com influências culturais de muitos outros povos. Este documento foi originalmente escrito por um português, mas agora é mantido por um brasileiro. Há uma mistura de sotaques que deve se manter por um bom tempo, pois não há razão para alterar o texto original senão para fazer de adições e correções, até por respeito ao primeiro autor. Termos ``estrangeiros'' desnecessários foram substituídos por correspondentes da Língua Portuguesa, preservando-se apenas os que já fazem parte do jargão da informática.

1.1 Finalidades deste HOWTO

Este HOWTO é um guia de referência de configuração do sistema operativo Linux e seus programas, teclados e fontes de caracteres, permitindo sua utilização mais confortável por pessoas que falem a Língua Portuguesa. Como os programas mais antigos não foram desenhados com suporte a caracteres acentuados ou internacionalização, às vezes é preciso usar algum artifício que permita contornar o problema, mas o resultado nem sempre é totalmente satisfatório e em alguns casos chega ser frustrante. As dificuldades encontradas centram-se essencialmente em torno de quatro pontos:

- a introdução de caracteres acentuados através do teclado, como seja a escrita de jo~ao em vez da sua forma correcta: joão;
- a exibição dos mesmos na tela do computador. O suporte resume-se normalmente à correcta localização das teclas, nada mais;
- o correto tratamento de convenções nacionais de formato de datas, horas e valores monetários;
- a exibição de mensagens, títulos e demais informações dos programas em outra língua que não o Inglês, que é a língua original da maioria desses programas.

Versões anteriores do Portuguese HOWTO concentravam-se em contornar as duas primeiras dificuldades mencionadas, de forma a fazer que, dentro do possível, tanto o sistema operativo como os programas nele utilizados aceitassem os caracteres acentuados e usassem os teclados com suporte para o Português. Embora esse objetivo ainda seja perseguido, os problemas de teclado e caracteres já têm soluções satisfatórias, pelo menos para as aplicações mais usadas. As próximas versões deverão se dedicar mais aos dois últimos problemas.

O texto contém uma já não tão breve discussão sobre o tratamento do teclado e das fontes de caracteres do console pelo Linux, bem como do suporte a várias línguas nacionais. O Sistema de Janelas X também é discutido, fazendo-se uma comparação entre ele e o modo de texto. Por fim, são fornecidas instruções para a configuração do sistema operativo e de diversos aplicativos importantes.

TOC

O documento tem se tornado menos *slackwariano*, incluindo cada vez mais informações relativas a outras distribuições. Slackware e Caldera ainda se baseiam na versão 5 da biblioteca de funções do sistema para Linux (*libc*), enquanto Debian, Red Hat, TurboLinux, S.u.S.E e Stampede já suportam a versão 6 (que na verdade é a glibc versão 2). Esta nova versão da biblioteca está um passo adiante no suporte a internacionalização e localização. A partir da versão 3.2 do Portuguese-HOWTO as particularidades de cada distribuição serão destacadas quando houver necessidade. A tentativa de tornar Debian uma distribuição ``de referência" a partir da versão 3.0 do HOWTO fracassou miseravelmente porque o número de diferenças entre as distribuições é muito grande.

A intenção original de não tentar concentrar toda a informação em um único texto, tratando apenas de alguns temas essenciais e fornecendo referências para outras fontes, também fracassou. O HOWTO em sua versão 3.1 já alcançava 38 páginas impressas em formato A4, o que está longe de poder ser chamado de sucinto. As principais razões para tal gigantismo são a necessidade de discutir diferenças entre as distribuições e a necessidade de fornecer informações mais detalhadas sobre aplicativos.

Excetuando-se aspectos muito específicos do Linux, como a configuração do modo texto e do *kernel*, a maioria das informações contidas neste documento pode ser aplicada a outros sistemas Unix. Exemplos são as configurações do Sistema de Janelas X e de vários aplicativos, que foram aplicadas em Solaris 2.{5,6,7} e Digital UNIX 3.2. Os mapas de teclado para terminais X e estações de trabalho Sun foram criados e são usados em máquinas que rodam Solaris. O documento poderá um dia se tornar o ``Unix Portuguese HOWTO", embora no momento não se tenha intenção ou meios de fazê-lo.

1.2 Onde encontrar a versão mais atual

Os Linux HOWTO fazem parte do Projeto de Documentação Linux (Linux Documentation Project -- LDP). Os documentos do LDP são mantidos em servidores da Universidade da Carolina do Norte (UNC) e do Instituto de Tecnologia de Massachusetts (MIT), nos Estados Unidos da América, e podem ser obtidos via FTP anônimo nos seguintes endereços:

- ftp://metalab.unc.edu/pub/Linux/docs/HOWTO (conhecida anteriormente como `sunsite.unc.edu`).
- ftp://tsx-11.mit.edu/pub/linux/docs/HOWTO

Pode-se também folhear os documentos HOWTO em formato HTML no endereço

- http://metalab.unc.edu/LDP/HOWTO

Muitas localidades mantêm cópias desses documentos. Deve-se dar preferência ao acesso à cópia mais próxima, para economizar o precioso tráfego internacional na Internet e também evitar a sobrecarga da máquina metalab.unc.edu. Uma lista completa dessas localidades pode ser obtida em

- http://metalab.unc.edu/LDP/mirrors.html

Eis uma lista dos endereços recomendados:

Brasil

- http://linusp.usp.br/LDP/
- http://www.conectiva.com.br/LDP/

- http://www.opensite.com.br/linux/
- http://taubate.valley-bbs.com.br/LDP/
- http://linux.unicamp.br/docs/

Portugal

- http://gil.di.uminho.pt/mirrors/LDP/
- http://deneb.cec.pt/LDP/
- http://ftp.dei.uc.pt/LDP/
- http://www.sc.uevora.pt/LDP/
- http://lwp.ualg.pt/docs/LDP/
- http://linux.global-one.pt/LDP/
- http://linux.ispgaya.pt/LDP/
- http://ae.fe.up.pt/LDP/

Os Linux HOWTO estão disponíveis em diversos formatos: texto simples, PostScript, DVI, HTML, RTF e Lyx. O formato original é SGML e os demais são gerados usando o SGMLtools (maiores informações em http://www.sgmltools.org/).

1.3 Página oficial na WWW

A página oficial do Linux Portuguese-HOWTO, assim como os diversos arquivos de configuração do sistema e programas aplicativos mencionados no texto, podem ser encontrados nos seguintes endereços:

- Brasil: http://linusp.usp.br/~casantos/
- Portugal: http://linux.fe.up.pt/howto/

1.4 Como enviar colaborações

Toda a formatação deste HOWTO, incluindo a numeração das seções é feita pelo SGMLtools. Por isso, ao enviar sugestões ou correções, mencione **os títulos das seções** onde as alterações sugeridas devem ocorrer e não os números. Nunca refira-se a um parágrafo como sendo ``o terceiro da seção 3.1''; cite as palavras inicias do trecho a ser modificado.

Envie sugestões e correções via correio eletrônico em texto simples, opcionalmente com acentuação no código ISO-8859-1. Se o texto não for composto em um sistema *NIX, certifique-se de usar a codificação correta. Todas as mensagens recebidas serão lidas, mas nem todas receberão resposta direta, devido à falta de tempo para tanto.

Não envie documentos no formato proprietário de processadores de texto nem em formato HTML (o que infelizmente parece ser o comportamento normal do Netscape Communicator e do Microsoft Outlook).

Mensagens perguntando como configurar seu modem, placa de vídeo ou rede não serão respondidas.

TOC

Para esse tipo de consulta é aconselhável procurar uma lista de discussão sobre Linux em Português acompanhar os grupos de discussão de Linux na USENET `comp.os.linux.*`.

``Flames'' terão o destino costumeiro: `/dev/null`.

---

2. Informações gerais

2.1 Conjuntos de caracteres

Um conjunto de caracteres é definido de acordo com os símbolos constantes no alfabeto utilizado para escrita em uma ou mais línguas. A Organização Internacional de Normas (*International Standards Organization* - ISO) padronizou através da norma ISO-8859 vários desses conjuntos, identificados por ISO-8859-*x* onde o *x* corresponde a um determinado alfabeto. O conjunto ISO-8859 utiliza 8 bits para representar cada caracter, o que permite uma gama de 256 sinais (valores de zero a 255). Em cada um dos conjuntos, os códigos zero a 127 correspondem ao conjunto ASCII (*American Standard Code for Information Interchange*) e os códigos 160 a 255 são usados para caracteres nacionais.

Para a língua portuguesa, recomenda-se o uso do conjunto ISO-8859-1, que compreende o alfabeto latino e letras acentuadas usadas pelas línguas do oeste da Europa e América. Este conjunto de caracteres também é frequentemente chamado de Latin-1 ou ISO Latin-1.

Um padrão mais recente é o Unicode, definido pela norma ISO-10646, que permite definir caracteres cuja representação interna no computador utiliza mais de um byte (ou *octeto* na nomenclatura ISO). Todas as versões mais recentes de sistemas Unix suportam Unicode (ou pelo menos seus fabricantes alegam suportar).

Além dos caracteres alfanuméricos e sinais de acentuação, é possível também gerar sinais *semigráficos* para desenho de linhas e bordas. Esses sinais podem aproveitar códigos não utilizados pelo conjunto oficial, tornando-os não portáveis.

O **Linux** foi desenhado internamente de modo a facilitar a sua fácil configuração e extensão em tempo de execução, não constituindo o tratamento do teclado e fontes de caracteres excepção. Ele possui uma implementação ``nível 1'' do padrão Unicode. Maiores detalhes podem ser encontrados nos manuais do Linux, que podem ser lido com os comandos

```
man unicode
man utf-8
man iso_8859_1
man ascii
```

mas antes disso certifique-se de que o *man* está configurado corretamente, conforme mostrado na seção Man, groff, troff.

2.2 Modo texto versus Sistema de Janelas X

Alguns sistemas operativos, tais como MacOS, Microsoft Windows e NeXT, possuem interfaces gráficas próprias. No Linux, assim como na maioria dos sistemas compatíveis com Unix, é de uso corrente um

ambiente gráfico criado para ser ``multiplataforma": o X Window System, que também foi projetado suportar diversos conjuntos de caracteres, idiomas e formatos de teclado, mas ainda **não** suporta totalmente o padrão Unicode e sim uma extensão do ISO-8859.

> A rigor o X Window System **não** é uma interface gráfica com o usuário, mas a combinação de um protocolo de comunicação com uma interface com programas aplicativos (API) *sobre a qual* se construíram várias interfaces gráficas. Tanto o protocolo quanto o Sistema de Janelas definem um conjunto de *mecanismos* e não *políticas* (elementos característicos da política de uma Interface Gráfica com o Usuário são menus, botões e caixas de diálogo). O uso do termo ``XWindows" é incorreto e deve ser evitado, dando-se preferência ao nome genérico ``X".

Existem muitas semelhanças entre os dois ambientes. Ambos se baseiam em padrões internacionais para definição de conjuntos de caracteres. Tanto no X quanto no console do Linux pode-se definir uma tecla chamada *Compose* cujo pressionamento seguido de duas outras gerará o caracter correspondente. Assim sendo, o pressionamento da seqüência **compose-,-c** gerará um c cedilhado.

O tratamento do console é feito diretamente pelo sistema operativo e aplicações comuns não se envolvem com o processamento dos códigos de varredura do teclado, recebendo um caracter ou uma seqüência deles ao ser pressionada cada tecla, de tal sorte que tendo sido corretamente configurados o teclado e a fonte de caracteres pouco mais se tem a fazer.

O X possui uma arquitetura muito diferente: tanto o teclado quanto a(s) tela(s) -- pode haver mais de uma tela -- são controlados por uma aplicação especial chamada *servidor X*. O pressionamento de uma tecla gera uma mensagem (chamada de *evento*) que é passada pelo servidor X à aplicação *cliente*. Há um programa muito útil chamado *xev* que permite observar cada evento a ele transmitido. Cliente e servidor se comunicam via rede usando um conjunto de regras chamado *protocolo X* e podem rodar em máquinas diferentes. A máquina onde roda o servidor é chamada *estação de trabalho* ou *terminal X* e a máquina onde roda a aplicação (programa cliente) também é chamada de *cliente*.

Foge ao escopo deste texto a discussão mais profunda do tratamento de eventos no X. Para maiores informações, sugere-se a consulta aos documentos mencionados na seção Leituras recomendadas, mas é importante esclarecer que o evento enviado pelo servidor ao cliente não contém o código numérico da tecla, chamado *keycode* na terminologia do X. Ao invés disso, é enviado um símbolo, chamado *keysymbol* ou *keysym*, obtido pela consulta a uma tabela de conversão mantida na memória do servidor. Esta tabela pode ser modificada total ou parcialmente a qualquer momento por meio de requisições definidas no protocolo X.

Existe um programa chamado *xmodmap* capaz de ler um arquivo contendo uma tabela de conversão *keycode-->keysym* e enviá-la, também na forma de mensagens no protocolo X, ao servidor. É responsabilidade do cliente e não do servidor interpretar o *keysym*, o que significa que a aplicação deve saber esperar mais um caracter ao receber uma ``tecla morta" para compor uma letra acentuada.

Desde a revisão 5 do X existe na bibliotaca de funções (*Xlib*) um mecanismo sofisticado de suporte à geração de caracteres em diversos códigos. A função de tratamento de entrada de texto que já existia nas versões anteriores, chamada **XLookupString**, entretanto, não processa as seqüências de acentos e letras de modo transparente às aplicações. Ao invés disso foi incluído um método de composição usando *contextos de entrada* (*input contexts*) por meio das funções **XmbLookupString** e **XwcLookupString**,

cujo uso é responsabilidade da aplicação -- ou do seu programador, melhor dizendo -- mesmo no cas tecla *Compose*. Isto deve-se so fato de o *X Consortium* ter chegado à conclusão de que o sistema de mapeamento de teclas não tratava de forma satisfatória toda a imensa variedade de línguas escritas nas várias partes do mundo. Deste modo, decidiu-se que o ``peso'' relativo à gestão do teclado fosse transferido para as aplicações X, o que cria uma dificuldade quando usamos aquelas mais antigas, que não usam o novo método de tratar a entrada.

Por essas razões é normalmente mais difícil conseguir acrescentar suporte à geração de caracteres acentuados em aplicativos que rodam sob o X, principalmente quando não se possui o código fonte. Aplicativos feitos para rodar apenas em modo texto, tais como **vi** e **minicom** dependerão totalmente dos recursos do emulador de terminal em uso quando rodando em uma janela do X. Se for usada uma versão atual do *xterm* ou *rxvt* o emulador fará o tratamento correto dos acentos.

O francês Thomas Quinot, cansado de esperar uma solução melhor para o problema da acentuação no X, resolveu implementar uma modificação para a Xlib proposta por André D. Balsa, que adiciona suporte à acentuação direta conforme mostrado na seção Contornando os limites do X. Isso permite usar aplicações como *xfig* ou *xedit* sem que seja necessário alterá-las. O truque parece funcionar apenas para caracteres do código ISO-8859-1 mas é suficiente para o Português e outras línguas. Como bem observa Balsa, a longo prazo todos os programas deverão ser modificados para usar os novos recursos do X, mas como a longo prazo estaremos todos mortos a solução de curto prazo de Quinot torna-se bastante atraente...

---

3. Configuração do console (modo texto)

O documento de referência sobre a configuração do console do Linux é o Keyboard and Console HOWTO, de Andries Brouwer, que pode ser encontrado nos repositórios do LDP. Conforme lá descrito, a configuração da fonte de caracteres e mapa de teclado é feita usando o pacote KBD, encontrado em todas as distribuições de Linux.

3.1 Mapas de teclado

Cada tecla do PC possui um código numérico. Ao pressionarmos uma delas o processador controlador do teclado envia ao computador esse código *de varredura*, também conhecido como *scancode*, junto com um sinal de que a tecla foi pressionada ou solta. As seqüências de eventos são então processadas pelo *driver* de teclado e armazenadas em uma fila de caracteres que é lida pelas aplicações por meio da chamada de funções do sistema operativo.

Um mapa de teclado é um arquivo de texto que estabelace as correspondências entre o *scancode* de tecla e o caracter (ou seqüência de caracteres) a gerar quando ela for pressionada, chamado *keycode*. Por exemplo:

```
# atribuição da tecla '-' do teclado numérico à tecla com código 74
keycode  74 = KP_Subtract

# atribuição da tecla '4' do teclado numérico à tecla com código 75
keycode  75 = KP_4

# etc...

keycode  76 = KP_5            # tecla 5
keycode  77 = KP_6            # tecla 6
keycode  78 = KP_Add          # soma
keycode  79 = KP_1            # tecla 1
keycode  80 = KP_2            # tecla 2
```

Além das teclas alfabéticas, numéricas e de símbolos, existem outras chamadas *modificadoras* que permitem gerar códigos que não correspondem a nenhum sinal gráfico: *Shift Control Alt* e *Meta*. Esta última normalmente não é encontrada em teclados de PCs, apenas em estações de trabalho de fabricantes como Sun, SGI, HP e Digital (eles não gostam de ser chamados de ``DEC''). O editor de texto Emacs usa muito a tecla *Meta*.

O arquivo de mapa permite também especificar teclas especiais chamadas ``teclas mortas'' (*deadkeys*). Quando pressionadas elas não resultam no aparecimento de um caracter na tela, limitando-se a alterar o comportamento da tecla pressionada a seguir para que, por exemplo, ao se digitar um **~** seguida de um **a**, seja gerado um `ã'.

TOC

3.2 Mapas de tradução de tela

Um mapa de tradução de tela permite especificar qual o caracter X a ser exibido na tela, quando um programa deseja exibir um caracter Y. Desta forma, poderíamos fazer com que ao escrever o caracter com o código do `c-cedilhado` na tela, fosse na realidade exibido um outro caracter de código diferente mas cuja *imagem* na nossa fonte de caracteres correspondesse à imagem de um `c-cedilhado`.

Este mapeamento é necessário apenas quando queremos usar uma fonte cujos caracteres não possuem códigos diretamente correspondentes aos do conjunto usado no mapa de teclado.

3.3 Comandos do pacote KBD

Loadkeys

Permite carregar um mapa de teclado. Por exemplo, o comando a seguir carrega o mapa armazenado no arquivo `portugal.map`.

```
loadkeys /usr/lib/kbd/keytables/portugal.map
```

Setfont

Permite o carregamento de uma fonte de caracteres de tela, possibilitando a alteração das fontes utilizadas em modo de texto. O comando a seguir, por exemplo, irá carregar uma fonte com o conjunto Latin-1:

```
setfont lat1u-16.psf
```

Showfont

mostra todos os caracteres existentes na fonte que está atualmente em uso no console. O X também tem um comanto chamado showfont, que serve para mostrar as características de uma determinada fonte, mas não os caracteres em si. Para esta última finalidade se usa o comando *xfd*. Se o programa showfont do pacote KBD for invocado em um emulador de terminal X, como *xterm*, ele gerará um erro ``GIO_SCRNMAP: Invalid argument'', mas não provocará nenhum dano.

Mapscr

Permite carregar um mapa de tradução de tela. Suponhamos que exista o arquivo `/etc/portugal.trad`. Se executarmos o comando

```
mapscrn /etc/portugal.trad
```

então a partir deste momento as traduções lá definidas serão usadas.

Loadunimap

Carrega um mapa de tradução de Unicode para a fonte de tela. O mapa padrão, chamado ``def.uni'' considera que estamos usando a fonte normal do IBM-PC. Este comando **não** é necessário quando usamos uma fonte com caracteres definidos no padrão Unicode, pois o programa setfont carrega automaticamente a tabela de mapeamento adequada. O comando a seguir carregará o mapa de tradução para as fontes latin-1:

```
loadunimap lat1
```

3.4 Carregamento de uma fonte de caracteres

O pacote KBD contém dois tipos de fontes com codificação latin-1:

- as que já possuem uma tabela de mapeamento unicode, carregada automaticamente pelo programa `setfont`. Estas fontes têm nomes `lat1u-*.psf`;
- as que não possuem tabela de mapeamento. Neste caso é preciso carregar uma com o programa `loadunimap`. Estas têm nomes `lat1-*.psf` (o ``u'' indica Unicode).

Nas versões mais novas do KBD os arquivos são comprimidos com `gzip`. É preciso carregar uma fonte que tenha os caractres latinos acentuados no padrão ISO 8859-1 e *também* os símbolos semigráficos. As fontes de nome `iso01.*` não possuem esses símbolos. A fonte mais recomendada é a `lat1u-16.psf`.

As versões mais antigas do pacote KBD mantinham essas fontes no diretório `/usr/lib/kbd/consolefonts`, que foi trocado depois da versão 0.92 por `/usr/share/consolefonts`. Dependendo da sua distribuição e do quanto ela esteja atualizada o diretório poderá ser um ou outro.

Slackware

Foi criado o script `/etc/rc.d/rc.font`, contendo o seguinte:

```
#!/bin/sh
#
# /etc/rc.d/rc.font
#
# Seleciona uma das fontes de caracteres disponiveis em
# /usr/lib/kbd/consolefonts.
#
setfont lat1u-16.psf

# O comando a seguir é totalmente desnecessário
# com a fonte lat1u-16.psf.
# loadunimap lat1u

# Remova o comentário da linha abaixo se estiver usando uma fonte sem
# tabela de mapeamento de Unicode para tela:
# loadunimap lat1
```

Debian

Edite o arquivo `/etc/kbd/config` e coloque uma linha contendo

```
CONSOLE_FONT=lat1u-16.psf
```

esse arquivo é processado pelo script `/etc/rc.boot/kbd`. Execute-o para ativar a nova fonte sem te dar ``reboot''. Lembre-se sempre: Linux **não** é Windows!

TOC

Red Hat

Edite o arquivo `/etc/sysconfig/i18n` e veja se contém o seguinte:

```
LANG=pt_BR
LINGUAS=pt_BR
LC_CTYPE=ISO-8859-1
LC_ALL=pt_BR
SYSFONT=lat1u-16
SYSTERM=linux-lat
```

Conectiva Red Hat Linux

Se você selecionar a língua correta durante a instalação não há nada mais a fazer. A configuração é feita extatamente como do Red Hat.

S.u.S.E.

Edite o arquivo `/etc/rc.config` e procure uma linha que começa com ``FONT=" (linha 64, em minha máquina com S.u.S.E. verão 5.6) e coloque

```
FONT=lat1u-16.psf
```

Edite os scripts `boot.setup` e `single` contidos no diretório `/etc/rc.d`. Procure a linha contendo o comando `/usr/bin/loadunimap` e remova-a ou ponha no seu início um ``#" para deixá-la como comentário. Veja comentários sobre este comando na seção Carregamento de uma fonte de caracteres.

Testando a fonte

Experimente algumas teclas como ``,.|!"#$%&/()=?", etc. e use o comando `showfont` para mostrar a fonte em uso.

3.5 Carregando um mapa de teclado

A seguir é necessário carregar o mapa de teclado adequado. Até a versão 0.92 do pacote KBD esses mapas ficavam no diretório `/usr/lib/kbd/keytables` passando mais tarde para `/usr/share/keytables`. Dependendo da distribuição você terá um diretório ou outro. Os mapas para diversos tipos de teclados são apresentados mais adiante.

Slackware.

Foi criado o script `/etc/rc.d/rc.keyboard`, contendo o seguinte:

```
#!/bin/sh
#
# /etc/rc.d/rc.keyboard
#
# Seleciona um dos mapas de teclado disponíveis no diretório
# /usr/lib/kbd/keytables
#
loadkeys abnt2
```

e acrescentei as seguintes linhas ao `/etc/rc.d/rc.S`, imediatamente antes do tratamento do `/etc/rc.d/rc.keyboard`:

```
# Carrega uma fonte de caracteres se existe um script rc.font.
if [ -x /etc/rc.d/rc.font ]; then
  /etc/rc.d/rc.font start
fi

# Carrega um mapa de teclado se existe um script rc.keyboard.
if [ -x /etc/rc.d/rc.keyboard ]; then
  /etc/rc.d/rc.keyboard start
fi
```

Debian.

Certifique-se de ter instalado o pacote *kbd* e depois faça o seguinte:

- copie os arquivos com os mapas de teclado fornecidos (veja a seção Ficheiros necessários) para o diretório `/usr/share/keytables/`. Não é necessário descomprimi-los;
- copie o mapa correspondente ao seu teclado para o arquivo `/etc/kbd/default.map.gz`.
- rode o script `/etc/init.d/keymaps.sh`.

Red Hat.

Certifique-se de ter instalado o pacote *kbd*. Copie os arquivos com os mapas de teclado fornecidos (veja a seção Ficheiros necessários) para o diretório `/usr/lib/kbd/keytables/` (ou `/usr/lib/kbd/keymaps/i386/qwerty/`, a pertir do Red Hat 5.2). Não é necessário descomprimi-los.

Edite o arquivo /etc/sysconfig/keyboard e coloque o nome do mapa a usar na variável KEYTABLE, como por exemplo

```
KEYTABLE="abnt2"
```

Conectiva Red Hat Linux.

O CRHL já vem com os mapas de teclado para Português. Se você selecionou o teclado correto durante a instalação, não há mais nada a fazer. Caso contrário, edite o arquivo `/etc/sysconfig/keyboard` e coloque

```
KEYTABLE="nome"
```

Onde ``nome'' é *br-abnt2*, *pt* ou *us-acentos* caso o desenho do seu teclado seja ABNT-2, português ou americano.

S.u.S.E..

Certifique-se de ter instalado o pacote *kbd*. Copie os arquivos com os mapas de teclado fornecidos (veja a seção Ficheiros necessários) para o diretório `/usr/lib/kbd/keytables/`. Não é necessário descomprimi-los.

Descomprima o mapa de teclado adequado, copiando para o arquivo `/etc/default.keytab`, usando por exemplo um comando como

```
zcat /usr/lib/kbd/keytables/pt.map.gz > /etc/default.keytab
```

Testando o teclado

Experimente pressionar a tecla **c-cedilhado** (se o teclado não tem esta tecla, digite `'c`). Verifique se todas as letras acentuadas, maiúsculas e minúsculas são geradas corretamente e aparecem na tela. Se aparecer algum caracter estranho verifique se carregou o mapa de teclado e a fonte de caracteres corretos, pois provavelmente uma dessas operações foi mal sucedida.

Se ao invés de c-cedilha minúsculo aparecer um maiúsculo é provável que uma tabela de mapeamento Unicode incorreta esteja a ser carregada. Este problema ocorrerá na distribuição S.u.S.E. se não se fizerem as alterações nos scripts do sistema mencionadas anteriormente.

3.6 Verificando erros

Mas, e se alguns dos caracteres continuarem a não aparecer? Bem, antes de mais nada verifique se a fonte e o mapa de teclado adequados foram carregados. Um caso especial é quando queremos usar uma fonte que não segue a codificação ISO Latin-1 (é o caso da fonte padrão do console do PC). Teríamos então de *convencer* a tela a mostrar os caracteres certos em cada caso.

Poderíamos recorrer ao comando *mapscrn*. O arquivo com a tabela de tradução teria no entanto de ser criado por nós, seguindo um processo moroso de tentativa e erro até encontrar o caracter cuja imagem desejávamos. Ou, de uma forma mais fácil, poderíamos usar o comando *showfont*.

Se usarmos a fonte de caracteres correta, o uso deste último comando será desnecessário. É até recomendável que não se use tal recurso, pois embora ele permita criar uma tabela de caracteres ``personalizada'' em um computador, será difícil que um documento acentuado produzido nessa máquina possa ser lido em outra que não tenha a mesma configuração.

4. Biblioteca de funções libc e aplicativos GNU

A biblioteca padrão de funções do Linux suporta Internacionalização e Localização segundo o padrão POSIX (Portable Operating System Interface). Trata-se de uma norma estabelecida pelo IEEE (Institute of Electrical and Electronics Engineers) para intercomunicação entre diferentes sistemas operativos. Existe também um padrão estabelecido pelo ANSI (American National Standards Institute) para a linguagem de programação C que permite escrever programas com suporte internacional.

4.1 Configurando o suporte internacional

Resumidamente, podemos dizer que as funções que tratam informações dependentes da língua ou do país podem ter seu comportamento modificado, bastando para tanto estabelecer algumas variáveis de ambiente. E que funções são essas? Podemos citar as de formatação de datas, valores monetários e mensagens de erro do sistema. Se você não sabe o que é uma variável de ambiente, sugiro que leia o manual com o comando

```
man environ
```

O padrão POSIX permite usar diversas variáveis de configuração, cada uma delas definindo o tratamento dado a um tipo de informação, a saber

LC_COLLATE

Define regras para comparação entre caracteres no alfabeto local. Por exemplo, a letra `ã' deve ser tratada como idêntica ao `a' no Português quando ordenamos alfabeticamente.

LC_CTYPE

Define regras para comparação entre caracteres maiúsculos e minúsculos. Se estivermos trabalhando com o código ASCII, sabemos que o código numérico de uma letra minúscula é 32 a mais do que a maiúscula correspondente, mas para uma letra acentuada esta regra não é válida!

LC_MONETARY

Muda o comportamento das funções de formatação de valores monetários e permite descobrir, por exemplo, o símbolo da moeda local, ou se usa ponto ou vírgula como separador de milhares e de casas decimais.

LC_MESSAGES

Estabelece a língua em que as mensagens serão apresentadas e como é uma resposta afirmativa ou negativa (S/N).

LC_NUMERIC

Estabelece o comportamento das funções de leitura/escrita de valores numéricos permitindo, por exemplo que usemos a vírgula decimal.

LC_TIME

Define a formatação de datas e horas.

LC_ALL

Define de uma só vez todas as categorias. Se usarmos LC_ALL, não precisaremos definir nenhuma das outras, a não ser que queiramos um comportamento diferente para aquele ítem específico.

LANG

Define de uma só vez todas as categorias, se LC_ALL não estiver definida. Também é usada pelo comando `man` para compor os caminhos pelos quais ele procurará as páginas do manual. Veja as observações a respeito desta variável nas seções Locale, Locale e Locale.

Maiores informações podem ser obtidas no manual do sistema com o comando

```
man 7 locale
```

O `7' no comando anterior é necessário para evitar confusão com uma função homônima da linguagem Perl. Note que as configurações acima afetam, em princípio, apenas as funções disponíveis na biblioteca de funções da linguagem C. Na prática, todos os programas que rodam em Linux usam aquela biblioteca, mas o sistema operativo não provê, catálogos de mensagens em diversas línguas para todos os programas.

A identificação da língua e do local é feita por dois códigos de duas letras, separados por um sinal ``_". Se omitirmos o código de país, assume-se o país padrão para a língua informada, mas deve haver um catálogo correspondente. Deste modo, o código ``pt_PT" significa Português de Portugal, enquanto ``pt_BR" significa Português do Brasil. Escolher uma combinação língua/local basta então colocar em seu arquivo `/etc/profile` uma linha contendo

```
LC_ALL="pt_PT"
export LC_ALL
```

Usuários brasileiros devem usar ``pt_BR" ao invés de ``pt_PT". Teste o resultado com os comandos a seguir (`/inexistente` é o nome de um arquivo que **não** existe):

```
echo ÁÉÍÓÚ | tr '[:upper:]' '[:lower:]'
tar tf /inexistente
ls -l /
date
cal 1 1999
```

O resultado do primeiro deverá ser ```áéíóú`". Os comandos `tar` e `ls` devem retornar mensagens em Português e `cal` deve mostrar um calendário de janeiro de 1999, com *do se te qu qu se sá* no cabeçalho dos dias da semana.

Na distribuição Debian, certifique-se de ter instalado o pacote ``locales", que pertence ao grupo ``admin". O nome do arquivo que contém o pacote (na versão 2.0 da distribuição) é `locales_2.0.7t-1.deb` e em meu CD está no diretório `debian/main/binary-i386/admin`.

4.2 Problemas com a libc 5

Até meados de 1997 todas as distribuições de Linux usavam a mesma biblioteca *libc*, que estava na versão 5, desenvolvida a partir de uma versão mais antiga da libc do projeto GNU. Esta biblioteca possuia suporte muito limitado à internacionalização e normalmente nenhum dos catálogos de locais era incluído na distribuição, ficando o inglês como língua padrão.

A versão 6 da libc do Linux baseia-se na versão 2 da libc do projeto GNU. O suporte à internacionalização foi muito melhorado, além de incluir outros atributos que não interessam para os fins deste HOWTO. Se sua distribuição for a Debian versão 2, Red Hat versão 5 ou qualquer outra que use a nova libc, então não há mais nada a fazer além do descrito na seção anterior.

Mas se você usa uma distribuição ainda baseada na libc 5 (Slackware, Caldera 1.x ou uma versão mais antiga de Debian ou Red Hat) é preciso instalar a coleção de locais. Não tente copiar a coleção de uma distribuição mais recente, pois os formatos dos arquivos são incompatíveis. Pegue na página do Portuguese HOWTO o arquivo `locales-pt.tgz`. Para instalá-lo na distribuição Slackware, basta (como usuário **root**) usar o comando

```
installpkg locales-pt.tgz
```

e para outras distribuições use o comando

```
tar xzf locales-pt.tgz -C /
```

Verifique se há dois subdiretórios do `/usr/share/locale` chamados `pt_BR` e `pt_PT`. Basta então configurar a variável de ambiente LC_ALL, como já descrito.

---

5. Configuração do X

Quando o servidor X está ativo, ele coloca o teclado do computador em um modo de operação chamado *raw* (cru), em oposição ao modo normal, chamado *cooked* (isto mesmo: cozido). No modo raw o sistema operativo não processa seqüências acento-letra. O servidor X é uma daquelas raras aplicações que, por dever de ofício, tem que tratar o teclado ``em baixo nível''.

O X vem equipado com um utilitário destinado à configuração do teclado, chamado *xmodmap*, que cumpre uma função correspondente à do comando *loadkeys*, ou seja, lê um arquivo de mapa de teclado do X, expecificando as equivalências entre os *keycodes* e respectivos *keysymbols*.

Eis um excerto deste arquivo:

```
keycode 47 = ccedilla Ccedilla dead_acute dead_doubleacute
keycode 48 = masculine ordfeminine dead_circumflex dead_caron
keycode 49 = backslash bar notsign
keycode 50 = Shift_L
keycode 51 = dead_tilde dead_circumflex dead_grave dead_breve
```

Observem que, ao contrário do loadkeys, o xmodmap não possui um diretório padrão onde o arquivo é procurado.

A configuração do X não interfere de forma alguma com a configuração do modo de texto. De facto, é possível ter o seu X bem configurado, e no entanto não ter realizado qualquer tipo de configuração ao modo de texto, e vice-versa. Outro ponto importante de se observar é que os códigos numéricos das teclas no X não correspondem aos do console. A tecla **Backspace**, por exemplo, tem o número 14 no console, e 22 no X (em um computador do tipo IBM-PC).

As versões 3.2 e posteriores do XFree86 permitam definir *dead-keys*, mas o seu tratamento é responsabilidade da aplicação, conforme visto na seção Modo texto versus Sistema de Janelas X. Há modos de contornar esta limitação e tornar o tratamento de *dead-keys* transparente às aplicações, conforme mostrado na seção Contornando os limites do X.

Dividimos a configuração em duas partes: Uma que deve ser feita antes do *login* do usuário e outra depois.

5.1 Configuração do xinit

Junto com este documento são fornecidos vários mapas de teclado para uso no X. Para automatizar o processo de configuração do teclado basta copiar o arquivo adequado para o diretório `/usr/X11R6/lib/X11/xinit`, onde normalmente ficam os arquivos de início da seção de trabalho no X. Na distribuição Slackware esse diretório é um link simbólico para `/var/X11R6/lib/xinit` e na Debian e na Red Hat para `/etc/X11/xinit`.

Nas distribuições Slackware e Red Hat, verifique se no referido diretório existe um arquivo chamado `.Xmodmap`. Se existir, copie o `Xmodmap.`<alguma-coisa> para ele, ou faça um link. Normalmente o arquivo de configuração `xinitrc` possui os comandos para carregá-lo automaticamente. Veja o seguinte trecho:

```
#!/bin/sh
# $XConsortium: xinitrc.cpp,v 1.4 91/08/22 11:41:34 rws Exp $

userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
sysresources=/usr/X11R6/lib/X11/xinit/.Xresources
sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap

# merge in defaults and keymaps

if [ -f $sysresources ]; then
    xrdb -merge $sysresources
fi

if [ -f $sysmodmap ]; then
    xmodmap $sysmodmap
fi
```

Na distribuição Debian o mapa de teclado padrão do X é `/etc/X11/Xmodmap`, basta copiar o mapa desejado para aquele arquivo. Ele será carregado pelo xinitrc, que por sua vez é um link simbólico para `/etc/X11/Xsession`, sempre que iniciar a seção de trabalho do usuário.

5.2 Configuração do XDM

Existe ainda um pequeno problema: no meu computador, por exemplo, o sistema carrega diretamente o X ao dar boot e o login é feito pelo X Display Manager (xdm). Como o xdm faz o login **antes** de iniciar a seção de trabalho, o mapa de teclado não será carregado, o que pode criar problemas se o usuário usa caracteres como ``[" ou ``]" em sua senha, pois nos teclados ABNT-2 e português esses símbolos são gerados por teclas cujos códigos numéricos não são os mesmos do teclado americano.

É preciso fazer uma pequena alteração no arquivo de configuração Xsetup_0. Esse arquivo deve estar no diretório `/usr/X11R6/lib/X11/xdm`, que na Slackware é um link simbólico para `/var/X11R6/lib/xdm` e `/etc/X11/xdm` na Debian e -- sujeito a confirmação -- RedHat. Eis o conteúdo completo desse arquivo:

```
#!/bin/sh
#
# /usr/X11R6/lib/X11/xdm/Xsetup_0
#
```

```
sysresources=/usr/X11R6/lib/X11/xinit/.Xresources
sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap

# merge in defaults and keymaps

if [ -r $sysresources ]; then
    xrdb -merge $sysresources
fi

if [ -r $sysmodmap ]; then
    xmodmap $sysmodmap
fi

xconsole -geometry 480x130-0-0 -daemon -notify -verbose -exitOnFail
```

A distribuição Debian tem um `Xsetup_0` um pouco diferente, em `/etc/X11/xdm/Xsetup_0`, mas basta acrescentar-lhe o seguinte:

```
sysmodmap=/etc/X11/Xmodmap
sysresources=/etc/X11/Xresources

if [ -f $sysresources ]; then
  xrdb -merge $sysresources
fi

if [ -f $sysmodmap ]; then
  xmodmap $sysmodmap
fi
```

Se no seu computador o login do usuário também for feito sempre via xdm, não é necessário mexer no xinitrc, como mostrado na seção anterior, já que o mapa de teclado será carregado antes do início na seção do usuário.

5.3 Compose

Uma das coisas mais importantes a definir quando vamos utilizar acentuação por meio de *dead-keys* é o conjunto de *regras de composição*. Essas regras determinam, por exemplo que a composição do caracter `'` com a letra **e** gerará um **é**.

Ao contrário do console, no qual podemos definir as regras de composição no próprio mapa de teclado, no X essas regras são colocadas no arquivo `/usr/X11R6/lib/X11/locale/???/Compose`, onde ??? é a codificação em uso (no nosso caso, iso8859-1).

Para facilitar o uso dos mapas para teclados que não têm o c-cedilhado, como o US+ (veja adiante) é conveniente definirmos uma nova regra de composição, permitindo que o **Ç** seja gerado pela seqüência `'C`. Se não fizermos isso, seremos obrigados a digitar `<dead_cedilla-C>`, sedo o *dead_cedilla*

produzido pela combinação `AltGR-=`, o que não é nada confortável. Além disso, no teclado americ somos obrigados a usar as aspas duplas para gerar o trema.

O arquivo `Compose` fornecido foi feito procurando imitar ao máximo o comportamento do console e possui as seguintes facilidades para geração de caracteres:

- C-cedilha: tecla `Ç` ou seqüência <acento-agudo>-C
- Apóstrofo: tecla `'` ou seqüências <acento-agudo>-<acento-agudo> e <acento-agudo>-<espaço>
- Acento grave: seqüências <acento-grave>-<acento-grave> e <acento-grave>-<espaço>
- Aspas: tecla `"` ou seqüência <trema>-<trema>
- Trema: seqüência <trema>-<espaço>
- Til: seqüências <til>-<espaço> e <til>-<til>
- Circumflexo: seqüências <circumflexo>-<espaço> e <circumflexo>-<circumflexo>

Para incluirmos as novas regras, basta aplicar uma alteração à definição original. O arquivo `Compose.patch` pode ser obtido via WWW na página do Portuguese HOWTO. Para aplicar a atualização, copie-o para o diretório `/usr/X11R6/lib/X11/locale/iso8859-1/`, faça uma cópia de reserva do `Compose` original e invoque o utilitário *patch*:

```
cp -p Compose Compose.backup
patch < Compose.patch
```

Caso você prefira não aplicar o ``patch'', um arquivo `Compose` pronto também é fornecido. Lembre-se de fazer cópia do original **antes** de substituí-lo!

5.4 Locale

Para os usuários brasileiros, pode haver mais uma alteração a fazer no X. Conforme vimos na seção Biblioteca libc, uma variável de ambiente (LANG ou LC_ALL) configura o suporte internacional existente na biblioteca de funções padrão do sistema (*libc*). A biblioteca de funções do X (*Xlib*) usa a variável LANG para identificar a língua em uso mas até a revisão 6.3, na qual é baseado o XFree86, não era incluída a combinação ``pt_BR'', correspondente a Português/Brasil. Como resultado, cada vez que executarmos um aplicativo X com LC_ALL definido como ``pt_BR'' ele emitirá a mensagem ``Warning: locale not supported by Xlib, locale set to C''.

Para incluir o local pt_BR no X11R6.3 basta alterar três arquivos existentes no diretório `/usr/X11R6/lib/X11/locale`. Pegue o arquivo `Xlocale.patch` via WWW na página do Portuguese HOWTO. Para aplicar a atualização, copie-o para o diretório `/usr/X11R6/lib/X11/locale`, faça uma cópia de reserva dos arquivos a serem alterados e invoque o utilitário *patch*:

```
cp -p compose.dir compose.dir.backup
cp -p locale.alias locale.alias.backup
cp -p locale.dir locale.dir.backup
patch < Xlocale.patch
```

Caso você prefira não aplicar o ``patch'', arquivos prontos também são fornecidos. Lembre-se de fazer cópias dos originais **antes** de substituí-los!

Em setembro de 1998 o fix-2 da revisão 6.4 do X Window System incluiu oficialmente o suporte ao local pt_BR. Infelizmente neste mesmo fix-2 foi introduzida uma falha em uma das funções de tratamento de *input contexts* da Xlib que provoca violações de acesso à memória. Uma das aplicações afetadas é o processador de textos LyX. Uma correção provavelmente será incluída no fix-4. O X11R6.4 deverá ser a base para a versão 4 do XFree86.

5.5 Geração de mapas com o XKeyCaps

O programa XKeyCaps, criado por Jamie Zawinski é uma interface gráfica para o `xmodmap`, que mostra na tela o desenho de um teclado e permite modificar interativamente os símbolos gerados por cada tecla e gerar automaticamente o arquivo `.Xmodmap` correspondente.

Ele pode ser obtido via WWW em http://www.jwz.org/xkeycaps/ e, a partir da versão 2.43, já incoropra também suporte para o desenho brasileiro (ABNT2), uma contribuição de Andre Gerhard.

5.6 Contornando os limites do X

Conforme foi explicado na seção Modo texto versus Sistema de Janelas X, o tratamento de acentos deve ser feito pela aplicação, mas ainda há muitos programas que não levam isto em consideração, tais como Netscape Navigator/Communicator e Nedit. Como não é possível modificar muitos deles, é preciso encontrar outro tipo de solução.

Alteração da biblioteca Xlib

Linux, assim como a maioria dos sistemas operacionais modernos, usa um sistema de vínculo de programas a bibliotecas de funções chamado ``ligação dinâmica'' (*dynamic binding*). Deste modo, podemos modificar o comportamento de um programa alterando uma dessas bibliotecas. Maiores informações sobre este tema pode ser obtida nos manuais do Linux com os comandos

```
man ld.so
man ldconfig
man ldd
man dlopen
```

Thomas Quinot criou uma alteração para a biblioteca de funções do X (*Xlib*) introduzindo o tratamento de acentos na função **XLookupString**, Tudo que se tem a fazer é substituir o arquivo contendo esta biblioteca por outro, que pode ser obtido via internet no endereço

http://web.fdn.fr/~tquinot/dead-keys.en.html

Existem duas versões do arquivo, uma para sistemas onde as bibliotecas do X suportam o uso seguro de *threads* (Debian 2.x, Red Hat 5.x, etc.) e outra para os sistemas que não possuem tal atributo (Slackware, Caldera 1.x). Se você não sabe o que são threads não se preocupe com isso, mas saiba que eles permitem

criar um programa capaz de se dividir em sub-processos que rodam concorrentemente em um computador. Para analisar sua *Xlib* rode o seguinte comando:

nm --dynamic /usr/X11R6/lib/libXext.so.6|grep _Xglobal_lock

Se aparecer ```U _Xglobal_lock`'' seu sistema suporta threads e o arquivo a obter é `libX11-XF3.3.1-TS.tar.gz`. Se não aparecer, seu sistema não suporta threads e o arquivo a obter é `libX11-XF3.3.1.tar.gz`. Tendo obtido o arquivo, copie-o para um diretório temporário e extraia seu conteúdo. Mova o arquivo `/usr/X11R6/libX11.so.6.1` para outro diretório, para preservá-lo. Não é suficiente renomeá-lo! Remova-o para um diretório cujo nome **não** esteja contido no arquivo `/etc/ld.so.conf`. Depois, mova o novo arquivo para o lugar do antigo e rode o programa *ldconfig* (isto deve ser feito pelo usuário **root**):

```
cp libX11-XF3.3.1-TS.tar.gz /tmp
cd /tmp
tar xzf libX11-XF3.3.1-TS.tar.gz
mkdir /usr/X11R6/oldlib
mv /usr/X11R6/lib/libX11.so.6.1 /usr/X11R6/oldlib
mv libX11.so.6.1 /usr/X11R6/lib
chown root:root /usr/X11R6/lib/libX11.so.6.1
chmod 755 /usr/X11R6/lib/libX11.so.6.1
/sbin/ldconfig
```

É aconselhável que a operação seja realizada quando nenhuma aplicação X estiver rodando. A seguir, edite o arquivo de configuração do servidor X, chamado `XF86Config`. Este arquivo fica no diretório `/etc` (Slackware) ou `/etc/X11` (Debian, Red Hat). Procure a seção ``Keyboard'' e inclua a opção ``XkbDisable'', conforme mostrado a seguir:

```
Section "Keyboard"
   Protocol        "Standard"
   XkbDisable
EndSection
```

A opção XkbDisable inabilita a extensão XKEYBOARD do servidor X, o que neste caso serve para sinalizar à função **XLookupString** que ela deve tratar os acentos. Se quisermos voltar ao comportamento normal, basta retirar a opção do `XF86Config`.

Configure o mapa de teclado do X conforme explicado nas seções anteriores deste documento. Para testar o resultado, rode o programa *xedit* e digite alguns caracteres acentuados.

Alteração no kernel do Linux

Esta seção é baseada em contribuição enviada por Bruno Barberi Gnecco e na documentação do diacrd. Podem haver ainda alguns erros e se alguém os detectar, por favor avise-me.

Seguindo a regra geral do mundo *nix, existe sempre mais de um modo se resolver o mesmo problema. Conforme vimos anteriormente, existem dois modos de operação do teclado, chamados *raw* e *cooked*. Estes modos de operação podem ser mudados com o programa **kbd_mode**, integrante do pacote kbd, o

que não é recomendável fazer a não ser para restaurar o estado do console após um desastre com o servidor X, por exemplo. No modo raw o kernel não processa os diacríticos (acentos).

Cedric Adjih criou uma alteração para o kernel que permitia tratar os acentos mesmo em modo raw, inicialmente voltada para o teclado de desenho francês. Enéas Queiroz, André D. Balsa e Claudemir Todo Bom fizeram melhorias e o adaptaram para os teclados internacional, português e ABNT. O tratemanto de acentos é feito parte pelo kernel, parte por um processo que roda em retaguarda (*daemon*) chamado **diacrd**, conforme descrito a seguir:

- Uma modificação no kernel faz que todos os dados lidos do teclado sejam redirecionados para um dispositivo especial do sistema chamado `/dev/rawkbd` quando o teclado estiver em modo *raw*.
- Os dados que o kernel grava no `/dev/rawkbd` são lidos pelo programa `diacrd`, que processa as seqüências acento-letra e reescreve no `/dev/rawkbd` os caracteres acentuados na forma de códigos de varredura (*scancodes*) que normalmente não existem no teclado.
- O kernel lê no `/dev/rawkbd` os códigos gravados pelo `diacrd` e os passa à aplicação, que no caso é o servidor X.
- O servidor X mapeia os códigos de varredura para caracteres. Um mapa de teclado especial permite então gerar as letras acentuadas.

O diacrd pode ser obtido via FTP anônimo no endereço

ftp://metalab.unc.edu/pub/Linux/system/keyboards

Há versões diferentes, de acordo com o tipo de teclado e, pelo menos por enquanto, não se pode reconfigurá-lo sem recompilar. Para fazer a instalação, de acordo com o arquivo ``README'', deve-se fazer o seguinte (pelo usuário **root**):

- Obtenha o arquivo -.tar.gz correspondente a seu teclado. Extraia seu conteúdo, o que deverá criar um diretório chamado `diacrd-<versão>`.
- Edite o arquivo `diacrd.c` e remova os comentários desejados, de acordo com suas preferências. Pode-se incluir capacidade de rodar o programa **xmodmap** e ejetar o CD-ROM sob controle do pressionamento de uma combinação de teclas.
- Aplique a modificação no kernel. Para isto, execute o comando ``make patch-usr-src''.
- Crie o dispositivo `/dev/rawkbd` executando o comando ``make rawkbd''.
- Compile o diacrd e instale-o executando os comandos ``make'' e ``make install''.
- Compile e instale o kernel, com os comandos

  ```
  cd /usr/src/linux
  make config
  make zlilo
  ```

  Para maiores informações sobre compilação/instalação do kernel, leia o Kernel-HOWTO, disponível via Internet nos repositórios do LDP mencionados na seção Onde encontrar a versão mais atual.
- Dê um ``reboot'' no computador. Verifique se o suporte ao dispositivo rawkbd foi instalado com sucesso com o comando

```
dmesg | grep RAWKBD
```

Deve aparecer

```
RAWKBD interface for diacriticals translation enabled...
```

e teste o dispositivo `/dev/rawkbd` com o comando

```
printf "### TESTANDO ###" >> /dev/rawkbd
```

Veja na seção Configuração do console como carregar um mapa com suporte à acentuação. O pacote diacrd vem com um arquivo chamada `usintl.map`, que corresponde ao nosso `us+.map`. Se seu teclado não for do tipo americano, use o `pt.map` ou `abnt-2.map`.

- Substitua o arquivo `/usr/X11R6/lib/X11/xkb/keycodes/xfree86` pelo arquivo fornecido junto com o diacrd.
- Se tudo correr bem, rode o programa **diacrd**. Depois ative o X e use o **xmodmap** para carregar o mapa `xmodmap.diacrd`.

Se a geração de acentos funcionar, é preciso tornar permanentes as alterações necessárias. Pode-se fazê-lo incluindo a carga do diacrd no arquivo `/etc/rc.d/rc.local` (Slackware).

Se algum dos leitores tiver criado um script de ativação do diacrd para a distribuição Debian ou Red Hat, por favor envie-o para que seja incluído aqui.

Mais informações sobre o diacrd podem ser encontradas no ``Dead keys Mini-HOWTO'' de Claudemir Todo Bom, disponível em http://linux.unicamp.br/docs/diversos/deadkeys.html.

Comparação entre as duas soluções

Em termos de resultados práticos, a Xlib modificada e o diacrd são equivalentes. Há porém algumas diferenças entre as duas soluções que merecem atenção.

Diacrd exige uma alteração no kernel do Linux para resolver um problema que o kernel já resolve, o que não deixa de parecer um tanto redundante. A dependência de um processo de usuário (o *daemon* kerneld) é uma desvantagem, pois se esse programa deixar de funcionar perde-se o suporte à acentuação no X. Trata-se de uma solução ainda em desenvolvimento e, segundo o conteúdo da documentação que o acompanha, o ideal seria que todo o tratamento fosse feito pelo kernel, sem depender de outro programa.

Até o momento da publicação deste HOWTO, o diacrd ainda não era compatível com a versão 2.2 do kernel do Linux.

A solução via Xlib é, pelo menos em teoria, totalmente transparente às aplicações e permite reconfiguração em tempo de execução, bastando alterar o mapa de teclado do X e as regras de composição. Já o diacrd precisa ser recompilado se quisermos trocar o tipo de teclado, mas é possível que novas versões incorporem algum recurso de reconfiguração.

O diacrd só funciona no teclado do próprio computador. Se quisermos usar um *display* remoto, seja um terminal X, seja um outro computador, não teremos suporte à acentuação. A troca da Xlib, por outro lado, enquadra-se plenamente na filosofia do X de que o servidor provê *mecanismo*, deixando a cargo da aplicação a definição de *políticas*. Exemplo da vantagem deste paradigma é poder usar as técnicas

descritas na seção Configuração do X em outros sistemas operativos e com terminais X.

A solução via Xlib parece-me mais ``limpa'' e é minha predileta, mas, como sempre, fica a critério do usuário ou do administrador do sistema qual alternativa escolher. De qualquer modo, sempre é bom lembrar qua ambas as soluções são remendos. Considerando-se a rápida evolução que está a ocorrer no campo do software livre, com o desenvolvimendo de ambientes de trabalho como KDE, GNOME e GNUStep, podemos supor que em breve nenhum desses remendos será mais necessário.

---

6. Configuração dos vários programas

6.1 Aplicativos

A maioria das aplicações que rodam no Unix usa algum tipo de arquivo de configuração que o usuário coloca em seu diretório de trabalho (home) e cujo nome normalmente é `.alguma-coisarc`. Tanto quanto possível, tentei evitar que isso fosse necessário, pois além de dar mais trabalho ao usuário (e ao administrador da rede ;-) pode dificultar um pouco as coisas. Por exemplo, aqui no CPMet temos o diretório *home* compartilhado entre um servidor Alpha rodando DEC UNIX com os PCs rodando Linux via NFS (até a maior parte do Linux está instalada no Alpha, os PCs só têm a partição raiz e uma área de swap). Os arquivos podem necessitar algum ajuste dependendo da plataforma e nem todos os programas possuem flexibilidade bastante para isso.

Uma opção que muitos programas também oferecem é especificar em uma variável de ambiente o nome do arquivo de configuração ou o uso de arquivos padrão que normalmente ficam em um diretório `/usr/lib/alguma-coisa` ou `/etc/alguma-coisa`.

Bash (biblioteca GNU readline)

Os programas que utilizam a biblioteca GNU readline para ler a linha de comando procuram por um arquivo chamado `.inputrc` no diretório ``HOME'' do usuário caso não exista uma variável de ambiente `INPUTRC` contendo o caminho para um arquivo de configuração.

Coloque uma linha no seu arquivo `/etc/profile` contendo

```
INPUTRC="/etc/inputrc"
export INPUTRC
```

e crie um arquivo /etc/inputrc contendo

```
set meta-flag on
set convert-meta off
set output-meta on
```

Outra alternativa é criar um arquivo `.inputrc` no diretório *home* do usuário com o conteúdo acima, mas é muito difícil manter atualizados os arquivos de todos os usuários, principalmente quando eles são muitos. Um `inputrc` mais completo pode ser obtido via WWW na página do Portuguese HOWTO. Ele contém opções para vários tipos de terminal e permite usar as teclas de movimento de cursor para percorrer o histórico de comandos (setas para cima e para baixo); ir para o primeiro e para o último comandos do histórico (teclas PageUp e PageDown); posicionar o cursor na linha (setas para a esquerda e direita) e posicionar o cursor no início e no fim da linha (teclas Home e End).

Para maiores informações leia os manuais do **bash** e da biblioteca **readline** com os comandos

```
man bash
```

```
man readline
```

Emacs

O pai de todos os editores pode ser configurado criando-se um arquivo chamado `.emacs` no diretório do usuário, contendo as seguintes linhas:

```
(set-input-mode nil nil 1)
(standard-display-european t)
(require 'iso-syntax)
```

Para tornar esta configuração global, na distribuição Slackware coloque os comandos no arquivo `/usr/lib/emacs/site-lisp/site-start.el`. Na distribuição Debian o emacs executa todos os scripts contidos no diretório `/etc/emacs/site-start.d` ao ser carregado. Tudo que se tem a fazer é colocar esses comandos em um arquivo chamado, por exemplo, `01portugues-emacs.el`.

Se o estimado leitor, assim como eu, não se agrada do tratamento dado pelo Emacs às teclas de *Delete*, *Home* e *End*, aproveite a oportunidade e acrescente ao mesmo arquivo o seguinte:

```
(global-unset-key [backspace] )
(global-set-key [backspace] 'delete-backward-char)
(global-unset-key [delete] )
(global-set-key [delete] 'delete-char)
(define-key global-map [home] 'beginning-of-line)
(define-key global-map [C-home] 'beginning-of-buffer)
(define-key global-map [end] 'end-of-line)
(define-key global-map [C-end] 'end-of-buffer)
```

Arquivos de configuração prontos podem ser obtidos via WWW na página do Portuguese HOWTO. Para Slackware, há um `site-start-emacs.el`, que deve ser copiado para o diretório `/usr/lib/emacs/site-lisp` com o nome de `site-start.el`. Para Debian, há um `01portugues-emacs.el` que deve ser copiado para o diretório `/etc/emacs/site-start.d`.

Certifique-se de estar usando a versão 24-out-1998 ou mais recente do arquivo de mapa de teclado para o X, pois ela possui uma correção no tratamento das teclas modificadoras Alt e Meta, que são muito usadas pelo Emacs.

flex

Especifique a opção **-8** se o *parser* a gerar necessitar de ler dados de 8 bit.

Fortune

Fortune é aquele programa que toda vez que é invocado apresenta uma pequena mensagem, geralmente bem humorada. Ele é inspirado nos biscoitos da fortuna chineses (em inglês *fortune cookies*, daí o nome). Eis algumas mensagens típicas:

```
dROGA!!oNDE ESTA O cAPSLOCK??
```

```
Mouse não encontrado, bater no gato? (S/N)

Que fio é ess<=V++088.../NO CARRIER

Quem ri por último está conectado a 2400Bit/s.
```

Tudo que o programa faz é escolher aleatoriamente uma mensagem em um repositório mantido no diretório `/usr/games/fortunes` (Slackware) ou `/usr/share/games/fortunes` (Debian). Neste diretório existem diversos arquivos com as ``fortunas'' e um arquivo índice para cada um deles, que possui a extensão `.dat`. O formato dos arquivos é muito simples: cada fortuna é composta de uma série de linhas de texto. As fortunas são separadas umas das outras por linhas contendo apenas um caracter **%**. Veja o trecho a seguir:

```
O que são quatro pontos na parede? Four migas. Ugh!
%
Errar é humano, botar a culpa no computador é mais humano ainda.
%
Aí ela me disse: Ou eu ou o modem! Sinto muitas saudades dela...
```

Tudo que temos a fazer é criar um arquivo com as fortunas chamado, digamos `fortunes` com o formato descrito acima. Depois basta usar o programa `strfile` para gerar o índice:

```
strfile fortunes
```

e um arquivo chamado `fortunes.dat` será criado. Claro que se quisermos que o fortune mostre apenas mensagens em Português, teremos que remover os arquivos existentes no diretório original. Sugiro simplesmente renomeá-lo para `fortunes-en` (de *En*glish) e criar outro vazio. Eu coletei algumas fortunas e as coloquei no arquivo `fortunes-pt.tar.gz` que pode ser obtido via WWW na página do Portuguese HOWTO. Não esqueça de colocar no seu /etc/profile algumas linhas contendo uma chamada ao fortune, por exemplo

```
if [ -x /usr/games/fortune -a ! -e $HOME/.hushlogin ]; then
   echo
   /usr/games/fortune
   echo
fi
```

Uma última informação: se o nome de um arquivo termina com o sufixo `-o` o fortune só o consulta se for chamado com a opção `-o`. Esses arquivos são os que contém mensagens cujo conteúdo pode ser considerado ofensivo por algumas pessoas, tais como

```
Só não mando a sogra pro inferno, com pena do Diabo.
```

Claro que existem coisas muito mais ofensivas por aí, mas este é um Linux HOWTO e não queremos realmente ofender ninguém, certo?

Ispell

Dicionários para o Português de Portugal podem ser obtidos via WWW na página do *Projecto Natura* em http://www.di.uminho.pt/~jj/pln/pln.html. Para o Brasil, há uma versão compilada pelo Ueda: http://www.ime.usp.br/~ueda/.

Eu gostaria de poder colocar maiores informações, mas ainda não tenho conhecimento suficiente sobre o Ispell e não posso ensinar o que não sei. Preciso de ajuda aqui.

TOC

JDK (inclui ICQJava)

Esta informação é baseada em uma mensagem da qual guardei o conteúdo mas não o remetente. Peço desculpas e espero que perdoe a falha. Não testei pessoalmente a informação e peço que me escrevam confirmando tanto a correção quanto, se possível, a identidade do autor.

O JDK utiliza fontes padrão que não suportam acentos. Isto quer dizer que letras com acentos são ignoradas e, geralmente, confundem o resto do texto. Resolver isto é extremamente fácil:

- entre no diretório `jdk1.1.5/lib` (no meu caso, `/jdk1.1.5/lib`);
- copie o arquivo `font.properties.hu` sobrescrevendo o `font.properties` atual (lembre-se de fazer backups!);
- entre no jdk e divirta-se! Fazendo isto, seu ICQ e todos os aplicativos que usem o jdk estarão aptos a lerem acentos! (fácil, não?)

Joe

Invoque o joe com a opção `-asis` na linha de comando ou altere os arquivos de configuração para ativar tal opção. Na Slackware eles estão no diretório `/usr/lib/joe`. Tudo que se tem a fazer é remover o espaço em branco existente no início de cada linha. Outra alternativa é acrescentar a seguinte linha ao arquivo `/etc/profile`:

```
alias joe='joe -asis'
```

Joe pode emular os editores Pico, emacs e WordStar. Um arquivo *joerc* está disponível via WWW na página do Portuguese HOWTO, contendo configurações que permitem usar as teclas *Home* e *End* para movimentar o cursor para o início e fim da linha.

Less

Coloque as seguintes linhas no seu arquivo `/etc/profile`:

```
LESS="-MM -i"
LESSCHARSET="latin1"
LESSKEY="/etc/lesskey"
LESSOPEN='|lesspipe.sh "%s"'
export LESS LESSCHARSET LESSKEY LESSOPEN
```

LESSKEY informa o nome de um arquivo contendo uma tabela de seqüências de caracteres geradas por cada tecla e as ações a serem tomadas pelo less. Para criar o arquivo `/etc/lesskey`, crie primeiro o arquivo `/etc/lesskey.in` contendo as seguintes linhas:

```
# Termianl ANSI (console do Linux, XTerm, etc)
\e[1~       goto-line
\e[4~       goto-end
\e[5~       back-screen
```

```
\e[6~       forw-screen
\e[7~       goto-line
\e[8~       goto-end
\e[A        back-line
\e[B        forw-line
# XTerm
\eOH        goto-line
\eOF        goto-end
\e[H        goto-line
\e[F        goto-end
# Console Sun (testado com teclados Type 4/5)
\e[214z     goto-line
\e[220z     goto-end
\e[216z     back-screen
\e[222z     forw-screen
# Arquivo seguinte/anterior
:n  next-file
:N  next-file
:p  prev-file
```

Depois ``compile-o'' usando o comando

```
# lesskey -o /etc/lesskey /etc/lesskey.in
```

Crie o arquivo `/usr/bin/lesspipe.sh` contendo

```
#!/bin/sh
# This is a preprocessor for 'less'.  It is used when this environment
# variable is set:   LESSOPEN="|lesspipe.sh %s"

case "$1" in
  *.rpm) rpm -qilp "$1" 2>/dev/null ;;
  *.tar) tar tvvf "$1" 2>/dev/null ;;
  *.tgz | *.tar.gz | *.taz | *.tar.Z | *.tar.z)
    tar tzvvf "$1" 2>/dev/null ;;
  *.tbz2 | *.tar.bz2)
    bzip2 -dc "$1" | tar tvvf - 2>/dev/null ;;
  *.Z) gzip -dc "$1"  2>/dev/null ;;
  *.z) gzip -dc "$1"  2>/dev/null ;;
  *.[1-9].gz | *.n.gz | *.man.gz)
    FILE=`file -Lz "$1" | cut -d ' ' -f 2`
    if [ "$FILE" = "troff" ]; then
      gzip -dc "$1" | groff -s -p -t -e -Tlatin1 -mandoc
    fi ;;
  *.gz) gzip -dc "$1"  2>/dev/null ;;
  *.zip) unzip -l "$1" 2>/dev/null ;;
  *.[1-9] | *.n | *.man)
    FILE=`file -L "$1" | cut -d ' ' -f 2`
    if [ "$FILE" = "troff" ]; then
```

```
        groff -s -p -t -e -Tlatin1 -mandoc "$1"
      fi ;;
    esac
```

Não esqueça de torná-lo executável:

```
chmod 755 /usr/bin/lesspipe.sh
```

Na distribuição Debian já existe um script `/usr/bin/lesspipe` (note a ausência da extensão `.sh`). Para os curiosos a respeito da referência a ``*.rpm", embora na máquina em questão se use Slackware, é possível ter o utilitário RPM instalado também, o que facilita tomar ``emprestados" pacotes do Red Hat, Caldera e S.u.S.E.. Existe um *RPM+Slackware Mini-HOWTO* que explica como fazer isso.

ls

Acrescente a seguinte linha ao arquivo `/etc/profile` :

```
alias ls="ls -N"
```

ou

```
alias ls="ls -b"
```

Se a sua distribuição de Linux usa o GNU ls (todas as que eu conheço usam) basta acrescentar ao arquivo `/etc/profile` ou `.profile` as seguintes linhas:

```
# ----------------------------------------
# Set up the color-ls environment variables
# ----------------------------------------
if [   "$SHELL" = "/bin/bash" -o \
       "$SHELL" = "/bin/sh" ]; then
   eval `dircolors -b`
elif [ "$SHELL" = "/bin/zsh" ]; then
   eval `dircolors -z`
elif [ "$SHELL" = "/bin/ash" ]; then
   eval `dircolors -s`
elif [ "$SHELL" = "/bin/ksh" -o \
       "$SHELL" = "/bin/pdksh" ]; then
   eval `dircolors -k`
elif [ "$SHELL" = "/bin/csh" -o \
       "$SHELL" = "/bin/tcsh" ]; then
   eval `dircolors -c`
else
   eval `dircolors -b`
fi
```

Se o seu shell é o csh ou tcsh, acrescente a seguinte linha ao arquivo `/etc/csh.login` ou `~/.login`:

```
alias ls 'ls --color'
```

LyX

Para aqueles que acham trabalhoso escrever documentos para o LaTeX usando um simples editor de texto (e realmente é) LyX é uma excelente opção. Este programa cria uma interface gráfica através da qual editamos os documentos que serão depois formatados pelo LaTeX. O ambiente é quase-WYSIWYG (What You See Is What You Get - O que tu vês é o que tu obténs). LyX não roda apenas em Linux, mas em qualquer Unix. Maiores informações podem ser obtidas em

http://www.lyx.org

A partir da versão 1.0.1 do LyX já tem a interface com o usuário em Português, traduzida por Pedro Kröger, que também está traduzindo os manuais, junto com Roberto Mello. Estas traduções já fazem parte da distribuição oficial do LyX.

Tendo o LyX instalado, é muito fácil criar documentos com acentuação em Português. Seguindo as seguintes regras:

- Se o teclado foi configurado para ter *dead keys* usando um dos mapas aqui fornecidos, não é necessário fazer mais nada. Basta digitar o texto normalmente usando as seqüências de acentuação. A única exceção é o c-cedilha que não pode ser gerado usando a seqüência `'c`, pois o LyX gera um c com acento. Temos que usar a seqüência Compose-vírgula-c.
- Se o teclado **não** foi configurado para ter *dead keys* ainda assim é possível acentuar no LyX. Selecione o *menu* `Options/Keyboard`. Na caixa de diálogo ``Key Mappings'', selecione no ítem `Language/Primary` a opção ``American''. Com isto o LyX fará a composição dos caracteres acentuados usando regras semelhantes às das *dead keys*.
  - A vírgula será tratada como cedilha. Para obter um `Ç'digite `,C` e para obter uma vírgula digite `,,`. Cuidado! A seqüência `,` <espaço> gerará uma cedilha isolada e não uma vírgula!
  - `~ ^ ' e ` ` serão tratados como acentos. Vale a mesma regra anterior: para obter apenas o acento, pressione a tecla duas vezes consecutivas.
  - `: ; . / ?` e `-` também serão tratados como acentos. `?a` gerará um `å' e assim por diante.

Para o LyX imprimir corretamente, é necessário que, ao criar um novo documento, sejam selecionados a língua e a codificação de caracteres adequadas. Crie um documento selecionando o *menu* `File/New`. Depois selecione o *menu* `Layout/Document`. Na caixa de diálogo ``Document Layout'' selecione no ítem `Language` a opção ``brazil'' ou ``portuges'' (sem o *u* mesmo); no ítem `Encoding` selecione ``latin1''.

Veja a observação sobre o pacote *algorithm* na seção TeX e LaTeX

Uma observação final sobre o LyX: a versão atual (1.0.x) utiliza a biblioteca XForms para construir a interface com o usuário. Como essa biblioteca não tem suporte para acentuação, não é possível digitar letras acentuadas nas caixas de diálogo, somente no corpo do documento editado. Segundo os desenvolvedores, nas novas versões do LyX será possível escolher o tipo de interface ao compilar o programa, o que permitirá o uso de *toolkits* mais flexíveis. Já existe uma versão de LyX portada para o toolkit Qt, usado no KDE, chamada KLyX. Os autores são Matthias Ettrich -- autor original do LyX -- e Kalle Dalheimer. Para maiores informações, consulte via WWW: http://www-pu.informatik.uni-tuebingen.de/users/ettrich/.

Man, groff, troff

Pode-se usar a opção de linha de comando `-Tlatin1` para o groff, mas é mais simples colocar uma linha no seu arquivo `/etc/profile` contendo

```
GROFF_TYPESETTER="latin1"
export GROFF_TYPESETTER
```

Para maiores informações leia o manual do groff com o comando

```
man groff
```

No Linux, o comando *man* usa o *groff* para formatar os manuais e deve ser configurado para usar o conjunto Latin 1, ou não será possível formatar satisfatoriamente manuais que contenham caracteres não pertencentes ao conjunto ASCII, como é o caso do ``man iso_8859_1''. Na distribuição Slackware, é preciso editar o arquivo `/usr/lib/man.config` e alterar as definições NROFF e NEQN, trocando a opção ``-Tascii'' para ``-Tlatin1'':

```
NROFF           /usr/bin/groff -Tlatin1 -mandoc
NEQN            /usr/bin/geqn -Tlatin1
```

ou, se usarmos a variável de ambiente GROFF_TYPESETTER, podemos simplesmente eliminar a opção -Tascii. Na distribuição Debian não é necessário fazer nenhuma configuração para o man, bastando configurar GROFF_TYPESETTER e na Red Hat (incluindo Conectiva) o arquivo é `/etc/man.config`

Groff também pode ser configurado para fazer a separação silábica em português, o que é muito útil se tivermos páginas de manual em Português. Isto é bastante simples, porque quando James Clarck pôs suporte a hifenização no groff ele usou o mesmo algoritmo de hifenização do TeX. Para ter hifenização em português, basta copiar o arquivo de regras do TeX e fazer o `groff` usá-lo. O arquivo `hyphen.pt` pode ser obtido junto com os demais na página do HOWTO. Trata-se do arquivo `pt8hyph.tex` (ver seção TeX e LaTeX) ao qual foi adicionado apenas um comentário.

Ele deve ser copiado para o diretório `/usr/share/groff/tmac/` ou `/usr/lib/groff/tmac/`, dependendo da distribuição. É conveniente criar um ``link'' simbólico para `hyphen.br`. Para usar esse arquivo, basta colocar no início do seu documento troff/groff as linhas a seguir:

```
.if \n(.g \{\
.hla pt
.hpf hyphen.pt
.\}
```

O `.if` não é necessário para a acentuação, mas ele testa se o processador usado é o groff. Deste modo pode-se usar o mesmo documento em outros UNIX nos quais se use o troff normal sem que ele gere mensagens de erro sobre comandos desconhecidos.

Midnight Comander (mc)

No menu `Options` sub-menu `Display bits...` ligue a opção ``Full 8 bits'' ou ``ISO 8859-1''. Na versão 3.2.11 isso permite que sejam mostrados nomes de arquivos contendo caracteres acentuados, mas não foi possível digitar tais caracteres na linha de comando ou nas caixas de diálogo.

Minicom

Coloque uma linha no seu arquivo `/etc/profile` contendo

```
MINICOM="-m -c on"
export MINICOM
```

Isso permitirá usar a tecla Alt para ativar os comandos (exatamente como o Telix) e também usar cores. Para maiores informações, leia o manual do Minicom usando o comando

```
man minicom
```

Mais uma dica sobre o Minicom: para fazê-lo usar usar corretamante a tecla *Meta* para ativação dos comandos rodando dentro de um *xterm*, deve-se invocá-lo com a opção ``-m'' e passar a opção ``-xrm "*eightBitInput: false"'' para o xterm. Se usarmos o *rxvt* então o minicom deve ser chamado com a opção ``-m'' e a tecla de ativação dos comandos será *Alt*. Fiz uma adaptação no script `xminicom` que pode ser obtido via WWW na página do Portuguese HOWTO.

Segundo Arnaldo Carvalho de Melo, as versões mais recentes do Minicom suportam internacionalização. Suporte para o Português foi acrescentado pelo pessoal da Conectiva. O código fonte pode ser obtido na página do Jukka (atual mantenedor do Minicom) em

http://www.clinet.fi/~walker/minicom.html

Netscape Communicator

O Communicator, assim como muitas aplicações que utilizam *toolkits* baseados no X Toolkit (Xt), permite que se modifiquem muitas de suas características por meio de arquivos de configuração (X Resources). No caso específico do Communicator, pode-se criar um arquivo chamado `Netscape` no diretório `/usr/X11R6/lib/X11/app-defaults` contendo estas opções.

Na distribuição brasileira Conectiva, o pacote do Communicator vem com um arquivo `/usr/lib/netscape/i18n/Netscape.ad.pt_BR` e um script `/usr/bin/netscape` que executa o Communicator fazendo-o ler este arquivo. Uma cópia do `Netscape.ad.pt_BR` pode ser obtida na página oficial do Portuguese-HOWTO. Para usá-lo, simplesmente copie-o para `/usr/X11R6/lib/X11/app-defaults/Netscape`, se sua distribuição não o possuir, mas fique atento que seu uso pode criar problemas com novas versões do programa.

Maiores informações sobre ``X Resources'' podem ser obtidas em [GET94], [McC94] e no manual do programa `xrdb`, com o comando

```
man xrdb
```

nn

Alô, alô, alguém usa nn? Informação mais atualizada será bem recebida.

Acrescente a seguinte linha ao arquivo `~/.nn/init`:

```
set data-bits 8
```

Pine e Pico

Para o Pine utilizar o conjunto de caracteres Latin 1, coloque uma linha no arquivo .pinerc, no diretório do usuário, contendo

```
character-set=ISO-8859-1
```

ou crie um arquivo geral de configuração contendo tal linha. Esse arquivo normalmente é `/usr/local/lib/pine.conf` ou `/usr/lib/pine.conf`

A configuração também pode ser feita usando o próprio programa. No *menu* de entrada selecione as opções Setup/Configuration. Vá até o ítem ``character-set'' e preencha-o com ``ISO-8859-1''. Para maiores informações leia o manual do pine com o comando

```
man pine
```

tcsh

Nenhuma medida especial é necessária se for feita a correta configuração das variáveis de ambiente ``LANG'' e ``LC_ALL'', conforme mostrado mostrado na seção Biblioteca libc.

Tcsh tem suporte a internacionalização e na distribuição Debian há um pacote chamado tcsh-i18n com suporte para French, German, Greek and Spanish.

TeX e LaTeX

Esta seção foi escrita com ajuda de Klaus Steding-Jessen.

O pacote Babel

O pacote *Babel*, criado por Johannes Braams provê suporte a um grande número de idiomas para o LaTeX. De acordo com o idioma selecionado ele define muitas coisa como, por exemplo, os títulos dos capítulos (Chapter, Capítulo, Kapitel) e o título das tabelas (Tabela, Table, Tabelle). Para usar o pacote, basta incluir no preâmbulo de seu documento LaTeX o comando

```
\usepackage[portuges]{babel}
```

ou

```
\usepackage[brazil]{babel}
```

Há diferenças sutis entre o Portugues do Brasil e o de Portugal, tais como as normas para escrita de datas e nomes de meses com a primeira letra maiúscula ou minúscula.

Pode-se misturar mais de um idioma no mesmo documento. Para Alemão e Português ficaria:

```
\usepackage[german,brazil]{babel}
```

Nesse caso a última opção (brazil) fica sendo o idioma corrente. Para mudar ao longo do texto, entre um e

outro, use:

```
\selectlanguage{german}

[...]

\selectlanguage{brazil}
```

Isto é muito útil também quando queremos que uma palavra não seja separada. Basta definir um novo idioma e usá-lo nas palavras que não podem ser separadas. Coloque no preâmbulo:

```
\newlanguage\nohyphen
\newcommand\nh[1]{{\language\nohyphen #1}}
```

E use com \nh{FOO BAR}. É melhor do que usar `\mbox`, que impede quebra no espaco em `\mbox{FOO BAR}`.

Separação silábica

Normalmente apenas os suportes a separação silábica para Inglês e Alemão são carregados. Para configurar hifenização no **teTeX**, execute o utilitário *texconfig*, que na distribuição Slackware deve ser o programa `/usr/lib/teTeX/bin/texconfig` e na Debian é `/usr/bin/texconfig`. Digite o comando

```
texconfig hyphen
```

O editor usado normalmente é o **vi**. Se o seu editor predileto for outro, crie uma variável de ambiente chamada **EDITOR** contendo o nome desse programa, como no exemplo a seguir:

```
EDITOR=pico
export EDITOR
```

O editor de texto será carregado, para editar o arquivo `language.dat`. Procure uma linha que começa por **%portuges** e remova o **%**. Grave o arquivo e saia do editor. O texconfig atualizará diversos arquivos de configuração (não se assuste com a quantidade de mensagens que aparecem na tela).

Se o nome da tabela de hifenização que consta no seu `language.dat` é `pthyph.tex` então provavelmente trata-se da versão 1.0, de 1987. A versão 1.2, de 1996, pode ser obtida via FTP anônimo em

ftp://ftp.tex.ac.uk/tex-archive/language/portuguese/pt8hyph.tex

Copie-o para o mesmo diretório onde se encontra o `pthyph.tex`, que dependendo de sua distribuição pode ser `/usr/lib/texmf/tex/generic/hyphen` `/usr/share/texmf/tex/generic/hyphen` ou `/usr/lib/teTeX/texmf/tex/generic/hyphen`. Execute o comando `texconfig~hash` e na mesma linha mencionada anteriormente insira um `8' no nome do arquivo, de modo que fique `pt8hyph.tex`.

Uso de Font Encoding T1

Se o conjunto de caracteres (*encoding*) de uma fonte não provê acesso direto a caracteres acentuados, (como é o caso do OT1, padrão do LaTeX) então o TeX gera estes caracteres sobrepondo o caracter base mais o acento. O algoritmo de hifenização, contudo, não hifeniza palavras formadas dessa forma.

Se uma fonte já provê acesso direto a caracteres acentuados, como as de codificação T1, estes caracteres são usados diretamente sem prejuízo do algoritmo de hifenização. O uso de *encoding* T1 é fundamental para a correta hifenização em Português.

Note que trata-se da mesma fonte, por falta a Computer Modern, apenas o que muda é o seu *encoding*, isto é, a ordem/disponibilidade dos caracteres dentro da mesma. Para selecionar estas fontes, inclua no preâmbulo do seu documento o comando

```
\usepackage[T1]{fontenc}
```

O antigo pacote `t1enc` não deve mais ser usado e existe hoje apenas por questões de compatibilidade com documentos antigos. O `fontenc` é mais atual, continua a ser mantido e é de uso mais geral, portanto preferível.

Edição de documentos

Normalmente a introdução de caracteres acentuados no texto exige o uso de *seqüências de escape* bastante trabalhosas. Para gerar um ``ö'' deve-se digitar \"o. Com *babel* pode-se digitar apenas "o, o que não deixa de ser inconveniente para ler o fonte do documento. Há um pacote chamado **inputenc** que permite especificar a codificação em que estão os caracteres de um documento. Lembre-se porém que se o seu documento for enviado para outro usuário que não possua o **inputenc** ele poderá não conseguir processá-lo, mas esse recurso já está disponível desde a liberação do LaTeX2e em dezembro 1994. Todas as distribuições de Linux atuais o incluem.

Sugestão: uma configuração do GNU Emacs para gerar caracteres acentuados, própria para o uso com o pacote inputenc:

```
(add-hook 'LaTeX-mode-hook
          (lambda ()
            (standard-display-european 1)
            (load-library "iso-acc")
            (iso-accents-mode 1)
            (iso-accents-customize "portuguese")
            (auto-fill-mode 1)
            ))
```

E outra que não requer o inputenc. O uso da biblioteca `iso-cvt' faz a transformação de ISO-8859-1 (no buffer do Emacs) para o padrão do LaTeX (no arquivo).

```
(add-hook 'LaTeX-mode-hook
          (lambda ()
            (standard-display-european 1)
            (load-library "iso-cvt")
```

```
(load-library "iso-acc")
(iso-accents-mode 1)
(iso-accents-customize "portuguese")
(auto-fill-mode 1)
))
```

Muitas vezes o usuario já possui muitos arquivos num formato de acento do LaTeX e gostaria de passar tudo para o formato ISO, próprio para uso do pacote inputenc. Uma boa opção para isto é usar o programa `recode`. Se você possui arquivos .tex e deseja converte-lo para formato ISO-8859-1 pode usar:

```
recode -d LaTeX:l1 file.tex
```

O código-fonte do GNU recode pode ser obtido via FTP anônimo em ftp://prep.ai.mit.edu/pub/gnu/recode/recode-3.4.tar.gz e em muitos espelhos do GNU mundo afora. A distribuição Debian tem um pacote pronto e outras também devem ter.

Teste da configuração do LaTeX

Para testar a nova configuração copie o seguinte trecho para um arquivo chamado, digamos, `exemplo.tex`:

```
\documentclass[a4paper]{article}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[portuges]{babel}
%%
%% ou \usepackage[brazil]{babel}
%%

\begin{document}
\title{Linux Portuguese-HOWTO}
\author{Carlos Augusto Moreira dos Santos}
\date{\today}

\maketitle

\section{Introdução}

Este documento pretende ser um guia de referência de configuração
do \textbf{Linux} e seus programas, teclados e fontes de
caracteres, permitindo sua internacionalização/utilização confortável
por pessoas que falem a Língua Portuguesa.

\end{document}
```

Esse texto contém de propósito uma ``palavra'' bastante longa para forçar a separação silábica. Ele está disponível via WWW no arquivo `exemplo.tex` na página do Portuguese HOWTO. Para processá-lo, use o comando `latex`, conforme mostrado a seguir:

```
bash$ latex exemplo.tex
This is TeX, Version 3.14159 (C version 6.1)
(exemplo.tex
LaTeX2e <1996/06/01>
Hyphenation patterns for english, german, portuges, loaded.
(/usr/lib/teTeX/texmf/tex/latex/base/article.cls
Document Class: article 1996/05/26 v1.3r Standard LaTeX document class
(/usr/lib/teTeX/texmf/tex/latex/base/size10.clo))
(/usr/lib/teTeX/texmf/tex/latex/base/inputenc.sty beta test version
(/usr/lib/teTeX/texmf/tex/latex/base/latin1.def))
(/usr/lib/teTeX/texmf/tex/latex/base/fontenc.sty
(/usr/lib/teTeX/texmf/tex/latex/base/T1enc.def))
(/usr/lib/teTeX/texmf/tex/generic/babel/babel.sty (portuges.ldf
(/usr/lib/teTeX/texmf/tex/generic/babel/babel.def))) (exemplo.aux) [1]
(exemplo.aux) )
(see the transcript file for additional information)
Output written on exemplo.dvi (1 page, 812 bytes).
Transcript written on exemplo.log.
```

A mensagem ``Hyphenation patterns for english, german, **portuges**, loaded." indica que a configuração foi bem sucedida. Se o seu computador está rodando o X o documento formatado poderá ser visto com o comando

```
xdvi exemplo.dvi
```

Observe que o ```\today`" gera a data corrente. No ``portugues" seria ``17 de Julho de 1998" mas no ``brazil" seria ``17 de julho de 1998".

Problemas com alguns pacotes do teTeX

Há um problema com o pacote *algorithm* do LaTeX, que não é suportado pelo babel, fazendo com que a lista de algoritmos saia com o título ``List of Algorithms" e o título de cada um deles seja impresso como ``Algorithm #". Para evitar esse problema, coloque no preâmbulo de seu documento LaTeX, após o comando ```\usepackage{algorithm}`" o seguinte:

```
\makeatletter
\renewcommand{\ALG@name}}{Algoritmo}}
\makeatother
\renewcommand{\listalgorithmname}}{Lista de Algoritmos}}
```

No teTeX 0.4 (versão 1.2h do portuges.ldf) o título do ambiente *proof*, encontrado nas classes *amsbook*, *amsart*, etc., sai como ``Proof.". Para corrigir isso, coloque no preâmbulo de seu documento o comando

```
\renewcommand{\proofname}}{Demonstra\c{c}\~ao}
```

mas o mais recomendável é que se atualize a versão do teTeX.

Usuários do LyX podem incluir esses comandos no preâmbulo do documento usando o menu `Layout/LaTeX Preamble`. Pode-se resolver o problema alterando o arquivo `portuges.ldf` que

contém as definições necessárias ao Português, mas esta solução não é portável, pois exigirá que tal arquivo seja alterado em todas os computadores em que o documento deva ser processado.

A licença do Babel não me permite distribuir o `portuges.ldf` separado do resto do pacote, mas para os interessados, coloquei à disposição na página do HOWTO um arquivo chamado `portuges.ldf.patch` que criei para o meu teTeX antigo (Slackware 3.4). Para aplicar a atualização, copie-o para o diretório `/usr/lib/teTeX/texmf/tex/generic/babel` (Slackware), faça uma cópia de reserva do `portuges.ldf` original e invoque o utilitário `patch`:

```
cp -p portuges.ldf portuges.ldf.backup
patch < portuges.ldf.patch
```

Use o patch por sua conta e risco! Se você deixar o backup no diretório original ele será incluído no arquivo `ls-R` da próxima vez que o programa *texhash* for executado. Isso não fará mal algum, mas pode-se mover o backup para algum lugar seguro (eu uso `/usr/backup`).

Lista TeX-BR

Há uma lista de discussão brasileira de usuários de TeX/LaTeX, chamada TeX-BR, que roda no servidor de listas da FURG. Para entrar da lista mande um mail contendo apenas a palavra ``subscribe" no corpo para < tex-br-request@listas.furg.br>. Esta lista é administrada por Rafael Rodrigues Obelheiro.

Há uma página na WWW em http://biquinho.furg.br/tex-br/ por meio da qual se pode ler o histórico de mensagens. Também há referências para muitos documentos sobre LaTeX, alguns em Português e outros em Inglês.

LaTeX-demo

Pode ser útil também um documento de exemplo para ter onde começar. Pensando nisso, Klaus Steding-Jessen preparou um pequeno documento em Português com o objetivo de ser um guia ``by example" para o usuário de LaTeX iniciante e intermediário, que pode ser obtido via WWW em http://biquinho.furg.br/doc/LaTeX-demo/.

Klaus ecreveu também uma série três artigos destinados a ``descrever o sistema LaTeX como uma alternativa mais eficiente aos processadores de texto WYSIWYG" que podem ser lidos via WWW em http://biquinho.furg.br/tex-br/doc/artigo-1-jessen/.

WordPerfect

O Corel Wordperfect para Linux versão 8 suporta a acentuação por teclas mortas. Não é necessário nenhum procedimento especial. Como um entusiasta de software livre, entretanto, recomendo enfaticamente que se use o LyX, que já foi inclusive traduzido para nossa língua, conforme mencionado na seção LyX.

Xemacs (antigo lucid emacs)

> Agradeço a colaboração de Judson Santos Santiago e Goedson Teixeira Paixão que ajudaram a identificar os problemas com o Xemacs.

O Xemacs já tem suporte à acentuação direta no teclado usando ``dead keys", mas há um erro na configuração

original que o impede de reconhecer o acento circunflexo. Este problema não ocorre se for usada a biblio *Xlib* aterada por Thomas Quinot, mencionada na seção Contornando os limites do X, mas mesmo que nã usemos, basta colocar os seguintes comandos no seu arquivo de configuração `.emacs`:

```
;; Ajuste para fazer o acento circunflexo funcionar
;; Contribuição de Goedson Teixeira Paixao <gopaixao@dcc.ufmg.br>
(require 'x-compose)
(define-key global-map 'dead-circumflex compose-circumflex-map)
```

Na distribuição Debian 2.0 o Xemacs 20.4 executa todos os scripts contidos no diretório `/etc/xemacs20/site-start.d` ao ser carregado. Tudo que se tem a fazer é colocar esses comandos em um arquivo chamado, por exemplo, `01portugues-xemacs.el`. Não é necessário instalar o *xemacs20-mule*, que possui extensões para línguas que não usam o alfabeto romano. Os pacotes a instalar são os seguintes:

- xemacs20-bin
- xemacs20-nomule
- xemacs20-support
- xemacs20-supportel

Se você instalou o Xemacs no Slackware ou outro Unix, à moda antiga (dowload, compilação, instação), então o arquivo a alterar é o `site-start.el`, que deve estar no diretório `/usr/lib/xemacs/site-lisp` ou `/usr/local/lib/xemacs/site-lisp`, dependendo de sua instalação.

Observe que a partir da versão 20.3 o Xemacs usa uma variável especial para controlar o comportamento da tecla ``Delete'', não existente nos outros emacs chamada **delete-key-deletes-forward**. Para ativar este comportamento coloque no seu arquivo `.emacs` uma linha contendo

```
(setq delete-key-deletes-forward t)
```

Arquivos de configuração prontos podem ser obtidos via WWW na página do Portuguese HOWTO. Para Slackware, há um `site-start-xemacs.el`, que deve ser copiado para o diretório correto com o nome de `site-start.el`. Para Debian, há um `01portugues-xemacs.el` que deve ser copiado para o diretório `/etc/xemacs20/site-start.d`.

Certifique-se de estar usando a versão 24-out-1998 ou mais recente do arquivo de mapa de teclado para o X, pois ela possui uma correção no tratamento das teclas modificadoras Alt e Meta, que são muito usadas pelo Xemacs.

6.2 Rede local e Internet

FTP (File Transfer Protocol)

Existem dois modos de transferência de arquivos: *binary* e *ASCII*, sendo este utilizado para textos. Deve-se tomar cuidado ao transferir um arquivo, pois o modo de transfêrencia `ASCII` remove o oitavo bit de cada caracter transmitido, o que terá como conseqüência a perda de todos os caracteres acentuados. Desta forma é aconselhado o envio de documentação em modo `binary` de forma a manter a integridade da mesma.

Cuidado! Algumas versões mais antigas do pacote net-tools do Linux têm um cliente FTP que não reconh corretamente quando o servidor remoto roda Unix. Deste modo ele não comutará o modo de transferência binário automaticamente. Além disso, alguns servidores FTP também não fornecem a informação corretamente. Certifique-se de digitar o comando `bin` antes de um `get` quando quiser que a transferência seja binária!

Correio eletrônico

O mesmo tipo de restrições do FTP se aplica ao envio de documentos contendo caracteres acentuados através de correio eletrônico. Embora isto não aconteça em todos os sistemas em uso na internet, bastará que o correio enviado passe no seu trajecto por um sistema que não suporte 8 bits de informação para que o nosso documento seja deturpado.

Para que não hajam problemas, deve-se utilizar um programa de emeil, que suporte o formato MIME (Multipurpose Internet Mail Extensions), formato este que permite o envio de documentação em modo 8 bits. Exemplos de programas de correio eletrônico com suporte para MIME, são o Eudora e o Pine.

Se o destinatário da mensagem não usa um agente com suporte para MIME, existe a opção de codificar os documentos com o utilitário UUENCODE. Para maiores informações a esse respeito, leia a documentação usando os comandos

```
man uuencode
man uudecode
```

---

7. Ficheiros necessários

Atenção! Os mapas de teclado para o X fornecidos mapeiam a função das teclas segundo a lista a seguir:

- Alt esquerdo: Alt
- Alt direito: AltGr
- Control esquerdo: Control
- Control direito: Control
- ScrollLock: ScrollLock
- Janela esquerda: Meta
- Janela direita: Compose
- Menu: Menu

Se o seu teclado não possuir as teclas *para Windows 95* então pode ser melhor restabelecer a distribuição padrão: edite o mapa e coloque um ponto de exclamação no início das linhas que definem os *keycodes* 64, 113, 115, 116 e 117.

Foram elaborados mapas com suporte à acentuação para seis modelos de teclado, tanto para uso no console quanto para o X. Eles podem ser obtidos via WWW em um dos repositórios da página do Portuguese HOWTO:

- http://linusp.usp.br/~casantos/

Ao instalar um dos mapas fornecidos, lembre-se de ler os comentários contidos neles, pois há informações importantes sobre opções de configuração e aproveitamento das teclas adicionais dos teclados padrão *Windows 95*. As instruções para instalação estão nas seções Configuração do console e Configuração do X.

O mapa US+ (us+.map e Xmodmap.us+)

Mapas para os teclados que seguem o padrão americano. Como não existem teclas especiais para gerar o c-cedilhado nem o trema, foi usado um pequeno truque: o c-cedilhado é gerado pela seqüência `'C`. No console pode-se fazê-lo com a seqüência `AltGR-C`. O trema é gerado pela tecla `"`. Para gerar as aspas duplas é necessário digitar a seqüência `"<espaço>` e no console pode-se usar `""`; opcionalmente pode-se usar `AltGR-"`, o que não é uma solução muito confortável, mas funciona...

Testados com teclados de várias marcas (e alguns sem marca :-).

O mapa Portugal (pt.map e Xmodmap.pt)

Mapas para teclados com desenho português. Esses teclados são os que possuem uma tecla com os caracteres « e ». Não são muito confortáveis, porque para gerar o símbolos **@ [ ] { }** e o trema é necessário usar a tecla **Alt-GR**.

Testado com um teclado da marca Key Tronic.

O mapa ABNT-2 (abnt-2.map e Xmodmap.abnt2)

Os computadores vendidos no Brasil fabricados pela IBM, Compaq e Itautec, entre outros, vêm com esses teclados. Eles também pode ser adquiridos avulsos e são fabricados pela UIS e Keytec (não confundir com Key Tronic). Este desenho é o mais confortável de todos, pois tem a mesma distribuição dos acentos encontrada nas máquinas de escrever.

Testado com teclados das marcas UIS e IBM.

Sun Type 4 e Type 5 (Xmodmap.Sun4+ e Xmodmap.Sun5c+)

Encontrados nas máquinas fabricadas pela Sun. Estes mapas foram testados em SPARCstations modelos 1, 4, IPC e Classic rodando SunOS 4.1, Solaris 2.5.1 e Linux 2.0.33 (Red Hat 4.2). Como esses teclados não possuem o c-cedilhado, foi usado o mesmo truque do mapa US+. Apenas o mapa para o X é fornecido.

NCD+ (Xmodmap.NCD+)

Este mapa é para o teclado modelo N107 e foi testado em terminais X modelo 16r fabricados pela Network Computing Devices Inc. (NCD). Embora a distribuição de teclas seja idêntica à do Sun *Type 4*, os *keycodes* das teclas são diferentes. Esses teclados também não possuem o c-cedilhado, por isso foi usado o mesmo truque do mapa US+. Apenas o mapa para o X é fornecido.

A geração dos caracteres acentuados é feita de acordo com a tabela a seguir:

```
Para gerar              Digitar
-------------------------------------------------
ç                       'c ou compose-,-c
á é í ó ú               'a 'e 'i 'o 'u
à è ì ò ù               `a `e `i `o `u
ã õ                     ~a ~o
â ê î ô û               ^a ^e ^i ^o ^u
ä ë ï ö ü               "a "e "i "o "u
" (aspas)               "" ou AltGR-" no mapa US+
¨ (trema)               "-espaço
' (apóstrofo)           '' ou AltGR-' no mapa US+
` (grave)               `` ou AltGR-` no mapa US+
®                       compose-O-R
©                       compose-O-C
-------------------------------------------------
```

8. Informações Adicionais

8.1 Fontes de informação sobre Linux em Português

Grupo de Investigação Linux

Grupo de usuários de linux da Universidade do Minho, em Portugal que pode ser visitado em http://gil.di.uminho.pt.

Grupo de Utilizadores de Linux do ISCTE

Pode ser visitado em http://www.students.iscte.pt/~a12593/gul.html.

Linux Brasil

Servidor da Universidade Estadual de Campinas, Brasil, contendo informações sobre Linux no Brasil, que absorveu o antigo projeto LDP-br. Pode ser visitado em http://linux.unicamp.br.

Projeto LinUSP

Desenvolvido na Universidade de São Paulo, Brasil, coordenado por Jorge L. deLyra. Pode ser visitado em http://linusp.usp.br/.

8.2 Free Translation Project

Programadores que quiserem desenvolver aplicações com suporte a mais de uma língua devem usar algum método de definir mensagens em tempo de execução, possivelmente usando uma biblioteca de funções para este fim. O projeto GNU criou o ``Free Translation Project", com esta finalidade. Muitos aplicativos GNU já permitem que se troquem as mensagens bastando criar um catálogo que é lido pelo programa ao iniciar. Há equipes de tradução responsáveis por cada língua.

Há uma lista de discussão via emeil para cada equipe de tradução, rodando em um servidor provido pela < Linux International>. Para fazer contato com a equipe de tradução para Português, envie um emeil para < pt@li.org> e para assinar a lista envie emeil para < pt-request@li.org> contendo apenas a palavra ``subscribe". Assine a lista somente se você quiser fazer parte da equipe de tradutores, pois ela **não** se destina ao esclarecimento de dúvidas de usuários!

Mais informações podem ser encontradas na documentação do pacote *gettext* disponível via FTP anônimo em qualquer repositório de software do GNU. Provavelmente sua distribuição já tem um pacote pronto para ser instalado, mas se você desejar, os fontes da versão mais recentes podem ser encontrados em

ftp://ftp.gnu.org/pub/gnu/

e o nome do arquivo a copiar é ``gettext-<versão>.tar.gz".

8.3 LIE-BR

A informação a seguir é baseada no conteúdo da página de apresentação da lie-br, escrita por Jorge Godoy.

A lie-br foi criada com o intuito de auxiliar a comunidade Linux brasileira a obter softwares e documentação em sua língua nativa: o português. Essa lista é uma ramificação de outra maior, a LIE que visa a internacionalização do Linux e sua posterior tradução para todos os idiomas (português incluso).

Resumo dos objetivos da lie-br:

- contribuir para o projeto de internacionalização do Linux fazendo a parte relativa ao português para o Brasil;
- internacionalizar programas, de modo a poderem ser usados nas línguas nativas de cada país, e não ficar preso ao idioma original ou ao inglês;
- disponibilizar (sic) material em português para consulta e documentação;
- aumentar o acervo de documentação disponível e, indiretamente, auxiliar no desenvolvimento iniciado pelo projeto LDP-BR.

Mais informações podem ser encontradas na página WWW da lie-br, no endereço

http://lie-br.conectiva.com.br/

Ainda há uma quantidade enorme de documentos por traduzir. Se você tem bom domínio do Inglês e deseja colaborar, então **primeiro** leia as instruções na página mencionada e assine a lista ``lie-br", enviando um emeil para < lie-br-subscribe@bazar.conectiva.com.br>. Não é preciso especificar o assunto (subject) nem colocar nada no corpo da mensagem.

8.4 Versões de software testadas

Todas as informações presentes neste documento foram testadas nas seguintes versões de software:

- Distribuições Slackware 3.2 (muito modificada), Debian 2.0, Red Hat 5.2, Conectiva 3.0 (Guarani) e S.u.S.E. 5.6
- XFree86 versões 3.3 a 3.3.3.1
- Fvwm 2.0.46
- Rxvt versão 2.4.5
- Kernel versões 2.0.33 a 2.0.36
- Kbd versões 0.92 a 0.96
- GNU emacs versões 19.34 a 20.3
- Less versões 321 e 332
- GNU Bash versões 1.14.7 a 2.01.1
- tcsh versões 6.07.02 e 6.07.06

TOC

- LyX 0.12.0 a 1.0.0pre6
- XForms 0.88
- Joe 2.8
- Pine 3.96 e 4.0
- Pico 2.9
- teTeX versões 0.4 e 0.9-9
- XEmacs versões 20.3 e 20.4

---

9. Observações finais

9.1 Futuras adições a este documento

As próximas versões do HOWTO deverão conter, entre outros assuntos, os seguintes:

- suporte à distribuição Caldera;
- informações sobre compartilhamento de arquivos em rede usando NFS, SAMBA e Mars-NWE;
- configuração de *toolkits* de interface gráfica: Qt, GTK, XForms, Tk (Tcl) e os baseados em Xt, como Motif, Lesstif, Xaw (*international ainda não está funcionando);
- uma seção sobre desenvolvimento de programas;
- referências ao Ispell (usá-lo para corrigir o próprio HOWTO :-);
- informações sobre X/Open, XPG4 e POSIX, ou pelo menos ponteiros para elas.

Aceitam-se voluntários para qualquer uma destas tarefas.

Parece cada vez mais claro que as informações sobre configuração de aplicativos devem ser movidas para outro documento, independente do HOWTO. Na forma como ele está hoje já chega a quase quarenta páginas impressas e com a adição de outras distribuições e aplicações em breve chegará às 50. Um documento tão grande foge ao espírito de um HOWTO, que deve ser sucinto e concentrado em um assunto específico.

9.2 Nota de Direitos de Autor

Este HOWTO teve como autor João Carlos Rodrigues Pereira, baseado em documentação escrita por José Bandeira além dos restantes HOWTO's do Linux. Atualmente ele é mantido por Carlos Augusto Moreira dos Santos, com a colaboração de muitos outros.

Os documentos HOWTO do Linux podem ser reproduzidos e distribuídos em todo ou em parte, segundo qualquer meio físico ou electrónico, desde que esta Nota de Direitos de Autor se mantenha intacta em todas as cópias dos mesmos. A distribuição comercial é autorizada e encorajada, no entanto, o autor gostaria de ser notificado de tais ocorrências.

Todas as traduções, trabalhos derivados, ou trabalhos agregando qualquer dos documentos HOWTO do Linux deverão estar abrangidos por esta Nota de Direitos de Autor, ou seja, não poderá ser imposta qualquer restrição adicional a trabalhos efectuados a partir de um dos documentos HOWTO do Linux nomeadamente no que diz respeito à sua distribuição.

Exceções a estas regras poderam ser obtidas. Para tal, dever-se-á contactar o coordenador dos documentos HOWTO do Linux no endereço < linux-howto@metalab.unc.edu>.

9.3 Garantia (inexistência de) e nota de responsabilidade

Apesar de ter sido feito o máximo esforço possível para que o conteúdo deste documento esteja correto, não há nenhuma garantia de que ele não possua erros, nem de que tenha algum tipo de utilidade ou aplicação comercial, técnica, educacional ou medicinal. O autor não se responsabiliza por prejuízos decorrentes do seu uso. Em resumo: se a informação aqui contida quebrar seu computador em mil pedacinhos, junte tudo e cole, mas **não reclame para mim**!

Documentos escritos por terceiros são de responsabilidade exclusiva deles e sua referência neste HOWTO não representa nenhum tipo de recomendação, abono ou garantia de suporte.

9.4 Agradecimentos

Deixo aqui os meus agradecimentos a todos os que de alguma forma me ajudaram quer através das suas sugestões quer através de contribuições de outro tipo, em especial a

João Carlos Rodrigues Pereira

Autor original deste documento que agora mantenho. Nunca tive contato com ele, nem sei por onde andará. Sua página no Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa (http://caravela.di.fc.ul.pt/~jcrp/) não existe mais. Lembro-me de tê-la visto, anos atrás.

Greg Hankins

Ex-coordenador dos **Linux HOWTO**, por me permitir assumir a manutenção deste documento e fornecer as primeiras dicas sobre autoria de documentos SGML.

As pessoas listadas a seguir enviaram mensagens diretamente para mim ou para as listas Linux-BR da UNICAMP e TeX-BR com informações, sugestões ou comentários que foram incluídas neste texto. Se alguém foi esquecido, por favor desculpe a falha.

Andre Gerhard <andre@lme.usp.br>
Arnaldo Carvalho de Melo <acme@conectiva.com.br>
Bruno Barberi Gnecco <brunobg@geocities.com>
Cees de Groot <cg@pobox.com>
Francisco Semeraro <semeraro@sti.com.br>
Goedson Teixeira Paixao <gopaixao@dcc.ufmg.br>
Joao Carvalho <foxfire@fe.up.pt>
Jorge Carvalho Pinto <jmcpinto@yahoo.com>
Judson S Santiago <judson@costeira.dimap.ufrn.br>
Ken MacLeod <ken@bitsko.slc.ut.us>
Klaus Steding-Jessen <jessen@acm.org>
Lamarque Vieira Souza <lamarque@dcc.ufmg.br>
Marcelo Malheiros <malheiro@dca.fee.unicamp.br>
Marcos Vinicius Lannes dos Santos <lannes@cnpgl.embrapa.br>

TOC

Pedro Kröger <kroger@e-net.com.br>
Rafael Caetano dos Santos <rcaetano@linux.ime.usp.br>
Rafael Rodrigues Obelheiro <obelix@biquinho.furg.br>
Ramiro Morales <morales@siscard.com.ar>
Raul Carvalho <rmpc@ip.pt>
Ricardo Y. Igarashi <iga@originet.com.br>
Roberto Mello <mello@internext.com.br>
Wanderlei Antonio Cavassin <cavassin@conectiva.com.br>

Os seguintes agradecimentos são do primeiro autor:

Carlos Ferreira

Pela luta que trava pela defesa da lingua portuguesa, bem patente na sua *Página Portuguesa* disponível no URL: `http://lila.dei.uc.pt/~cjrf/po/` [Nota do mantenedor atual: esta página não existe]

João C. Silva

Pelo apoio e incentivo e criticas (bem como por me deixar testar os meus conhecimentos no SEU computador).

José Bandeira

Autor dos ficheiros `port.map` e `xmodmap`. Pelo seu apoio e por ter escrito alguma da documentação mais elucidativa que eu já li sobre o assunto.

---

10. Bibliografia comentada

Esta seção ainda está incompleta e deve ser melhorada nas próximas versões. Os HOWTOs mencionados aqui geralmente são distribuídos nos CDs de instalação e todas as distribuições possuem pelo menos alguns deles em pacotes prontos para instalar. Na Slackwre esses pacotes são os da série F e os documentos ficam instalados no diretório `/usr/doc/faq/howto` e na Debian o diretório é `/usr/doc/HOWTO`. Eles também podem ser obtidos via WWW em http://metalab.unc.edu/LDP/HOWTO/.

[BAL97] BALSA, André D. Linux and X11 internationalization

Disponível via WWW em http://wauug.erols.com/~balsa/linux/deadkeys/

Uma discussão sobre o problema da acentuação no X. Também contém referências para outros documentos que tratam de internacionalização.

[BRO98] Brower Andries. The Linux Keyboard and Console HOWTO

Disponível via WWW em http://metalab.unc.edu/LDP/HOWTO/Keyboard-and-Console-HOWTO.html.

Descreve o tratamento teclado e console no Linux (kernel versão 2.0) e tem várias referências ao X. Leitura obrigatória para quem quer entender a base do assunto.

[GET94] Gettys, James et al. Xlib - C Language X Interface

Cambridge, X Consortium, Inc., 1994. 473p.

Referência definitiva sobre programação com `Xlib`.

[JON89] Jones, Oliver. Introduction to the X Window System

Englewood Cliffs, Prentice Hall, 1990.

Um livro bastante antigo, mas contendo uma excelente e didática introdução ao X e à programação com `Xlib`.

[McC94] McCormack, Joel et al.X Toolkit Intrinsics - C Language Interface

Digital Equipment Corporation/X Consortium, 1994.

A referência básica para programadores interessados em usar Xt.

[OSS92] Ossana, Joseph & Kernighan, Brian. Troff user's manual

Computer Science Technical Report No. 54. Murray Hill, AT&T Bell Laboratories, 1992.

Descrição detalhada de como usar o Troff para formatação de documentos.

[KER82] Kernighan, Brian. A Typesetter-independent TROFF

TOC

Murray Hill, AT&T Bell Laboratories, 1982.

O artigo de um dos pais do UNIX sobre o Troff.

[QUI98] Quinot, Thomas. Dead keys under X11

Disponível via WWW em http://www.fdn.fr/~tquinot/index.en.html

A versão em inglês do artigo sobre a modificação da *Xlib* para dar suporte à acentuação independente da aplicação.

[RAY98] Raymond, Eric S. The Linux XFree86 HOWTO

Disponível via WWW em http://metalab.unc.edu/LDP/HOWTO/XFree86-HOWTO.html.

Descreve como obter, instalar e configurar o XFree86. Todas as distribuições de Linux já vêm com pacotes do XFree86 prontos para instalar, mas as informações sobre configuração podem ser muito úteis.

[SCH86] Scheifler, Robert; Gettys, James. The X Window System

ACM Transactions on Graphics, New York, ACM, *5* (2): 79-109, april, 1986.

O artigo seminal de Scheifler e Gettys descrevendo o X Window System.

[SUN90a] Sun Microsystems. Using NROFF & TROFF.

Sun Microsystems, 1990.

Parte da documentação que acompanhava as estações de trabalho Sun® e o sistema SunOS®, no tempo em que os fabricantes de computadores distribuiam documentação de verdade.

[SUN90b] Sun Microsystems. Formatting documents

Sun Microsystems, 1990.

Veja o ítem anterior.

Arquivos cuja leitura pode ser útil:

`/usr/src/linux/Documentation/unicode.txt`

Explica como ativar os diversos tipos de fontes no console. Também explica onde obter fontes para o alfabeto Klingon, o que pode ser muito útil se o leitor for um habitante daquele Império ou admirador de Jornada nas Estrelas. Depois das recentes aventuras espaciais do Linux, não duvido de mais nada...

A respeito deste assunto é de fundamental importância saber que, segundo nosso consultor interestrelar Klaus Steding-Jessen, ``Klingon é o nome da raça e do seu idioma. Kling é o nome do planeta. Mas note que o Império Klingon é formado não apenas por esse planeta, mas por várias colônias, teoricamente todas falando Klingon". Tenho certeza de que sua vida nunca mais será a mesma depois de saber disto.

`/usr/src/linux/include/linux/keyboard.h`

Contém as definições de constantes, funções e macros utilizadas por programas que fazem

tratamento de teclado sob Linux. Normalmente esse arquivo é instalado com o pacote que con o código fonte do kernel. As distribuições geralmente possuem um pacote apenas com os arqu *include* e outros com o resto dos programas-fonte do kernel.

`/usr/X11R6/include/X11/keysymdef.h`

Contém as definições de constantes, funções e macros utilizadas por programas que fazem tratamento de teclado sob o X Window System. Normalmente esse arquivo é instalado com o pacote que contém as bibliotecas de desenvolvimento de aplicações para X.

Database-SQL-RDBMS HOW-TO document for Linux (PostgreSQL Object Relational Database System)

Al Dev (Alavoor Vasudevan) alavoor@yahoo.com

v11.0, 8 January 1999

This document is a "practical guide" to very quickly setup a SQL Database engine and front end tools on a Unix system. It also discusses the International standard language ANSI/ISO SQL and reviews the merits/advantages of the SQL database engine developed by the world-wide internet in an "open development" environment. It is about HOW-TO setup a next generation Object Relational SQL Database "PostgreSQL" on Unix system which can be used as a Application Database Server or as a Web Database Server. PostgreSQL implements a subset of International ISO/ANSI SQL standards of years 1998,92,89. This document also gives information on the database interface programs like Front End GUIs, RAD tools (Rapid Application Development), ODBC, JDBC drivers, "C", "C++", Java, Perl programming interfaces and Web Database Tools. Information given here applies to all Unix platforms and to all other SQL databases. It will be very useful for people who are new to Databases, SQL language and PostgreSQL. This document also has SQL tutorial, SQL syntax which would be very helpful for beginners. Experienced people will find this document as a useful reference guide. For students, the information given here will enable them to get the source code for PostgreSQL relational database system, from which they can learn as to how a RDBMS SQL database engine is created.

1. Introduction

2. Other Formats of this Document

3. Laws of Physics applies to Software!

4. What is PostgreSQL ?

5. Where to get it ?

6. PostgreSQL Supports Extremely Large Databases greater than 200 Gig

7. How can I trust PostgreSQL ? Regression Test Package builds customer confidence

8. GUI FrontEnd Tool for PostgreSQL (Graphical User Interface)

9. Integrated Development Environment Tools for PostgreSQL (GUI IDE)

10. Interface Drivers for PostgreSQL

- 10.1 ODBC Drivers for PostgreSQL
- 10.2 UDBC Drivers for PostgreSQL
- 10.3 JDBC Drivers for PostgreSQL
- 10.4 Java for PostgreSQL

11. Perl Database Interface (DBI) Driver for PostgreSQL

- 11.1 Perl 5 interface for PostgreSQL
- 11.2 Perl Database Interface DBI

12. PostgreSQL Management Tools

- 12.1 PGACCESS - A GUI Tool for PostgreSQL Management
- 12.2 Windows Interactive Query Tool for PostgreSQL (WISQL or MPSQL)
- 12.3 Interactive Query Tool (ISQL) for PostgreSQL called PSQL

- 12.4 MPMGR - A Database Management Tool for PostgresSQL

13. Setting up multi-boxes PostgreSQL with just one monitor

14. Applications and Tools for PostgreSQL

- 14.1 PostgreSQL 4GL for web database applications - AppGEN Development System
- 14.2 WWW Web interface for PostgresSQL - DBENGINE
- 14.3 Apache Webserver Module for PostgreSQL - NeoSoft NeoWebScript
- 14.4 HEITML server side extension of HTML and a 4GL language for PostgreSQL
- 14.5 America On-line AOL Web server for PostgreSQL
- 14.6 Problem/Project Tracking System Application Tool for PostgreSQL
- 14.7 Convert dbase dbf files to PostgreSQL

15. Web Database Design/Implementation tool for PostgreSQL - EARP

- 15.1 What is EARP ?
- 15.2 Implementation
- 15.3 What you need to run EARP
- 15.4 How does it work ?
- 15.5 Where to get EARP ?

16. PHP Hypertext Preprocessor - Server-side html-embedded scripting language for PostgreSQL

- 16.1 Major Features
- 16.2 Credits
- 16.3 PHP 3 - Brief History
- 16.4 So, what can I do with PHP/FI ?
- 16.5 A simple example
- 16.6 CGI Redirection
- 16.7 Running PHP/FI from the command line

TOC

17. Python Interface for PostgreSQL

- 17.1 Where to get PyGres ?
- 17.2 Information and support

18. Gateway between PostgreSQL and the WWW - WDB-P95

- 18.1 About wdb-p95
- 18.2 Does the PostgreSQL server, pgperl, and httpd have to be on the same host?

19. "C", "C++", ESQL/C language Interfaces and Bitwise Operators for PostgreSQL

- 19.1 "C" interface
- 19.2 "C++" interface
- 19.3 ESQL/C
- 19.4 BitWise Operators for PostgreSQL

20. Japanese Kanji Code for PostgreSQL

21. PostgreSQL Port to Windows 95/Windows NT

22. Mailing Lists

- 22.1 Get a Free e-mail account
- 22.2 English Mailing List
- 22.3 Archive of Mailing List
- 22.4 Spanish Mailing List

TOC

23. Documentation and Reference Books

- 23.1 User Guides and Manuals
- 23.2 Online Documentation
- 23.3 Useful Reference Textbooks
- 23.4 ANSI/ISO SQL Specifications documents - SQL 1992, SQL 1998
- 23.5 Syntax of ANSI/ISO SQL 1992
- 23.6 Syntax of ANSI/ISO SQL 1998
- 23.7 SQL Tutorial for beginners
- 23.8 Temporal Extension to SQL92
- 23.9 Part 0 - Acquiring ISO/ANSI SQL Documents
- 23.10 Part 1 - ISO/ANSI SQL Current Status
- 23.11 Part 2 - ISO/ANSI SQL Foundation
- 23.12 Part 3 - ISO/ANSI SQL Call Level Interface
- 23.13 Part 4 - ISO/ANSI SQL Persistent Stored Modules
- 23.14 Part 5 - ISO/ANSI SQL/Bindings
- 23.15 Part 6 - ISO/ANSI SQL XA Interface Specialization (SQL/XA)
- 23.16 Part 7 - ISO/ANSI SQL Temporal
- 23.17 Part 8 - ISO/ANSI SQL MULTIMEDIA (SQL/MM)

24. Technical support for PostgreSQL

25. Economic and Business Aspects

26. List of Other Databases

27. Internet World Wide Web Searching Tips

28. Conclusion

TOC

29. FAQ - Questions on PostgreSQL

30. Copyright Notice

Appendix

31. Appendix A - Syntax of ANSI/ISO SQL 1992

32. Appendix B - SQL Tutorial for beginners

- 32.1 Tutorial for PostgreSQL
- 32.2 Internet URL pointers

33. Appendix C - Linux Quick Install Instructions

---

1. Introduction

The purpose of this document is to provide comprehensive list of pointers/URLs to quickly setup PostgreSQL and also to advocate the benefits of Open Source Code system like PostgreSQL, Linux.

Each and every computer system in the world needs a database to store/retrieve the information. The primary reason you use the computer is to store, retrieve and process information and do all these very quickly, thereby saving you time. At the same time, the system must be simple, robust, fast, reliable, economical and very easy to use. Database is the most **VITAL SYSTEM** as it stores mission critical information of every company in this world. The most popular database systems are based on the International Standard Organisation (ISO) SQL specifications, which in turn is based on ANSI SQL (American) standards. The most current specifications widely used in the industry are ISO/ANSI SQL 1992. Upcoming standard is the SQL 1998/99 which is also called SQL-3 is still under development. Popular database like Oracle, Sybase and Informix systems are based on these standards or are trying to implement these standards.

There are more than 20 varieties of commercial/internet database systems which are being used in the world and many more will be coming in the near future. Without a standard like ANSI/ISO SQL, it would be very difficult for the customer to develop a application once and run on all the database systems. Customer wants to develop an application ONCE using ISO SQL, ODBC, JDBC and deploy on all varieties of database systems in the world.

The world's most popular FREE Database which implements some of the ISO SQL, ANSI SQL/98, SQL/92 and ANSI SQL/89 RDBMS is PostgreSQL. PostgreSQL is next generation Object relational database and the future ANSI SQL standards like SQL 1998 (SQL-3) and beyond will increasingly deal with Object databases and Object data types. PostgreSQL is the only free RDBMS in the world which supports Object databases and SQL. This document will tell you how-to install the database, how to set up the Web database, application database, front end GUIs and interface programs. It is strongly advised that you MUST write your database applications 100 % compliant to standards of ISO/ANSI SQL, ODBC, JDBC so that your application is portable across multiple databases like PostgreSQL, Oracle, Sybase, Informix etc.

You get the highest quality, and lot many features with PostgreSQL as it follows 'Open Source Code development model'. Open Source Code model is the one where the complete source code is given to you and the development takes place on the internet by a extremely vast network of human brains. Future trend shows that most of the software development will take place on the so called "Information Super-Highway" which spans the whole globe. In the coming years, internet growth will be explosive which will further fuel rapid adoption of PostgreSQL by the database community.

By applying the principles of statistics, mathematics and science to software quality, you get the best quality of software only in a 'Open Source Code System' like PostgreSQL, wherein the source code is open to a very vast number of human brains inter-connected by the information super-highway. Greater the number of human brains working, the better will be the quality of software. Open Source Code model will also prevent **RE-INVENTION OF WHEELS**, eliminates **DUPLICATION OF WORK** and will be very economical, saves time in distribution and follows the modern economic laws of optimizing the national and global resources. Once a software work is done by others, than you **DO NOT** need to re-do that again. You will not be wasting your valuable time on something which had already been **WELL DONE**. Your time is extremely precious and it must be utilized efficiently, because you have only 8 hours per day for doing work!! As we will be entering the 21st century, there will be a change in the way that you get software for your use. Customers will give first preference for the open systems software like PostgreSQL, Linux, etc...

If you buy binaries, you will not get any equity and ownership of source code. Source code is a very valuable asset and binaries have no value. Buying software *may* become a thing of the past. You only need to buy good hardware, it is worth spending money on the hardware and get the software from internet. Important point is that it is the computer hardware which is **doing bulk of the work**. Hardware is the real work horse and software is just driving it. Computer hardware is so much more complex that only 6 nations out of 180 countries in the world so far have demonstrated the capability of designing and manufacturing computer chips/hardware. Design and manufacturing of computer chips is a advanced technology. It is a very complex process, capital intensive, requires large investments in plant and production machines which deal with 0.18 micron technology. On a single small silicon chip millions of transistors/circuits are densely packed. Companies like Applied Material, AMD, Intel, Cyrix, Hitachi, IBM and others spent significant number of man-years to master the high-technology like Chip Design, Micro-electronics and Nano-electronics. Micro means (one-millionth of meter 10^-6), Nano means (one-billionth of meter 10^-9). Current technology uses micro-electronics of about 0.35 micron using aluminum as conductors and 0.25 micron sizes using copper as conductors of electrons. In near future the technology of 0.10 micron with copper and even nano-electronics will be used to make computer chips. Aluminum conductors will be phased out by copper on computer chips, as copper is a better conductor of electrons. In photolithography process extreme ultraviolet, X-ray or electron-beam techniques will be used to etch circuits for feature size less than 0.15 micron. In about 20 years from now, silicon chips

will be phased out by molecular computers and bio chips which will be billions of times faster than silicon chips. Molecules are a of atoms. And atoms are tiny particles which makes up everything that you see in this world. Molecular computers will use the molecules of matter as ultra-fast electronic on/off switches. When the switch is ON it indicates 1, and when it is OFF it indicates 0. All the computer programs in this world are based on binary (numbers 1 and 0). Table below shows the progress and future advancement trends of computer chips.

Advancement of chip capabilities in future

Item/Year	1997	1999	2001	2003	2012	2020
Feature size(micron)	0.25	0.18	0.15	0.13	0.05	< 0.00001
Wafer size(mm)	200	300	300	300	450	Mol/Bio
Min Operating Voltage	1.8-2.5	1.5-1.8	1.2-1.5	1.2-1.5	0.5-0.6	< 0.001
Max power dissipation	70	90	110	130	175	600
On-chip frequencey (MHz)	750	1,250	1,500	2,100	10,000	> 50,000
DRAM capacity	256 MB	1 GB	2 GB	4 GB	256 GB	> 1000GB

As you can see, it is hardware that is high technology and important and software is a less difficult technology. Hence, manufacturing hardware/hard-goods is vital for national economy! Companies like Compaq, Dell, Sun Microsystems, HP, IBM who manufacture computers are major contributors to U.S economy today and in the future!!

On other hand, each and every country in the world develops/makes software. In fact, any person in this world with a small low-cost PC can create a Oracle database server system. But it would take him about 10 years (Oracle database server is about 10 man-years of work). One man-year is one person working full-time for one full year. If 10 people work for a year than it is 10 man-years spent.

Databases like Oracle, Informix, Sybase, IBM DB2 (Unix) are written using the "C" language and binaries are created by compiling the source code and than they are shipped out to customers. Oracle, Sybase, Informix databases are 100 % "C" programs!!

Since a lot of work had been done on PostgreSQL for the past 12 years, it does not make sense to re-create from scratch another database system which satisfies ANSI/ISO SQL. It will be a great advantage to take the existing code and add missing features or enhancements to PostgreSQL and start using it immediately.

PostgreSQL is not just a free database but it is a good quality 'Internet Product'. Prediction is that demand for "Made in Internet" products will grow exponentially as it is capable of maintaining a high quality, low cost, extremely large user-base and developer-base. Those nations who do not use the 'Made in Internet' products will be seriously missing "World-wide Internet Revolution" and will be left far behind other countries. The reason is "Internet" itself is the world's **LARGEST** software company!

2. Other Formats of this Document

This document is published in 10 different formats namely - DVI, Postscript, Latex, LyX, GNU-info, HTML, RTF(Rich Text Format), Plain-text, Unix man pages and SGML.

- You can get this HOWTO document as a single file tar ball in HTML, DVI, Postscript or SGML formats from - ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/other-formats/
- Plain text format is in: ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO
- Translations to other languages like French, German, Spanish, Chinese, Japanese are in ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO Any help from you to translate to other languages is welcome.

The document is written using a tool called "SGML tool" which can be got from - http://www.xs4all.nl/~cg/sgmltools/ Compiling the source you will get the following commands like

- sgml2html databasehowto.sgml (to generate html file)
- sgml2rtf databasehowto.sgml (to generate RTF file)
- sgml2latex databasehowto.sgml (to generate latex file)

This document is located at -

- http://sunsite.unc.edu/LDP/HOWTO/PostgreSQL-HOWTO.html

Also you can find this document at the following mirrors sites -

- http://www.caldera.com/LDP/HOWTO/PostgreSQL-HOWTO.html
- http://www.WGS.com/LDP/HOWTO/PostgreSQL-HOWTO.html
- http://www.cc.gatech.edu/linux/LDP/HOWTO/PostgreSQL-HOWTO.html
- http://www.redhat.com/linux-info/ldp/HOWTO/PostgreSQL-HOWTO.html
- Other mirror sites near you (network-address-wise) can be found at http://sunsite.unc.edu/LDP/hmirrors.html select a site and go to directory /LDP/HOWTO/PostgreSQL-HOWTO.html

In order to view the document in dvi format, use the xdvi program. The xdvi program is located in tetex-xdvi*.rpm package in Redhat Linux which can be located through ControlPanel | Applications | Publishing | TeX menu buttons.

```
To read dvi document give the command -
        xdvi -geometry 80x90 howto.dvi
And resize the window with mouse. See man page on xdvi.
To navigate use Arrow keys, Page Up, Page Down keys, also
you can use 'f', 'd', 'u', 'c', 'l', 'r', 'p', 'n' letter
keys to move up, down, center, next page, previous page etc.
To turn off expert menu press 'x'.
```

You can read postscript file using the program 'gv' (ghostview) or 'ghostscript'. The ghostscript program in ghostscript*.rpm package and gv program is in gv*.rpm package in Redhat Linux which can be located through ControlPanel | Applications | Graphics menu buttons. The gv program is much more user friendly than ghostscript. Also ghostscript and gv are available on other platforms like OS/2, Windows 95 and NT, you view this document even on those platforms.

```
To read postscript document give the command -
        gv howto.ps

To use ghostscript give -
        ghostscript howto.ps
```

CAUTION: This document is large, total number of pages (postscript) if printed will be approximately 113 pages.

You can read HTML format document using Netscape Navigator, Microsoft Internet explorer, Redhat Baron Web browser or any of the 10 other web browsers.

You can read the latex, LyX output using LyX a X-Windows front end to latex.

---

3. Laws of Physics applies to Software!

In this chapter, it will be shown how science plays a important role in the creation of various objects like software, this universe, mass, atoms, energy and even yourself! This chapter also shows why knowledge of science is very important BEFORE you start using the products of science. Objects include everything - for example PostgreSQL, time, mass, energy, planets, sun, moon, stars, galaxies, super-clusters, humans etc... are objects made by science. This chapter also shows how laws of science and statistics favour the open-source code system like PostgreSQL and Linux. As the internet speed is increasing everyday, and internet is becoming more and MORE reliable, the open-source code system will gain very rapid momentum. And, if rules of statistics and laws of physics are correct than the closed source-code systems will eventually VANISH from this planet.

The paragraphs given below will show you - "how vast science is, how important it is for man and how it impacts software projects like PostgreSQL, Linux".

Developing a project like PostgreSQL requires resources like energy and time, hence PostgreSQL is a product of energy and time. Since energy and time can be explained only by science, there is a direct co-relation between physics and software projects like PostgreSQL, Linux.

Laws of science (Physics) applies everywhere and at all the times, to anything that you do, even while you are developing the software projects. Physics is in action even while you are talking (sound waves), walking (friction between ground and your feet), reading a book or writing software. Various branches of sciences like physics, chemistry etc all merge into one grand region called Mathematics (which is also known as the Queen of all Sciences). Everything in this world has a deep root in mathematics, including PostgreSQL. PostgreSQL uses 'Modern Algebra' which is a tiny branch of mathematics. Modern algebra deals with 'Set Theory', 'Relational Algebra', science of Groups, Rings, Collections, Sets, Unions, Intersections, Exclusions, Domains, Lists, etc...

The software like PostgreSQL is existing today because of the energy and time. It is the energy which made this world, human brains and many other things! And mass and energy are ONE and the **SAME** entity! The fact that mass and energy are same was unknown to people 100 years ago!

Cells in the human brains consume energy while processing (creating software), by converting the chemical energy from food into electrical and heat energy. Even while you are reading this paragraph, the cells in your brain are burning out the fuel and are using tiny amounts of energy. So STOP READING NOW ! The energy activity of neurons (brain cells) can be measured in the laboratory. For example, there are many instruments like 'Lie Detectors' and other medical instruments which can measure the energy activities of brain. All of these implies that human brain is a thermodynamic heat engine. Because human brain is a thermodynamic engine, the laws of thermodynamics applies to brain and hence thermodynamics has indirect effects on software like PostgreSQL.

As per science, it is impossible to build any system or theory (including a database software system) which will be 100 % perfect and bug free. It is like chasing a mirage, we will **NEVER** reach the goal of perfect system or theory. Detailed mathematical equations/discussions to prove that 'perfect system' (as

well as 'imperfect system') is impossible, is a advanced topic and is beyond the scope of this docume And such a mathematical discussion deals with infinite number of dimensions (as well as primary dimensions) which are existing in nature. Unfortunately humans can see or feel only 4 dimensions but mathematics can easily explore many other dimensions. Other dimensions are **'infinitely smaller'** than the atoms and atoms themselves are very minute which human eyes cannot see them! Mathematics is very powerful as it can analyze and explain the birth/death of our universe. Our universe is almost **zero size** if you look from other universe and vice versa. That is, our universe is not visible (does not exist) for persons in other universe! And vice versa!! Theoretically, you can exit out of our universe and travel vast distances (billions of light years) in zero time and re-enter universe at a different point of space-time! Distance between our universe and other universes is actually zero!

Even though there are infinite number of dimensions, they all can be derived/generated from a small number of PRIMARY dimensions. That is, infinite number of dimensions can combine and collapse into primary dimensions. Primary dimensions simply absorb other dimensions without themselves getting destroyed. Mathematicians use these primary dimensions to understand the birth and death of universes. The universe where you are currently living started with a **BIG BANG** billions of years ago (roughly 20 billion years ago) which was caused by the interactions of atomic particles of other dimensions. Just before the big bang there was a tiny point where length, breadth, height and time was ZERO (that is, our universe was NOT THERE!!) and other universes and primary dimensions were existing. Time itself was NOT there and atoms, stars, planets, galaxies were NOT there! The atoms inside your body were NOT there!! So many things happened even BEFORE the time was born!

Big bang and hence birth of our universe was caused by few atoms of primary dimensions. **SOMEONE** (something?) caused the dashing of few tiny atoms of other dimensions to create our universe, and new dimensions time, length, breadth, height was born! We see someone's hand in this process. That process is **not very well understood** by man. Man is trying to generate another universe in the lab by simulating the big bang event (Huge accelerator is under construction in Europe, another construction in Dallas, USA was stopped by US congress due to budget cuts). There are atoms of other dimensions just as we have atoms in our universe. Theoretically, you can create/generate infinite number of universes! This process is reversible, that is our universe can completely close down and vanish into few atoms of other dimensions! It is similar in anology as to how YOU were born from two tiny cells which DASHED against each other to create one single cell. That single tiny cell divided and multiplied into 6 trillion cells to become a 6 foot tall human (that is you!). There are some similarities between humans and universes, universes are born and later die, very much similar to humans.

Since PERFECT state (as well as IMPERFECT state) is impossible, universes like ours are born and later die down in a cyclic process. BUT there can NO PERFECT death, only transformation is permitted by science! Our universe is currently expanding at a tremendous rate, it is not static but is very dynamic. This universe will keep expanding untill something interferes to collapse this universe by applying pressures with external dimensions to aborb and annihilate our universe!! Two possibilities are - removing the mass from our universe via black holes (a slow process), or injecting more mass into our universe via white holes (a rapid big crunch process). There can be millions of universes in existence but there can be very few universes which are built from primary dimensions. You are not only made up of atoms of this universe but also by atoms of other dimensions! In black holes the atoms and particles from our universe get sucked into and are completely transformed and converted to particles of other dimensions! In the center of black hole time is zero and length, breadth, height is zero! And black hole is EXIT/ENTRY door to and from other universes. Entry/Exit doors to other universes can open up

TOC

anywhere, even inside your body!

There can be infinite number of colors, computer langauages, computer chip designs and theories but there CANNOT be ONE SINGLE PERFECT color, computer language, design or system! What you can have is only a NEAR PERFECT color(wavelength), system, database, or theory! Nature is like a Kaliedoscope.

By combining the energies of millions of people around the world via internet it is possible to achieve a **NEAR PERFECT** system (including a database software system). Individually, the energy of each person will be minute, but by networking a large number of people, the total energy will be huge which can be focused on a project to generate a near perfect system. Human beings are creatures of finite energy and resources, frequently guilty of errors ranging from the trivial to the profound. Because of our propensity for error varies broadly with particular skills, particular experience and the vagaries of the moment, the discovery and elimination of software bugs produced by ordinary human error can be greatly facilitated by bringing a great many minds to bear on the problem. In addition, more individuals represent extra person hours of code and its asocciated function.

It is very clear that internet can network a vast number of people, which implies internet has a lot of energy and time which can produce much higher quality software products in much shorter time as compared to commercial companies. Even big companies like Microsoft, IBM cannot overpower and overrule the laws of Physics but will eventually **SURRENDER UNTO** laws of science!

Today, there are too many SQL RDBMS databases in the world aiming at only one specification ANSI/ISO SQL. **Man must not waste his time creating too many SIMILAR/IDENTICAL software packages** and there are many other topics in science which need good attention.

Conclusion is - because of laws of science, 'open source code' system like PostgreSQL, Linux will be always much better than 'closed source code' system and it is possible to prove this statement scientifically.

---

4. What is PostgreSQL ?

PostgreSQL Version 6.4 is a free database, complete source code is given to you and is a Object-Relational Database System near compliant (getting closer) with ANSI SQL1998,92,89 and runs on diverse hardware platforms and Operating systems.

```
Sometimes emergency bug fix patches are released after the
GA release of PostgreSQL. You can apply these optional patches
depending upon the needs of your application. Follow these
steps to apply the patches -
Change directory to postgresql source directory
        cd /usr/src/postgresql6.4
        patch -p0 < patchfile
        make clean
        make
See also manual page do 'man patch'
```

The patch files are located in

- PostgreSQL patches : ftp://ftp.postgresql.org/pub/patches

The ultimate objective and the final goal of PostgreSQL is to become 100 % compliant to ANSI/ISO SQL and also to become the number ONE open generic Database in the world. PostgreSQL will also guide, steer, control, monitor and dictate the future of ANSI/ISO SQL. That is, the implementation and ideas first take place in PostgreSQL and than later be incorporated into the ANSI/ISO SQL.

Informix Universal server (released 1997) is based on earlier version of PostgreSQL because Informix bought Illustra Inc. and integrated with Informix. Illustra database was completely based on Postgres (earlier version of PostgreSQL).

PostgreSQL is an enhancement of the POSTGRES database management system, a next-generation DBMS research prototype. While PostgreSQL retains the powerful data model and rich data types of POSTGRES, it replaces the PostQuel query language with an extended subset of SQL.

PostgreSQL development is being performed by a team of Internet developers who all subscribe to the PostgreSQL development mailing list. The current coordinator is Marc G. Fournier

- scrappy@postgreSQL.org

This team is now responsible for all current and future development of PostgreSQL. Ofcourse, the database customer himself is the developer of PostgreSQL! The development load is distributed among a very large number of database end-users on internet.

The authors of PostgreSQL 1.01 were Andrew Yu and Jolly Chen. Many others have contributed to the porting, testing, debugging and enhancement of the code. The original Postgres code, from which PostgreSQL is derived, was the effort of many graduate students, undergraduate students, and staff programmers working under the direction of Professor Michael Stonebraker at the University of California, Berkeley.

The original name of the software at Berkeley was Postgres. When SQL functionality was added in 1995, its name was changed to Postgres95. The name was changed at the end of 1996 to PostgreSQL.

Millions of PostgreSQL is installed as Database servers, Web database servers and Application data servers. It is much more advanced and is a object oriented relational database (ORDBMS).

PostgreSQL can store more data types than traditional datatypes like integer, characters, etc. - you get to create user-defined types, functions, inheritance etc. PostgreSQL runs on Solaris, SunOS, HPUX, AIX, Linux, Irix, Dig Unix, BSDi,NetBSD, FreeBSD, SCO unix, NEXTSTEP, Unixware and all and every flavor of Unix. Port to Windows 95/NT is underway.

- Title: PostgreSQL SQL RDBMS Database (Object Relational Database Management System)
- Current Version: 6.4
- Age: PostgreSQL is 12 years old. Developed since 1985
- Authors: Developed by millions/universities/companies on internet for the past 12 YEARS

PostgreSQL and related items in this document are subject to the following COPYRIGHT.


```
PostgreSQL Data Base Management System

Copyright (c) 1994-6 Regents of the University of California

Permission to use, copy, modify, and distribute this software and its
documentation for any purpose, without fee, and without a written agreement
is hereby granted, provided that the above copyright notice and this
paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR
DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING
LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS
DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON
AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO
PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.
```

5. Where to get it ?

You can buy Redhat Linux CDROM, Debian Linux CDROM or Slackware Linux CDROM which already contains the Postgresql in package form (both source code and binaries) from :

- Linux System Labs Web site: http://www.lsl.com/ 7 (U.S. dollars)
- Cheap Bytes Inc Web site: http://www.cheapbytes.com/ 7 (U.S. dollars)
- Debian Main Web site : http://www.debian.org/vendors.html

PostgreSQL organisation is also selling 'PostgreSQL CDROM' which contains the complete source code and binaries for many Unix operating systems as well as full documentation.

- PostgreSQL CDROM from main Web site at : http://www.postgresql.org 30 (U.S. dollars)

Binaries only distribution of PostgreSQL:

- You can run PostgreSQL without compiling the source. Get binaries for Intel-Linux from ftp://www.redhat.com/pub/contrib/i386/ file is postgresql-6.4.i386.rpm. This is in the redhat package 'rpm' format and it contains both source and binaries for PostgreSQL.
- Binaries site for Solaris, HPUX, AIX, IRIX, Linux : ftp://ftp.postgresql.org/pub/bindist If you compile on any platform please upload to site, so that it will be useful for others.
- ftp site : Get binaries for Intel-Linux from ftp://ftp.redhat.com/pub/contrib/i386/ file is postgresql-6.4.i386.rpm. This is in the redhat package 'rpm' format and it contains both source and binaries for PostgreSQL.

WWW Web sites:

- Primary Web site: http://www.postgresql.org/
- Secondary Web site: http://logical.thought.net/postgres95/
- http://www.itm.tu-clausthal.de/mirrors/postgres95/
- http://s2k-ftp.cs.berkeley.edu:8000/postgres95/
- http://xenium.pdi.net/PostgreSQL/
- http://s2k-ftp.cs.berkeley.edu:8000/postgres95/

The ftp sites are listed below :-

- Primary FTP: ftp://ftp.postgresql.org/pub
- Secondary FTP: ftp://ftp.chicks.net/pub/postgresql
- ftp://ftp.emsi.priv.at/pub/postgres/
- ftp://ftp.itm.tu-clausthal.de/pub/mirrors/postgres95
- ftp://rocker.sch.bme.hu/pub/mirrors/postgreSQL

- ftp://ftp.jaist.ac.jp/pub/dbms/postgres95
- ftp://ftp.luga.or.at/pub/postgres95
- ftp://postgres95.vnet.net:/pub/postgres95
- ftp://ftpza.co.za/mirrors/postgres
- ftp://sunsite.auc.dk/pub/databases/postgresql
- ftp://ftp.task.gda.pl/pub/software/postgresql
- ftp://xenium.pdi.net/pub/PostgreSQL

PostgreSQL source code is also available at all the mirror sites of sunsite unc (total of about 1000 sites around the globe). It is inside the Red Hat Linux distribution in /pub/contrib/i386/postgresql.rpm file.

- For list of mirror sites go to ftp://sunsite.unc.edu

---

6. PostgreSQL Supports Extremely Large Databases greater than 200 Gig

Performance of 32-bit cpu machines will decline rapidly when the database size exceeds 5 GigaByte. You can run 30 gig database on 32-bit cpu but the performance will be degraded. Machines with 32-bit cpu imposes a limitation of 2 GB on RAM, 2 GB on file system sizes and other limitations on the operating system.

For extremely large databases, it is strongly advised to use 64-bit machines like Digital Alpha cpu, Sun Ultra-sparc 64-bit cpu, Silicon graphics 64-bit cpu, Intel Merced IA-64 cpu, HPUX 64bit machines or IBM 64-bit machines. Compile PostgreSQL under 64-bit cpu and it can support huge databases and large queries. Performance of PostgreSQL for queries on large tables and databases will be several times faster than PostgreSQL on 32-bit cpu machines. Advantage of 64-bit machines are that you get very large memory addressing space and the operating system can support very large file-systems, provide better performance with large databases, support much larger memory (RAM), have more capabilities etc..

7. How can I trust PostgreSQL ? Regression Test Package builds customer confidence

To validate PostgreSQL, regression test package (src/test/regress) is included in the distribution. Regression test package will verify the standard SQL operations as well as the extensibility capabilities of PostgreSQL. The test package already contains hundreds of SQL test programs.

You should use the computer's high-speed power to validate the PostgreSQL, instead of using human brain power. Computers can carry out software regression tests millions or even billions of times faster than humans can. Modern computers can run billions of SQL tests in a very short time. In the near future the speed of computer will be several zillion times faster than human brain! Hence, it makes sense to use the power of computer to validate the software.

You can add more tests just in case you need to, and can upload to the primary PostgreSQL web site if you feel that it will be useful to others on internet. Regression test package helps build customer confidence and trust in PostgreSQL and facilitates rapid deployment of PostgreSQL on production systems.

Regression test package can be taken as a **"VERY SOLID"** technical document mutually agreed upon between the developers and end-users. PostgreSQL developers extensively use the regression test package during development period and also before releasing the software to public to ensure good quality.

Capablilities of PostgreSQL are directly reflected by the regression test package. If a functionality, syntax or feature exists in the regression package than it is supported, and all others which are NOT listed in the package MAY not be supported by PostgreSQL!! You may need to verify those and add it to regression package.

8. GUI FrontEnd Tool for PostgreSQL (Graphical User Interface)

PostgreSQL has Tcl/Tk interface library in the distribution called 'pgTcl'. Tcl/Tk is a Rapid Application Development tool and is a scripting language. Using Tcl/TK, you can develop applications once and run it everywhere on NT, Win 95, Linux, Apple Macintosh iMac, OS/2 and all unixes. Tcl stands for 'Tool Command Language' and Tk is 'Tool Kit'. There is a IDE (integrated development environment) for Tcl/Tk called SpecTcl. Check for this rpm package under Redhat linux distribution or at the sunscript site given below.

Perl is strongly recommended as a scripting language, since it is the most widely used and very powerful and is available on every hardware and OS platforms. Perl-Tk or Perl-Qt or C++ QtEZ or C++ Lesstiff is strongly recommended. A combination of Perl, C and Tk will satisfy the needs of most projects, considering the time resources, speed of executable and ease of programming. Note that Java programs run very slow. Even C++ programs run slower than C programs. The C programs run the fastest as compared to other languages. Perl programs can be converted to "C" program using Perl2C program and than compiled which will run fast.

Perl is used for programming 90 % of internet scripts in the world. So you will have only one language ("Perl") for all your needs - in applications as well as internet. See Perl Database Interface of this document at Perl Database Interface

You should ponder over the strong points of Perl vis-a-vis Tcl/Tk before starting any project on Tcl. Tcl/TK is usually shipped with every linux cdrom. Also you can get it from these sites -

- Tcl/Tk http://www.scriptics.com
- Object oriented extension of Tcl called INCR at http://www.tcltk.com
- Visual TCL site http://www.neuron.com
- Visual TCL Redhat rpm at ftp://ftp.redhat.com/pub/contrib/i386/visualtcl*.rpm
- http://sunscript.sun.com/
- http://sunscript.sun.com/TclTkCore/
- ftp://ftp.sunlabs.com/pub/tcl/tcl8.0a2.tar.Z
- Reference text book: Many textbooks on TCL/TK are available in the market.

9. Integrated Development Environment Tools for PostgreSQL (GUI IDE)

Check out the following development tools which you can use in conjunction with ODBC/JDBC drivers. These are similar to Borland C++ Builder, Borland JBuilder.

Vibe is a Java and C++ IDE (Integrated Development Environment) that won Unix Review's IDE of the year.

- More information on Vibe is at http://www.LinuxMall.com/products/00487.html
- 'QT' an application framework for Windows95/NT and Unix available at http://www.troll.no and ftp site is at ftp://ftp.troll.no

You can also use Borland C++ Builder, Delphi, Borland JBuilder, PowerBuilder on Windows95 connecting to PostgreSQL on unix box through ODBC/JDBC drivers.

See also **PERL** Database Interface of this document at Perl Database Interface Perl and C/C++ is strongly recommended as the choice of langauge for IDE.

Free of cost IDE Tools -

Check the CDROM of redhat linux distribution.

- Lesstiff Motif tool ftp://ftp.redhat.com/pub/contrib/i386/lesstiff*.rpm
- FreeBuilder ftp://ftp.redhat.com/pub/contrib/i386/free*.rpm
- SpecTCL ftp://ftp.redhat.com/pub/contrib/i386/spec*.rpm
- JccWarrior ftp://ftp.redhat.com/pub/contrib/i386/jcc*.rpm
- Kanchenjunga Java RAD Tool for PostgreSQL http://www.man.ac.uk/~whaley/kj/kanch.html
- Applixware Tool http://www.redhat.com
- XWPE X Windows Programming Environment http://www.rpi.edu/~payned/xwpe/ ftp://ftp.redhat.com/pub/contrib/i386/xwpe*.rpm
- XWB X Windows Work Bench ftp://ftp.redhat.com/pub/contrib/i386/xwb*.rpm
- NEdit ftp://ftp.redhat.com/pub/contrib/i386/nedit*.rpm

10. Interface Drivers for PostgreSQL

10.1 ODBC Drivers for PostgreSQL

ODBC stands for 'Open DataBase Connectivity' is a popular standard for accessing information from various databases from different vendors. Applications written using the ODBC drivers are guaranteed to work with various databases like PostgreSQL, Oracle, Sybase, Informix etc..

- PostODBC is already included in the distribution. See main web site http://www.postgresql.org. It is included on the PostgreSQL CDROM.
- http://www.openlinksw.com Open Link Software Corporation is selling ODBC for PostgreSQL and other databases. Open Link also is giving away free ODBC (limited seats) check them out.
- Insight ODBC for PostgreSQL http://www.insightdist.com/psqlodbc This is the official PostODBC site.

There is a project called the FreeODBC Pack Package . There's no PostgreSQL Version there, maybe you can help.

- http://www.ids.net/~bjepson/freeODBC/ This is a free of cost version of ODBC.

10.2 UDBC Drivers for PostgreSQL

UDBC is a static version of ODBC independent of driver managers and DLL support, used to embed database connectivity support directly into applications.

- http://www.openlinksw.com Open Link Software Corporation is selling UDBC for PostgreSQL and other databases. Open Link also is giving away free UDBC (limited seats) check them out.

10.3 JDBC Drivers for PostgreSQL

JDBC stands for 'Java DataBase Connectivity'. Java is a platform independent programming language developed by Sun Microsystems. Java programmers are encouraged to write database applications using the JDBC to facilitate portability across databases like PostgreSQL, Oracle, informix, etc. If you write Java applications you can get JDBC drivers for PostgreSQL from the following sites:

JDBC driver is already included in the PostgreSQL distribution.

- http://www.demon.co.uk/finder/postgres/index.html Sun's Java connectivity to PostgreSQL
- ftp://ftp.ai.mit.edu/people/rst/rst-jdbc.tar.gz
- http://www.openlinksw.com Open Link Software Corporation is selling JDBC for PostgreSQL and other databases. Open Link also is giving away free JDBC (limited seats) check them out.

- JDBC UK site http://www.retep.org.uk/postgres
- JDBC FAQ site http://eagle.eku.edu/tools/jdbc/faq.html

The JDBC home, guide and FAQ are located at -

- JDBC HOME http://splash.javasoft.com/jdbc
- JDBC guide http://www.javasoft.com/products/jdk/1.1/docs/guide/jdbc
- JDBC FAQ http://javanese.yoyoweb.com/JDBC/FAQ.txt

10.4 Java for PostgreSQL

Java programmers can find these for PostgreSQL very useful.

- ftp://ftp.redhat.com/pub/contrib/i386 and see postgresql-jdbc-*.rpm
- http://www.blackdown.org

---

11. Perl Database Interface (DBI) Driver for PostgreSQL

11.1 Perl 5 interface for PostgreSQL

PERL is an acronym for 'Practical Extraction and Report Language'. Perl is available on each and every operating system and hardware platform in the world. You can use Perl on Windows95/NT, Apple Macintosh iMac, all flavors of Unix (Solaris, HPUX, AIX, Linux, Irix, SCO etc..), mainframe MVS, desktop OS/2, OS/400, Amdahl UTS and many others. Perl runs **EVEN** on many unpopular or generally-unknown operating systems and hardware!! So do not be surprised if you see perl running on a very rarely used operating system. You can imagine the vast extent of the user base and developer base of Perl.

Perl interface for PostgreSQL is included in the distribution of PostgreSQL. Check in src/pgsql_perl5 directory.

- Pgsql_perl5 contact Email: E.Mergl@bawue.de
- Another source from - ftp://ftp.kciLink.com/pub/PostgresPerl-1.3.tar.gz
- Perl Home page : http://www.perl.com/perl/index.html
- Perl tutorial, look for Tutorial title at : http://reference.perl.com/
- Perl FAQ is at : http://www.yahoo.com/Computers_and_Internet/Programming_Languages/Perl/
- Perl GUI User Interfaces Perl-Qt rpm : ftp://ftp.redhat.com/pub/contrib/i386 and look for PerlQt-1.06-1.i386.rpm
- Perl GUI User Interfaces Perl-Qt : http://www.accessone.com/~jql/perlqt.html
- Perl GUI User Interfaces Perl-XForms : ftp://ftp.redhat.com/pub/contrib/i386 and look for Xforms4Perl-0.8.4-1.i386.rpm
- Perl GUI User Interfaces Perl-Tk : ftp://ftp.redhat.com/pub/contrib/i386
- Perl GUIkits : http://reference.perl.com/query.cgi?ui
- Perl Database Interfaces : http://reference.perl.com/query.cgi?database
- Perl to "C" translator : http://www.perl.com/CPAN-local/modules/by-module/B/ and look for Compiler-a3.tar.gz
- Bourne shell to Perl translator : http://www.perl.com/CPAN/authors/id/MERLYN/sh2perl-0.02.tar.gz
- awk to Perl a2p and sed to Perl s2p is included with the PERl distribution.
- See also the newsgroups for PERL at comp.lang.perl.*

11.2 Perl Database Interface DBI

WHAT IS DBI ?

The Perl Database Interface (DBI) is a database access Application Programming Interface (API) for the Perl Language. The Perl DBI API specification defines a set of functions, variables and conventions that provide a consistent database interface independent of the actual database being used. The information for this DBI

section is obtained from 'DBI FAQ' doc whose author is Alligator Descartes and reproduced here with his permission.

- Alligator Descartes Hermetica is at descarte@hermetica.com

DBI driver for PostgreSQL DBD-Pg-0.89

Get DBD-Pg-0.89.tar.gz from below

- DBD-Pg-0.89 : http://www.perl.com/CPAN/modules/by-module/DBD/
- Comprehensive Perl Archive Network CPAN http://www.perl.com/CPAN
- DBI drivers list and DBI module pages http://www.hermetica.com/technologia/perl/DBI
- DBI information is at http://www.fugue.com/dbi/
- Primary ftp site ftp://ftp.demon.co.uk/pub/perl/db
- Miscellaneous DBI link http://www-ccs.cs.umass.edu/db.html
- Miscellaneous DBI link http://www.odmg.org/odmg93/updates_dbarry.html
- Miscellaneous DBI link http://www.jcc.com/sql_stnd.html
- PostgreSQL database http://www.postgresql.org

REQUIREMENTS:

- - build, test and install Perl 5 (at least 5.002)
- - build, test and install the DBI module (at least 0.89)
- - build, test and install PostgreSQL (at least 6.2)

Technical support for DBI

Please send comments and bug-reports to

- E.Mergl@bawue.de

Please include the output of perl -v, and perl -V, the version of PostgreSQL, the version of DBD-Pg, and the version of DBI in your bug-report.

What is DBI, DBperl, Oraperl and *perl?

To quote Tim Bunce, the architect and author of DBI:

``DBI is a database access Application Programming Interface (API) for the Perl Language. The DBI API Specification defines a set of functions, variables and conventions that provide a consistent database interface independent of the actual database being used."

In simple language, the DBI interface allows users to access multiple database types transparently. So, if you connecting to an Oracle, Informix, mSQL, Sybase or whatever database, you don't need to know the underlying mechanics of the 3GL layer. The API defined by DBI will work on all these database types.

A similar benefit is gained by the ability to connect to two different databases of different vendor within the one perl script, ie, I want to read data from an Oracle database and insert it back into an Informix database all within one program. The DBI layer allows you to do this simply and powerfully.

DBperl is the old name for the interface specification. It's usually now used to denote perl4 modules on data interfacing, such as, oraperl, isqlperl, ingperl and so on. These interfaces didn't have a standard API and are generally not supported.

Here's a list of DBperl modules, their corresponding DBI counterparts and support information. DBI driver queries should be directed to the dbi-users mailing list.

Module Name	Database Required	Author	DBI
Sybperl	Sybase	Michael Peppler <mpeppler@datamig.com> http://www.mbay.net/~mpeppler	DBD::Sybase
Oraperl	Oracle 6 & 7	Kevin Stock <dbi-users@fugue.com>	DBD::Oracle
Ingperl	Ingres	Tim Bunce & Ted Lemon <dbi-users@fugue.com>	DBD::Ingres
Interperl	Interbase	Buzz Moschetti <buzz@bear.com>	DBD::Interbase
Uniperl	Unify 5.0	Rick Wargo <rickers@coe.drexel.edu>	None
Pgperl	Postgres	Igor Metz <metz@iam.unibe.ch>	DBD::Pg
Btreeperl	NDBM	John Conover <john@johncon.com>	SDBM?
Ctreeperl	C-Tree	John Conover <john@johncon.com>	None
Cisamperl	Informix C-ISAM	Mathias Koerber <mathias@unicorn.swi.com.sg>	None
Duaperl	X.500 Directory User Agent	Eric Douglas	None

However, some DBI modules have DBperl emulation layers, so, DBD::Oracle comes with an Oraperl emulation layer, which allows you to run legacy oraperl scripts without modification. The emulation layer translates the oraperl API calls into DBI calls and executes them through the DBI switch.

Here's a table of emulation layer information:

Module	Emulation Layer	Status
DBD::Oracle	Oraperl	Complete
DBD::Informix	Isqlperl	Under development
DBD::Sybase	Sybperl	Working? (Needs verification)
DBD::mSQL	Msqlperl	Experimentally released with DBD::mSQL-0.61

The Msqlperl emulation is a special case. Msqlperl is a perl5 driver for mSQL databases, but does not conform to the DBI Specification. It's use is being deprecated in favour of DBD::mSQL. Msqlperl may be downloaded from CPAN via:

- http://www.perl.com/cgi-bin/cpan_mod?module=Msqlperl

DBI specifications

There are a few information sources on DBI.

- DBI Specification http://www.hermetica.com/technologia/perl/DBI/doc/dbispec

There are two specifications available at this link, the new DBI Draft Specification which is a rapidly changing document as the development team drive towards a stable interface, and the old historical DBperl Specification out of which the current DBI interface evolved.

The later document should be regarded as being of historical interest only and should not serve as a programming manual, or authoratative in any sense. However, it is still a very useful reference source.

POD documentation PODs are chunks of documentation usually embedded within perl programs that document the code ``in place'', providing a useful resource for programmers and users of modules. POD for DBI and drivers is beginning to become more commonplace, and documentation for these modules can be read with the following commands.

The DBI Specification The POD for the DBI Specification can be read with the command

perldoc DBI

Oraperl Users of the Oraperl emulation layer bundled with DBD::Oracle, may read up on how to program with the Oraperl interface by typing:

perldoc Oraperl

This will produce an updated copy of the original oraperl man page written by Kevin Stock for perl4. The oraperl API is fully listed and described there.

DBD::mSQL Users of the DBD::mSQL module may read about some of the private functions and quirks of that driver by typing:

perldoc DBD::mSQL

Frequently Asked Questions The Frequently Asked Questions is also available as POD documentation. Read this by typing:

perldoc DBI::FAQ

This may be more convenient to people not permanently, or conveniently, connected to the Internet.

POD in general Information on writing POD, and on the philosophy of POD in general, can be read by typing:

perldoc perlpod

Users with the Tk module installed may be interested to learn there is a Tk-based POD reader available called tkpod, which formats POD in a convenient and readable way.

Rambles, Tidbits and Observations : There are a series of occasional rambles from various people on the DBI mailing lists.

- http://www.hermetica.com/technologia/perl/DBI/tidbits

``DBI -- The perl5 Database Interface'' This is an article written by Alligator Descartes and Tim Bunce on the structure of DBI. It was published in issue 5 of ``The Perl Journal''. It's extremely good. Go buy the magazine. In

fact, buy all of them. ``The Perl Journal''s WWW site is:

- http://www.tpj.com

``DBperl'' This article, published in the November 1996 edition of ``Dr. Dobbs Journal'' concerned DBperl.

``The Perl5 Database Interface'' This item is a book to be written by Alligator Descartes and published by O'Reilly and Associates.

Mailing Lists There are three mailing lists for DBI run by Ted Lemon. These can all be subscribed to and unsubscribed from via the World Wide Web at

- Mailing lists http://www.fugue.com/dbi

The lists that users may participate in are:

dbi-announce This mailing list is for announcements only. If you cannot successfully use the form on the above WWW page, please subscribe to the list in the following manner:

- Email: dbi-announce-request@fugue.com with a message body of 'subscribe'

dbi-dev This mailing list is intended for the use of developers discussing ideas and concepts for the DBI interface, API and driver mechanics. Only any use for developers, or interested parties. If you cannot successfully use the form on the above WWW page, please subscribe to the list in the following manner:

- Email: dbi-dev-request@fugue.com with a message body of 'subscribe'

dbi-users This mailing list is a general discussion list used for bug reporting, problem discussion and general enquiries. If you cannot successfully use the form on the above WWW page, please subscribe to the list in the following manner:

- Email: dbi-users-request@fugue.com with a message body of 'subscribe'

Mailing List Archives

- US Mailing List Archives http://outside.organic.com/mail-archives/dbi-users/
- European Mailing List Archives http://www.rosat.mpe-garching.mpg.de/mailing-lists/PerlDB-Interest

Compilation problems or "It fails the test"

If you have a core dump, try the Devel::CoreStack module for generating a stack trace from the core dump. Devel::CoreStack can be found on CPAN at:

- http://www.perl.com/cgi-bin/cpan_mod?module=Devel::CoreStack

Email the dbi-users Mailing List stack trace, module versions, perl version, test cases, operating system versions and any other pertinent information. The more information you send, the quicker developers can track problems down. If you send us nothing, expect nothing back.

Is DBI supported under Windows 95 / NT platforms?

The DBI and DBD::Oracle Win32 ports are now a standard part of DBI, so, downloading DBI of version higher than 0.81 should work fine. You can access Microsoft Access and SQL-Server databases from DBI via ODBC. Supplied with DBI-0.79 (and later) is an experimental DBI 'emulation layer' for the Win32::ODBC module. It's called DBI::W32ODBC. You will need the Win32::ODBC module.

- Win32 DBI http://www.hermetica.com/technologia/perl/DBI/win32

- Win32 ODBC http://www.roth.net

What's DBM? And why use DBI instead ?

UNIX was originally blessed with simple file-based ``databases", namely the dbm system. dbm lets you store data in files, and retrieve that data quickly. However, it also has serious drawbacks.

File Locking

The dbm systems did not allow particularly robust file locking capabilities, nor any capability for correcting problems arising through simultaneous writes [to the database].

Arbitrary Data Structures

The dbm systems only allows a single fixed data structure: key-value pairs. That value could be a complex object, such as a [C] struct, but the key had to be unique. This was a large limitation on the usefulness of dbm systems.

However, dbm systems still provide a useful function for users with simple datasets and limited resources, since they are fast, robust and extremely well-tested. Perl modules to access dbm systems have now been integrated into the core Perl distribution via the AnyDBM_File module.

To sum up, DBM is a perfectly satisfactory solution for essentially read-only databases, or small and simple datasets. However, for more powerful and scaleable datasets, not to mention robust transactional locking, users are recommended to use DBI.

Is < insert feature here > supported in DBI?

Given that we're making the assumption that the feature you have requested is a non-standard database-specific feature, then the answer will be no.

DBI reflects a generic API that will work for most databases, and has no database-specific functionality.

However, driver authors may, if they so desire, include hooks to database-specific functionality through the func method defined in the DBI API. Script developers should note that use of functionality provided via the func methods is unlikely to be portable across databases.

Is DBI any use for CGI programming?

In a word, yes! DBI is hugely useful for CGI programming! In fact, CGI programming is one of two top uses for DBI.

DBI confers the ability to CGI programmers to power WWW-fronted databases to their users, which provides users with vast quantities of ordered data to play with. DBI also provides the possibility that, if a site is receiving far too much traffic than their database server can cope with, they can upgrade the database server behind the scenes with no alterations to the CGI scripts.

How do I get faster connection times with DBD Oracle and CGI?

The Apache httpd maintains a pool of httpd children to service client requests. Using the Apache mod_perl module by Doug MacEachern, the perl interpreter is embedded with the httpd children. The CGI, DBI, and your other favorite modules can be loaded at the startup of each child. These modules will not be reloaded unless

changed on disk. For more information on Apache, see the Apache Project's WWW site:

- Apache Project WWW site http://www.apache.org
- Mod_perl module http://www.perl.com/cgi-bin/cpan_mod?module=mod_perl

How do I get persistent connections with DBI and CGI?

Using Edmund Mergl's Apache::DBI module, database logins are stored in a hash with each of these httpd child. If your application is based on a single database user, this connection can be started with each child. Currently, database connections cannot be shared between httpd children. Apache::DBI can be downloaded from CPAN via:

- http://www.perl.com/cgi-bin/cpan_mod?module=Apache::DBI

``When I run a perl script from the command line, it works, but, when I run it under the httpd, it fails!'' Why?

Basically, a good chance this is occurring is due to the fact that the user that you ran it from the command line as has a correctly configured set of environment variables, in the case of DBD::Oracle, variables like $ORACLE_HOME, $ORACLE_SID or TWO_TASK. The httpd process usually runs under the user id of nobody, which implies there is no configured environment. Any scripts attempting to execute in this situation will correctly fail. To solve this problem, set the environment for your database in a BEGIN ( ) block at the top of your script. This will solve the problem. Similarly, you should check your httpd error logfile for any clues, as well as the ``Idiot's Guide To Solving Perl / CGI Problems'' and ``Perl CGI Programming FAQ'' for further information. It is unlikely the problem is DBI-related. Read BOTH these documents carefully!

- Idiot's Guide to Solving Perl / CGI problems http://www.perl.com/perl/faq/index.html

Can I do multi-threading with DBI?

As of the current date, no. Perl does not support multi-threading. However, multi-threading is expected to become part of the perl core distribution as of version 5.005, which implies that DBI may support multi-threading fairly soon afterwards. For some OCI example code for Oracle that has multi-threaded SELECT statements, see:

- http://www.hermetica.com/technologia/oracle/oci/orathreads.tar.gz

How can I invoke stored procedures with DBI?

Assuming that you have created a stored procedure within the target database, eg, an Oracle database, you can use $dbh->do to immediately execute the procedure. For example,

$dbh->do("BEGIN someProcedure END");

How can I get return values from stored procedures with DBI?

Remember to perform error checking, though!

```
$sth = $dbh->prepare( "BEGIN foo(:1, :2, :3); END;" );
$sth->bind_param(1, $a);
$sth->bind_param_inout(2, \$path, 2000);
```

```
$sth->bind_param_inout(3, \$success, 2000);
$sth->execute;
```

How can I create or drop a database with DBI?

Database creation and deletion are concepts that are entirely too abstract to be adequately supported by DBI. For example, Oracle does not support the concept of dropping a database at all! Also, in Oracle, the database server essentially is the database, whereas in mSQL, the server process runs happily without any databases created in it. The problem is too disparate to attack. Some drivers, therefore, support database creation and deletion through the private func methods. You should check the documentation for the drivers you are using to see if they support this mechanism.

How are NULL values handled by DBI?

NULL values in DBI are specified to be treated as the value undef. NULLs can be inserted into databases as NULL, for example:

```
$rv = $dbh->do( "INSERT INTO table VALUES( NULL )" );
```

but when queried back, the NULLs should be tested against undef. This is standard across all drivers.

What are these func methods all about?

The func method is defined within DBI as being an entry point for database-specific functionality, eg, the ability to create or drop databases. Invoking these driver-specific methods is simple, for example, to invoke a createDatabase method that has one argument, we would write:

```
$rv = $dbh->func( 'argument', 'createDatabase' );
```

Software developers should note that the func methods are non-portable between databases.

Commercial Support and Training

The Perl5 Database Interface is FREE software. IT COMES WITHOUT WARRANTY OF ANY KIND. However, some organizations are providing either technical support or training programs on DBI.

PERL CLINIC : The Perl Clinic can arrange commercial support contracts for Perl, DBI, DBD::Oracle and Oraperl. Support is provided by the company with whom Tim Bunce, author of DBI, works. For more information on their services, please see :

- http://www.perl.co.uk/tpc

---

12. PostgreSQL Management Tools

12.1 PGACCESS - A GUI Tool for PostgreSQL Management

PgAccess is a Tcl/Tk interface to PostgreSQL. It is already included in the distribution of PostgreSQL. You may want to check out this web site for a newer copy

- http://www.flex.ro/pgaccess
- If you have any comment, suggestion for improvements, please feel free to e-mail to : teo@flex.ro

Features of PgAccess

PgAccess windows - Main window, Table builder, Table(query) view, Visual query builder.

Tables

- opening tables for viewing, max 200 records (changed by preferences menu)
- column resizing, dragging the vertical grid line (better in table space rather than in the table header)
- text wrap in cells - layout saved for every table
- import/export to external files (SDF,CSV)
- filter capabilities (enter filter like (price>3.14)
- sort order capabilities (enter manually the sort field(s))
- editing in place
- improved table generator assistant
- improved field editing

Queries

- define , edit and stores "user defined queries"
- store queries as views
- execution of queries
- viewing of select type queries result
- query deleting and renaming
- Visual query builder with drag & drop capabilities. For any of you who had installed the Tcl/Tk plugin for Netscape Navigator, you can see it at work clicking here

Sequences

- defines sequences, delete them and inspect them Functions

- define, inspect and delete functions in SQL language

Future implementation will have

- table design (add new fields, renaming, etc.)
- function definition
- report generator
- basic scripting

If you have any comment, suggestion for improvements e-mail to :

- teo@flex.ro

Information about libgtcl

You will need the PostgreSQL to Tcl interface library libgtcl, lined as a Tcl/Tk 'load'-able module. The libpgtcl and the source is located in the PostgreSQL directory /src/interfaces/libpgtcl. Specifically, you will need a libpgtcl library that is 'load'-able from Tcl/Tk. This is technically different from an ordinary PostgreSQL loadable object file, because libpgtcl is a collection of object files. Under Linux, this is called libpgtcl.so. You can download from the above site a version already compiled for Linux i386 systems. Just copy libpgtcl.so into your system library director (/usr/lib). One of the solutions is to remove from the source the line containing load libpgtcl.so and to load pgaccess.tcl not with wish, but with pgwish (or wishpg) that wish that was linked with libpgtcl library.

12.2 Windows Interactive Query Tool for PostgreSQL (WISQL or MPSQL)

MPSQL provides users with a graphical SQL interface to PostgresSQL. MPSQL is similar to Oracle's SQL Worksheet and Microsoft SQL Server's query tool WISQL. It has nice GUI and has history of commands. Also you can cut and paste and it has other nice features to improve your productivity.

- http://www.troubador.com/~keidav/index.html
- Email: keidav@whidbey.com
- http://www.ucolick.org/~de/ in file tcl_syb/wisql.html
- http://www.troubador.com/~keidav/index.html
- Email: de@ucolick.org

12.3 Interactive Query Tool (ISQL) for PostgreSQL called PSQL

ISQL is For Character command line terminals. This is included in the distribution, and is called PSQL. Very similar to Sybase ISQL, Oracle SQLplus. At unix prompt give command 'psql' which will put you in psql> prompt.

```
Type \h to see help of commands.
```

Very user friendly and easy to use. Can also be accessed from shell scripts.

12.4 MPMGR - A Database Management Tool for PostgresSQL

MPMGR will provide a graphical management interface for PostgresSQL. You can find it at

- http://www.mutinybaysoftware.com/
- Email: keidav@mutinybaysoftware.com
- http://www.troubador.com/~keidav/index.html
- Email: keidav@whidbey.com
- http://www.ucolick.org/~de in file tcl_syb/wisql.html
- WISQL for PostgreSQL http://www.ucolick.org/~de/Tcl/pictures
- Email: de@ucolick.org

---

13. Setting up multi-boxes PostgreSQL with just one monitor

You can stack up the cpu-boxes and connect to just one monitor and use the KVM (Keyboard, Video, Monitor) switch box to select the host. This saves space and you eliminate monitor, keyboard and the mouse (saving anywhere from 100 to 500 US dollars per set) and also avoid lot of clutter.

Using this switch box, you can stack up many PostgreSQL servers (development, test, production), Web servers, ftp servers, Intranet servers, Mail servers, News servers in a tower shelf. The switch box can be used for controlling Windows 95/NT or OS/2 boxes as well.

Please check out these sites:

- DataComm Warehouse Inc Call 24 hours a day, 7 days a week at 1-800-328-2261. They supply all varieties of computer hardware http://www.warehouse.com 4-port Manual KVM switch (PS/2) is about $89.99 Part No. DDS1354
- Network Technologies Inc http://www.networktechinc.com/servswt.html (120 dollars/PC 8 ports) which lists 'Server Switches' and 'Video only switches'
- Scene Double Inc, England http://www.scene.demon.co.uk/qswitch.htm
- Cybex corporation http://www.cybex.com
- Raritan Inc http://www.raritan.com
- RealStar Solutions Inc http://www.real-star.com/kvm.htm
- Belkin Inc http://www.belkin.com
- Better Box Communications Ltd. http://www.betterbox.com/info.html
- Go to nearest hardware store and ask for "Server Switch" also known as "KVM Auto Switches".

Search engine yahoo to find more companies with "Server Switches" or "KVM Switches".

It is strongly recommended to have a dedicated unix box for each PostgreSQL data-server for better performance. No other application program/processes should run on this box. See the Business section of your local newspapers for local vendors selling only intel box, 13" monochrome monitor (very low cost monitor). Local vendors sell just the hardware **without** any Microsoft Windows/DOS (saves you about $ 150). You do not need a color monitor for the database server, as you can do remote administration from color PC workstation. Get RedHat (or some other distribution of) Linux cdrom from below -

- Linux System Labs Web site: http://www.lsl.com/ 7 (U.S. dollars)
- Cheap Bytes Inc Web site: http://www.cheapbytes.com/ 7 (U.S. dollars)

Make sure that the hardware you purchase is supported by Redhat Linux. Check the ftp site of Redhat for recommended hardware like SCSI adapters, video cards before buying. For just $ 600 you will get a powerful intel box with Redhat Linux running PostgreSQL. Use odbc/jdbc/perl/tcl to connect to

PostgreSQL from Windows95, OS/2, Unix Motif or web browser (e.g. Redbaron, Opera, Netscape, Z others). (Web browsers are very fast becoming the standard client).

Using this idea you can control many cpu boxes by just one monitor and one keyboard!

Below is the **extract from networktechnic Inc** for the "Server Switches"

Order now call 800-742-8324 (toll free in USA)

To receive our catalog please Email your address to: sales@networktechinc.com

CONTROL MULTIPLE PC'S with one keyboard monitor & mouse These electronic switches will allow one keyboard, monitor & mouse to control up to 64 PC's. Embedded microcomputers simulate the presence of keyboard, monitor & mouse to all attached PC's 100% of the time.

Features and Applications

- Keyboard, front panel or remote control
- 9 pin D Serial or 6 pin miniDIN mouse support
- 5 pin DIN or 6 pin miniDIN keyboard support
- 1600x1200 video resolution with no degradation
- Uses standard cables
- Desktop or rackmount

Specifications - Keyboard

- All connectors are female
- 5 pin DIN or 6 pin miniDIN
- Will hard or soft boot all PC's 100% of the time

Controls - Buttons on Front

- Touch a button and be connected to that PC
- Hold any button in for more than 0.5 second and go into SCAN, BROADCAST or COMMAND mode
- LEDs on front indicate mode of operation Keyboard
- Type CTRL+* and go into COMMAND mode
- SCAN, BROADCAST or COMMAND are available

Mouse

* 9 pin D serial

- o NTI switch emulates Microsoft serial mouse to all PC's
- o 9 pin D male for mouse
- o 9 pin D female for PC's

* 6 pin miniDIN

- o NTI switch emulates IBM PS/2 style mouse to all PC's

- o All connectors are female

Wired remote

- Optional--must be purchased
- Operates same as "Buttons on Front"

Monitor-VGA

- Bandwidth is 150 MHz
- 1600X1200 resolution with no degradation
- All connectors are female

---

14. Applications and Tools for PostgreSQL

14.1 PostgreSQL 4GL for web database applications - AppGEN Development System

AppGEN can be downloaded from

- http://www.man.ac.uk/~whaley/ag/appgen.html
- ftp://ftp.mcc.ac.uk/pub/linux/ALPHA/AppGEN.

AppGEN is a high level fourth generation language and application generator for producing World Wide Web (WWW) based applications. These applications are typically used over the internet or within a corporate intranet. AppGEN applications are implemented as C scripts conforming to the Common Gateway Interface (CGI) standard supported by most Web Servers.

To use AppGEN you will need the following :-

PostgresSQL, relational database management system

A CGI compatible web server such as NCSA's HTTPD

An ansi C compiler such as GCC

AppGEN consists of the following Unix (Linux) executables :-

- defgen, which produces a basic template application from a logical data structure. The applications are capable of adding, updating, deleting and searching for records within the database whilst automatically maintaining referential integrity.
- appgen, the AppGEN compiler which compiles the appgen source code into CGI executable C source and HTML formatted documents ready for deployment on a web server.
- dbf2sql, a utility fo converting dBase III compatible .dbf files into executable SQL scripts. This enables data stored in most DOS/Windows based database packages to be migrated to a SQL server such as PostgresSQL.
- In addition, AppGEN comprises of a collection of HTML documents, GIF files and Java applets which are used at runtime by the system. And of course, like all good software, the full source code is included.

The author, Andrew Whaley, can be contacted on

- andrew@arthur.smuht.nwest.nhs.uk

14.2 WWW Web interface for PostgresSQL - DBENGINE

dbengine a plug 'n play Web interface for PostgreSQL created by Ingo Ciechowski. It is at

- http://www.cis-computer.com/dbengine/

About DBENGINE : dbengine is an interface between the WWW and Postgres95 which provides simple access to any existing database within just a few minutes.

PHP 3 gives you a Perl like language in your documents, but no real Perl while AppGen and wdb-p95 require that you create some configuration file for each of your databases -- sound's like you'll first of all have to learn some sort of new mata language before you can get started.

Unlike other tools you don't have to learn any special programming or scripting language to get started with dbengine. Also there's no configuration file for each database, so you don't have to get familiar with such a new structure. However - in case you want to gain access to the full features of dbengine it'd be a good idea to know the Perl language.

The whole system can be configured by simple manipulations of an additional database that contains closer information about how to visualize your database access. You can even specify virtual Fields which are calculated on the fly right before they're displayed on the screen.

License : dbengine is free software under the same terms as Perl. Read its licence if you aren't sure what you can or can't do. The bottom line is that this is a kinder and gentler version of the GNU licence -- one that doesn't infect your work if you care to borrow from dbengine or package up pieces of it as part of a commercial product.

14.3 Apache Webserver Module for PostgreSQL - NeoSoft NeoWebScript

Apache is a well-known Web Server. And a module to interface PostgreSQL to Apache Webserver is at -

- http://www.neosoft.com/neowebscript/

NeoWebScript is a programming language that allows both simple and complex programs to be embedded into HTML files.

When an HTML page containing embedded NeoWebScript is requested, the NeoWebScript-enabled webserver executes the embedded script(s), producing a webpage containing customized content created by the program.

NeoWebScript is a fast, secure, easy to learn way to do powerful, server-based interactive programming directly in the HTML code in web pages. With NeoWebScript, counters, email forms, graffiti walls, guest books and visitor tracking are all easy, even for a beginning programmer. See how well NeoWebScript holds its' own vs. PERL and JavaScript.

If you'd like to install NeoWebScript on your webserver, your Webmaster needs to read our Sysop FAQ

to get started. Theory of Operations will explain how NeoWebScript works, while installation will ta them through the steps. Management deals with configuration issues and running the server, tests let verify correct NeoWebScript operation, and troubleshooting deals with server problems.

There is no cost to you to use NeoWebScript-2.2 for your ISP, your intranet, or your extranet. You'll see a full license when you register to download, but it costs $ 99 if you want to embed it in your own product or use it in a commerce (eg. SSL) server.

NeoWebScript is a module for the Apache webserver that allows you to embed the Tcl/Tk programming language in your webpages as a scripting tool. It was invented by Karl Lehenbauer, NeoSoft's Chief Technical Officer, and documented, enhanced and extended by NeoSoft's programmers and technical writers.

The Apache webserver is the world's most popular webserver, accounting for 68 % of the sites polled.

Tcl/Tk is the powerful, free, cross-platform scripting language developed by Dr. John Ousterhout. In his own words

"Tcl/Tk lets software developers get the job done ten times faster than with toolkits based on C or C++. It's also a great glue language for making existing applications work together and making them more graphical and Internet-aware."

Karl Lehenbauer, Founder and Chief Technical Officer of NeoSoft, has been part of Tcl/Tk development from the very beginning. Together with Mark Diehkans, they authored Extended Tcl, also known as TclX or NeoSoft Tcl, a powerful set of extensions to the language. Many of the current core Tcl commands originated in Extended Tcl, and were then imported into the core language by Dr. Ousterhout.

NeoSoft Inc., 1770 St. James Place, Suite 500, Houston, TX 77056 USA

14.4 HEITML server side extension of HTML and a 4GL language for PostgreSQL

Tool heitml is another way to interface postgres with the world wide web. For more details contact

```
Helmut Emmelmann H.E.I. Informationssyteme GmbH
Wimpfenerstrasse 23 Tel. 49-621-795141
68259 Mannheim Germany Fax. 49-621-795161
```

- E-mail Mr.Helmut Emmelmann at emmel@h-e-i.de
- Heitml main web site http://www.heitml.com
- Heitml secondary web site http://www.h-e-i.deom

heitml is a server side extension of HTML and a 4GL language at the same time. People can write web applications in the HTML style by using new HTML-like tags.

heitml (pronounced "Hi"-TML) is an extension of HTML and a full-featured 4th generation language that enables Web-based Applications to interact with data stored in SQL databases, without resorting to

complex CGI scripts.

heitml extends HTML on the sever side, dynamically converting ".hei" files to HTML format and so is compatible with any web browser.It embraces the familiar, easy-to-use HTML syntax and provides a large assortment of pre-developed Tags and Libraries to take care of tasks that formerly required CGI. As XML, heitml provides user defined tags. With heitml the user defined markup can be translated to HTML and send to a browser.

heitml targets both HTML designers and professional programmers alike. HTML designers can use heitml Tags to build dynamic web pages, access SQL databases, or create complete web applications. Counters, registration databases, search forms, email forms, or hierarchical menues can all be created simply by using the pre-developed HTML-like Tags found in the many Component Libraries.

For programmers heitml embeds a complete forth generation language in HTML

```
(e.g. <if>, <while>, and <let> Tags),
```

plus powerful expression evaluation with integer, real, boolean, string, and tuple data types. Tuples have reference semantics as in modern object oriented languages and are stored on a heap. heitml variables including all complex data structures stored on the heap maintain their values between pages using the Session Mode. It is possible to define your own tags or environment tags and even re-define HTML-tags.

heitml makes it possible to

- - - develop Web Sites in a structured and modular way, drastically reducing maintenance overhead.

- - - develop intelligent and interactive Web Sites, with content that dynamically adapts itself to user needs.

- - - show the content of SQL databases with no programming other than to use our library of prefined "dba" Tags.

- - - develop complex database and Catalog Shopping applications using Session Variables

heitml runs on Linux with any Web Server using the CGI interface, and is especially fast (avoiding the CGI overhead) within the APACHE Web Server using the apache API. Currently MSQL (Version 1 and 2), PostgreSQL (Version 6), mysql, and the yard databases are supported). heitml also works on Linux, BSDi, Solaris and SunOS, as well as Windows NT with CGI and ISAPI and ODBC and Windows 95.

heitml (on linux) is free for research, non-commercial and private usage. Commercial Web Sites must pay a licensing fee. The fully operational version of heitml is available for a trial period downloaded freely. (Note, however, that each ".hei" Web Page you develop will display a message identifying it as the version for non-commercial use. After registration, you will receive a key to switch off the message without having to re-install the program.)

heitml (pronounced "Hi"-TML) significantly extends and enhances the functionality of HTML by definable tags and full programming features. This makes dynamic content and database applications possible simply within the HTML world, without CGI and without external scripting or programming languages. This means you, as an HTML author, can embed applications in your web pages, simply by using some new tags without CGI and without programming. As an advanced user or programmer on the

other hand you can create and program powerful tag libraries. This approach makes heitml suitable f HTML newcomers and professional programmers alike. heitml runs on the web server and dynamica generates HTML, so heitml is compatible with the internet standards and with any web browser. It allows full access to databases while shielding the user from any unneccessary CGI complexity. heitml has been developed according to the newst research and in compiler construction and transaction systems.

heitml pages are developed just the same way as HTML pages, with a text editor or HTML editor, and placed on the web server as usual. However now pages can contain dynamic heitml tags and access tag libraries. You can use these tags to access the database, to create dynamic content, to send emails, and even to create powerful applications like registration databases and shopping systems.

HTML newcomers and professional programmers alike will be amazed at how quickly and easily they can design exciting applications like our Interactive Guestbook without resorting to complex and difficult to learn CGI scripts, simply by using the tools provided in our dba Library.

heitml is accompanied by a wide range of tag libraries, to create guestbooks, database maintenance applications, extensible query forms, powerful email forms or structure your web site using a hierarchic menu. These tools are ready to go, just add the corresponding tags to your web site.

As an experienced programmer you can make fully use of the heitml persistent dynamic tuple architecture : heitml is not just a scripting language with dynamic typing, full power expression evaluation, recursive procedures and extensive parameter passing features, but it also features persistent dynamic tuples to automatically keep session data of any size.

14.5 America On-line AOL Web server for PostgreSQL

The no-cost commercial webserver, AOLserver version 2.3 supports database connections to PostgreSQL 6.2.1 and higher. for more info see

- AOL Web Server http://www.aolserver.com

14.6 Problem/Project Tracking System Application Tool for PostgreSQL

This is at

- http://www.homeport.org/~shevett/pts/

14.7 Convert dbase dbf files to PostgreSQL

The program dbf2msql works fine with mSQL and PostgreSQL. You can find it at

- ftp://ftp.nerosworld.com/pub/SQL/dbf2sql/
- ftp://ftp.postgresql.org/pub/incoming/dbf2pg-3.0.tar.gz

This program was written by Maarten Boekhold, Faculty of Electrical Engineering TU Delft, NL Computer Architecture and Digital Technique section

- M.Boekhold@et.tudelft.nl

You can also use a python method to read dbf files and load into a postgres database.

- See http://www.python.org

---

15. Web Database Design/Implementation tool for PostgreSQL - EARP

- http://www.oswego.edu/Earp
- ftp://ftp.oswego.edu in the directory 'pub/unix/earp'.

15.1 What is EARP ?

The "Easily Adjustable Response Program" (EARP) created by David Dougherty. EARP is a Web Database Design/Implementation tool, built on top of the PostgreSQL database system. Its functionality includes:

- A Visual Design System.
- A sendmail interface. (can handle incoming and outgoing mail)
- An Enhanced Security Mechanism.
- A cgi driver.

15.2 Implementation

The main implementation of EARP is a CGI binary which runs under the http daemon to provide access to the database server. All of the design tools are built into the driver, no design takes place over anything but the web. The tools themselves require a graphical browser, the compatibility of objects designed with the tools is implementation independent, based on designing individuals preferences.

15.3 What you need to run EARP

EARP will likely run on a variety of platforms with little or no porting. The known working platforms consist of the following:

- Solaris 2.5
- Linux 1.2.13+
- GNU C++
- PostgreSQL (Version 1.01 / 1.02)
- netsite server
- NCSA httpd
- GNU C++
- PostgreSQL (Version 1.01 / 1.02)

- NCSA httpd
- Apache httpd

The current (1.3) release of Earp was designed on top of the libpq release that came with PostgreSQL v1.01/1.02. If you are using a more recent version of Postgres, expect that the program will require some porting to work correctly. In the development version (Earp 2.0), libpq support is being incorporated as a module.

15.4 How does it work ?

One of the main features of EARP is that it uses an Object Oriented approach to producing html pages which interface to the database. Most pages will consist of several objects. Each object is produced by some sort of tool and given a name, objects are then linked together in a callable sequence by the page tool. Objects are also reusable across multiple pages. Basic tools exist for HTML, Querys, Grabbing input from forms, Extendable Formatting of Query and Input objects, and Linking together of objects into other objects. More advanced tools include the mail tool and the multithreaded query tool.

Another feature of EARP is advanced security. Access to various areas of the EARP system can be limited in a variety of ways. To facilitate its advanced security, EARP performs checks for each connection to the system, determining what ids and groups the connecting agent belongs to. Access to areas is defined seperately, and the combination decides if access to a specific area of Earp is allowed. Moreover, all that is required to implement the security features is an http server that supports basic (or better) user authentication.

15.5 Where to get EARP ?

EARP is available via anonymous ftp from

- ftp://ftp.oswego.edu in the directory 'pub/unix/earp'.

---

16. PHP Hypertext Preprocessor - Server-side html-embedded scripting language for PostgreSQL

WWW Interface Tool is at -

- http://www.php.net
- http://www.vex.net/php

Old name is Professional Home Pages (PHP 3) and new name is PHP Hypertext Pre-Processor

- Mirror sites are in many countries like www.COUNTRYCODE.php.net
- http://www.fe.de.php.net
- http://www.sk.php.net
- http://php.iquest.net/

Questions e-mail to :

- rasmus@lerdorf.on.ca

PHP 3 is a server-side html-embedded scripting language. It lets you write simple scripts right in your .HTML files much like JavaScript does, except, unlike JavaScript PHP 3 is not browser-dependant. JavaScript is a client-side html-embedded language while PHP 3 is a server-side language. PHP 3 is similar in concept to Netscape's LiveWire Pro product. If you have the money, you run Netscape's Commerce Server and you run one of the supported operating systems, you should probably have a look at LiveWire Pro. If you like free fast-moving software that comes with full source code you will probably like PHP 3.

16.1 Major Features

Standard CGI, FastCGI and Apache module Support As a standard CGI program, PHP 3 can be installed on any Unix machine running any Unix web server. With support for the new FastCGI standard, PHP 3 can take advantage of the speed improvements gained through this mechanism. As an Apache module, PHP 3 becomes an extremely powerful and lightning fast alternative to CGI programmimg.

- Access Logging With the access logging capabilities of PHP 3, users can maintain their own hit counting and logging. It does not use the system's central access log files in any way, and it provides real-time access monitoring. The Log Viewer Script provides a quick summary of the accesses to a set of pages owned by an individual user. In addition to that, the package can be configured to generate a footer on every page which shows access information. See the bottom of this page for an example of this.
- Access Control A built-in web-based configuration screen handles access control configuration. It is possible to create rules for all or some web pages owned by a certain person which place various restrictions on who can view these pages and how they will be viewed. Pages can be password

protected, completely restricted, logging disabled and more based on the client's domain, brow e-mail address or even the referring document.

- Postgres Support Postgres is an advanced free RDBMS. PHP 3 supports embedding Postgres95 and PostgreSQL SQL queries directly in .html files.
- RFC-1867 File Upload Support File Upload is a new feature in Netscape 2.0. It lets users upload files to a web server. PHP 3 provides the actual Mime decoding to make this work and also provides the additional framework to do something useful with the uploaded file once it has been received.
- HTTP-based authentication control PHP 3 can be used to create customized HTTP-based authentication mechanisms for the Apache web server.
- Variables, Arrays, Associative Arrays PHP 3 supports typed variables, arrays and even Perl-like associative arrays. These can all be passed from one web page to another using either GET or POST method forms.
- Conditionals, While Loops PHP 3 supports a full-featured C-like scripting language. You can have if/then/elseif/else/endif conditions as well as while loops and switch/case statements to guide the logical flow of how the html page should be displayed.
- Extended Regular Expressions Regular expressions are heavily used for pattern matching, pattern substitutions and general string manipulation. PHP 3 supports all common regular expression operations.
- Raw HTTP Header Control The ability to have web pages send customized raw HTTP headers based on some condition is essential for high-level web site design. A frequent use is to send a Location: URL header to redirect the calling client to some other URL. It can also be used to turn off cacheing or manipulate the last update header of pages.
- On-the-fly GIF image creation PHP 3 has support for Thomas Boutell's GD image library which makes it possible to generate GIF images on the fly.
- ISP "Safe Mode" support PHP 3 supports a unique "Safe Mode" which makes it safe to have multiple users run PHP scripts on the same server.
- It's Free! One final essential feature. The package is completely free. It is licensed under the GPL which allows you to use the software for any purpose, commercial or otherwise. See the GNU Public License document for complete details.

16.2 Credits

* Large parts of this code were developed at and for the University of Toronto. Many thanks to Lee Oattes of the Network Development Department at the university for constant constructive criticism.

* The PostgreSQL support code was written by Adam Sussman

- asussman@vidya.com

* Countless others have helped test and debug the package.

16.3 PHP 3 - Brief History

PHP began life as a simple little cgi wrapper written in Perl. It was never intended to go beyond own private use. The name of this first package was Personal Home Page Tools, which later became Personal Home Page Construction Kit.

A tool was written to easily embed SQL queries into web pages. It was basically another CGI wrapper that parsed SQL queries and made it easy to create forms and tables based on these queries. This tool was named FI (Form Interpreter).

PHP/FI version 2.0 is a complete rewrite of these two packages combined into a single program. It has now evolved to the point where it is a simple programming language embedded inside HTML files. The original acronym, PHP, has stuck. It isn't really appropriate any longer. PHP/FI is used more for entire web sites today than for small Personal Home Page setups. By whatever name, it eliminates the need for numerous small Perl cgi programs by allowing you to place simple scripts directly in your HTML files. This speeds up the overall performance of your web pages since the overhead of forking Perl several times has been eliminated. It also makes it easier to manage large web sites by placing all components of a web page in a single html file. By including support for various databases, it also makes it trivial to develop database enabled web pages. Many people find the embedded nature much easier to deal with than trying to create separate HTML and CGI files.

Throughout this documentation any references to PHP, FI or PHP/FI all refer to the same thing. The difference between PHP and FI is only a conceptual one. Both are built from the same source distribution. Now PHP/FI is renamed as PHP 3.

16.4 So, what can I do with PHP/FI ?

The first thing you will notice if you run a page through PHP/FI is that it adds a footer with information about the number of times your page has been accessed (if you have compiled access logging into the binary). This is just a very small part of what PHP/FI can do for you. It serves another very important role as a form interpreter cgi, hence the FI part of the name. For example, if you create a form on one of your web pages, you need something to process the information on that form. Even if you just want to pass the information to another web page, you will have to have a cgi program do this for you. PHP/FI makes it extremely easy to take form data and do things with it.

16.5 A simple example

Suppose you have a form:

```
<FORM ACTION="/cgi-bin/php.cgi/~userid/display.html" METHOD=POST>
<INPUT TYPE="text" name="name">
<INPUT TYPE="text" name="age">
<INPUT TYPE="submit">
<FORM>
```

Your display.html file could then contain something like:

```
< ?echo "Hi $ name, you are $ age years old!<p>" >
```

It's that simple! PHP/FI automatically creates a variable for each form input field in your form. You can then use these variables in the ACTION URL file.

The next step once you have figured out how to use variables is to start playing with some logical flow tags in your pages. For example, if you wanted to display different messages based on something the user inputs, you would use if/else logic. In our above example, we can display different things based on the age the user entered by changing our display.html to:

```
<?
    if($age>50);
        echo "Hi $name, you are ancient!<p>";
    elseif($age>30);
        echo "Hi $name, you are very old!<p>";
    else;
        echo "Hi $name.";
    endif;
>
```

PHP/FI provides a very powerful scripting language which will do much more than what the above simple example demonstrates. See the section on the PHP/FI Script Language for more information.

You can also use PHP/FI to configure who is allowed to access your pages. This is done using a built-in configuration screen. With this you could for example specify that only people from certain domains would be allowed to see your pages, or you could create a rule which would password protect certain pages. See the Access Control section for more details.

PHP/FI is also capable of receiving file uploads from any RFC-1867 compliant web browser. This feature lets people upload both text and binary files. With PHP/FI's access control and logical functions, you have full control over who is allowed to upload and what is to be done with the file once it has been uploaded. See the File Upload section for more details.

PHP/FI has support for the PostgreSQL database package. It supports embedded SQL queries in your .HTML files. See the section on PostgreSQL Support for more information.

PHP/FI also has support for the mysql database package. It supports embedded SQL queries in your .HTML files. See the section on mysql Support for more information.

16.6 CGI Redirection

Apache 1.0.x Notes

A good way to run PHP/FI is by using a cgi redirection module with the Apache server. Please note that you do not need to worry about redirection modules if you are using the Apache module version of PHP/FI. There are two of these redirection modules available. One is developed by Dave Andersen

- angio@aros.net

and it is available at

- ftp://ftp.aros.net/pub/util/apache/mod_cgi_redirect.c

and the other comes bundled with Apache and is called mod_actions.c. The modules are extremely similar. They differ slightly in their usage. Both have been tested and both work with PHP/FI.

Check the Apache documentation on how to add a module. Generally you add the module name to a file called Configuration. The line to be added if you want to use the mod_actions module is:

Module action_module mod_actions.o

If you are using the mod_cgi_redirect.c module add this line:

Module cgi_redirect_module mod_cgi_redirect.o

Then compile your httpd and install it. To configure the cgi redirection you need to either create a new mime type in your mime.types file or you can use the AddType command in your srm.conf file to add the mime type. The mime type to be added should be something like this:

```
application/x-httpd-php phtml
```

If you are using the mod_actions.c module you need to add the following line to your srm.conf file:

```
Action application/x-httpd-php /cgi-bin/php.cgi
```

If you are using mod_cgi_redirect.c you should add this line to srm.conf:

```
CgiRedirect application/x-httpd-php /cgi-bin/php.cgi
```

Don't try to use both mod_actions.c and mod_cgi_redirect.c at the same time.

Once you have one of these cgi redirection modules installed and configured correctly, you will be able to specify that you want a file parsed by php/fi simply by making the file's extension .phtml. Furthermore, if you add index.phtml to your DirectoryIndex configuration line in your srm.conf file then the top-level page in a directory will be automatically parsed by php if your index file is called index.phtml.

Netscape HTTPD

You can automatically redirect requests for files with a given extension to be handled by PHP/FI by using the Netscape Server CGI Redirection module. This module is available in the File Archives on the PHP/FI Home Page. The README in the package explicitly explains how to configure it for use with PHP/FI.

NCSA HTTPD

NCSA does not currently support modules, so in order to do cgi redirection with this server you need to modify your server source code. A patch to do this with NCSA 1.5 is available in the PHP/FI file archives.

16.7 Running PHP/FI from the command line

If you build the CGI version of PHP/FI, you can use it from the command line simply typing: php.cgi filename where filename is the file you want to parse. You can also create standalone PHP/FI scripts by making the first line of your script look something like:

```
#!/usr/local/bin/php.cgi -q
```

The "-q" suppresses the printing of the HTTP headers. You can leave off this option if you like.

---

17. Python Interface for PostgreSQL

Python in an interpreted, object orientated scripting language. It is simple to use (light syntax, simple and straighforward statements), and has many extensions for building GUIs, interfacing with WWW, etc. An intelligent web browser (HotJava like) is currently under development (november 1995), and this should open programmers many doors. Python is copyrighted by Stichting S Mathematisch Centrum, Amsterdam, The Netherlands, and is freely distributable. It contains support for dynamic loading of objects, classes, modules, and exceptions. Adding interfaces to new system libraries through C code is straightforward, making Python easy to use in custom settings. Python is a very high level scripting language with X interface. Python package is distributed on Linux cdroms includes most of the standard Python modules, along with modules for interfacing to the Tix widget set for Tk.

PyGreSQL is a python module that interfaces to a PostgreSQL database. It embeds the PostgreSQL query library to allow easy use of the powerful PostgreSQL features from a Python script. PyGreSQL is written by D'Arcy J.M. Cain and Pascal Andre.

- New site of PyGreSQL http://www.druid.net/pygresql/
- Maintained by D'Arcy at http://www.druid.net/~darcy/
- Old site is at ftp://ftp.via.ecp.fr/pub/python/contrib/Database/PyGres95.README
- D'Arcy J.M. Cain darcy@druid.net
- Pascal Andre andre@chimay.via.ecp.fr
- Pascal Andre andre@via.ecp.fr

17.1 Where to get PyGres ?

The home sites of the differents packages are:

- Python ftp://ftp.python.org:/pub/www.python.org/1.5/python1.5b2.tar.gz
- PyGreSQL ftp://ftp.druid.net/pub/distrib/PyGreSQL-2.1.tgz
- Old site ftp://ftp.via.ecp.fr/pub/python/contrib/Database/PyGres95-1.0b.tar.gz

You should anyway try to find some mirror site closer of your site. Refer to the information sources to find these sites. PyGreSQL should reside in the contrib directories of Python and PostgreSQL sites.

17.2 Information and support

If you need information about these packages please check their web sites:

- Python : http://www.python.org/
- PostgreSQL : http://epoch.cs.berkeley.edu:8000/postgres95/index.html

- PyGreSQL ftp://ftp.druid.net/pub/distrib/PyGreSQL-2.1.tgz
- Old site PyGreSQL : http://www.via.ecp.fr/via/products/pygres.html

For support :

- Mailing list for PyGreSQL. You can join by sending email to majordomo@vex.net with the line "subscribe pygresql name@domain" in the body replacing "name@domain" with your own email address.
- Newsgroup for Python : newsgroup comp.lang.python
- PyGreSQL : contact Andre at andre@via.ecp.fr for bug reports, ideas, remarks

---

18. Gateway between PostgreSQL and the WWW - WDB-P95

18.1 About wdb-p95

WDB-P95 - A Web interface to PostgreSQL Databases was created by J. Douglas Dunlop It is at

- New WDB from J Rowe is at http://www.lava.net/beowulf/programming/wdb
- New versions of WWW-WDB is at http://www.eol.ists.ca/~dunlop/wdb-p95/
- For questions or to join Mailing lists contact dunlop@eol.ists.ca

This is a modified version of wdb-1.3a2 which provides a gateway to a the WWW for PostgreSQL. This version also requires a Browser that is capable of handling HTML Tables for the tabular output. This is not required by the original wdb and can be fairly easily reverted.

You can try out CASI Tape and Image Query. You can have a peek at the Form Definition File (FDF) which is used to create the CASI Tape and Image Query too, which includes a JOIN of 2 tables.

This release contains all files necessary to install and run WDB-P95 as an interface to your PostgreSQL databases. To port this system to other database should be relatively easy - provided that it supports standard SQL and has a Perl interface.

18.2 Does the PostgreSQL server, pgperl, and httpd have to be on the same host?

No - the PostgreSQL server does not have to be on the same host. As WDB-P95 is called by the http daemon, they have to be on the same host. - And as WDB-P95 was written to use Pg.pm - pgperl has to be on the same host too. Pgperl was written using the libpq library, so it will be able to access any PostgreSQL server anywhere in the net, just like any other PostgreSQL client. As illustrated below

(WWW Client (Netscape)) => (HTTP Server (NCSA's http) + WDB-P95 + pgperl + libpq)=> (PostgreSQL server)

Curly brackets () represent machines.

Each machine can be of a different type : NT, SUN, HP, ... but you need the libpq interface library for the machine type where you plan to use WDB-P95, as you need it to compile pgperl. (The system was designed to use HTML tables so a recent WWW client is best)

19. "C", "C++", ESQL/C language Interfaces and Bitwise Operators for PostgreSQL

19.1 "C" interface

It is included in distribution and is called 'libpq'. Similar to Oracle OCI, Sybase DB-lib, Informix CLI libraries.

19.2 "C++" interface

It is included in distribution and is called 'libpq++'.

19.3 ESQL/C

ESQL/C 'Embedded C Pre-compiler' for PostgreSQL ESQL/C is like Oracle Pro*C, Informix ESQL/C. The PostgreSQL ESQL/C is an SQL application-programming interface (API) enables the C programmer to create custom applications with database-management capabilities. The PostgreSQL ESQL/C allows you to use a third-generation language with which you are familiar and still take advantage of the Structured Query Language (SQL).

ESQL/C consists of the following pieces of software:

- The ESQL/C libraries of C functions provide access to the database server.
- The ESQL/C header files provide definitions for the data structures, constants, and macros useful to the ESQL/C program.
- The ESQL/C preprocessor, is a source-code preprocessor that converts a C file containing SQL statements into an executable file.

It is at

- ESQL/C for PostgreSQL is already included in the distribution.
- Main site ftp://ftp.lysator.liu.se/pub/linus
- Email : linus@epact.se

19.4 BitWise Operators for PostgreSQL

Bitwise operators was written by Nicolas Moldavsky

- nico@overnet.com.ar

"C" functions that implement bitwise operators (AND, OR, XOR, bit complement) on pgsql. Get them by

anonymous FTP from

- ftp://ftp.overnet.com.ar/pub/utils/linux/bitpgsql.tgz

Makefile for Linux is included.

---

20. Japanese Kanji Code for PostgreSQL

It is at the following site

- ftp://ftp.sra.co.jp/pub/cmd/postgres/

21. PostgreSQL Port to Windows 95/Windows NT

Port to Windows 95/Windows NT is underway. Porting is being done using gcc, gmake for Win NT/95. To compile source code on win32 gnu-win32 program is used. GNU gcc is available for win32. Check this site -

- http://www.cygnus.com/misc/gnu-win32

At this site and get the file cdk.exe (self-extractor file for gnu-win32)

Porting can also be done using the following "Unix-Emulator on NT" tool from

- http://www.softway.com

22. Mailing Lists

22.1 Get a Free e-mail account

Check out the free deals offered - get free e-mail accounts from

- In Yahoo http://www.yahoo.com click on e-mail
- In Lycos http://www.lycos.com click on new e-mail accounts
- In hotmail http://www.hotmail.com click on new e-mail accounts

Subscribe to PostgreSQL mailing list and Yahoo has additional feature of creating a seperate folder for PostgreSQL e-mails, so that your regular e-mail is not cluttered. Select menu Email- > Options- > Filters and pick seperate folder for email. With this e-mail account you can access mail from anywhere in the world as long as you have access to a web page.

If you have any other e-mail, you can use "Mail Filters" to receive automatically the PostgreSQL mails into a seperate folder. This will avoid mail cluttering.

22.2 English Mailing List

See the Mailing Lists Item on the main web page at :

- http://www.postgresql.org/
- Email questions to: pgsql-questions@postgresql.org
- Developers pgsql-hackers@postgresql.org
- Port specific questions pgsql-ports@postgresql.org
- Documentation questions pgsql-docs@postgresql.org

You will get the answers/replies back by e-mail in less than a day!!

You can also subscribe to mailing lists. See also the section 'Get a Free e-mail account' above. To subscribe or unsubscribe from the list, send mail to

- pgsql-questions-request@postgresql.org
- pgsql-hackers-request@postgresql.org
- pgsql-ports-request@postgresql.org
- pgsql-docs-request@postgresql.org

The body of the message should contain the single line

subscribe

(or)

unsubscribe

22.3 Archive of Mailing List

Also mailing lists are archived in html format at the following location -

- Date-wise listing available via MHonarc via the WWW at http://www.postgresql.org/mhonarc/pgsql-questions
- ftp://ftp.postgresql.org directory is /pub/majordomo

22.4 Spanish Mailing List

Now there is an "unofficial" list of postgreSQL in Spanish. See also the section 'Free Account to Organise your PostgreSQL e-mails' above. To subscribe the user has to send a message to:

- majordomo@tlali.iztacala.unam.mx

The body of the message should contain the single line:

inscripcion pgsql-ayuda

---

23. Documentation and Reference Books

23.1 User Guides and Manuals

The following are included in the PostgreSQL distribution in the postscript, HTML formats and unix man-pages. If you have access to internet, you can find the documents listed below at http://www.postgresql.org/docs

- "Installation Guide"
- "User Guide" for PostgreSQL
- "Implementation Guide" detailing database internals of PostgreSQL.
- Online manuals.
- Online manuals in HTML formats.
- Also manuals in Postscript format for printing hard copies.

23.2 Online Documentation

- Listing and description of default data types and operators

  ```
  Is a a part of PSQL command in the release 6.4.
  ```

- Listing of supported SQL keywords

  ```
  There is a script in the /tools directory to do that
  ```

- Listings of supported statements -

  ```
  Use the command psql \h
  ```

- Basic relational database concepts under PostgreSQL (implementation) and several online examples (queries) -

  ```
  Look at the regression tests at src/test. There you can find the directories
  regress/sql and suite/*.sql.
  ```

- Tutorial for PostgreSQL.

  ```
  SQL tutorial scripts is in the directory src/tutorial
  ```

  See also "SQL Tutorial for beginners" in Appendix B of this document Appendix B

23.3 Useful Reference Textbooks

- "Understanding the New SQL: A Complete Guide" - by Jim Melton and Alan R.Simon

  ```
  Morgan Kaufman Publisher is one of best SQL books. This deals with SQL92.
  ```

- "A Guide to THE SQL STANDARD" - by C.J.Date

  ```
  Addison-Wesley Publishing company is also a good book. Very popular book for SQL.
  ```

- SQL - The Standard Handbook, November 1992

  ```
  Stephen Cannan and Gerard Otten
  McGraw-Hill Book Company Europe , Berkshire, SL6 2QL, England
  ```

- SQL Instant Reference, 1993

```
Martin Gruber, Technical Editor: Joe Celko
SYBEX Inc.  2021 Challenger Drive Alameda, CA 94501
```

- C.J.Date, "An introduction to Database Systems" (6th Edition), Addison-Wesley, 1995, ISBN 0-201-82458-2

```
This book is the Bible of Database Management Systems.
The book details normalization, SQL, recovery, concurrency, security,
integrity, and extensions to the original relational model, current issues
like client/server systems and the Object Oriented model(s). Many
references are included for further reading. Recommended for most users.
```

- Stefan Stanczyk, "Theory and Practice of Relational Databases", UCL Press Ltd, 1990, ISBN 1-857-28232-9

```
Book details theory of relational databases, relational algebra, calculus
and normalisation. But it does not cover real world issues and examples
beyond simple examples. Recommended for most users.
```

- "The Practical SQL Handbook" Third Edition, Addison Wesley Developers Press ISBN 0-201-44787-8

```
Recommended for most users.
```

- Michael Stonebraker, "Readings in Database Systems", Morgan Kaufmann, 1988, ISBN 0-934613-65-6

```
This book is a collection of papers that have been published over the
years on databases. It's not for the casual user but it is really a
reference for advanced (post-graduate) students or database system
developers.
```

- C.J.Date, "Relational Database - Selected Readings", Addison-Wesley, 1986, ISBN 0-201-14196-5

```
This book is a collection of papers that have been published over the
years on databases. It's not for the casual user but it is really a
reference for advanced (post-graduate) students or database system
developers.
```

- Nick Ryan and Dan Smith, "Database Systems Engineering", International Thomson Computer Press, 1995, ISBN 1-85032-115-9

```
This book goes into the details of access methods, storage techniques.
```

- Bipin C. Desai, "An introduction to Database Systems", West Publishing Co., 1990, ISBN 0-314-66771-7

```
It's not for the casual user but it is for advanced (post-graduate)
students or database system developers.
```

- Joe Celko "INSTANT SQL Programming"

```
Wrox Press Ltd.
Unit 16, 20 James Road, Tyseley
Birmingham, B11 2BA, England
1995
```

- Michael Gorman "Database Management Systems: Understanding and Applying Database"

```
Technology
QED and John Wiley
1991
```

- Michael Gorman "Enterprise Database for a Client/Server Environment" QED and John Wiley

```
Presents the requirements of building client/server database
applications via repository metamodels and the use of ANSI standard SQL
```

1993

Hundreds of other titles on SQL are available! Check out a bookstore.

23.4 ANSI/ISO SQL Specifications documents - SQL 1992, SQL 1998

ANSI/ISO SQL specifications documents can be found at these sites listed below -

- http://www.naiua.org/std-orgs.html
- http://www.ansi.org/docs and click on file cat_c.html and search with "Database SQL"
- SQL92 standard http://www.jcc.com and click on file sql_stnd.html
- ANSI/ISO SQL specifications http://www.contrib.andrew.cmu.edu/~shadow/sql.html You will find SQL Reference here.

23.5 Syntax of ANSI/ISO SQL 1992

See Appendix A of this document Appendix A

23.6 Syntax of ANSI/ISO SQL 1998

The SQL 1998 (SQL 3) specification is still under development. See 'Electronic Access to the SQL3 Working Draft' of this document at SQL 1998

23.7 SQL Tutorial for beginners

See Appendix B of this document Appendix B

23.8 Temporal Extension to SQL92

- Document for Temporal Extension to SQL-92 ftp://FTP.cs.arizona.edu/tsql/tsql2/
- Temporal SQL-3 specification ftp://FTP.cs.arizona.edu/tsql/tsql2/sql3/

This directory contains the language specification for a temporal extension to the SQL-92 language standard. This new language is designated TSQL2.

The language specification present here is the final version of the language.

Correspondence may be directed to the chair of the TSQL2 Language Design Committee, Richard T.Snodgrass, Department of Computer Science, University of Arizona, Tucson, AZ 85721,

- rts@cs.arizona.edu

The affiliations and e-mail addresses of the TSQL2 Language Design Committee members may be found in a separate section at the end of the language specification.

The contents of this directory are as follows.

spec.dvi,.ps TSQL2 Language Specification, published in September, 1994

bookspec.ps TSQL2 Language Specification, as it appears in the TSQL2 book, published in September, 1995 (see below).

sql3 change proposals submitted to the ANSI and ISO SQL3 committees.

Associated with the language specification is a collection of commentaries which discuss design decisions, provide examples, and consider how the language may be implemented. These commentaries were originally proposals to the TSQL2 Language Design Committee. They now serve a different purpose: to provide examples of the constructs, motivate the many decisions made during the language design, and compare TSQL2 with the many other language proposals that have been made over the last fifteen years. It

should be emphasized that these commentaries are not part of the TSQL2 language specification per se, but rather supplement a elaborate upon it. The language specification proper is the final word on TSQL2.

The commentaries, along with the language specification, several indexes, and other supporting material, has been published as a book:

Snodgrass, R.T., editor, The TSQL2 Temporal Query Language, Kluwer Academic Publishers, 1995, 674+xxiv pages.

The evaluation commentary appears in the book in an abbreviated form; the full commentary is provided in this directory as file eval.ps

The file tl2tsql2.pl is a prolog program that tranlates allowed temporal logic to TSQL2. This program was written by Michael Boehlen

- boehlen@iesd.auc.dk

He may be contacted for a paper that describes this translation. This is a rather dated version of that program. Newer versions are available at

- http://www.cs.auc.dk/general/DBS/tdb/TimeCenter/Software

(the TimeDB and Tiger systems).

23.9 Part 0 - Acquiring ISO/ANSI SQL Documents

This document shows you how to (legally) acquire a copy of the SQL-92 standard and how to acquire a copy of the "current" SQL3 Working Draft.

The standard is copyrighted ANSI standard by ANSI, the ISO standard by ISO.

There are two (2) current SQL standards, an ANSI publication and an ISO publication. The two standards are word-for-word identical except for such trivial matters as the title of the document, page headers, the phrase "International Standard" vs "American Standard", and so forth.

Buying the SQL-92 Standard

The ISO standard, ISO/IEC 9075:1992, Information Technology - Database Languages - SQL, is currently (March, 1993) available and in stock from ANSI at:

```
American National Standards Institute
1430 Broadway
New York, NY 10018 (USA)
Phone (sales): +1.212.642.4900
```

at a cost of US$230.00. The ANSI version, ANSI X3.135-1992, American National Standard for Information Systems - Database Language SQL, was not available from stock at this writing, but was expected to be available by some time between late March and early May, 1993). It is expected to be be priced at US$225.00.

If you purchase either document from ANSI, it will have a handling charge of 7% added to it (that is, about US$9.10). Overseas shipping charges will undoubtedly add still more cost. ANSI requires a hardcopy of a company purchase order to accompany all orders; alternately, you can send a check drawn on a US bank in US dollars, which they will cash and clear before shipping your order. (An exception exists: If your organization is a corporate member of ANSI, then ANSI will ship the documents and simply bill your company.)

The ISO standard is also available outside the United States from local national bodies (country standardization bodies) that are members of either ISO (International Organization for Standardization) or IEC (International Electrotechnical Commission). Copies of the list of national bodies and their addresses are available from ANSI or from other national bodies. They are also available from ISO:

```
International Organization for Standardization
Central Secretariat
1, rue de Varembi
CH-1211 Genhve 20
Switzerland
```

If you prefer to order the standard in a more convenient and quick fashion, you'll have to pay for the privilege. You can order ISO/IEC 9075:1992, Information Technology - Database Languages - SQL, from:

```
Global Engineering Documents
2805 McGaw Ave
Irvine, CA 92714 (USA)
USA
Phone (works from anywhere): +1.714.261.1455
Phone (only in the USA): (800)854-7179
```

for a cost of US$308.00. I do not know if that includes shipping or not, but I would guess that international shipping (at least) would cost extra. They will be able to ship you a document fairly quickly and will even accept "major credit cards". Global does not yet have the ANSI version nor do they have a price or an expected date (though I would expect it within a few weeks following the publication by ANSI and at a price near US$300.00).

Buying a copy of the SQL3 Working Draft

You can purchase a hardcopy of the SQL3 working draft from the ANSI X3 Secretariat, CBEMA (Computer and Business Equipment Manufacturers Association). They intend to keep the "most recent" versions of the SQL3 working draft available and sell them for about US$60.00 to US$65.00. You can contact CBEMA at:

```
CBEMA, X3 Secretariat
Attn: Lynn Barra
1250 Eye St.
Suite 200
Washington, DC 20005 (USA)
```

Lynn Barra can also be reached by telephone at +1.202.626.5738 to request a copy, though mail is probably more courteous.

Electronic Access to the SQL3 Working Draft

The most recent version (as of the date of this writing) of the SQL3 (both ANSI and ISO) working draft (and all of its Parts) is available by "anonymous ftp" or by "ftpmail" on:

```
gatekeeper.dec.com
```

at

```
/pub/standards/sql/
```

In this directory are a number of files. There are PostScript. files and "plain text" (not prettily formatted, but readable on a screen without special software).

In general, you can find files with names like:

```
sql-bindings-mar94.ps
sql-bindings-mar94.txt
sql-cli-mar94.ps
sql-cli-mar94.txt
sql-foundation-mar94.ps
sql-foundation-mar94.txt
sql-framework-mar94.ps
sql-framework-mar94.txt
sql-psm-mar94.ps
sql-psm-mar94.txt
```

As new versions of the documents are produced, the "mar94" will change to indicate the new date of publication (e.g., "aug94" is the expected date of the next publication after "mar94").

In addition, for those readers unable to get a directory listing by FTP, we have placed a file with the name:

```
ls
```

into the same directory. This file (surprise!) contains a directory listing of the directory.

Retrieving Files Directly Using ftp

This is a sample of how to use FTP. Specifically, it shows how to connect to gatekeeper.dec.com, get to the directory where the base document is kept, and transfer the document to your host. Note that your host must have Internet access to do this. The login is 'ftp' and the password is your email address (this is sometimes referred to as 'anonymous ftp'). The command 'type binary' is used to ensure that no bits are stripped from the file(s) received. 'get' gets one file at a time. Comments in the script below are inside angle brackets < like so > .

```
% ftp gatekeeper.dec.com
Connected to gatekeeper.dec.com.
220- *** /etc/motd.ftp ***
     Gatekeeper.DEC.COM is an unsupported service of DEC Corporate Research.
     <...this goes on for a while...>
220 gatekeeper.dec.com FTP server (Version 5.83 Sat ... 1992) ready.
Name (gatekeeper.dec.com:<yourlogin here>): ftp  <anonymous also works>
331 Guest login ok, send ident as password.
Password: <enter your email address here >
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.  <or whatever>
Using binary mode to transfer files.
ftp> cd pub/standards/sql
250 CWD command successful.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 9529
-r--r--r--  1 root     system     357782 Feb 25 10:18 x3h2-93-081.ps
-r--r--r--  1 root     system     158782 Feb 25 10:19 x3h2-93-081.txt
-r--r--r--  1 root     system     195202 Feb 25 10:20 x3h2-93-082.ps
-r--r--r--  1 root     system      90900 Feb 25 10:20 x3h2-93-082.txt
-r--r--r--  1 root     system    5856284 Feb 25 09:55 x3h2-93-091.ps
-r--r--r--  1 root     system    3043687 Feb 25 09:57 x3h2-93-091.txt
226 Transfer complete.
ftp> type binary
200 Type set to I.
ftp> get x3h2-93-082.txt
200 PORT command successful.
150 Opening BINARY mode data connection for x3h2-93-082.txt (90900 bytes).
226 Transfer complete.
90900 bytes received in 0.53 seconds (166.11 Kbytes/s)
ftp> quit
% <the file is now in your directory as x3h2-93-082.txt>
```

Retrieving Files Without Direct ftp Support

Digital Equipment Corporation, like several other companies, provides ftp email service. The response can take several days, but it does provide a service equivalent to ftp for those without direct Internet ftp access. The address of the server is:

ftpmail@decwrl.dec.com

The following script will retrieve the PostScript for the latest version of the SQL3 document:

```
reply joe.programmer@imaginary-corp.com
connect gatekeeper.dec.com anonymous
binary
compress
```

The following script will retrieve the PostScript for the latest version of the SQL3 document:

```
reply joe.programmer@imaginary-corp.com
connect gatekeeper.dec.com anonymous
binary
compress
uuencode
chdir /pub/standards/sql
get x3h2-93-091.ps
quit
```

The first line in the script commands the server to return the requested files to you; you should replace "joe.programmer@imaginary-corp.com" with your Internet address. The file in this example, x3h2-93-091.ps, is returned in "compress"ed "uuencode"d format as 34 separate email messages. If your environment does not provide tools for reconstructing such files, then you could retrieve the file as plain text with the following script:

```
reply joe.programmer@imaginary-corp.com
connect gatekeeper.dec.com anonymous
chdir /pub/standards/sql
get x3h2-93-091.ps
quit
```

But be warned, the .ps file will probably be sent to you in more than 70 parts!

To retrieve any particular file, other than x3h2-93-091.ps, simply replace "x3h2-93-091.ps" with the name of the desired file. To get a directory listing of all files available, replace "get x3h2-93-091.ps" with "dir".

23.10 Part 1 - ISO/ANSI SQL Current Status

This chapter is a source of information about the SQL standards process and its current state.

Current Status:

Development is currently underway to enhance SQL into a computationally complete language for the definition and management of persistent, complex objects. This includes: generalization and specialization hierarchies, multiple inheritance, user defined data types, triggers and assertions, support for knowledge based systems, recursive query expressions, and additional data administration tools. It also includes the specification of abstract data types (ADTs), object identifiers, methods, inheritance, polymorphism, encapsulation, and all of the other facilities normally associated with object data management.

In the fall of 1996, several parts of SQL3 went through a ISO CD ballot. Those parts were SQL/Framework, SQL/Foundation, and SQL/Bindings. Those ballots failed (as expected) with 900 or so comments. In Late January, there was an ISO DBL editing meeting that processed a large number of problem solutions that were either included with ballot comments or submitted as separate papers. Since the DBL editing meeting was unable to process all of the comments, the editing meeting has been extended. The completion of the editing meeting is scheduled for the end of July, 1997, in London.

Following the July editing meeting, the expectation is that a Final CD ballot will be requested for these parts of SQL. The Final CD process will take about 6 months and a DBL editing meeting, after which there will be a DIS ballot and a fairly quick IS ballot.

The ISO procedures have changed since SQL/92, so the SQL committees are still working through the exact details of the process.

If everything goes well, these parts of SQL3 will become an official ISO/IEC standard in late 1998, but the schedule is very tight.

In 1993, the ANSI and ISO development committees decided to split future SQL development into a multi-part standard. The Parts are:

- Part 1: Framework A non-technical description of how the document is structured.
- Part 2: Foundation The core specification, including all of the new ADT features.
- Part 3: SQL/CLI The Call Level Interface.
- Part 4: SQL/PSM The stored procedures specification, including computational completeness.
- Part 5: SQL/Bindings The Dynamic SQL and Embedded SQL bindings taken from SQL-92.
- Part 6: SQL/XA An SQL specialization of the popular XA Interface developed by X/Open
- Part 7:SQL/TemporalAdds time related capabilities to the SQL standards.

In the USA, the entirety of SQL3 is being processed as both an ANSI Domestic ("D") project and as an ISO project. The expect time frame for completion of SQL3 is currently 1999.

The SQL/CLI and SQL/PSM are being processed as fast as possible as addendums to SQL-92. In the USA, these are being processed only as International ("I") projects. SQL/CLI was completed in 1995. SQL/PSM should be completed sometime in late 1996.

In addition to the SQL3 work, a number of additional projects are being persued:

- SQL/MM An ongoing effort to define standard multi-media packages using the SQL3 ADT capabilities.
- Remote Data Access (RDA)

Standards Committee and Process

There are actually a number of SQL standards committees around the world. There is an international SQL standards group as a part of ISO. A number of countries have committees that focus on SQL. These countries (usually) send representatives to ISO/IEC JTC1/SC 21/WG3 DBL meetings. The countries that actively participate in the ISO SQL standards process are:

- Australia
- Brazil
- Canada
- France
- Germany
- Japan
- Korea
- The Netherlands
- United Kingdom
- United States

NIST Validation

SQL implementations are validated (in the Unites States) by the National Institute of Standards and Testing (NIST). NIST currently has a validation test suite for entry level SQL-92. The exact details of the NIST validation requirements are defined as a Federal Information Processing Standard (FIPS). The current requirements for SQL are defined in FIPS 127-2. The Postscript and Text versions of this document can be retrieved from NIST. The current SQL Validated Products List can also be retrieved from NIST.

Standard SQL Publications and Articles

There are two versions of the SQL standard. Both are available from ANSI:

- ISO/IEC 9075:1992, "Information Technology --- Database Languages --- SQL"
- ANSI X3.135-1992, "Database Language SQL"

The two versions of the SQL standard are identical except for the front matter and references to other standards. Both versions are available from:

```
American National Standards Institute
1430 Broadway
New York, NY 10018
USA
Phone (sales): +1.212.642.4900
```

In additon to the SQL-92 standard, there is now a Technical Corrigendum (bug fixes):

```
* Technical Corrigendum 1:1994 to ISO/IEC 9075:1992
```

TC 1 should also be available from ANSI. There is only an ISO version of TC 1 -- it applies both to the ISO and ANSI versions of SQL-92.

In addition to the standards, several books have been written about the 1992 SQL standard. These books provide a much more readable description of the standard than the actual standard.

Related Standards

A number of other standards are of interest to the SQL community. This section contains pointers to information on those effor These pointers will be augmented as additional information becomes available on the web.

- SQL Environments (FIPS 193)
- Next Generation Repository Systems (X3H4) - a News Release calling for particpation in "Developing Standards for the Next Generation Repository Systems."

23.11 Part 2 - ISO/ANSI SQL Foundation

A significant portion of the SQL3 effort is in the SQL Foundation document:

- Base SQL/PSM capabilities (moved form SQL/PSM-92)
- New data types
- Triggers
- Subtables
- Abstract Data Types (ADT)
- Object Oriented Capabilities

There are several prerequisites to the object oriented capabilities:

- Capability of defining complex operations
- Store complex operations in the database
- External procedure calls - Some operations may not be in SQL, or may require external interactions

These capabilities are defined as a part of SQL/PSM

A great deal of work is currently being done to refine the SQL-3 object model and align it with the object model proposed by ODMG. This effort is described in the X3H2 and ISO DBL paper: Accomodating SQL3 and ODMG. A recent update on the SQL3/OQL Merger is also available.

SQL3 Timing

Work on SQL3 is well underway, but the final standards is several years away.

- International ballot to progress SQL3 Foundation from Working Draft to Committee Draft (CD) taking place fall, 1996.
- Ballot is expected to generate numerous comments
- A second CD ballot is likely to be required
- Draft International Standard ballot is likely to be take place in mid 1998
- International Standard could be completed by mid 1999.

The ANSI version of the standard will be on a similar schedule.

23.12 Part 3 - ISO/ANSI SQL Call Level Interface

The SQL/CLI is a programing call level interface to SQL databases. It is designed to support database access from shrink-wrapped applications. The CLI was originally created by a subcommittee of the SQL Access Group (SAG). The SAG/CLI specification was published as the Microsoft Open DataBase Connectivity (ODBC) specification in 1992. In 1993, SAG submitted the CLI to the ANSI and ISO SQL committees. (The SQL Access Group has now merged with X/Open consortium.)

SQL/CLI provides an international standard for:

- Implementation-independent CLI to access SQL databases
- Client-server tools can easily access database through dynamic Link Libraries
- Supports and encourages rich set of Client-server tools

SQL/CLI Timing

For the standards process, SQL/CLI is being processed with blinding speed.

- SQL/CLI is an addendum to 1992 SQL standard (SQL-92)
- Completed as an ISO standard in 1995

- ISO/IEC 9075-3:1995 Information technology -- Database languages -- SQL -- Part 3: Call-Level Interface (SQL/CLI)
- Current SQL/CLI effort is adding support for SQL3 features

23.13 Part 4 - ISO/ANSI SQL Persistent Stored Modules

SQL/PSM expands SQL by adding:

- Procedural language extensions
- Multi-statement and Stored Procedures
- External function and procedure calls

In addition to being a valuable application development tool, SQL/PSM provides the foundation support for the object oriented capabilities in SQL3.

Multi-statement and Stored Procedures

Multi-statement and stored procedures offer a variety of advantages in a client/server environment:

- Performance - Since a stored procedure can perform multiple SQL statements, network interaction with the client are reduced.
- Security - A user can be given the right to call a stored procedure that updates a table or set of tables but denied the right to update the tables directly
- Shared code - The code in a stored procedure does not have to be rewritten and retested for each client tool that accesses the database.
- Control - Provides a single point of definition and control for application logic.

Procedural Language Extensions

Procedural language add the power of a traditional programming language to SQL through flow control statements and a variety of other programming constructs.

Flow Control Statements

- If-then-else
- Looping constructs
- Exception handling
- Case statement
- Begin-End blocks

The procedural language extensions include other programming language constructs:

- Variable declarations
- Set statements for value assignment
- Get diagnostics for process and status information

In addition, all of the traditional SQL statements can be included in multi-statement procedures.

External Procedure and Function Calls

One feature frequently mentioned in the wish lists for many database products, and implemented in some, is a capability augmenting the built-in features with calls to user-written procedures external to the database software.

- Allows a particular site or application to add their own database functions
- Can be used throughout the database applications

The benefit of this capability is that it gives the database (and therefore database applications) access to a rich set of procedures and functions too numerous to be defined by a standards committee.

SQL/PSM Timing

SQL/PSM is proceeding quickly:

- SQL/PSM is an addendum to SQL-92
- International ballot to progress SQL/PSM from a Draft International Standard to an International Standard ended January,

1996.

- Editing meeting in May, 1996 did not resolve all of the comments
- Continuation of PSM Editing meeting is scheduled for September 30 through October 4, 1996
- The schedule is tight but there is a chance that PSM will be published with a 1996 date.
- The official designation will be: ISO/IEC DIS 9075-4:199? Information technology -- Database languages -- SQL -- Part 4: SQL Persistent Stored Modules (SQL/PSM)
- Work is well underway on adding SQL/PSM support for SQL3 features.

23.14 Part 5 - ISO/ANSI SQL/Bindings

For ease of reference, the programming language bindings have been pulled out into a separate document. The current version is simply an extract of the dynamic and embedded bindings from SQL-92.

A variety of issues remain unresolved for the programming language bindings.

For traditional programming language, mappings exist for the SQL-92 datatypes. However, mappings must be defined between SQL objects and programming language variables.

For object oriented languages, mapping must be defined for the current SQL datatypes and between the SQL object model and the object model of the object-oriented language.

The object model needs to stabilize before these can be addressed.

The language bindings will be completed as a part of SQL3.

23.15 Part 6 - ISO/ANSI SQL XA Interface Specialization (SQL/XA)

This specification would standardize an application program interface (API) between a global Transaction Manager and an SQL Resource Manager. It would standardize the function calls, based upon the semantics of ISO/IEC 10026, "Distributed Transaction Processing", that an SQL Resource Manager would have to support for two-phase commit. The base document is derived from an X/Open publication, with X/Open permission, that specifies explicit input and output parameters and semantics, in terms of SQL data types, for the following functions: xa_close, xa_commit, xa_complete, xa_end, xa_forget, xa_open, xa_prepare, xa_recover, xa_rollback, and xa_start.

ISO is currently attempting to fast-track the X/Open XA specification. The fast-track process adopts a current industry specification with no changes. The XA fast-track ballot at the ISO SC21, JTC 1 level started on April 27, 1995 and ends on October 27, 1995. If the XA specification is approved by 75% of the votes, and by 2/3 of the p-members of JTC 1, it will become an International Standard. If the fast-track ballot is approved, SQL/XA could become a standard in 1996.

23.16 Part 7 - ISO/ANSI SQL Temporal

Temporal SQL deals with time-related data. The concept is that it is useful to query data to discover what it looked like at a particular point in time. Temporal SQL is a December, 1994 paper by Rick Snodgrass describing the concepts.

X3 Announces the Approval of a New Project, ISO/IEC 9075 Part 7: SQL/Temporal is a press release related to SQL/Temporal.

```
---------------------------------------------------------------------------
                                Temporal SQL
                                ************
Rick Snodgrass (chair of the TSQL2 committee)
31-Dec-1994
```

Several people have questioned the need for additional support for time in SQL3 (as proposed by DBL RIO-75, requesting a new part of SQL to support temporal databases). The claim is that abstract data types (ADT's) are sufficient for temporal support. In this informational item, I argue, using concrete examples, that using columns typed with abstract data types is inadequate for temporal queries. In particular, many common temporal queries are either difficult to simulate in SQL, or require embedding SQL in a procedural language. Alternatives are expressed in TSQL2, a temporal extension to SQL-92.

INTRODUCTION

Valid-time support goes beyond that of a temporal ADT. With the later, a column is specified as of a temporal domain, such as DATE or INTERVAL (examples will be given shortly). With valid time, the rows of a table vary over time, as reality changes. The timestamp associated with a row of a valid-time table is interpreted by the query language as the time when the combination of values of the columns in the row was valid. This implicit timestamp allows queries to be expressed succinctly and intuitively.

A CASE STUDY - STORING CURRENT INFORMATION

The University of Arizona's Office of Appointed Personnel has some information in a database, including each employee's name, their current salary, and their current title. This can be represented by a simple table.

```
Employee(Name, Salary, Title)
```

Given this table, finding an employee's salary is easy.

```
SELECT Salary
FROM Employee
WHERE Name = 'Bob'
```

Now the OAP wishes to record the date of birth. To do so, a column is added to the table, yielding the following schema.

```
Employee(Name, Salary, Title, DateofBirth DATE)
```

Finding the employee's date of birth is analogous to determining the salary.

```
SELECT DateofBirth
FROM Employee
WHERE Name = 'Bob'
```

A CASE STUDY - STORING HISTORY INFORMATION

The OAP wishes to computerize the employment history. To do so, they append two columns, one indicating when the information in the row became valid, the other indicating when the information was no longer valid.

Employee (Name, Salary, Title, DateofBirth, Start DATE, Stop DATE)

To the data model, these new columns are identical to DateofBirth. However, their presence has wide-ranging consequences.

A CASE STUDY - PROJECTION

To find an employee's current salary, things are more difficult.

```
SELECT Salary
FROM Employee
WHERE Name = 'Bob' AND Start <= CURRENT_DATE AND CURRENT_DATE <= Stop
```

This query is more complicated than the previous one. The culprit is obviously the two new columns. The OAP wants to distribute to each employee their salary history. Specifically, for each person, the maximal intervals at each salary needs to be determined. Unfortunately, this is not possible in SQL. An employee could have arbitrarily many title changes between salary changes.

Name	Salary	Title	DateofBirth	Start	Stop
Bob	60000	Assistant Provost	1945-04-09	1993-01-01	1993-05-30
Bob	70000	Assistant Provost	1945-04-09	1993-06-01	1993-09-30
Bob	70000	Provost	1945-04-09	1993-10-01	1994-01-31
Bob	70000	Professor	1945-04-09	1994-02-01	1994-12-31

Figure 1

Note that there are three rows in which Bob's salary remained constant at $70,000. Hence, the result should be two rows for Bob.

```
Name    Salary  Start           Stop
----    ------  -----           ----
Bob     60000   1993-01-01      1993-05-30
Bob     70000   1993-06-01      1994-12-31
```

One alternative is to give the user a printout of Salary and Title information, and have user determine when his/her salary changed. This alternative is not very appealing or realistic. A second alternative is to use SQL as much as possible.

```
CREATE TABLE Temp(Salary, Start, Stop)
AS      SELECT Salary, Start, Stop
        FROM Employee;
```

repeat

```
        UPDATE Temp T1
        SET (T1.Stop) = (SELECT MAX(T2.Stop)
                         FROM Temp AS T2
                         WHERE T1.Salary = T2.Salary AND T1.Start < T2.Start
                                AND T1.Stop >= T2.Start AND T1.Stop < T2.Stop)
        WHERE EXISTS (SELECT *
                      FROM Temp AS T2
                      WHERE T1.Salary = T2.Salary AND T1.Start < T2.Start
                                AND T1.Stop >= T2.Start AND T1.Stop < T2.Stop)
        until no rows updated;

DELETE FROM Temp T1

WHERE EXISTS (SELECT *
              FROM Temp AS T2
              WHERE T1.Salary = T2.Salary
                        AND ((T1.Start > T2.Start AND T1.Stop <= T2.Stop)
                        OR (T1.Start >= T2.Start AND T1.Stop < T2.Stop))
```

The loop finds those intervals that overlap or are adjacent and thus should be merged. The loop is executed log N times in the worst case, where N is the number of rows in a chain of overlapping or adjacent value-equivalent rows. The reader can simulate the query on the example table to convince him/herself of its correctness.

A third alternative is to use SQL only to open a cursor on the table. A linked list of periods is maintained, each with a salary. This linked list should be initialized to empty.

```
DECLARE emp_cursor CURSOR FOR
        SELECT Salary, Title, Start, Stop
        FROM Employee;
OPEN emp_cursor;
loop:
        FETCH emp_cursor INTO :salary, :start, :stop;
        if no-data returned then goto finished;
        find position in linked list to insert this information;
        goto loop;
finished:
CLOSE emp_cursor;
```

iterate through linked list, printing out dates and salaries

The linked list may not be necessary in this case if the cursor is ORDER BY Start.

In any case, the query, a natural one, is quite difficult to express using the facilities present in SQL-92. The query is trivial in TSQL2.

```
        SELECT Salary
        FROM Employee
```

A CASE STUDY - JOIN

A more drastic approach is to avoid the problem of extracting the salary history by reorganizing the schema to separate salary, title, and date of birth information (in the following, we ignore the date of birth, for simplicity).

```
Employee1 (Name, Salary, Start DATE, Stop DATE)
Employee2 (Name, Title, Start DATE, Stop DATE)
```

The Employee1 table is as follows.

```
Name    Salary  Start           Stop
----    ------  -----           ----
Bob     60000   1993-01-01      1993-05-30
Bob     70000   1993-06-01      1993-12-31
```

Here is the example Employee2 table.

```
Name    Title                   Start           Stop
----    ------                  -----           ----
Bob     Assistant Provost       1993-01-01      1993-09-30
Bob     Provost                 1993-10-01      1994-01-31
Bob     Professor               1994-02-01      1994-12-31
```

With this change, getting the salary information for an employee is now easy.

```
SELECT Salary, Start, Stop
FROM Employee1
WHERE Name = 'Bob'
```

But what if the OAP wants a table of salary, title intervals (that is, suppose the OAP wishes a table to be computed in the form of Figure 1)? One alternative is to print out two tables, and let the user figure out the combinations. A second alternative is to use SQL entirely. Unfortunately, this query must do a case analysis of how each row of Employee1 overlaps each row of Employee2; there are four possible cases.

```
SELECT Employee1.Name, Salary, Dept, Employee1.Start, Employee1.Stop
FROM Employee1, Employee2
WHERE Employee1.Name = Employee2.Name
     AND Employee2.Start <= Employee1.Start AND Employee1.Stop < Employee2.Stop
UNION
SELECT Employee1.Name, Salary, Dept, Employee1.Start, Employee2.Stop
FROM Employee1, Employee2
WHERE Employee1.Name = Employee2.Name
     AND Employee1.Start >= Employee2.Start AND Employee2.Stop < Employee1.Stop
        AND Employee1.Start < Employee2.Stop
UNION
SELECT Employee1.Name, Salary, Dept, Employee2.Start, Employee1.Stop
FROM Employee1, Employee2
WHERE Employee1.Name = Employee2.Name
     AND Employee2.Start > Employee1.Start AND Employee1.Stop < Employee2.Stop
        AND Employee2.Start < Employee1.Stop
UNION
SELECT Employee1.Name, Salary, Dept, Employee2.Start, Employee2.Stop
FROM Employee1, Employee2
WHERE Employee1.Name = Employee2.Name
     AND Employee2.Start > Employee1.Start AND Employee2.Stop < Employee1.Stop
```

Getting all the cases right is a challenging task. In TSQL2, performing a temporal join is just what one would expect.

```
SELECT Employee1.Name, Salary, Dept
```

```
FROM Employee1, Employee2
WHERE Employee1.Name = Employee2.Name
```

A CASE STUDY - AGGREGATES

Now the OAP is asked, what is the maximum salary? Before adding time, this was easy.

```
SELECT MAX(Salary)
FROM Employee
```

Now that the salary history is stored, we'd like a history of the maximum salary over time. The problem, of course, is that SQL does not provide temporal aggregates. The easy way to do this is to print out the information, and scan manually for the maximums. An alternative is to be tricky and convert the snapshot aggregate query into a non-aggregate query, then convert that into a temporal query. The non-aggregate query finds those salaries for which a greater salary does not exist.

```
SELECT Salary
FROM Employee AS E1
WHERE NOT EXISTS (SELECT *
                  FROM Employee AS E2
                  WHERE E2.Salary > E1.Salary)
```

Converting this query into a temporal query is far from obvious. The following is one approach.

```
CREATE TABLE Temp (Salary, Start, Stop)
AS      SELECT Salary, Start, Stop
        FROM Employee;
INSERT INTO Temp
        SELECT T.Salary, T.Start, E.Start
        FROM Temp AS T, Employee AS E
        WHERE E.Start >= T.Start AND E.Start < T.Stop AND E.Salary > T.Salary;

INSERT INTO Temp
        SELECT T.Salary, T.Stop, E.Stop
        FROM Temp AS T, Employee AS E
        WHERE E.Stop > T.Start AND E.Stop <= T.Stop AND E.Salary > T.Salary;
DELETE FROM Temp T
WHERE EXISTS (SELECT *
              FROM Employee AS E
              WHERE ((T.Start => E.Start AND T.Start < E.Stop)
                        OR (E.Start >= T.Start AND E.Start < T.Stop))
                    AND E.Salary > T.Salary;
```

This approach creates an auxiliary table. We add to this table the lower period of a period subtraction and the upper period of a period subtraction. We then delete all periods that overlap with some row defined by the subquery, thereby effecting the NOT EXISTS. Finally we generate from the auxiliary table maximal periods, in the same way that the salary information was computed above. As one might imagine, such SQL code is extremely inefficient to execute, given the complex nested queries with inequality predicates.

A third alternative is to use SQL as little as possible, and instead compute the desired maximum history in a host language using cursors.

The query in TSQL2 is again straightforward and intuitive.

```
SELECT MAX(Salary)
FROM Employee
```

SUMMARY

Time-varying data is manipulated in most database applications. Valid-time support is absent in SQL. Many common temporal queries are either difficult to simulate in SQL, or require embedding SQL in a procedural language, due to SQL's lack of support for valid-time tables in its data model and query constructs.

Elsewhere, we showed that adding valid-time support requires few changes to the DBMS implementation, can dramatically sim some queries and enable others, and can later enable optimizations in storage structures, indexing methods, and optimization strategies that can yield significant performance improvements.

With a new part of SQL3 supporting time-varying information, we can begin to address such applications, enabling SQL3 to better manage temporal data.

```
-----------------------------------------------------------------------------
           Accredited Standards Committee* X3, Information Technology
NEWS RELEASE

Doc. No.:       PR/96-0002

Reply to:       Barbara Bennett at bbennett@itic.nw.dc.us

             X3 Announces the Approval of a New Project, ISO/IEC

                        9075 Part 7:  SQL/Temporal

Washington D.C., January 1996
-----------------------------------------------------------------------------
```

-- Accredited Standards Committee X3, Information Technology is announcing the approval of a new project on SQL/Temporal Support, ISO/IEC 9075 Part 7, with the work being done in Technical Committee X3H2, Database. The scope of this proposed standard specifies a new Part of the emerging SQL3 standard, e.g., Part 7, Temporal SQL, to be extensions to the SQL language supporting storage, retrieval, and manipulation of temporal data in an SQL database environment. The next X3H2 meeting is scheduled for March 11-14, 1996 in Kansas.

Inquiries regarding this project should be sent to the

```
Chairman of X3H2,
Dr. Donald R. Deutsch,
Sybase, Inc., Suite 800,
6550 Rock Spring
Drive, Bethesda, MD  20817.
Email: deutsch@sybase.com.
```

An initial call for possible patents and other pertinent issues (copyrights, trademarks) is now being issued. Please submit information on these issues to the

```
X3 Secretariat at
1250 Eye Street
NW, Suite 200,
Washington DC  20005.
Email: x3sec@itic.nw.dc.us
FAX:  (202)638-4922.
```

23.17 Part 8 - ISO/ANSI SQL MULTIMEDIA (SQL/MM)

A new ISO/IEC international standardization project for development of an SQL class library for multimedia applications was approved in early 1993. This new standardization activity, named SQL Multimedia (SQL/MM), will specify packages of SQL abstract data type (ADT) definitions using the facilities for ADT specification and invocation provided in the emerging SQL3 specification. SQL/MM intends to standardize class libraries for science and engineering, full-text and document processing, and methods for the management of multimedia objects such as image, sound, animation, music, and video. It will likely provide an SQL language binding for multimedia objects defined by other JTC1 standardization bodies (e.g. SC18 for documents, SC24 for images, and SC29 for photographs and motion pictures).

The Project Plan for SQL/MM indicates that it will be a multi-part standard consisting of an evolving number of parts. Part 1 will be a Framework that specifies how the other parts are to be constructed. Each of the other parts will be devoted to a specific SQL

application package. The following SQL/MM Part structure exists as of August 1994:

- Part 1: Framework A non-technical description of how the document is structured.
- Part 2: Full Text Methods and ADTs for text data processing. About 45 pages.
- Part 3: Spatial Methods and ADTs for spatial data management. About 200 pages with active contributions from Spatial Data experts from 3 national bodies.
- Part 4: General Purpose Methods and ADTs for complex numbers, Facilities include trig and exponential functions, vectors, sets, etc. Currently about 90 pages.

There are a number of standards efforts in the area of Spatial and Geographic information:

- ANSI X3L1 - Geographic Information Systems. Mark Ashworth of Unisys is the liason between X3L1 and ANSI X3H2. He is also the editor for parts 1, 3, and 4 of the SQL/MM draft.
- ISO TC 211 - Geographic information/Geomatics

---

24. Technical support for PostgreSQL

If you have any technical question or encounter any problem you can e-mail to:

- pgsql-questions@postgresql.org

and expect e-mail answer in less than a day. As the user-base of internet product is very vast, and users support other users, internet will be capable of giving technical support to billions of users easily. Email support is much more convenient than telephone support as you can cut and paste error messages, program output etc.. and easily transmit to mailing list/newsgroup.

In the near future, PostgreSQL organisation will be selling technical support to large/small companies, the revenue generated will be used for maintaining several mirror sites (web and ftp) around the world. The revenue will also be used to produce printed documentation, guides, textbooks which will help the customers.

You can also take help from professional consulting firms like Anderson, WGS (Work Group Solutions). Contact them for help, since they have very good expertise in "C", "C++" (PostgreSQL is written in "C") -

- Work Group Solutions http://www.wgs.com
- Anderson Consulting http://www.ac.com

25. Economic and Business Aspects

Commercial databases pay many taxes like federal, state, sales, employment, social security, medicare taxes, health care for employees, bunch of benefits for employees, marketing and advertisement costs. All these costs do not go directly for the development of the database. When you buy a commercial database, some portion of the amount goes for overheads like taxes, expenses and balance for database R&D costs.

Also commercial databases have to pay for buildings/real-estates and purchase Unix machines, install and maintain them. All of these costs are passed onto customers.

PostgreSQL has the advantage over commercial databases as there is no direct tax since it is made on the internet. A very vast group of people contribute to the development of the PostgreSQL. For example, in a hypothetical case, if there are one million companies in U.S.A and each contribute about $ 10 (worth of software to PostgreSQL) than each and every company will get ten million dollars!! This is the **GREAT MAGIC** of software development on internet.

Currently, PostgreSQL source code is about 2,00,000 lines of "C", "C++" code. If cost of each line of "C" code is $ 2 than the total cost of PostgreSQL as of today is $ 4,00,000 (four hundred thousand dollars!).

Many companies already develop in-house vast amount of "C", "C++" code. Hence by taking in the source code of PostgreSQL and collaborating with other companies on internet will greatly benefit the company saving time and efforts.

26. List of Other Databases

Listed below are other SQL databases for Unix, Linux.

- Click and go to Applications->databases. http://www.caldera.com/tech-ref/linuxapps/linapps.html
- Click and go to Applications->databases. http://www.xnet.com/~blatura/linapps.shtml
- Database resources http://linas.org/linux/db.html This was written by Linas Vepstas: linas@fc.net
- Free Database List http://cuiwww.unige.ch:80/~scg/FreeDB/FreeDB.list.html
- Browne's RDBMS List http://www.hex.net/~cbbrowne/rdbms.html written by Christopher B. Browne cbbrowne@hex.net
- SAL's List of Relational DBMS http://SAL.KachinaTech.COM/H/1/
- SAL's List of Object-Oriented DBMS http://SAL.KachinaTech.COM/H/2/
- SAL's List of Utilites and Other Databases http://SAL.KachinaTech.COM/H/3/
- ACM SIGMOD Index of Publicly Available Database Software http://bunny.cs.uiuc.edu/sigmod/databaseSoftware/

27. Internet World Wide Web Searching Tips

Internet is very vast and it has vast number of software and has a ocean of information underneath. It is growing at the rate of 300% annually world wide. It is estimated that there are about 10 million Web sites world wide!

To search for a information you would use search engines like "Yahoo", "Netscape", "Lycos" etc. Go to Yahoo, click on search. Use filtering options to narrow down your search criteria. The default search action is "Intelligent search" which is more general and lists all possiblities. Click on "Options" to select "EXACT phrase" search, "AND" search, "OR" search, etc.. This way you would find the information you need much faster. Also in the search menu, there are radio-buttons for searching in Usenet, Web-sites and Yahoo sites.

28. Conclusion

After researching all the available databases which are **free** and source code is available, it was found that ONLY PostgreSQL is the MOST mature, most widely used and robust RDBMS SQL free database (object relational) in the world.

PostgreSQL is very appealing since lot of work had already been done. It has ODBC and JDBC drivers, using these it is possible to write applications independent of the databases. The applications written in PostgreSQL using ODBC, JDBC drivers are easily portable to other databases like Oracle, Sybase and Informix and vice versa.

You may ask "But why PostgreSQL ?" The answer is, since it takes lot more time to develop a database system from scratch, it makes sense to pick up a database system which satisfies the following conditions -

A database system

- Whose source code is available - Must be a 'Open Source Code' system
- Has no license strings, no ownership strings attached to it
- Which can be distributed on internet
- Which had been on development for several years.
- Which satisfies standards like ISO/ANSI SQL 92 (and SQL 89)
- Which can satisfy future needs like SQL 3 (SQL 98)
- Which has advanced capabilities

And it just happens to be 'PostgreSQL' which satisfies all these conditions and is an appropriate software for this situation.

29. FAQ - Questions on PostgreSQL

Please refer to the latest version of FAQ for General, Linux and Irix at

- http://www.postgresql.org/docs/faq-english.shtml

30. Copyright Notice

Copyright (c) 1997 Al Dev (Alavoor Vasudevan). All rights reserved.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. In no event shall the author of this document be liable for any damages whatsoever (including without limitation, special, incidental, consequential, or direct/indirect damages for personal injury, loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of this document.

Author offers no warranties or guarantees on fitness, usability, merchantability of this document. Brands, companies and product names mentioned in this document are trademarks or registered trademarks of their respective holders. Please refer to individual copyright notices of brands, companies and products mentioned in this document. It is your responsibility to read and understand the copyright notices of the organisations/companies/products/authors mentioned in this document before using their respective information.

31. Appendix A - Syntax of ANSI/ISO SQL 1992

```
This file contains a depth-first tree traversal of the BNF
for the  language done at about 27-AUG-1992 11:03:41.64.
The specific version of the BNF included here is:  ANSI-only, SQL2-only.


<SQL terminal character> ::=
      <SQL language character>
    | <SQL embedded language character>

<SQL language character> ::=
      <simple Latin letter>
    | <digit>
    | <SQL special character>

<simple Latin letter> ::=
      <simple Latin upper case letter>
    | <simple Latin lower case letter>

<simple Latin upper case letter> ::=
          A | B | C | D | E | F | G | H | I | J | K | L | M | N | O
    | P | Q | R | S | T | U | V | W | X | Y | Z

<simple Latin lower case letter> ::=
          a | b | c | d | e | f | g | h | i | j | k | l | m | n | o
    | p | q | r | s | t | u | v | w | x | y | z

<digit> ::=
    0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<SQL special character> ::=
      <space>
    | <double quote>
    | <percent>
    | <ampersand>
    | <quote>
    | <left paren>
    | <right paren>
    | <asterisk>
    | <plus sign>
    | <comma>
    | <minus sign>
    | <period>
    | <solidus>
    | <colon>
    | <semicolon>
    | <less than operator>
    | <equals operator>
    | <greater than operator>
    | <question mark>
    | <underscore>
    | <vertical bar>
```

```
<space> ::= !! <EMPHASIS>(space character in character set in use)

<double quote> ::= "

<percent> ::= %

<ampersand> ::= &

<quote> ::= '

<left paren> ::= (

<right paren> ::= )

<asterisk> ::= *

<plus sign> ::= +

<comma> ::= ,

<minus sign> ::= -

<period> ::= .

<solidus> ::= /

<colon> ::= :

<semicolon> ::= ;

<less than operator> ::= <

<equals operator> ::= =

<greater than operator> ::= >

<question mark> ::= ?

<underscore> ::= _

<vertical bar> ::= |

<SQL embedded language character> ::=
      <left bracket>
    | <right bracket>

<left bracket> ::= [

<right bracket> ::= ]

<token> ::=
      <nondelimiter token>
    | <delimiter token>

<nondelimiter token> ::=
      <regular identifier>
    | <key word>
```

```
      | <unsigned numeric literal>
      | <national character string literal>
      | <bit string literal>
      | <hex string literal>

<regular identifier> ::= <identifier body>

<identifier body> ::=
    <identifier start> [ ( <underscore> | <identifier part> )... ]

<identifier start> ::= <EMPHASIS>(!! See the Syntax Rules)

<identifier part> ::=
      <identifier start>
    | <digit>

<key word> ::=
      <reserved word>
    | <non-reserved word>

<reserved word> ::=
      ABSOLUTE | ACTION | ADD | ALL
    | ALLOCATE | ALTER | AND
    | ANY | ARE
    | AS | ASC
    | ASSERTION | AT
    | AUTHORIZATION | AVG
    | BEGIN | BETWEEN | BIT | BIT_LENGTH
    | BOTH | BY
    | CASCADE | CASCADED | CASE | CAST
    | CATALOG
    | CHAR | CHARACTER | CHAR_LENGTH
    | CHARACTER_LENGTH | CHECK | CLOSE | COALESCE
    | COLLATE | COLLATION
    | COLUMN | COMMIT
    | CONNECT
    | CONNECTION | CONSTRAINT
    | CONSTRAINTS | CONTINUE
    | CONVERT | CORRESPONDING | COUNT | CREATE | CROSS
    | CURRENT
    | CURRENT_DATE | CURRENT_TIME
    | CURRENT_TIMESTAMP | CURRENT_USER | CURSOR
    | DATE | DAY | DEALLOCATE | DEC
    | DECIMAL | DECLARE | DEFAULT | DEFERRABLE
    | DEFERRED | DELETE | DESC | DESCRIBE | DESCRIPTOR
    | DIAGNOSTICS
    | DISCONNECT | DISTINCT | DOMAIN | DOUBLE | DROP
    | ELSE | END | END-EXEC | ESCAPE
    | EXCEPT | EXCEPTION
    | EXEC | EXECUTE | EXISTS
    | EXTERNAL | EXTRACT
    | FALSE | FETCH | FIRST | FLOAT | FOR
    | FOREIGN | FOUND | FROM | FULL
    | GET | GLOBAL | GO | GOTO
    | GRANT | GROUP
    | HAVING | HOUR
    | IDENTITY | IMMEDIATE | IN | INDICATOR
    | INITIALLY | INNER | INPUT
```

```
      | INSENSITIVE | INSERT | INT | INTEGER | INTERSECT
      | INTERVAL | INTO | IS
      | ISOLATION
      | JOIN
      | KEY
      | LANGUAGE | LAST | LEADING | LEFT
      | LEVEL | LIKE | LOCAL | LOWER
      | MATCH | MAX | MIN | MINUTE | MODULE
      | MONTH
      | NAMES | NATIONAL | NATURAL | NCHAR | NEXT | NO
      | NOT | NULL
      | NULLIF | NUMERIC
      | OCTET_LENGTH | OF
      | ON | ONLY | OPEN | OPTION | OR
      | ORDER | OUTER
      | OUTPUT | OVERLAPS
      | PAD | PARTIAL | POSITION | PRECISION | PREPARE
      | PRESERVE | PRIMARY
      | PRIOR | PRIVILEGES | PROCEDURE | PUBLIC
      | READ | REAL | REFERENCES | RELATIVE | RESTRICT
      | REVOKE | RIGHT
      | ROLLBACK | ROWS
      | SCHEMA | SCROLL | SECOND | SECTION
      | SELECT
      | SESSION | SESSION_USER | SET
      | SIZE | SMALLINT | SOME | SPACE | SQL | SQLCODE
      | SQLERROR | SQLSTATE
      | SUBSTRING | SUM | SYSTEM_USER
      | TABLE | TEMPORARY
      | THEN | TIME | TIMESTAMP
      | TIMEZONE_HOUR | TIMEZONE_MINUTE
      | TO | TRAILING | TRANSACTION
      | TRANSLATE | TRANSLATION | TRIM | TRUE
      | UNION | UNIQUE | UNKNOWN | UPDATE | UPPER | USAGE
      | USER | USING
      | VALUE | VALUES | VARCHAR | VARYING | VIEW
      | WHEN | WHENEVER | WHERE | WITH | WORK | WRITE
      | YEAR
      | ZONE

<non-reserved word> ::=

        ADA
      | C | CATALOG_NAME
      | CHARACTER_SET_CATALOG | CHARACTER_SET_NAME
      | CHARACTER_SET_SCHEMA | CLASS_ORIGIN | COBOL | COLLATION_CATALOG
      | COLLATION_NAME | COLLATION_SCHEMA | COLUMN_NAME | COMMAND_FUNCTION
      | COMMITTED
      | CONDITION_NUMBER | CONNECTION_NAME | CONSTRAINT_CATALOG | CONSTRAINT_NAME
      | CONSTRAINT_SCHEMA | CURSOR_NAME
      | DATA | DATETIME_INTERVAL_CODE
      | DATETIME_INTERVAL_PRECISION | DYNAMIC_FUNCTION
      | FORTRAN
      | LENGTH
      | MESSAGE_LENGTH | MESSAGE_OCTET_LENGTH | MESSAGE_TEXT | MORE | MUMPS
      | NAME | NULLABLE | NUMBER
      | PASCAL | PLI
      | REPEATABLE | RETURNED_LENGTH | RETURNED_OCTET_LENGTH | RETURNED_SQLSTATE
```

```
      | ROW_COUNT
      | SCALE | SCHEMA_NAME | SERIALIZABLE | SERVER_NAME | SUBCLASS_ORIGIN
      | TABLE_NAME | TYPE
      | UNCOMMITTED | UNNAMED

<unsigned numeric literal> ::=
      <exact numeric literal>
    | <approximate numeric literal>

<exact numeric literal> ::=
      <unsigned integer> [ <period> [ <unsigned integer> ] ]
    | <period> <unsigned integer>

<unsigned integer> ::= <digit>...

<approximate numeric literal> ::= <mantissa> E <exponent>

<mantissa> ::= <exact numeric literal>

<exponent> ::= <signed integer>

<signed integer> ::= [ <sign> ] <unsigned integer>

<sign> ::= <plus sign> | <minus sign>

<national character string literal> ::=
    N <quote> [ <character representation>... ] <quote>
      [ ( <separator>... <quote> [ <character representation>... ] <quote> )... ]

<character representation> ::=
      <nonquote character>
    | <quote symbol>

<nonquote character> ::= !! <EMPHASIS>(See the Syntax Rules.)

<quote symbol> ::= <quote><quote>

<separator> ::= ( <comment> | <space> | <newline> )...

<comment> ::=
    <comment introducer> [ <comment character>... ] <newline>

<comment introducer> ::= <minus sign><minus sign>[<minus sign>...]

<comment character> ::=
      <nonquote character>
    | <quote>

<newline> ::= !! <EMPHASIS>(implementation-defined end-of-line indicator)

<bit string literal> ::=
    B <quote> [ <bit>... ] <quote>
      [ ( <separator>... <quote> [ <bit>... ] <quote> )... ]

<bit> ::= 0 | 1

<hex string literal> ::=
    X <quote> [ <hexit>... ] <quote>
```

```
      [ ( <separator>... <quote> [ <hexit>... ] <quote> )... ]

<hexit> ::= <digit> | A | B | C | D | E | F | a | b | c | d | e | f

<delimiter token> ::=
      <character string literal>
    | <date string>
    | <time string>
    | <timestamp string>
    | <interval string>
    | <delimited identifier>
    | <SQL special character>
    | <not equals operator>
    | <greater than or equals operator>
    | <less than or equals operator>
    | <concatenation operator>
    | <double period>
    | <left bracket>
    | <right bracket>

<character string literal> ::=
    [ <introducer><character set specification> ]
    <quote> [ <character representation>... ] <quote>
      [ ( <separator>... <quote> [ <character representation>... ] <quote> )... ]

<introducer> ::= <underscore>

<character set specification> ::=
      <standard character repertoire name>
    | <implementation-defined character repertoire name>
    | <user-defined character repertoire name>
    | <standard universal character form-of-use name>
    | <implementation-defined universal character form-of-use name>

<standard character repertoire name> ::= <character set name>

<character set name> ::= [ <schema name> <period> ]
      <SQL language identifier>

<schema name> ::=
    [ <catalog name> <period> ] <unqualified schema name>

<catalog name> ::= <identifier>

<identifier> ::=
    [ <introducer><character set specification> ] <actual identifier>

<actual identifier> ::=
      <regular identifier>
    | <delimited identifier>

<delimited identifier> ::=
    <double quote> <delimited identifier body> <double quote>

<delimited identifier body> ::= <delimited identifier part>...

<delimited identifier part> ::=
      <nondoublequote character>
```

```
    | <doublequote symbol>

<nondoublequote character> ::= <EMPHASIS>(!! See the Syntax Rules)

<doublequote symbol> ::= <double quote><double quote>

<unqualified schema name> ::= <identifier>

<SQL language identifier> ::=
    <SQL language identifier start>
       [ ( <underscore> | <SQL language identifier part> )... ]

<SQL language identifier start> ::= <simple Latin letter>

<SQL language identifier part> ::=
      <simple Latin letter>
    | <digit>

<implementation-defined character repertoire name> ::=
    <character set name>

<user-defined character repertoire name> ::= <character set name>

<standard universal character form-of-use name> ::=
    <character set name>

<implementation-defined universal character form-of-use name> ::=
    <character set name>

<date string> ::=
    <quote> <date value> <quote>

<date value> ::=
    <years value> <minus sign> <months value>
        <minus sign> <days value>

<years value> ::= <datetime value>

<datetime value> ::= <unsigned integer>

<months value> ::= <datetime value>

<days value> ::= <datetime value>

<time string> ::=
    <quote> <time value> [ <time zone interval> ] <quote>

<time value> ::=
    <hours value> <colon> <minutes value> <colon> <seconds value>

<hours value> ::= <datetime value>

<minutes value> ::= <datetime value>

<seconds value> ::=
      <seconds integer value> [ <period> [ <seconds fraction> ] ]

<seconds integer value> ::= <unsigned integer>
```

```
<seconds fraction> ::= <unsigned integer>

<time zone interval> ::=
    <sign> <hours value> <colon> <minutes value>

<timestamp string> ::=
    <quote> <date value> <space> <time value>
        [ <time zone interval> ] <quote>

<interval string> ::=
    <quote> ( <year-month literal> | <day-time literal> ) <quote>

<year-month literal> ::=
      <years value>
    | [ <years value> <minus sign> ] <months value>

<day-time literal> ::=
      <day-time interval>
    | <time interval>

<day-time interval> ::=
    <days value>
      [ <space> <hours value> [ <colon> <minutes value>
        [ <colon> <seconds value> ] ] ]

<time interval> ::=
      <hours value> [ <colon> <minutes value> [ <colon> <seconds value> ] ]
    | <minutes value> [ <colon> <seconds value> ]
    | <seconds value>

<not equals operator> ::= <>

<greater than or equals operator> ::= >=

<less than or equals operator> ::= <=

<concatenation operator> ::= ||

<double period> ::= ..

<module> ::=
    <module name clause>
    <language clause>
    <module authorization clause>
    [ <temporary table declaration>... ]
    <module contents>...

<module name clause> ::=
    MODULE [ <module name> ]
      [ <module character set specification> ]

<module name> ::= <identifier>

<module character set specification> ::=
    NAMES ARE <character set specification>

<language clause> ::=
```

```
    LANGUAGE <language name>

<language name> ::=
    ADA | C | COBOL | FORTRAN | MUMPS | PASCAL | PLI

<module authorization clause> ::=
      SCHEMA <schema name>
    | AUTHORIZATION <module authorization identifier>
    | SCHEMA <schema name>
          AUTHORIZATION <module authorization identifier>

<module authorization identifier> ::=
    <authorization identifier>

<authorization identifier> ::= <identifier>

<temporary table declaration> ::=
    DECLARE LOCAL TEMPORARY TABLE
        <qualified local table name>
      <table element list>
      [ ON COMMIT ( PRESERVE | DELETE ) ROWS ]

<qualified local table name> ::=
    MODULE <period> <local table name>

<local table name> ::= <qualified identifier>

<qualified identifier> ::= <identifier>

<table element list> ::=
      <left paren> <table element> [ ( <comma> <table element> )... ] <right paren>

<table element> ::=
      <column definition>
    | <table constraint definition>

<column definition> ::=
    <column name> ( <data type> | <domain name> )
    [ <default clause> ]
    [ <column constraint definition>... ]
    [ <collate clause> ]

<column name> ::= <identifier>

<data type> ::=
      <character string type>
           [ CHARACTER SET <character set specification> ]
    | <national character string type>
    | <bit string type>
    | <numeric type>
    | <datetime type>
    | <interval type>

<character string type> ::=
      CHARACTER [ <left paren> <length> <right paren> ]
    | CHAR [ <left paren> <length> <right paren> ]
    | CHARACTER VARYING <left paren> <length> <right paren>
    | CHAR VARYING <left paren> <length> <right paren>
```

```
      | VARCHAR <left paren> <length> <right paren>

<length> ::= <unsigned integer>

<national character string type> ::=
      NATIONAL CHARACTER [ <left paren> <length> <right paren> ]
    | NATIONAL CHAR [ <left paren> <length> <right paren> ]
    | NCHAR [ <left paren> <length> <right paren> ]
    | NATIONAL CHARACTER VARYING <left paren> <length> <right paren>
    | NATIONAL CHAR VARYING <left paren> <length> <right paren>
    | NCHAR VARYING <left paren> <length> <right paren>

<bit string type> ::=
      BIT [ <left paren> <length> <right paren> ]
    | BIT VARYING <left paren> <length> <right paren>

<numeric type> ::=
      <exact numeric type>
    | <approximate numeric type>

<exact numeric type> ::=
      NUMERIC [ <left paren> <precision> [ <comma> <scale> ] <right paren> ]
    | DECIMAL [ <left paren> <precision> [ <comma> <scale> ] <right paren> ]
    | DEC [ <left paren> <precision> [ <comma> <scale> ] <right paren> ]
    | INTEGER
    | INT
    | SMALLINT

<precision> ::= <unsigned integer>

<scale> ::= <unsigned integer>

<approximate numeric type> ::=
      FLOAT [ <left paren> <precision> <right paren> ]
    | REAL
    | DOUBLE PRECISION

<datetime type> ::=
      DATE
    | TIME [ <left paren> <time precision> <right paren> ]
          [ WITH TIME ZONE ]
    | TIMESTAMP [ <left paren> <timestamp precision> <right paren> ]
          [ WITH TIME ZONE ]

<time precision> ::= <time fractional seconds precision>

<time fractional seconds precision> ::= <unsigned integer>

<timestamp precision> ::= <time fractional seconds precision>

<interval type> ::= INTERVAL <interval qualifier>

<interval qualifier> ::=
      <start field> TO <end field>
    | <single datetime field>

<start field> ::=
    <non-second datetime field>
```

```
          [ <left paren> <interval leading field precision> <right paren> ]

<non-second datetime field> ::= YEAR | MONTH | DAY | HOUR
    | MINUTE

<interval leading field precision> ::= <unsigned integer>

<end field> ::=
      <non-second datetime field>
    | SECOND [ <left paren> <interval fractional seconds precision> <right paren> ]

<interval fractional seconds precision> ::= <unsigned integer>

<single datetime field> ::=
      <non-second datetime field>
          [ <left paren> <interval leading field precision> <right paren> ]
    | SECOND [ <left paren> <interval leading field precision>
          [ <comma> <interval fractional seconds precision> ] <right paren> ]

<domain name> ::= <qualified name>

<qualified name> ::=
    [ <schema name> <period> ] <qualified identifier>

<default clause> ::=
      DEFAULT <default option>

<default option> ::=
      <literal>
    | <datetime value function>
    | USER
    | CURRENT_USER
    | SESSION_USER
    | SYSTEM_USER
    | NULL

<literal> ::=
      <signed numeric literal>
    | <general literal>

<signed numeric literal> ::=
    [ <sign> ] <unsigned numeric literal>

<general literal> ::=
      <character string literal>
    | <national character string literal>
    | <bit string literal>
    | <hex string literal>
    | <datetime literal>
    | <interval literal>

<datetime literal> ::=
      <date literal>
    | <time literal>
    | <timestamp literal>

<date literal> ::=
    DATE <date string>
```

```
<time literal> ::=
    TIME <time string>

<timestamp literal> ::=
    TIMESTAMP <timestamp string>

<interval literal> ::=
    INTERVAL [ <sign> ] <interval string> <interval qualifier>

<datetime value function> ::=
      <current date value function>
    | <current time value function>
    | <current timestamp value function>

<current date value function> ::= CURRENT_DATE

<current time value function> ::=
      CURRENT_TIME [ <left paren> <time precision> <right paren> ]

<current timestamp value function> ::=
      CURRENT_TIMESTAMP [ <left paren> <timestamp precision> <right paren> ]

<column constraint definition> ::=
    [ <constraint name definition> ]
    <column constraint>
      [ <constraint attributes> ]

<constraint name definition> ::= CONSTRAINT <constraint name>

<constraint name> ::= <qualified name>

<column constraint> ::=
      NOT NULL
    | <unique specification>
    | <references specification>
    | <check constraint definition>

<unique specification> ::=
    UNIQUE | PRIMARY KEY

<references specification> ::=
    REFERENCES <referenced table and columns>
      [ MATCH <match type> ]
      [ <referential triggered action> ]

<referenced table and columns> ::=
     <table name> [ <left paren> <reference column list> <right paren> ]

<table name> ::=
      <qualified name>
    | <qualified local table name>

<reference column list> ::= <column name list>

<column name list> ::=
    <column name> [ ( <comma> <column name> )... ]
```

```
<match type> ::=
      FULL
    | PARTIAL

<referential triggered action> ::=
      <update rule> [ <delete rule> ]
    | <delete rule> [ <update rule> ]

<update rule> ::= ON UPDATE <referential action>

<referential action> ::=
      CASCADE
    | SET NULL
    | SET DEFAULT
    | NO ACTION

<delete rule> ::= ON DELETE <referential action>

<check constraint definition> ::=
    CHECK
        <left paren> <search condition> <right paren>

<search condition> ::=
      <boolean term>
    | <search condition> OR <boolean term>

<boolean term> ::=
      <boolean factor>
    | <boolean term> AND <boolean factor>

<boolean factor> ::=
    [ NOT ] <boolean test>

<boolean test> ::=
    <boolean primary> [ IS [ NOT ]
          <truth value> ]

<boolean primary> ::=
      <predicate>
    | <left paren> <search condition> <right paren>

<predicate> ::=
      <comparison predicate>
    | <between predicate>
    | <in predicate>
    | <like predicate>
    | <null predicate>
    | <quantified comparison predicate>
    | <exists predicate>
    | <unique predicate>
    | <match predicate>
    | <overlaps predicate>

<comparison predicate> ::=
    <row value constructor> <comp op>
        <row value constructor>

<row value constructor> ::=
```

```
         <row value constructor element>
     | <left paren> <row value constructor list> <right paren>
     | <row subquery>

<row value constructor element> ::=
      <value expression>
     | <null specification>
     | <default specification>

<value expression> ::=
      <numeric value expression>
     | <string value expression>
     | <datetime value expression>
     | <interval value expression>

<numeric value expression> ::=
      <term>
     | <numeric value expression> <plus sign> <term>
     | <numeric value expression> <minus sign> <term>

<term> ::=
      <factor>
     | <term> <asterisk> <factor>
     | <term> <solidus> <factor>

<factor> ::=
    [ <sign> ] <numeric primary>

<numeric primary> ::=
      <value expression primary>
     | <numeric value function>

<value expression primary> ::=
      <unsigned value specification>
     | <column reference>
     | <set function specification>
     | <scalar subquery>
     | <case expression>
     | <left paren> <value expression> <right paren>
     | <cast specification>

<unsigned value specification> ::=
      <unsigned literal>
     | <general value specification>

<unsigned literal> ::=
      <unsigned numeric literal>
     | <general literal>

<general value specification> ::=
      <parameter specification>
     | <dynamic parameter specification>
     | <variable specification>
     | USER
     | CURRENT_USER
     | SESSION_USER
     | SYSTEM_USER
     | VALUE
```

```
<parameter specification> ::=
    <parameter name> [ <indicator parameter> ]

<parameter name> ::= <colon> <identifier>

<indicator parameter> ::=
    [ INDICATOR ] <parameter name>

<dynamic parameter specification> ::= <question mark>

<variable specification> ::=
    <embedded variable name> [ <indicator variable> ]

<embedded variable name> ::=
    <colon><host identifier>

<host identifier> ::=
      <Ada host identifier>
    | <C host identifier>
    | <COBOL host identifier>
    | <Fortran host identifier>
    | <MUMPS host identifier>
    | <Pascal host identifier>
    | <PL/I host identifier>

<Ada host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<C host identifier> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<COBOL host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<Fortran host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<MUMPS host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<Pascal host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<PL/I host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<indicator variable> ::=
    [ INDICATOR ] <embedded variable name>

<column reference> ::= [ <qualifier> <period> ] <column name>

<qualifier> ::=
      <table name>
    | <correlation name>

<correlation name> ::= <identifier>

<set function specification> ::=
      COUNT <left paren> <asterisk> <right paren>
    | <general set function>

<general set function> ::=
      <set function type>
```

```
            <left paren> [ <set quantifier> ] <value expression> <right paren>

<set function type> ::=
    AVG | MAX | MIN | SUM | COUNT

<set quantifier> ::= DISTINCT | ALL

<scalar subquery> ::= <subquery>

<subquery> ::= <left paren> <query expression> <right paren>

<query expression> ::=
      <non-join query expression>
    | <joined table>

<non-join query expression> ::=
      <non-join query term>
    | <query expression> UNION  [ ALL ]
          [ <corresponding spec> ] <query term>
    | <query expression> EXCEPT [ ALL ]
          [ <corresponding spec> ] <query term>

<non-join query term> ::=
      <non-join query primary>
    | <query term> INTERSECT [ ALL ]
          [ <corresponding spec> ] <query primary>

<non-join query primary> ::=
      <simple table>
    | <left paren> <non-join query expression> <right paren>

<simple table> ::=
      <query specification>
    | <table value constructor>
    | <explicit table>

<query specification> ::=
    SELECT [ <set quantifier> ] <select list> <table expression>

<select list> ::=
      <asterisk>
    | <select sublist> [ ( <comma> <select sublist> )... ]

<select sublist> ::=
      <derived column>
    | <qualifier> <period> <asterisk>

<derived column> ::= <value expression> [ <as clause> ]

<as clause> ::= [ AS ] <column name>

<table expression> ::=
    <from clause>
    [ <where clause> ]
    [ <group by clause> ]
    [ <having clause> ]

<from clause> ::= FROM <table reference>
```

```
    [ ( <comma> <table reference> )... ]

<table reference> ::=
      <table name> [ [ AS ] <correlation name>
          [ <left paren> <derived column list> <right paren> ] ]
    | <derived table> [ AS ] <correlation name>
          [ <left paren> <derived column list> <right paren> ]
    | <joined table>

<derived column list> ::= <column name list>

<derived table> ::= <table subquery>

<table subquery> ::= <subquery>

<joined table> ::=
      <cross join>
    | <qualified join>
    | <left paren> <joined table> <right paren>

<cross join> ::=
    <table reference> CROSS JOIN <table reference>

<qualified join> ::=
    <table reference> [ NATURAL ] [ <join type> ] JOIN
      <table reference> [ <join specification> ]

<join type> ::=
      INNER
    | <outer join type> [ OUTER ]
    | UNION

<outer join type> ::=
      LEFT
    | RIGHT
    | FULL

<join specification> ::=
      <join condition>
    | <named columns join>

<join condition> ::= ON <search condition>

<named columns join> ::=
    USING <left paren> <join column list> <right paren>

<join column list> ::= <column name list>

<where clause> ::= WHERE <search condition>

<group by clause> ::=
    GROUP BY <grouping column reference list>

<grouping column reference list> ::=
    <grouping column reference>
        [ ( <comma> <grouping column reference> )... ]

<grouping column reference> ::=
```

```
    <column reference> [ <collate clause> ]

<collate clause> ::= COLLATE <collation name>

<collation name> ::= <qualified name>

<having clause> ::= HAVING <search condition>

<table value constructor> ::=
    VALUES <table value constructor list>

<table value constructor list> ::=
    <row value constructor> [ ( <comma> <row value constructor> )... ]

<explicit table> ::= TABLE <table name>

<query term> ::=
      <non-join query term>
    | <joined table>

<corresponding spec> ::=
    CORRESPONDING [ BY <left paren> <corresponding column list> <right paren> ]

<corresponding column list> ::= <column name list>

<query primary> ::=
      <non-join query primary>
    | <joined table>

<case expression> ::=
      <case abbreviation>
    | <case specification>

<case abbreviation> ::=
      NULLIF <left paren> <value expression> <comma>
            <value expression> <right paren>
    | COALESCE <left paren> <value expression>
            ( <comma> <value expression> )... <right paren>

<case specification> ::=
      <simple case>
    | <searched case>

<simple case> ::=
    CASE <case operand>
      <simple when clause>...
      [ <else clause> ]
    END

<case operand> ::= <value expression>

<simple when clause> ::= WHEN <when operand> THEN <result>

<when operand> ::= <value expression>

<result> ::= <result expression> | NULL

<result expression> ::= <value expression>
```

```
<else clause> ::= ELSE <result>

<searched case> ::=
    CASE
      <searched when clause>...
      [ <else clause> ]
    END

<searched when clause> ::= WHEN <search condition> THEN <result>

<cast specification> ::=
    CAST <left paren> <cast operand> AS
        <cast target> <right paren>

<cast operand> ::=
      <value expression>
    | NULL

<cast target> ::=
      <domain name>
    | <data type>

<numeric value function> ::=
      <position expression>
    | <extract expression>
    | <length expression>

<position expression> ::=
    POSITION <left paren> <character value expression>
        IN <character value expression> <right paren>

<character value expression> ::=
      <concatenation>
    | <character factor>

<concatenation> ::=
    <character value expression> <concatenation operator>
        <character factor>

<character factor> ::=
    <character primary> [ <collate clause> ]

<character primary> ::=
      <value expression primary>
    | <string value function>

<string value function> ::=
      <character value function>
    | <bit value function>

<character value function> ::=
      <character substring function>
    | <fold>
    | <form-of-use conversion>
    | <character translation>
    | <trim function>
```

```
<character substring function> ::=
    SUBSTRING <left paren> <character value expression> FROM <start position>
                [ FOR <string length> ] <right paren>

<start position> ::= <numeric value expression>

<string length> ::= <numeric value expression>

<fold> ::= ( UPPER | LOWER )
     <left paren> <character value expression> <right paren>

<form-of-use conversion> ::=
    CONVERT <left paren> <character value expression>
        USING <form-of-use conversion name> <right paren>

<form-of-use conversion name> ::= <qualified name>

<character translation> ::=
    TRANSLATE <left paren> <character value expression>
        USING <translation name> <right paren>

<translation name> ::= <qualified name>

<trim function> ::=
    TRIM <left paren> <trim operands> <right paren>

<trim operands> ::=
    [ [ <trim specification> ] [ <trim character> ] FROM ] <trim source>

<trim specification> ::=
      LEADING
    | TRAILING
    | BOTH

<trim character> ::= <character value expression>

<trim source> ::= <character value expression>

<bit value function> ::=
    <bit substring function>

<bit substring function> ::=
    SUBSTRING <left paren> <bit value expression> FROM <start position>
        [ FOR <string length> ] <right paren>

<bit value expression> ::=
      <bit concatenation>
    | <bit factor>

<bit concatenation> ::=
    <bit value expression> <concatenation operator> <bit factor>

<bit factor> ::= <bit primary>

<bit primary> ::=
      <value expression primary>
    | <string value function>
```

```
<extract expression> ::=
    EXTRACT <left paren> <extract field>
        FROM <extract source> <right paren>

<extract field> ::=
      <datetime field>
    | <time zone field>

<datetime field> ::=
      <non-second datetime field>
    | SECOND

<time zone field> ::=
      TIMEZONE_HOUR
    | TIMEZONE_MINUTE

<extract source> ::=
      <datetime value expression>
    | <interval value expression>

<datetime value expression> ::=
      <datetime term>
    | <interval value expression> <plus sign> <datetime term>
    | <datetime value expression> <plus sign> <interval term>
    | <datetime value expression> <minus sign> <interval term>

<interval term> ::=
      <interval factor>
    | <interval term 2> <asterisk> <factor>
    | <interval term 2> <solidus> <factor>
    | <term> <asterisk> <interval factor>

<interval factor> ::=
    [ <sign> ] <interval primary>

<interval primary> ::=
      <value expression primary> [ <interval qualifier> ]

<interval term 2> ::= <interval term>

<interval value expression> ::=
      <interval term>
    | <interval value expression 1> <plus sign> <interval term 1>
    | <interval value expression 1> <minus sign> <interval term 1>
    | <left paren> <datetime value expression> <minus sign>
          <datetime term> <right paren> <interval qualifier>

<interval value expression 1> ::= <interval value expression>

<interval term 1> ::= <interval term>

<datetime term> ::=
      <datetime factor>

<datetime factor> ::=
      <datetime primary> [ <time zone> ]

<datetime primary> ::=
```

```
        <value expression primary>
      | <datetime value function>

<time zone> ::=
    AT <time zone specifier>

<time zone specifier> ::=
        LOCAL
      | TIME ZONE <interval value expression>

<length expression> ::=
        <char length expression>
      | <octet length expression>
      | <bit length expression>

<char length expression> ::=
    ( CHAR_LENGTH | CHARACTER_LENGTH )
        <left paren> <string value expression> <right paren>

<string value expression> ::=
        <character value expression>
      | <bit value expression>

<octet length expression> ::=
    OCTET_LENGTH <left paren> <string value expression> <right paren>

<bit length expression> ::=
    BIT_LENGTH <left paren> <string value expression> <right paren>

<null specification> ::=
    NULL

<default specification> ::=
    DEFAULT

<row value constructor list> ::=
    <row value constructor element>
        [ ( <comma> <row value constructor element> )... ]

<row subquery> ::= <subquery>

<comp op> ::=
        <equals operator>
      | <not equals operator>
      | <less than operator>
      | <greater than operator>
      | <less than or equals operator>
      | <greater than or equals operator>

<between predicate> ::=
    <row value constructor> [ NOT ] BETWEEN
        <row value constructor> AND <row value constructor>

<in predicate> ::=
    <row value constructor>
        [ NOT ] IN <in predicate value>

<in predicate value> ::=
```

```
        <table subquery>
      | <left paren> <in value list> <right paren>

<in value list> ::=
    <value expression> ( <comma> <value expression> )...

<like predicate> ::=
    <match value> [ NOT ] LIKE <pattern>
      [ ESCAPE <escape character> ]

<match value> ::= <character value expression>

<pattern> ::= <character value expression>

<escape character> ::= <character value expression>

<null predicate> ::= <row value constructor>
    IS [ NOT ] NULL

<quantified comparison predicate> ::=
    <row value constructor> <comp op> <quantifier> <table subquery>

<quantifier> ::= <all> | <some>

<all> ::= ALL

<some> ::= SOME | ANY

<exists predicate> ::= EXISTS <table subquery>

<unique predicate> ::= UNIQUE <table subquery>

<match predicate> ::=
    <row value constructor> MATCH [ UNIQUE ]
        [ PARTIAL | FULL ] <table subquery>

<overlaps predicate> ::=
    <row value constructor 1> OVERLAPS <row value constructor 2>

<row value constructor 1> ::= <row value constructor>

<row value constructor 2> ::= <row value constructor>

<truth value> ::=
      TRUE
    | FALSE
    | UNKNOWN

<constraint attributes> ::=
      <constraint check time> [ [ NOT ] DEFERRABLE ]
    | [ NOT ] DEFERRABLE [ <constraint check time> ]

<constraint check time> ::=
      INITIALLY DEFERRED
    | INITIALLY IMMEDIATE

<table constraint definition> ::=
    [ <constraint name definition> ]
```

```
    <table constraint> [ <constraint attributes> ]

<table constraint> ::=
      <unique constraint definition>
    | <referential constraint definition>
    | <check constraint definition>

<unique constraint definition> ::=
            <unique specification> even in SQL3)
    <unique specification>
      <left paren> <unique column list> <right paren>

<unique column list> ::= <column name list>

<referential constraint definition> ::=
    FOREIGN KEY
        <left paren> <referencing columns> <right paren>
      <references specification>

<referencing columns> ::=
    <reference column list>

<module contents> ::=
      <declare cursor>
    | <dynamic declare cursor>
    | <procedure>

<declare cursor> ::=
    DECLARE <cursor name> [ INSENSITIVE ] [ SCROLL ] CURSOR
      FOR <cursor specification>

<cursor name> ::= <identifier>

<cursor specification> ::=
    <query expression> [ <order by clause> ]
      [ <updatability clause> ]

<order by clause> ::=
    ORDER BY <sort specification list>

<sort specification list> ::=
    <sort specification> [ ( <comma> <sort specification> )... ]

<sort specification> ::=
    <sort key> [ <collate clause> ] [ <ordering specification> ]

<sort key> ::=
      <column name>
    | <unsigned integer>

<ordering specification> ::= ASC | DESC

<updatability clause> ::=
    FOR
        ( READ ONLY |
          UPDATE [ OF <column name list> ] )

<dynamic declare cursor> ::=
```

```
    DECLARE <cursor name> [ INSENSITIVE ] [ SCROLL ] CURSOR
        FOR <statement name>

<statement name> ::= <identifier>

<procedure> ::=
    PROCEDURE <procedure name>
        <parameter declaration list> <semicolon>
      <SQL procedure statement> <semicolon>

<procedure name> ::= <identifier>

<parameter declaration list> ::=
      <left paren> <parameter declaration>
          [ ( <comma> <parameter declaration> )... ] <right paren>
    | <parameter declaration>...

<parameter declaration> ::=
      <parameter name> <data type>
    | <status parameter>

<status parameter> ::=
    SQLCODE | SQLSTATE

<SQL procedure statement> ::=
      <SQL schema statement>
    | <SQL data statement>
    | <SQL transaction statement>
    | <SQL connection statement>
    | <SQL session statement>
    | <SQL dynamic statement>
    | <SQL diagnostics statement>

<SQL schema statement> ::=
      <SQL schema definition statement>
    | <SQL schema manipulation statement>

<SQL schema definition statement> ::=
      <schema definition>
    | <table definition>
    | <view definition>
    | <grant statement>
    | <domain definition>
    | <character set definition>
    | <collation definition>
    | <translation definition>
    | <assertion definition>

<schema definition> ::=
    CREATE SCHEMA <schema name clause>
      [ <schema character set specification> ]
      [ <schema element>... ]

<schema name clause> ::=
      <schema name>
    | AUTHORIZATION <schema authorization identifier>
    | <schema name> AUTHORIZATION
          <schema authorization identifier>
```

```
<schema authorization identifier> ::=
    <authorization identifier>

<schema character set specification> ::=
    DEFAULT CHARACTER
        SET <character set specification>

<schema element> ::=
      <domain definition>
    | <table definition>
    | <view definition>
    | <grant statement>
    | <assertion definition>
    | <character set definition>
    | <collation definition>
    | <translation definition>

<domain definition> ::=
    CREATE DOMAIN <domain name>
        [ AS ] <data type>
      [ <default clause> ]
      [ <domain constraint>... ]
      [ <collate clause> ]

<domain constraint> ::=
    [ <constraint name definition> ]
    <check constraint definition> [ <constraint attributes> ]

<table definition> ::=
    CREATE [ ( GLOBAL | LOCAL ) TEMPORARY ] TABLE
        <table name>
      <table element list>
      [ ON COMMIT ( DELETE | PRESERVE ) ROWS ]

<view definition> ::=
    CREATE VIEW <table name> [ <left paren> <view column list>
                                  <right paren> ]
      AS <query expression>
      [ WITH [ <levels clause> ] CHECK OPTION ]

<view column list> ::= <column name list>

<levels clause> ::=
    CASCADED | LOCAL

<grant statement> ::=
   GRANT <privileges> ON <object name>
     TO <grantee> [ ( <comma> <grantee> )... ]
       [ WITH GRANT OPTION ]

<privileges> ::=
      ALL PRIVILEGES
    | <action list>

<action list> ::= <action> [ ( <comma> <action> )... ]

<action> ::=
```

```
        SELECT
      | DELETE
      | INSERT [ <left paren> <privilege column list> <right paren> ]
      | UPDATE [ <left paren> <privilege column list> <right paren> ]
      | REFERENCES [ <left paren> <privilege column list> <right paren> ]
      | USAGE

<privilege column list> ::= <column name list>

<object name> ::=
        [ TABLE ] <table name>
      | DOMAIN <domain name>
      | COLLATION <collation name>
      | CHARACTER SET <character set name>
      | TRANSLATION <translation name>

<grantee> ::=
        PUBLIC
      | <authorization identifier>

<assertion definition> ::=
    CREATE ASSERTION <constraint name> <assertion check>
        [ <constraint attributes> ]

<assertion check> ::=
    CHECK
          <left paren> <search condition> <right paren>

<character set definition> ::=
    CREATE CHARACTER SET <character set name>
          [ AS ]
        <character set source>
        [ <collate clause> | <limited collation definition> ]

<character set source> ::=
        GET <existing character set name>

<existing character set name> ::=
        <standard character repertoire name>
      | <implementation-defined character repertoire name>
      | <schema character set name>

<schema character set name> ::= <character set name>

<limited collation definition> ::=
    COLLATION FROM <collation source>

<collation source> ::=
        <collating sequence definition>
      | <translation collation>

<collating sequence definition> ::=
        <external collation>
      | <schema collation name>
      | DESC <left paren> <collation name> <right paren>
      | DEFAULT

<external collation> ::=
```

```
    EXTERNAL <left paren> <quote> <external collation name> <quote> <right paren>

<external collation name> ::=
      <standard collation name>
    | <implementation-defined collation name>

<standard collation name> ::= <collation name>

<implementation-defined collation name> ::= <collation name>

<schema collation name> ::= <collation name>

<translation collation> ::=
    TRANSLATION <translation name>
        [ THEN COLLATION <collation name> ]

<collation definition> ::=
    CREATE COLLATION <collation name> FOR
        <character set specification>
      FROM <collation source>
        [ <pad attribute> ]

<pad attribute> ::=
      NO PAD
    | PAD SPACE

<translation definition> ::=
    CREATE TRANSLATION <translation name>
      FOR <source character set specification>
        TO <target character set specification>
      FROM <translation source>

<source character set specification> ::= <character set specification>

<target character set specification> ::= <character set specification>

<translation source> ::=
      <translation specification>

<translation specification> ::=
      <external translation>
    | IDENTITY
    | <schema translation name>

<external translation> ::=
    EXTERNAL <left paren> <quote> <external translation name> <quote> <right paren>

<external translation name> ::=
      <standard translation name>
    | <implementation-defined translation name>

<standard translation name> ::= <translation name>

<implementation-defined translation name> ::= <translation name>

<schema translation name> ::= <translation name>

<SQL schema manipulation statement> ::=
```

```
        <drop schema statement>
      | <alter table statement>
      | <drop table statement>
      | <drop view statement>
      | <revoke statement>
      | <alter domain statement>
      | <drop domain statement>
      | <drop character set statement>
      | <drop collation statement>
      | <drop translation statement>
      | <drop assertion statement>

<drop schema statement> ::=
    DROP SCHEMA <schema name> <drop behavior>

<drop behavior> ::= CASCADE | RESTRICT

<alter table statement> ::=
    ALTER TABLE <table name> <alter table action>

<alter table action> ::=
        <add column definition>
      | <alter column definition>
      | <drop column definition>
      | <add table constraint definition>
      | <drop table constraint definition>

<add column definition> ::=
    ADD [ COLUMN ] <column definition>

<alter column definition> ::=
    ALTER [ COLUMN ] <column name> <alter column action>

<alter column action> ::=
        <set column default clause>
      | <drop column default clause>

<set column default clause> ::=
    SET <default clause>

<drop column default clause> ::=
    DROP DEFAULT

<drop column definition> ::=
    DROP [ COLUMN ] <column name> <drop behavior>

<add table constraint definition> ::=
    ADD <table constraint definition>

<drop table constraint definition> ::=
    DROP CONSTRAINT <constraint name> <drop behavior>

<drop table statement> ::=
    DROP TABLE <table name> <drop behavior>

<drop view statement> ::=
    DROP VIEW <table name> <drop behavior>
```

```
<revoke statement> ::=
    REVOKE [ GRANT OPTION FOR ]
        <privileges>
        ON <object name>
      FROM <grantee> [ ( <comma> <grantee> )... ] <drop behavior>

<alter domain statement> ::=
    ALTER DOMAIN <domain name> <alter domain action>

<alter domain action> ::=
      <set domain default clause>
    | <drop domain default clause>
    | <add domain constraint definition>
    | <drop domain constraint definition>

<set domain default clause> ::= SET <default clause>

<drop domain default clause> ::= DROP DEFAULT

<add domain constraint definition> ::=
    ADD <domain constraint>

<drop domain constraint definition> ::=
    DROP CONSTRAINT <constraint name>

<drop domain statement> ::=
    DROP DOMAIN <domain name> <drop behavior>

<drop character set statement> ::=
    DROP CHARACTER SET <character set name>

<drop collation statement> ::=
    DROP COLLATION <collation name>

<drop translation statement> ::=
    DROP TRANSLATION <translation name>

<drop assertion statement> ::=
    DROP ASSERTION <constraint name>

<SQL data statement> ::=
      <open statement>
    | <fetch statement>
    | <close statement>
    | <select statement: single row>
    | <SQL data change statement>

<open statement> ::=
    OPEN <cursor name>

<fetch statement> ::=
    FETCH [ [ <fetch orientation> ] FROM ]
      <cursor name> INTO <fetch target list>

<fetch orientation> ::=
      NEXT
    | PRIOR
    | FIRST
```

```
      | LAST
      | ( ABSOLUTE | RELATIVE ) <simple value specification>

<simple value specification> ::=
        <parameter name>
      | <embedded variable name>
      | <literal>

<fetch target list> ::=
      <target specification> [ ( <comma> <target specification> )... ]

<target specification> ::=
        <parameter specification>
      | <variable specification>

<close statement> ::=
      CLOSE <cursor name>

<select statement: single row> ::=
      SELECT [ <set quantifier> ] <select list>
        INTO <select target list>
          <table expression>

<select target list> ::=
      <target specification> [ ( <comma> <target specification> )... ]

<SQL data change statement> ::=
        <delete statement: positioned>
      | <delete statement: searched>
      | <insert statement>
      | <update statement: positioned>
      | <update statement: searched>

<delete statement: positioned> ::=
      DELETE FROM <table name>
        WHERE CURRENT OF <cursor name>

<delete statement: searched> ::=
      DELETE FROM <table name>
        [ WHERE <search condition> ]

<insert statement> ::=
      INSERT INTO <table name>
        <insert columns and source>

<insert columns and source> ::=
        [ <left paren> <insert column list> <right paren> ]
              <query expression>
      | DEFAULT VALUES

<insert column list> ::= <column name list>

<update statement: positioned> ::=
      UPDATE <table name>
        SET <set clause list>
          WHERE CURRENT OF <cursor name>

<set clause list> ::=
```

```
    <set clause> [ ( <comma> <set clause> )... ]

<set clause> ::=
    <object column> <equals operator> <update source>

<object column> ::= <column name>

<update source> ::=
      <value expression>
    | <null specification>
    | DEFAULT

<update statement: searched> ::=
    UPDATE <table name>
      SET <set clause list>
      [ WHERE <search condition> ]

<SQL transaction statement> ::=
      <set transaction statement>
    | <set constraints mode statement>
    | <commit statement>
    | <rollback statement>

<set transaction statement> ::=
    SET TRANSACTION <transaction mode>
        [ ( <comma> <transaction mode> )... ]

<transaction mode> ::=
      <isolation level>
    | <transaction access mode>
    | <diagnostics size>

<isolation level> ::=
    ISOLATION LEVEL <level of isolation>

<level of isolation> ::=
      READ UNCOMMITTED
    | READ COMMITTED
    | REPEATABLE READ
    | SERIALIZABLE

<transaction access mode> ::=
      READ ONLY
    | READ WRITE

<diagnostics size> ::=
    DIAGNOSTICS SIZE <number of conditions>

<number of conditions> ::= <simple value specification>

<set constraints mode statement> ::=
    SET CONSTRAINTS <constraint name list>
        ( DEFERRED | IMMEDIATE )

<constraint name list> ::=
      ALL
    | <constraint name> [ ( <comma> <constraint name> )... ]
```

```
<commit statement> ::=
    COMMIT [ WORK ]

<rollback statement> ::=
    ROLLBACK [ WORK ]

<SQL connection statement> ::=
      <connect statement>
    | <set connection statement>
    | <disconnect statement>

<connect statement> ::=
    CONNECT TO <connection target>

<connection target> ::=
      <SQL-server name>
        [ AS <connection name> ]
          correspondence with Tony Gordon)
        [ USER <user name> ]
    | DEFAULT

<SQL-server name> ::= <simple value specification>

<connection name> ::= <simple value specification>

<user name> ::= <simple value specification>

<set connection statement> ::=
    SET CONNECTION <connection object>

<connection object> ::=
      DEFAULT
    | <connection name>

<disconnect statement> ::=
    DISCONNECT <disconnect object>

<disconnect object> ::=
      <connection object>
    | ALL
    | CURRENT

<SQL session statement> ::=
      <set catalog statement>
    | <set schema statement>
    | <set names statement>
    | <set session authorization identifier statement>
    | <set local time zone statement>

<set catalog statement> ::=
    SET CATALOG <value specification>

<value specification> ::=
      <literal>
    | <general value specification>

<set schema statement> ::=
    SET SCHEMA <value specification>
```

```
<set names statement> ::=
    SET NAMES <value specification>

<set session authorization identifier statement> ::=
    SET SESSION AUTHORIZATION
        <value specification>

<set local time zone statement> ::=
    SET TIME ZONE
        <set time zone value>

<set time zone value> ::=
      <interval value expression>
    | LOCAL

<SQL dynamic statement> ::=
      <system descriptor statement>
    | <prepare statement>
    | <deallocate prepared statement>
    | <describe statement>
    | <execute statement>
    | <execute immediate statement>
    | <SQL dynamic data statement>

<system descriptor statement> ::=
      <allocate descriptor statement>
    | <deallocate descriptor statement>
    | <set descriptor statement>
    | <get descriptor statement>

<allocate descriptor statement> ::=
    ALLOCATE DESCRIPTOR <descriptor name>
       [ WITH MAX <occurrences> ]

<descriptor name> ::=
    [ <scope option> ] <simple value specification>

<scope option> ::=
      GLOBAL
    | LOCAL

<occurrences> ::= <simple value specification>

<deallocate descriptor statement> ::=
    DEALLOCATE DESCRIPTOR <descriptor name>

<set descriptor statement> ::=
    SET DESCRIPTOR <descriptor name>
        <set descriptor information>

<set descriptor information> ::=
      <set count>
    | VALUE <item number>
        <set item information> [ ( <comma> <set item information> )... ]

<set count> ::=
    COUNT <equals operator> <simple value specification 1>
```

```
<simple value specification 1> ::= <simple value specification>

<item number> ::= <simple value specification>

<set item information> ::=
    <descriptor item name> <equals operator> <simple value specification 2>

<descriptor item name> ::=
      TYPE
    | LENGTH
    | OCTET_LENGTH
    | RETURNED_LENGTH
    | RETURNED_OCTET_LENGTH
    | PRECISION
    | SCALE
    | DATETIME_INTERVAL_CODE
    | DATETIME_INTERVAL_PRECISION
    | NULLABLE
    | INDICATOR
    | DATA
    | NAME
    | UNNAMED
    | COLLATION_CATALOG
    | COLLATION_SCHEMA
    | COLLATION_NAME
    | CHARACTER_SET_CATALOG
    | CHARACTER_SET_SCHEMA
    | CHARACTER_SET_NAME

<simple value specification 2> ::= <simple value specification>

<item number> ::= <simple value specification>

<get descriptor statement> ::=
    GET DESCRIPTOR <descriptor name> <get descriptor information>

<get descriptor information> ::=
      <get count>
    | VALUE <item number>
        <get item information> [ ( <comma> <get item information> )... ]

<get count> ::=
    <simple target specification 1> <equals operator>
         COUNT

<simple target specification 1> ::= <simple target specification>

<simple target specification> ::=
      <parameter name>
    | <embedded variable name>

<get item information> ::=
    <simple target specification 2> <equals operator> <descriptor item name>>

<simple target specification 2> ::= <simple target specification>

<prepare statement> ::=
```

```
    PREPARE <SQL statement name> FROM <SQL statement variable>

<SQL statement name> ::=
      <statement name>
    | <extended statement name>

<extended statement name> ::=
    [ <scope option> ] <simple value specification>

<SQL statement variable> ::= <simple value specification>

<deallocate prepared statement> ::=
    DEALLOCATE PREPARE <SQL statement name>

<describe statement> ::=
      <describe input statement>
    | <describe output statement>

<describe input statement> ::=
    DESCRIBE INPUT <SQL statement name> <using descriptor>

<using descriptor> ::=
    ( USING | INTO ) SQL DESCRIPTOR <descriptor name>

<describe output statement> ::=
    DESCRIBE [ OUTPUT ] <SQL statement name> <using descriptor>

<execute statement> ::=
    EXECUTE <SQL statement name>
      [ <result using clause> ]
      [ <parameter using clause> ]

<result using clause> ::= <using clause>

<using clause> ::=
      <using arguments>
    | <using descriptor>

<using arguments> ::=
    ( USING | INTO ) <argument> [ ( <comma> <argument> )... ]

<argument> ::= <target specification>

<parameter using clause> ::= <using clause>

<execute immediate statement> ::=
    EXECUTE IMMEDIATE <SQL statement variable>

<SQL dynamic data statement> ::=
      <allocate cursor statement>
    | <dynamic open statement>
    | <dynamic fetch statement>
    | <dynamic close statement>
    | <dynamic delete statement: positioned>
    | <dynamic update statement: positioned>

<allocate cursor statement> ::=
    ALLOCATE <extended cursor name> [ INSENSITIVE ]
```

```
          [ SCROLL ] CURSOR
        FOR <extended statement name>

<extended cursor name> ::=
    [ <scope option> ] <simple value specification>

<dynamic open statement> ::=
    OPEN <dynamic cursor name> [ <using clause> ]

<dynamic cursor name> ::=
      <cursor name>
    | <extended cursor name>

<dynamic fetch statement> ::=
    FETCH [ [ <fetch orientation> ] FROM ] <dynamic cursor name>
        <using clause>

<dynamic close statement> ::=
    CLOSE <dynamic cursor name>

<dynamic delete statement: positioned> ::=
    DELETE FROM <table name>
      WHERE CURRENT OF
          <dynamic cursor name>

<dynamic update statement: positioned> ::=
    UPDATE <table name>
      SET <set clause>
          [ ( <comma> <set clause> )... ]
        WHERE CURRENT OF
            <dynamic cursor name>

<SQL diagnostics statement> ::=
    <get diagnostics statement>

<get diagnostics statement> ::=
    GET DIAGNOSTICS <sql diagnostics information>

<sql diagnostics information> ::=
      <statement information>
    | <condition information>

<statement information> ::=
    <statement information item> [ ( <comma> <statement information item> )... ]

<statement information item> ::=
    <simple target specification> <equals operator> <statement information item name>

<statement information item name> ::=
      NUMBER
    | MORE
    | COMMAND_FUNCTION
    | DYNAMIC_FUNCTION
    | ROW_COUNT

<condition information> ::=
    EXCEPTION <condition number>
      <condition information item> [ ( <comma> <condition information item> )... ]
```

```
<condition number> ::= <simple value specification>

<condition information item> ::=
    <simple target specification> <equals operator> <condition information item name>

<condition information item name> ::=
      CONDITION_NUMBER
    | RETURNED_SQLSTATE
    | CLASS_ORIGIN
    | SUBCLASS_ORIGIN
    | SERVER_NAME
    | CONNECTION_NAME
    | CONSTRAINT_CATALOG
    | CONSTRAINT_SCHEMA
    | CONSTRAINT_NAME
    | CATALOG_NAME
    | SCHEMA_NAME
    | TABLE_NAME
    | COLUMN_NAME
    | CURSOR_NAME
    | MESSAGE_TEXT
    | MESSAGE_LENGTH
    | MESSAGE_OCTET_LENGTH

<embedded SQL host program> ::=
      <embedded SQL Ada program>
    | <embedded SQL C program>
    | <embedded SQL COBOL program>
    | <embedded SQL Fortran program>
    | <embedded SQL MUMPS program>
    | <embedded SQL Pascal program>
    | <embedded SQL PL/I program>

<embedded SQL Ada program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL C program> ::=
      !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL COBOL program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL Fortran program> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL MUMPS program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL Pascal program> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL PL/I program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL declare section> ::=
      <embedded SQL begin declare>
        [ <embedded character set declaration> ]
        [ <host variable definition>... ]
      <embedded SQL end declare>
    | <embedded SQL MUMPS declare>
```

```
<embedded SQL begin declare> ::=
    <SQL prefix> BEGIN DECLARE SECTION
        [ <SQL terminator> ]

<SQL prefix> ::=
      EXEC SQL
    | <ampersand>SQL<left paren>

<SQL terminator> ::=
      END-EXEC
    | <semicolon>
    | <right paren>

<embedded character set declaration> ::=
    SQL NAMES ARE <character set specification>

<host variable definition> ::=
      <Ada variable definition>
    | <C variable definition>
    | <COBOL variable definition>
    | <Fortran variable definition>
    | <MUMPS variable definition>
    | <Pascal variable definition>
    | <PL/I variable definition>

<Ada variable definition> ::=
    <Ada host identifier> [ ( <comma> <Ada host identifier> )... ] :
    <Ada type specification> [ <Ada initial value> ]

<Ada type specification> ::=
      <Ada qualified type specification>
    | <Ada unqualified type specification>

<Ada qualified type specification> ::=
      SQL_STANDARD.CHAR [ CHARACTER SET
         [ IS ] <character set specification> ]
          <left paren> 1 <double period> <length> <right paren>
    | SQL_STANDARD.BIT
          <left paren> 1 <double period> <length> <right paren>
    | SQL_STANDARD.SMALLINT
    | SQL_STANDARD.INT
    | SQL_STANDARD.REAL
    | SQL_STANDARD.DOUBLE_PRECISION
    | SQL_STANDARD.SQLCODE_TYPE
    | SQL_STANDARD.SQLSTATE_TYPE
    | SQL_STANDARD.INDICATOR_TYPE

<Ada unqualified type specification> ::=
      CHAR
          <left paren> 1 <double period> <length> <right paren>
    | BIT
          <left paren> 1 <double period> <length> <right paren>
    | SMALLINT
    | INT
    | REAL
    | DOUBLE_PRECISION
    | SQLCODE_TYPE
    | SQLSTATE_TYPE
```

```
    | INDICATOR_TYPE

<Ada initial value> ::=
    <Ada assignment operator> <character representation>...

<Ada assignment operator> ::= <colon><equals operator>

<C variable definition> ::=
      [ <C storage class> ]
      [ <C class modifier> ]
      <C variable specification>
    <semicolon>

<C storage class> ::=
      auto
    | extern
    | static

<C class modifier> ::= const | volatile

<C variable specification> ::=
      <C numeric variable>
    | <C character variable>
    | <C derived variable>

<C numeric variable> ::=
    ( long | short | float | double )
      <C host identifier> [ <C initial value> ]
            [ ( <comma> <C host identifier> [ <C initial value> ] )... ]

<C initial value> ::=
    <equals operator> <character representation>...

<C character variable> ::=
    char [ CHARACTER SET
             [ IS ] <character set specification> ]
      <C host identifier>
        <C array specification> [ <C initial value> ]
        [ ( <comma> <C host identifier>
          <C array specification>
                 [ <C initial value> ] )... ]

<C array specification> ::=
    <left bracket> <length> <right bracket>

<C derived variable> ::=
      <C VARCHAR variable>
    | <C bit variable>

<C VARCHAR variable> ::=
    VARCHAR [ CHARACTER SET [ IS ]
        <character set specification> ]
        <C host identifier>
            <C array specification> [ <C initial value> ]
          [ ( <comma> <C host identifier>
              <C array specification>
                      [ <C initial value> ] )... ]
```

```
<C bit variable> ::=
    BIT <C host identifier>
        <C array specification> [ <C initial value> ]
      [ ( <comma> <C host identifier>
        <C array specification>
                   [ <C initial value> ] )... ]

<COBOL variable definition> ::=
    (01|77) <COBOL host identifier> <COBOL type specification>
      [ <character representation>... ] <period>

<COBOL type specification> ::=
      <COBOL character type>
    | <COBOL bit type>
    | <COBOL numeric type>
    | <COBOL integer type>

<COBOL character type> ::=
    [ CHARACTER SET [ IS ]
          <character set specification> ]
    ( PIC | PICTURE ) [ IS ] ( X [ <left paren> <length> <right paren> ] )...

<COBOL bit type> ::=
    ( PIC | PICTURE ) [ IS ]
        ( B [ <left paren> <length> <right paren> ] )...

<COBOL numeric type> ::=
    ( PIC | PICTURE ) [ IS ]
      S <COBOL nines specification>
    [ USAGE [ IS ] ] DISPLAY SIGN LEADING SEPARATE

<COBOL nines specification> ::=
      <COBOL nines> [ V [ <COBOL nines> ] ]
    | V <COBOL nines>

<COBOL nines> ::= ( 9 [ <left paren> <length> <right paren> ] )...

<COBOL integer type> ::=
      <COBOL computational integer>
    | <COBOL binary integer>

<COBOL computational integer> ::=
    ( PIC | PICTURE ) [ IS ] S<COBOL nines>
      [ USAGE [ IS ] ] ( COMP | COMPUTATIONAL )

<COBOL binary integer> ::=
    ( PIC | PICTURE ) [ IS ] S<COBOL nines>
      [ USAGE [ IS ] ] BINARY

<Fortran variable definition> ::=
    <Fortran type specification>
    <Fortran host identifier>
        [ ( <comma> <Fortran host identifier> )... ]

<Fortran type specification> ::=
      CHARACTER [ <asterisk> <length> ]
          [ CHARACTER SET [ IS ]
                <character set specification> ]
```

```
      | BIT [ <asterisk> <length> ]
      | INTEGER
      | REAL
      | DOUBLE PRECISION

<MUMPS variable definition> ::=
    ( <MUMPS numeric variable> | <MUMPS character variable> )
        <semicolon>

<MUMPS numeric variable> ::=
    <MUMPS type specification>
      <MUMPS host identifier> [ ( <comma> <MUMPS host identifier> )... ]

<MUMPS type specification> ::=
      INT
    | DEC
          [ <left paren> <precision> [ <comma> <scale> ] <right paren> ]
    | REAL

<MUMPS character variable> ::=
    VARCHAR <MUMPS host identifier> <MUMPS length specification>
      [ ( <comma> <MUMPS host identifier> <MUMPS length specification> )... ]

<MUMPS length specification> ::=
    <left paren> <length> <right paren>

<Pascal variable definition> ::=
    <Pascal host identifier> [ ( <comma> <Pascal host identifier> )... ] <colon>
      <Pascal type specification> <semicolon>

<Pascal type specification> ::=
      PACKED ARRAY
          <left bracket> 1 <double period> <length> <right bracket>
        OF CHAR
          [ CHARACTER SET [ IS ]
                <character set specification> ]
    | PACKED ARRAY
          <left bracket> 1 <double period> <length> <right bracket>
        OF BIT
    | INTEGER
    | REAL
    | CHAR [ CHARACTER SET
                                [ IS ] <character set specification> ]
    | BIT

<PL/I variable definition> ::=
    (DCL | DECLARE)
        (   <PL/I host identifier>
          | <left paren> <PL/I host identifier>
                [ ( <comma> <PL/I host identifier> )... ] <right paren> )
    <PL/I type specification>
    [ <character representation>... ] <semicolon>

<PL/I type specification> ::=
      ( CHAR | CHARACTER ) [ VARYING ]
          <left paren><length><right paren>
          [ CHARACTER SET
                [ IS ] <character set specification> ]
```

```
      | BIT [ VARYING ] <left paren><length><right paren>
      | <PL/I type fixed decimal> <left paren> <precision>
            [ <comma> <scale> ] <right paren>
      | <PL/I type fixed binary> [ <left paren> <precision> <right paren> ]
      | <PL/I type float binary> <left paren> <precision> <right paren>

<PL/I type fixed decimal> ::=
        ( DEC | DECIMAL ) FIXED
      | FIXED ( DEC | DECIMAL )

<PL/I type fixed binary> ::=
        ( BIN | BINARY ) FIXED
      | FIXED ( BIN | BINARY )

<PL/I type float binary> ::=
        ( BIN | BINARY ) FLOAT
      | FLOAT ( BIN | BINARY )

<embedded SQL end declare> ::=
      <SQL prefix> END DECLARE SECTION
          [ <SQL terminator> ]

<embedded SQL MUMPS declare> ::=
      <SQL prefix>
        BEGIN DECLARE SECTION
          [ <embedded character set declaration> ]
          [ <host variable definition>... ]
        END DECLARE SECTION
      <SQL terminator>

<embedded SQL statement> ::=
      <SQL prefix>
        <statement or declaration>
      [ <SQL terminator> ]

<statement or declaration> ::=
        <declare cursor>
      | <dynamic declare cursor>
      | <temporary table declaration>
      | <embedded exception declaration>
      | <SQL procedure statement>

<embedded exception declaration> ::=
      WHENEVER <condition> <condition action>

<condition> ::=
      SQLERROR | NOT FOUND

<condition action> ::=
      CONTINUE | <go to>

<go to> ::=
      ( GOTO | GO TO ) <goto target>

<goto target> ::=
        <host label identifier>
      | <unsigned integer>
      | <host PL/I label variable>
```

```
<host label identifier> ::= !!<EMPHASIS>(See the Syntax Rules.)

<host PL/I label variable> ::= !!<EMPHASIS>(See the Syntax Rules.)

<preparable statement> ::=
      <preparable SQL data statement>
    | <preparable SQL schema statement>
    | <preparable SQL transaction statement>
    | <preparable SQL session statement>
    | <preparable implementation-defined statement>

<preparable SQL data statement> ::=
      <delete statement: searched>
    | <dynamic single row select statement>
    | <insert statement>
    | <dynamic select statement>
    | <update statement: searched>
    | <preparable dynamic delete statement: positioned>
    | <preparable dynamic update statement: positioned>

<dynamic single row select statement> ::= <query specification>

<dynamic select statement> ::= <cursor specification>

<preparable dynamic delete statement: positioned> ::=
   DELETE [ FROM <table name> ]
      WHERE CURRENT OF <cursor name>

<preparable dynamic update statement: positioned> ::=
   UPDATE [ <table name> ]
      SET <set clause list>
      WHERE CURRENT OF <cursor name>

<preparable SQL schema statement> ::=
      <SQL schema statement>

<preparable SQL transaction statement> ::=
      <SQL transaction statement>

<preparable SQL session statement> ::=
      <SQL session statement>

<preparable implementation-defined statement> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<direct SQL statement> ::=
    <directly executable statement> <semicolon>

<directly executable statement> ::=
      <direct SQL data statement>
    | <SQL schema statement>
    | <SQL transaction statement>
    | <SQL connection statement>
    | <SQL session statement>
    | <direct implementation-defined statement>

<direct SQL data statement> ::=
```

```
        <delete statement: searched>
      | <direct select statement: multiple rows>
      | <insert statement>
      | <update statement: searched>
      | <temporary table declaration>

<direct select statement: multiple rows> ::=
    <query expression> [ <order by clause> ]

<direct implementation-defined statement> ::=
    !!<EMPHASIS>(See the Syntax Rules)

<SQL object identifier> ::=
    <SQL provenance> <SQL variant>

<SQL provenance> ::= <arc1> <arc2> <arc3>

<arc1> ::= iso | 1 | iso <left paren> 1 <right paren>

<arc2> ::= standard | 0 | standard <left paren> 0 <right paren>

<arc3> ::= 9075

<SQL variant> ::= <SQL edition> <SQL conformance>

<SQL edition> ::= <1987> | <1989> | <1992>

<1987> ::= 0 | edition1987 <left paren> 0 <right paren>

<1989> ::= <1989 base> <1989 package>

<1989 base> ::= 1 | edition1989 <left paren> 1 <right paren>

<1989 package> ::= <integrity no> | <integrity yes>

<integrity no> ::= 0 | IntegrityNo <left paren> 0 <right paren>

<integrity yes> ::= 1 | IntegrityYes <left paren> 1 <right paren>

<1992> ::= 2 | edition1992 <left paren> 2 <right paren>

<SQL conformance> ::= <low> | <intermediate> | <high>

<low> ::= 0 | Low <left paren> 0 <right paren>

<intermediate> ::= 1 | Intermediate <left paren> 1 <right paren>

<high> ::= 2 | High <left paren> 2 <right paren>
```

32. Appendix B - SQL Tutorial for beginners

32.1 Tutorial for PostgreSQL

SQL tutorial is also distributed with PostgreSQL. The SQL tutorial scripts is in the directory src/tutorial

32.2 Internet URL pointers

The SQL tutorial for beginners can be found at

- http://w3.one.net/~jhoffman/sqltut.htm

Comments or suggestions? Mail to

- Jim Hoffman jhoffman@one.net

The following are the sites suggested by John Hoffman:

- SQL Reference http://www.contrib.andrew.cmu.edu/~shadow/sql.html
- Ask the SQL Pro http://www.inquiry.com/techtips/thesqlpro/
- SQL Pro's Relational DB Useful Sites http://www.inquiry.com/techtips/thesqlpro/usefulsites.html
- Programmer's Source http://infoweb.magi.com/~steve/develop.html
- DBMS Sites http://info.itu.ch/special/wwwfiles Go here and see file comp_db.html
- DB Ingredients http://www.compapp.dcu.ie/databases/f017.html
- Web Authoring http://www.stars.com/Tutorial/CGI/
- Computing Dictionary http://wfn-shop.princeton.edu/cgi-bin/foldoc
- DBMS Lab/Links http://www-ccs.cs.umass.edu/db.html
- SQL FAQ http://epoch.CS.Berkeley.EDU:8000/sequoia/dba/montage/FAQ Go here and see file SQL_TOC.html
- SQL Databases http://chaos.mur.csu.edu.au/itc125/cgi/sqldb.html
- RIT Database Design Page http://www.it.rit.edu/~wjs/IT/199602/icsa720/icsa720postings.html
- Database Jump Site http://www.pcslink.com/~ej/dbweb.html
- Programming Tutorials on the Web http://www.eng.uc.edu/~jtilley/tutorial.html
- Development Resources http://www.ndev.com/ndc2/support/resources.htp
- Query List http://ashok.pair.com/sql.htm
- IMAGE SQL Miscellaneous http://jazz.external.hp.com/training/sqltables/main.html
- Internet Resource List http://www.eit.com/web/netservices.html

33. Appendix C - Linux Quick Install Instructions

If you are planning to use PostgreSQL on Linux, and need help in installing Linux, then please visit the pointers given in this Appendix. They cover the following topics -

- Salient Features of Linux - Why Linux is better as a database server when compared with Windows 95/NT
- 10 minutes Linux Quick Install Instructions
- Microsoft-Linux Analogy List
- Quick Steps to Recompile the Linux Kernel
- Main Site is at http://members.spree.com/technology/aldev/
- Mirror site http://aldev.8m.com
- Mirror site http://aldev.webjump.com
- Mirror site http://homepages.infoseek.com/~aldev1/index.html
- Mirror site http://www3.bcity.com/aldev/

The Linux Printing Usage HOWTO

by Mark Komarinski <markk@auratek.com>

v1.2.2, 6 February 1998

1. Introduction

- 1.1 Linux Printing HOWTO History
- 1.2 Version History
- 1.3 Copyrights and Trademarks
- 1.4 Downloading the Linux Printing HOWTOs
- 1.5 Feedback
- 1.6 Acknowledgments

2. Printing Under Linux

- 2.1 History of Linux Printing
- 2.2 Printing a File Using *lpr*
- 2.3 Viewing the Print Queue with *lpq*
- 2.4 Canceling a Print Job Using *lprm*
- 2.5 Controlling the lpd program with *lpc*
- 2.6 The RedHat printtool

3. Printing files

- 3.1 Printing graphics files
- 3.2 Printing PostScript files
- 3.3 Printing PDF files
- 3.4 Printing TeX files
- 3.5 Printing `troff` formatted files
- 3.6 Printing `man` pages

TOC

4. Miscellaneous Items

- 4.1 Formatting Before Printing
- 4.2 The PRINTER Environment Variables

5. Answers to Frequently Asked Questions

6. Troubleshooting

7. References

---

1. Introduction

This document describes how to use the line printer spooling system provided with the Linux operating system. This HOWTO is the supplementary document to the Linux Printing Setup HOWTO, which discusses the installation and setup of the Linux printing system. The material presented in this HOWTO should be equally relevent for all flavors of the BSD operating system in addition to the Linux operating system.

1.1 Linux Printing HOWTO History

Note from Mark Komarinski <markk@auratek.com>:

I'd like to thank Matt Foster for doing a lot of work in the re-write of this HOWTO. I'm keeping his style, and adding when necessary to keep everything updated.

Note from Matt Foster <mwf@engr.uark.edu>:

This version of the Linux Printing HOWTO is a complete rewrite of the one originally written by Grant Taylor <grant@god.tufts.edu> and Brian McCauley <B.A.McCauley@bham.ac.uk>. I have tried to keep with the coverage of material presented by Grant and Brian's HOWTO, but I have drastically modified the style of presentation and the depth of material covered. I feel that this makes the HOWTO more complete and easier to read. I can only hope that you agree.

1.2 Version History

v1.2.2

- Re-indexed, other changes to fit in the new RedHat docs. Thanks Ed!

v1.2.1

- updates, some changes for Dr. Linux publication

v1.2

- Windows Printers
- Changing max size of print files

v1.11

- new maintainter!
- Added lpc info
- Added some info for troubleshooting
- A start on printing graphics files!

v1.1

TOC

- revised some of the wording
- developed section on PostScript printing
- attempted to clarify some of the examples 8-)
- fleshed the discussion of the basic Linux printing utilities

v1.0

- initial public release of the Printing Usage HOWTO

1.3 Copyrights and Trademarks

Some names mentioned in this HOWTO are claimed as copyrights and/or trademarks of certain persons and/or companies. These names appear in full or initial caps in this HOWTO.

(c) 1995 Matt Foster (mwf@engr.uark.edu)
(c) 1996-1997 Mark F. Komarinski (markk@auratek.com)

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice.

That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at <linux-howto@sunsite.unc.edu>. You may finger this address for phone number and additional contact information.

1.4 Downloading the Linux Printing HOWTOs

I recommend that if you want to print a copy of this HOWTO that you download the PostScript version. It is formatted in a fashion that is aesthetically appealing and easier to read. You can get the PostScript version from one of the many Linux distribution sites (such as SunSITE ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/).

1.5 Feedback

Questions, comments, or corrections for this HOWTO may be directed to <markk@auratek.com>.

1.6 Acknowledgments

Thanks go out to all of the people who took the time to read the alpha version of this HOWTO and respond with many helpful comments and suggestions---some of you may see your comments reflected in the version.

I'd also like to thank Matt Foster who did the original re-write.

2. Printing Under Linux

This section discusses how to print files, examine the print queue, remove jobs from the print queue, format files before printing them, and configure your printing environment.

2.1 History of Linux Printing

The Linux printing system---the *lp* system---is a port of the source code written by the Regents of the University of California for the Berkeley Software Distribution version of the UNIX operating system.

2.2 Printing a File Using *lpr*

By far, the most simplistic way to print in the Linux operating system is to send the file to be printed directly to the printing device. One way to do this is to use the *cat* command. As the `root` user, one could do something like

```
# cat thesis.txt > /dev/lp
```

In this case, `/dev/lp` is a symbolic link to the actual printing device---be it a dot-matrix, laser printer, typesetter, or plotter. (See *ln*(1) for more information on symbolic links.)

For the purpose of security, only the `root` user and users in the same group as the print daemon are able to write directly to the printer. This is why commands such as *lpr*, *lprm*, and *lpq* have to be used to access the printer.

Because of this, users have to use *lpr* to print a file. The *lpr* command takes care of all the initial work needed to print the file, and then it hands control over to another program, *lpd*, the line printing daemon. The line printing daemon then tells the printer how to print the file.

When *lpr* is executed, it first copies the specified file to a certain directory (the spool directory) where the file remains until *lpd* prints it. Once *lpd* is told that there is a file to print, it will spawn a copy of itself (what we programmers call forking). This copy will print our file while the original copy waits for more requests. This allows for multiple jobs to be queued at once.

The syntax of *lpr*(1) is a very familiar one,

```
$ lpr [ options ] [ filename ... ]
```

If `filename` is not specified, *lpr* expects input to come from standard input (usually the keyboard, or another program's output). This enables the user to redirect a command's output to the print spooler. As such,

```
$ cat thesis.txt | lpr
```

TOC

or,

```
$ pr -l60 thesis.txt | lpr
```

The *lpr* command accepts several command-line arguments that allow a user to control how it works. Some of the most widely used arguments are: **-P**printer specifies the printer to use, **-h** suppresses printing of the burst page, **-s** creates a symbolic link instead of copying the file to the spool directory (useful for large files), and **-#**num specifies the number of copies to print. An example interaction with *lpr* might be something like

```
$ lpr -#2 -sP dj thesis.txt
```

This command will create a symbolic link to the file `thesis.txt` in the spool directory for the printer named *dj*, where it would be processed by *lpd*. It would then print a second copy of `thesis.txt`.

For a listing of all the options that *lpr* will recognize, see *lpr*(1).

2.3 Viewing the Print Queue with *lpq*

To view the contents of the print queue, use the *lpq* command. Issued without arguments, it returns the contents of the default printer's queue.

The returned output of *lpq* can be useful for many purposes.

```
$ lpq
lp is ready and printing
Rank   Owner      Job  Files                                 Total Size
active mwf        31   thesis.txt                            682048 bytes
```

2.4 Canceling a Print Job Using *lprm*

Another useful feature of any printing system is the ability to cancel a job that has been previously queued. To do this, use *lprm*.

```
$ lprm -
```

The above command cancels all of the print jobs that are owned by the user who issued the command. A single print job can be canceled by first getting the job number as reported by *lpq* and then giving that number to *lprm*. For example,

```
$ lprm 31
```

would cancel job 31 (`thesis.txt`) on the default printer.

TOC

2.5 Controlling the lpd program with *lpc*

The *lpc*(8) program is used to control the printers that lpd serves. you can enable or disable a printer or its queues, rearrange entries within a queue, and get a status report on the printers and their queues. Lpc is mostly used in a setup where there are multiple printers hanging off one machine.

```
$ lpc
```

The above will start the lpc program. By default, this enters you into an interactive mode, and you can begin issuing commands. The other option is to issue an lpc command on the command line.

```
$ lpc status all
```

A list of the available commands are in the *lpd* man page, but here are a few of the major commands you'll want to know about. Any commands marked with *option* can either be a printer name (lp, print, etc) or the keyword `all`, which means all printers.

- disable *option* - prevents any new printer job from being entered
- down *option* - disables all printing on the printer
- enable *option* - allow new jobs to enter the print queue
- quit (or exit) - leave lpc
- restart *option* - restarts lpd for that printer
- status *option* - print status of printer
- up *option* - enable everything and start a new lpd

2.6 The RedHat printtool

Just a quick note here on RedHat's amazing printtool program. It seems to do everything that a magicfilter would do. RedHat already installs many of the programs to do the filtering. Here's how I have my printer set up under RH 4.0 with an HP LJ 4L connected to my parallel port (should be the same for other versions of RH as well).

- Become root and fire up printtool (if you su'ed, you remembered to SETENV DISPLAY :0.0 and xhost +, right?)
- Click "Add", and hit "OK" for a local printer.
- Fill in the printer device (/dev/lp1 for me)
- Fill in the input filter - Select a printer type, resolution, and paper size (ljet4, 300x300, and letter)
- Hit "OK" all the way back, and restart the lpd.

Just like rolling an /etc/printcap file by hand, you can have multiple printer definitions for each physical printer. One for different paper sizes, resolutions, etc.

---

3. Printing files

This section covers printing the kinda of files that you'll run across in a Linux setup.

3.1 Printing graphics files

Printing graphics files through a printer usually depends on the kind of graphics you're converting, and the kind of printer you want to send to. Dot matrix is usually out of the question due to differences in the way dot-matrix handles graphics. Your best bet in this situation is to see if your printer is compatable with an Epson or an IBM ProPrinter, then convert the graphics file to PostScript, then use Ghostscript (see next section) to print the graphics.

If you have a laser printer, things are a bit easier since many are compatable with PCL. This now gives you a few options. Some programs may output directly in PCL. If not, programs like *NetPBM* can convert into PCL. Last option is to use ghostscript (see next section).

Your absolutely best option is to install packages like NetPBM and Ghostscript then installing a magic filter to process the graphics files automagically.

3.2 Printing PostScript files

Printing PostScript files on a printer that has a PostScript interpreter is simple; just use *lpr*, and the printer will take care of all of the details for you. For those of us that don't have printers with PostScript capabilities, we have to resort to other means. Luckily, there are programs available that can make sense of PostScript, and translate it into a language that most printers will understand. Probably the most well known of these programs is Ghostscript.

Ghostscript's responsibility is to convert all of the descriptions in a PostScript file to commands that the printer will understand. To print a PostScript file using Ghostscript, you might do something like

```
$ gs -dSAFER -dNOPAUSE -sDEVICE=deskjet -sOutputFile=\|lpr thesis.ps
```

Notice in the above example that we are actually piping the output of Ghostscript to the *lpr* command by using the **-sOutputFile** option.

Ghostview is an interface to Ghostscript for the X Window System. It allows you to preview a PostScript file before you print it. Ghostview and Ghostscript can both be swiped from ftp://prep.ai.mit.edu/pub/gnu/.

3.3 Printing PDF files

Adobe has released an Acrobat reader for Linux, and it's available on the Adobe home page http://www.adobe.com. Its predecessor, xpdf, is also available. Both should print to a postscript device.

TOC

3.4 Printing TeX files

One of the easiest ways to print TeX files is to convert them to PostScript and then print them using Ghostscript. To do this, you first need to convert them from TeX to a format known as DVI (which stands for device-independent). You can do this with the *tex*(1) command. Then you need to convert the DVI file to a PostScript file using *dvips*. All of this would look like the following when typed in.

```
$ tex thesis.tex
$ dvips thesis.dvi
```

Now you are ready to print the resulting PostScript file as described above.

3.5 Printing `troff` formatted files

```
$ groff -Tascii thesis.tr | lpr
```

or, if you prefer,

```
$ groff thesis.tr > thesis.ps
```

and then print the PostScript file as described above.

3.6 Printing `man` pages

```
$ man man | col -b | lpr
```

The man pages contain pre-formatted `troff` data, so we have to strip out any highlighting, underlines, etc. The 'col' program does this just nicely, and since we're piping data, the `man` program won`t use `more`.

---

4. Miscellaneous Items

This covers topics not in any of the others.

4.1 Formatting Before Printing

Since most ASCII files are not formatted for printing, it is useful to format them in some way before they are actually printed. This may include putting a title and page number on each page, setting the margins, double spacing, indenting, or printing a file in multiple columns. A common way to do this is to use a print preprocessor such as *pr*.

```
$ pr +4 -d -h"Ph.D. Thesis, 2nd Draft" -l60 thesis.txt | lpr
```

In the above example, *pr* would take the file `thesis.txt` and skip the first three pages (+4), set the page length to sixty lines (-l60), double space the output (-d), and add the phrase "Ph.D. Thesis, 2nd Draft" to the top of each page (-h). *Lpr* would then queue *pr*'s output. See its on-line manual page for more information on using *pr*.

4.2 The PRINTER Environment Variables

All of the commands in the Linux printing system accept the **-P** option. This option allows the user to specify which printer to use for output. If a user doesn't specify which printer to use, then the default printer will be assumed as the output device.

Instead of having to specify a printer to use every time that you print, you can set the PRINTER environment variable to the name of the printer that you want to use. This is accomplished in different ways for each shell. For *bash* you can do this with

```
$ PRINTER="printer_name"; export PRINTER
```

and *csh*, you can do it with

```
% setenv PRINTER "printer_name"
```

These commands can be placed in your login scripts (.profile for *bash*, or .cshrc for *csh*), or issued on the command-line. (See *bash*(1) and *csh*(1) for more information on environment variables.)

5. Answers to Frequently Asked Questions

Q1. How do I prevent the staircase effect?

A1. The staircase effect is caused by the way some printers expect lines to be terminated. Some printers want lines that end with a carriage-return/line-feed sequence (DOS-style) instead of the line-feed sequence used for UNIX-type systems. The easiest way to fix this is to see if your printer can switch between the two styles somehow---either by flipping a DIP switch, or by sending an escape sequence at the start of each print job. To do the latter, you need to create a filter (see Q2).

A quick fix is to use a filter on the command-line. An example of this might be

```
$ cat thesis.txt | todos | lpr
```

Q2. What is a filter?

A2. A filter is a program that reads from standard input (*stdin*), performs some action on this input, and writes to standard output (*stdout*). Filters are used for a lot of things, including text processing.

Q3. What is a *magic* filter?

A3. A magic filter is a filter that performs an action based on a file's type. For example, if the file is a plain, text file, it would simply print the file using the normal methods. If the file is a PostScript file, or any other format, it would print it using another method (ghostscript). Two examples of this is magicfilter and APSfilter. One caveat of these filters is that the appropriate programs have to be installed before you install the filter.

The reason for this is that when the magicfilter gets installed, it queries your system for specific programs (such as ghostscript - if it finds it, then it knows it can handle PostScript data), then builds itself based on what it finds. To handle all the printer files, you should probably have at least the following installed:

- GhostScript
- TeX
- NetPBM
- jpeg utilities
- gzip

Q4. What about the Windows Printing System? Will Linux work with that?

A4. Maybe. Printers that accept only the WPS commands will not work with Linux. Printers that accept WPS and other commands (such as the Canon BJC 610) will work, as long as they're set to something other than WPS format. Other printers, such as some HP DeskJet 820Cxi/Cse, will *not* work with Linux. That being said, Linux can act as a print server (See Samba) for Win95 machines, since Win95 has drivers for those printers.

TOC

Q5. What kinda cheey system is this? I can't print more than 6 pages or else I get a "file too large" er

A5. One of the options in the /etc/printcap file relates to the maximum size of a print file. The default is 1000 disk blocks (about 500k?). For PostScript files and the like, this will give you maybe 6-8 pages with graphics and all. Be sure to add the following line in the printer definition:

```
mx=0
```

The primary reason for this is to keep the spool partition from getting filled. There is another way to do it, by making lpr create a soft link from the spool directory to your print file. But you have to remember to add the `-s` option to lpr every time.

---

6. Troubleshooting

This section covers some common things that can go wrong with your printing system.

If your printer doesn't work:

- Do other print jobs work? (application problem?)
- Is lpd running? (check it using lpc) (print controller?)
- Can root send something directly to the printer? (print services?)
- Can you print from DOS? (cable/printer problem?)

Answering these questions can help find a solution.

Send other suggestions for this section to <markk@auratek.com>.

7. References

This is a section of references on the Linux printing system. I have tried to keep the references section of this HOWTO as focused as possible. If you feel that I have forgotten a significant reference work, please do not hesitate to contact me.

Before you post your question to a USENET group, consider the following:

- Is the printer accepting jobs? (Use *lpc*(8) to verify.)
- Is the answer to your question covered in this HOWTO or Grant Taylor's Printing HOWTO?

If any of the above are true, you may want to think twice before you post your question. And, when you do finally post to a newsgroup, try to include pertinent information. Try not to just say something like, "I'm having trouble with lpr, please help." These types of posts will most definitely be ignored by many. Also try to include the kernel version that you're running, how the error occured, and, if any, the specific error message that the system returned.

On-Line Manual Pages

- *cat*(1) concatenate and print files
- *dvips*(1) convert a TeX DVI file to PostScript
- *ghostview*(1) view PostScript documents using Ghostscript
- *groff*(1) front-end for the groff document formatting system
- *gs*(1) Ghostscript interpreter/viewer
- *lpc*(8) line printer control program
- *lpd*(8) line printer spooler daemon
- *lpq*(1) spool queue examination program
- *lpr*(1) off-line printer
- *lprm*(1) remove jobs from the line printer spooling queue
- *pr*(1) convert text files for printing
- *tex*(1) text formatting and typesetting

USENET newsgroups

- `comp.os.linux.*` a plethora of information on Linux
- `comp.unix.*` discussions relating to the UNIX operating system

Linux Quake HOWTO

Bob Zimbinski bobz@mr.net

v1.0.1.14, 30 August 1998

This document explains how to install, run and troubleshoot Quake, QuakeWorld and Quake II on an Intel Linux system.

1. Introduction

- 1.1 Feedback,Comments, Corrections
- 1.2 Acknowledgments **
- 1.3 Other Sources of Information

2. Quake/Quakeworld

- 2.1 Minimum Requirements
- 2.2 Installing Quake
- 2.3 Adding the Linux Binaries
- 2.4 Setting Permissions
- 2.5 X11 Quake
- 2.6 SVGAlib Quake
- 2.7 GLQuake
- 2.8 Linux-Specific Command Line Options
- 2.9 QuakeWorld **
- 2.10 Servers
- 2.11 Mods & Addons

TOC

3. Quake II

- 3.1 Prerequisites
- 3.2 Installing Quake II
- 3.3 Adding the Linux Binaries
- 3.4 Setting Permissions **
- 3.5 The X Renderer
- 3.6 The SVGAlib Renderer
- 3.7 The OpenGL Renderer
- 3.8 The GLX Renderer
- 3.9 Linux-Specific Command Line Options
- 3.10 Quake II Servers
- 3.11 Mods & Addons

4. Related Software

- 4.1 QStat
- 4.2 XQF
- 4.3 QuickSpy
- 4.4 QPlug for Linux
- 4.5 qkHacklib
- 4.6 GiMd2Viewer
- 4.7 QIPX
- 4.8 Ice
- 4.9 Q2getty
- 4.10 rcon
- 4.11 qlog ++
- 4.12 Cheapo **
- 4.13 qgraph **

5. Troubleshooting/FAQs

- 5.1 General
- 5.2 Quake/QuakeWorld

- 5.3 Quake II

6. Tips & Tricks

- 6.1 Running X and GL games without setuid
- 6.2 Running SVGA and GL games from X
- 6.3 Keeping the mouse inside the window in X
- 6.4 3Dfx "tweak" settings work in Linux too
- 6.5 The Poor Man's Server Browser
- 6.6 Using `lib3dfxgl.so` for Quake I

7. Administrivia

- 7.1 New Versions of This Document
- 7.2 Other Formats of This Document
- 7.3 Distribution Policy
- 7.4 Revision History

---

1. Introduction

Quake, QuakeWorld and Quake II are tremendously popular 3D action games developed by id Software.

If you're not familiar with the Quake games, there are better places than this HOWTO to learn about the basics. See section Other Sources of Information below for a list of some of these better places.

This document assumes you have Linux up and running, and in some cases the X Window System as well. X is not required to run these games, but it's a nice way to test a basic installation. If you are not running X, you may safely skip over any references to it.

Sections of this document that were updated in the last revision have a ** after the section heading. Sections updated in the revision prior to the last are marked with a ++.

1.1 Feedback,Comments, Corrections

This document certainly does **not** contain everything there is to know about Linux Quake. With your help, though, we can bring it closer to that ideal. We want this HOWTO to be as complete and accurate as possible, so if you notice mistakes or omissions, please bring them to our attention.

Questions, comments, or corrections should be sent to Bob Zimbinski (bobz@mr.net) or Mike Hallock (mikeh@medina.net). Constructive criticism is welcome. Flames are not.

1.2 Acknowledgments **

The original version of this document was written by Brett A.Thomas (quark@baz.com) and Mike Hallock (mikeh@medina.net). Bob Zimbinski (bobz@mr.net) rewrote and expanded the original document.

Special thanks to the following people for bringing us Quake for Linux:

- John Carmack and the rest of id Software for these great games.
- Dave 'Zoid' Kirsch (zoid@idsoftware.com) for the Linux ports.
- Dave Taylor (ddt@crack.com) for starting this whole wacky Linux port business.
- Daryll Strauss (daryll@harlot.rb.ca.us) for the Linux glide ports.
- Brian Paul (brianp@elastic.avid.com) for the Mesa graphics library.
- David Bucciarelli (tech.hmw@plus.it) for the Mesa/glide driver.

Thanks to the following people for contributions to this Howto:

- Mike Brunson (brunson@l3.net) for the vid_restart tip.

TOC

- Joey Hess (joey@kite.ml.org) for info on running svga & gl games from X
- Joe S. (jszabo@eden.rutgers.edu) for a steaming pile of suggestions
- Brad Lambert (bradl@dial.pipex.com) for the `-noudp` reminder.
- agx (gguenthe@iris.rz.uni-konstanz.de) for Quake installation info and pointing out QIPX.
- Derrik Pates (dmp8309@silver.sdsmt.edu) for making me think about security.
- Michael Dwyer (michael_dwyer@mwiworks.com) for the "OS considerations" ideas.
- Derek Simkowiak (dereks@kd-dev.com) for the Quake I CD installation procedure.
- sunstorm (sunstorm@glasscity.net) Quake Mission Pack 2 information.
- Neil Marshall (marshall@pssnet.com) for screen info.

1.3 Other Sources of Information

Linux-Specific Quake Information **

- LQ:Linux Quake & Utilites http://www.linuxquake.com/.
- Linux Quake Page http://captured.com/threewave/linux/
- QuakeWorld.net http://www.quakeworld.net

General Quake Information

- id Software http://www.idsoftware.com
- PlanetQuake http://www.planetquake.com
- QuakeWorld Central http://qwcentral.stomped.com
- 3Dfx's GL Quake FAQ (somewhat out of date) http://www.3dfx.com/game_dev/quake_faq.html
- Farenheit 176 Console Command Listing http://www.planetquake.com/f176
- rec.games.computer.quake.* newsgroups

Linux Gaming Information

- LinuxGames http://www.linuxgames.com/
- The Linux Game Tome http://www.cs.washington.edu/homes/tlau/tome/linux-game.html

---

2. Quake/Quakeworld

To install Quake on your Linux system, you'll need some flavor of the official Quake distribution from id. This will be either the retail DOS/Windows CD-ROM that you bought at your favorite software store, or the shareware version you downloaded from the net (see below for details on acquiring the shareware version). Alternatively, if you've already got Quake installed on a DOS/Windows machine, you can use the relevant files from that installation.

2.1 Minimum Requirements

You will need, as a bare minimum, the following:

- A Pentium 90 or better (133 recommended) computer
- 16 MB RAM (24 recommended)
- The Quake CD-ROM **or** the shareware version (quake106.zip)
- Linux kernel version 2.0.24 or later
- libc 5.2.18 or later
- One of the following:
  - X11R5 or later (for xquake)
  - SVGAlib 1.2.0 or later (for squake and glquake)
- 30-80 megabytes free disk space (depends on how you install)
- Access to the root account of the machine you're installing on

Optional:

- A supported soundcard
- A 3Dfx VooDoo Graphics or VooDoo2 3D graphics accelerator card.
- Mesa 2.6 or later (for glquake)

2.2 Installing Quake

Download the Necessary Files **

All the necessary files for Linux Quake are available at id Software's ftp site, ftp.idsoftware.com. This site can be quite busy at times,so you may want to use one of these mirror sites instead:

- ftp.cdrom.com/pub/idgames/idstuff (California, USA)
- ftp.gamesnet.net/idsoftware (California, USA)
- ftp.linuxquake.com/lqstuff (Michigan, USA)
- ftp.stomped.com/pub/mirror/idstuff (Minnesota, USA)
- mirrors.telepac.pt/pub/idgames (Lisbon, Portugal)
- download.netvision.net.il/pub/mirrors/idsoftware (Haifa, Israel)

The Quake files mentioned in in this section are:

- Shareware Quake for Windows distribution ftp://ftp.idsoftware.com/idstuff/quake/quake106.zip
- X11 Quake binary ftp://ftp.idsoftware.com/idstuff/unsup/unix/quake.x11-1.0-i386-unknown-linux2.0.tar.gz

- SVGAlib Quake binary ftp://ftp.idsoftware.com/idstuff/unsup/squake-1.1-i386-unknown-linux2.0.tar.gz
- OpenGL/Mesa Quake binary ftp://ftp.idsoftware.com/idstuff/unsup/unix/glquake-0.97-i386-unknown-linux2.0.tar.gz
- Linux QuakeWorld clients (X11, SVGAlib and GL clients are all included in each package)
  - libc5 tar.gz package ftp://ftp.idsoftware.com/idstuff/quakeworld/unix/qwcl-2.30-i386-unknown-linux2.0.tar.gz
  - glibc tar.gz package ftp://ftp.idsoftware.com/idstuff/quakeworld/unix/qwcl-2.30-glibc-i386-unknown-linux2.0.tar.gz
  - libc5 rpm package ftp://ftp.idsoftware.com/idstuff/quakeworld/unix/qwcl-2.30-1.i386.rpm
  - glibc rpm package ftp://ftp.idsoftware.com/idstuff/quakeworld/unix/qwcl-2.30-glibc-1.i386.rpm
- Linux QuakeWorld server
  - libc5 tar.gz package ftp://ftp.idsoftware.com/idstuff/quakeworld/unix/qwsv-2.30-i386-unknown-linux2.0.tar.gz
  - glibc tar.gz package ftp://ftp.idsoftware.com/idstuff/quakeworld/unix/qwsv-2.30-glibc-i386-unknown-linux2.0.tar.gz
  - libc5 rpm package ftp://ftp.idsoftware.com/idstuff/quakeworld/unix/qwsv-2.30-1.i386.rpm
  - glibc rpm package ftp://ftp.idsoftware.com/idstuff/quakeworld/unix/qwsv-2.30-glibc-1.i386.rpm
- Capture the Flag client package ftp://ftp.cdrom.com/pub/quake/planetquake/threewave/ctf/client/3wctfc.zip

Other software mentioned:

- lha archive utility ftp://sunsite.unc.edu/pub/Linux/utils/compress/lha-1.00.tar.Z.
- SVGAlib graphics library http://sunsite.unc.edu/pub/Linux/libs/graphics/svgalib-1.3.0.tar.gz
- SVGAlib libc5 binary http://sunsite.unc.edu/pub/Linux/libs/graphics/svgalib-1.3.0.libc5.bin.tar.gz
- Glide runtime libraries http://glide.xxedgexx.com/3DfxRPMS.html

Create the Installation Directory ++

The first thing you'll need to do is decide where you want to install Quake. Lots of folks like to put it in `/usr/games/quake`. Anal system administrator that I am, I choose to install anything that's not part of my Linux distribution under `/usr/local`. So for me, Quake goes in `/usr/local/games/quake`. If you choose to install somewhere else, please substitute the appropriate path wherever `/usr/local/games/quake` is mentioned.

Note to Redhat users: If you plan on installing QuakeWorld from the `rpm` packages, you should probably install Quake in `/usr/local/games/quake`, since the `rpm`s install to this directory by default.

So go ahead and create the directory you'll install Quake in, and cd to it. The rest of these instructions will assume that this is your current directory.

```
mkdir /usr/local/games/quake
cd /usr/local/games/quake
```

Installing From a Quake CD ++

If you're installing from a Quake CD-ROM, read on. Otherwise you have permission to skip this section.

There are at least two versions of the Quake CD in circulation. I've got one from the early days that has Quake version 1.01 on it. Iv'e seen other CDs that contain version 1.06. You have 1.01 if you see files on your CD called `quake101.1` and `quake101.2`. If instead you see a file called `resource.1`, you have a newer CD. Mount your Quake CD now and determine which version you've got. In the example below, replace `/dev/cdrom` and `/mnt/cdrom` with the device file and mount point appropriate for your sysem:

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
ls /mnt/cdrom
```

- If you have a `resource.1` file on your CD, you can skip ahead to the next bullet. For a version 1.01 CD, you'll need to download the Quake shareware package to update the `.pak` files after the installation. The last bullet in this section explains this.
  - Concatenate the two resource files from your CD to a single file on your hard disk:

    ```
    cat /mnt/cdrom/quake101.1 /mnt/cdrom/quake101.2 > resource.1
    ```

  - Now continue on to the next paragraph, but when I refer to `/mnt/cdrom/resource.1`, you should use `/usr/local/games/quake/resource.1` instead.
- Now it's time to extract the Quake files. The `resource.1` file on your CD is really an lha archive (lha is a file compression and archiving format like zip or tar). We'll use the `lha(1)` command to extract it. If lha is not already installed on your system, you can get it from ftp://sunsite.unc.edu/pub/Linux/utils/compress/lha-1.00.tar.Z.

  ```
  lha e /mnt/cdrom/resource.1
  ```

  When lha is done, your Quake directory will contain a bunch of new files. A directory called `id1/` will also be created. The files in this directory are the only ones that are important for Linux Quake, so you can safely remove everything else. If you're totally new to Quake, or even if you're not, you may want to hang on to the `*.txt` files. On my system, I throw all the readmes that accumulate into a `doc/` directory. So:

  ```
  cd /usr/local/games/quake
  mkdir doc
  mv *.txt doc
  rm -f *
  ```

- If you installed from a version 1.01 CD, now you need to overwrite your `id1/pak0.pak` file with the one from the Quake shareware version. Install the shareware version as described in Shareware Version Install, only install it in some temporary directory so you don't overwrite your real Quake files. When you've extracted all the shareware files, copy the `id1/pak0.pak` file from the temporary shareware directory to your `/usr/local/games/quake/id1` directory. After that, you can erase the temporary shareware files.

That's it for installing from the CD. You can jump ahead to section " Installing the Linux Binaries" now.

DOS/Windows to Linux Install

If you have Quake installed under Windows or DOS on a different machine, you can transfer the files in `quake\id1\` to your Linux system via FTP or some other mechanism. Keep in mind that the filenames on your Linux system must be in lower case for Quake to find them, so you may have to rename them after the transfer. Also note that it may be necessary to delete your DOS/Win installation after you do this to remain in compliance with the terms of id's software license. It's not my fault if you do something illegal.

If your DOS/Win and Linux systems are on the same machine, you have two options: copy the files from your DOS/Windows partition to your Linux partition, or link to the necessary files from Linux. Both options will work equally well. You just save around 50 megabytes of disk space when you link instead of copy.

Whatever you choose to do, start by cd'ing to your Quake directory and creating a new directory below it called `id1`:

```
cd /usr/local/games/quake
mkdir id1
```

- If you want to copy the files from your DOS/Windows partition, do something like this:

```
cp /win95/games/quake/id1/*.pak id1
```

- To create links to your DOS/Windows Quake files instead, do this:

```
cd id1
ln -s /win95/games/quake/id1/*.pak .
```

Replace `/win95/games/quake` with the correct path to your DOS/Windows partition and Quake installation directory.

The Quake data files are now installed. Move ahead to " Installing the Linux Binaries".

Shareware Version Install

The single-episode shareware version of Quake is freely available for download from id's ftp site. It has all the features of the full version, with a couple of major limitations: You can't play QuakeWorld with it, and you can't play custom or modified levels.

Installing the shareware version of Quake isn't really much different than installing from the CD.

See section Download the Necessary Files for the location of the shareware distribution. Download it and extract it to your Quake directory:

```
cd /usr/local/games/quake
unzip -L /wherever/you/put/it/quake106.zip
```

Now you've got (among others) a file called `resource.1` that's really an lha archive (lha is a file compression and archiving format like zip or tar). We'll use the **lha(1)** command to extract it. If lha is not already installed on your system, you can get it from ftp://sunsite.unc.edu/pub/Linux/utils/compress/lha-1.00.tar.Z.

```
lha e resource.1
```

When lha is done, your Quake directory will contain a bunch of new files. A directory called `id1/` will also be created. The files in this directory are the only ones that are important for Linux Quake, so you can safely remove everything else. If you're totally new to Quake, or even if you're not, you may want to hang on to the `*.txt` files. On my system, I throw all the readmes that accumulate into a `doc/` subdirectory. So:

```
cd /usr/local/games/quake
mkdir doc
mv *.txt doc
rm -f *
```

Now you're ready to install the Linux binaries.

TOC

2.3 Adding the Linux Binaries

Decide which of the three flavors of Quake you'd like to install:

- X11 Quake allows you to run Quake in a window on your X desktop. It's the least exciting client, but it's a great, safe way to test your installation.
- Squake is the SVGAlib Quake client. It runs full screen on your console.
- GLQuake is the OpenGL Quake client, the One True Way to play Quake if you have a 3Dfx accellerator card.

Download the packages you want (see section Download the Necessary Files) and extract them to your Quake directory like so:

```
cd /usr/local/games/quake
tar -xzf XXXX-i386-unknown-linux2.0.tar.gz
```

2.4 Setting Permissions

Quake and QuakeWorld servers can be run by any user. The Quake clients, however, need access to your sound and graphics cards, which requires privileges that normal users don't have. One (bad) way to deal with this is to always run Quake as root. Responsible system administrators will cringe at this filthy suggestion. Making the Quake binaries setuid root is a more acceptable solution. Quake can then be run by regular users and still have the privileges it needs to access the sound and graphics devices. Setuid presents a security risk, though. A clever user could exploit a bug or security hole in Quake to gain root access to your system. Of course, if you don't run a multi-user system, this is may not be a big concern.

`squake` is the only Quake client that *must* be run with root permissions. With a little work, you can run the X and GL clients without setuid. Running X and GL games without setuid in the Tips and Tricks section tells how to make this work.

If you plan to run `squake`, make it setuid root with the following commands:

```
chown root squake
chmod 4755 squake
```

If you've decided it's ok to run `quake.x11` and `glquake` setuid root on your system, you can repeat the above commands for these binaries as well.

2.5 X11 Quake

If you installed the X11 client, now's a good time to try it out. You may need to do further configuration for `glquake` and `squake`, but at this point `quake.x11` should be ready to go.

```
cd /usr/local/games/quake
./quake.x11
```

If all is well, a small Quake window should appear with the first demo running in it. You should hear sound effects and possibly music, if the CD is mounted. If any of this fails to occur, please see the Troubleshooting section for help.

2.6 SVGAlib Quake

Both squake and glquake require SVGAlib to run (glquake uses SVGAlib to process keyboard and mouse input, in case you're wondering). SVGAlib comes with most modern Linux distributions, and **must** be properly configured before squake or glquake will run correctly.

`libvga.config` is SVGAlib's configuration file. On most systems you'll find it in either `/etc` or `/etc/vga`. Make sure the mouse, monitor, and video card settings in this file are correct for your system. See the SVGAlib documentation for more details.

If you don't already have SVGAlib on your system, download it from the location mentioned in the files section above. If you have a RedHat 5.x or other glibc-based Linux distribution, see Glibc, RedHat 5.x, Debian 2 considerations in the Troubleshooting/FAQs section for important information about compiling libraries for use with Quake. A precompiled libc5 SVGAlib binary is available at http://sunsite.unc.edu/pub/Linux/libs/graphics/svgalib-1.3.0.libc5.bin.tar.gz for those who don't want to deal with the hassle of compiling for libc5.

You should run squake from a virtual console. It won't run from X unless you're root when you start it. And running a game as the root user is something that should be avoided. So if you're in X, do a CTRL+ALT+F1, login and then:

```
cd /usr/local/games/quake
./squake
```

Running SVGA and GL games from X in the Tips & Tricks section below explains how to launch SVGA and GL Quake from X without manually switching to a virtual console.

2.7 GLQuake

Hardware-accelerated OpenGL Quake is Quake the way God intended it to be. There is no substitute, and once you've experienced it there's no going back.

To run `glquake`, you need a 3D card with the Voodoo, Voodoo2 or Voodoo Rush graphics chipset on it. There are specific issues to be dealt with if you have a Voodoo Rush card, and I won't go into them now because frankly, I wouldn't know what I was talking about. A future version of this HOWTO will cover Rush issues (If somebody wants to write about Voodoo Rush issues, I'll gladly include it here).

The SVGAlib, Glide, and Mesa libraries must all be installed and configured properly on your system for `glquake` to work. The following sections will very briefly cover what you need to do to get them going.

Bernd Kreimeier's (bk@gamers.org) Linux 3Dfx HOWTO (http://www.gamers.org/dEngine/xf3D/howto/3Dfx-HOWTO.html) is good source for further information.

The 3dfx.glide.linux newsgroup on the 3dfx news server (news.3dfx.com) is another good source of information about the intersection of Linux, glide, Mesa and Quake.

SVGAlib

glquake uses SVGAlib to get input from the mouse and keyboard, so you'll need to configure it as outlined in section SVGAlib Quake.

TOC

Glide

Glide is a library that provides an API for programming 3Dfx based cards. If you want the Mesa graphics library to use your 3Dfx card, you've gotta have it.

The latest version of glide can always be found at http://glide.xxedgexx.com/3DfxRPMS.html. Select the package(s) appropriate for your system, and install according to the instructions on the web page.

Note that unless you download the 3Dfx device driver package in addition to the Glide library, you will only be able to run Glide applications (like GLQuake) as root. Install the `/dev/3dfx` module and you can play GLQuake as a regular user.

Once you have glide installed, try out the test program that comes with it. Remember this program: it's a good way to reset your display if you ever have a glide application (like GLQuake) crash and leave your screen switched off. **NOTE: run this test from a VC, *not* X!** It's possible for the test app to lose mouse and keyboard focus in X, and then you'll have no way of shutting it down.

```
/usr/local/glide/bin/test3Dfx
```

Your screen should turn blue and prompt you to hit any key. After you press a key you should be returned to the prompt. 3dfx.glide.linux on 3dfx's news server (news.3dfx.com) is a great source of information for Linux glide-specific problems.

Mesa

Once glide's installed, you need to install Mesa, a free OpenGL implementation by Brian Paul (brianp@elastic.avid.com). Luckily, you won't have to look far, because Mesa 2.6 is included with the QLQuake & QuakeWorld binaries. All you have to do is move it to the right place:

```
cd /usr/local/games/quake
cp libMesaGL.so.2.6 /usr/local/lib
ldconfig
```

If you want to upgrade Mesa to a more recent version (Mesa 3.0 is the most recent version as of this writing), you can download the latest from ftp://iris.ssec.wisc.edu/pub/Mesa If you have a RedHat 5.x or other glibc-based Linux distribution, see Glibc, RedHat 5.x, Debian 2 considerations in the Troubleshooting/FAQs section for important information about compiling libraries for Quake.

After you've built it according to the instructions, you will have to do two things:

- Remove your old Mesa installation. If you previously installed a `libMesaGL.so.2.6` as described above, you must remove it or Quake may not use the new version.

  ```
  cd /usr/local/lib/
  rm -f libMesaGL.so.2*
  ```

- If the new Mesa has a major version number that's greater than 2, you need to create a link to it with the name `libMesaGL.so.2`:

  ```
  cd /usr/local/lib/
  ln -s /wherever/you/installed/it/libMesaGL.so.3.0 libMesaGL.so.2
  ldconfig
  ```

Now switch to a VC (CTRL+ALT+F1) and start glquake.

```
cd /usr/local/games/quake
./glquake
```

2.8 Linux-Specific Command Line Options

This section covers command line options that are specific to the Linux versions of Quake. There are plenty of other Quake options, but they're beyond the scope of this HOWTO. Check out some of the sites listed in section General Quake Information for this kind of information.

-mem *num*

Specify memory in megabytes to allocate (default is 8MB, which should be fine for most needs).

-nostdout

Don't do any output to stdout. Use this if you don't want all the console output dumped to your terminal.

-mdev *device*

Mouse device, default is `/dev/mouse`

-mrate *speed*

Mouse baud rate, default is 1200

-cddev *device*

CD device, default is `/dev/cdrom`

-mode *num*

Use indicated video mode (squake only)

-nokdb

Don't initialize keyboard

-sndbits *8 or 16*

Set sound bit sample size. Default is 16 if supported.

-sndspeed *speed*

Set sound speed. Usual values are 8000, 11025, 22051 and 44100. Default is 11025.

-sndmono

Set mono sound

-sndstereo

Set stereo sound (default if supported)

2.9 QuakeWorld **

http://www.quakeworld.net says it better than I could:

> QuakeWorld is an Internet multi-player specific version of Quake. While the original version of Quake can be played over the Internet, modem users - the majority of players, had less than satisfactory play. Symptoms like excessive lag - actions actually happening much later than you did them; packet loss - the game would freeze and resume several seconds later; and various other difficulties plagued users. After realizing how many people played Quake on the internet, and how many wanted to, but couldn't due to the play being

unsatisfactory, John Carmack of id Software decided to create a version of Quake that was optimized for the average modem Internet player. This Internet specific version does only 1 thing, play deathmatch games over a TCP/IP network such as the Internet. It has no support for solo play, and you can't do anything with out connecting to a special server.

TOC

You need the full, registered or retail version of Quake to play QuakeWorld, and a Linux QuakeWorld client. QuakeWorld clients come in the same flavors (X11, SVGAlib and Mesa) as normal Quake, but they're all bundled together in one package, so you only need to download one file. However, you've got four packages to choose from:

- a libc5 tar.gz package
- a glibc tar.gz package
- a libc5 rpm package
- a glibc rpm package

Install just one of these packages. Each contains the same files, they're just linked against different libraries. Redhat 5.x users should choose the the glibc rpm package. Users of glibc based systems without rpm support should use the glibc tar package. The libc5 rpm is for Redhat distributions prior to 5.0 and other distributions that use the rpm package format. The libc5 tar.gz package is for Slackware and everyone else.

See the Download the Necessary Files section for the location of the Linux Quakeworld files.

The prerequisites and configuration for these binaries are the same as for Quake, so refer to the previous sections for help on setting up SVGAlib or glide/Mesa.

Installing the RPM packages ++

Installation of the rpm packages should be as simple as:

```
su root
rpm -Uvh qwcl-xxxxx.i386.rpm
```

`qwcl`, `glqwcl` and `glqwcl.glx` will be installed setuid root so that they can access the graphics devices on your system. The X and GL clients can be run without root privileges if you follow the instructions in Running X and GL games without setuid below.

Rpm may complain that it can't find `libglide2x.so`. The Glide library is only necessary if you have a 3Dfx card and want to run QuakeWorld in GL mode (`glqwcl`). If you don't plan to use the GL mode, you can override the glide dependency with the `--nodeps` option:

```
su root
rpm -Uvh qwcl-xxxxx.i386.rpm --nodeps
```

Installing the tar.gz packages ++

To install, just untar the file in your Quake directory. Do it as root so the proper file permissions get set:

```
cd /usr/local/games/quake
su root
tar -xzf qwcl2.21-i386-unknown-linux2.0.tar.gz
```

`qwcl`, `glqwcl` and `glqwcl.glx` will be installed setuid root so that they can access the graphics devices on your system. The GL and X clients can be run without root privileges if you follow the instructions in Running X and GL games

without setuid below.

Running QuakeWorld ++

Once QuakeWorld is installed alongside your Quake files, you can start it up like:

```
./qwcl +connect some.server.address
```

See section Related Software for info about some nifty front ends for QuakeWorld that make finding servers easy.

- **lib3dfxgl.so** **

  With Quakeworld version 2.30, an alternative to the Mesa library is available. `lib3dfxgl.so` is a mini-GL driver optimized for Quake that provides slightly better framerates than Mesa. This is a port of a driver that 3Dfx developed for Quake under Windows, and apparently not all of its features work properly yet. So hopefully we can expect its performance to improve with time.

  Like Mesa, `lib3dfxgl.so` requires Glide in order to access your 3Dfx card. The Quakeworld packages come with a script, `glqwcl.3dfxgl` for running Quakeworld with this library on glibc systems. The next paragraph explains how to run Quakeworld with `lib3dfxgl.so` on a libc5 system. On a glibc system in order for this script to work,the `glqwcl` executable **must not be setuid**, nor should you run it as root. `glqwcl` will silently load Mesa rather than `lib3dfxgl.so` if it runs with root permissions. This non-root requirement implies that you have the `/dev/3dfx` driver installed.

  On a libc5 system, you need to create a symbolic link to `lib3dfxgl.so` called `libMesaGL.so.2` like so:

```
cd /usr/local/games/quake
ln -sf lib3dfxgl.so libMesaGL.so.2
```

  Then start Quakeworld from a script that tells `$LD_LIBRARY_PATH` to look in the current directory:

```
#!/bin/sh

LD_LIBRARY_PATH=".:$LD_LIBRARY_PATH" ./glqwcl $*
```

  You can tell which driver is being loaded by looking for output like the following in your console as Quakeworld starts:

```
GL_VENDOR: 3Dfx Interactive Inc.
GL_RENDERER: 3Dfx Interactive Voodoo^2(tm)
GL_VERSION: 1.1
GL_EXTENSIONS: 3DFX_set_global_palette WGL_EXT_swap_control
GL_EXT_paletted_texture GL_EXT_shared_texture_palette GL_SGIS_multitexture
```

  If `GL_VENDOR` says `Brian Paul` rather than `3Dfx Interactive Inc.`, that means Mesa is still being used rather than the miniport dirver.

- **glqwcl.glx** **

  `glqwcl.glx` is linked against standard OpenGL libraries instead of Mesa. This allows Quakeworld to run on other 3D hardware that is supported by other OpenGL implementations. At this time, I dont' know of any OpenGL

TOC

implementations that support hardware other than 3Dfx, but this renderer ensures that when they appear, we'll b able to play Quakeworld with them.

This is a GLX application, and as such, must be run from X.

You can use this client with Mesa/3Dfx if you install Mesa and Glide as explained in the previous section, then set the `$MESA_GLX_FX` environment variable to "fullscreen" before you run `quake2`:

```
export MESA_GLX_FX=fullscreen
./glqwcl.glx +_windowed_mouse 1
```

Why the `+_windowed_mouse 1` option? Remember that this is an X application which happens to use your 3Dfx card. Even though the display takes up your entire screen, Quakeworld is stil running in a window. This means that if you're not *very* careful, you could move the mouse pointer outside the Quakeworld window, and Quakeworld will suddenly stop responding to mouse and keyboard input. `+_windowed_mouse 1` avoids this problem by telling `glqwcl.glx` to grab the mouse and not let it move outside its window.

2.10 Servers

Most, if not all, existing information about running a DOS/Windows QW server is equally applicable to running a Linux server.

To start a QuakeWorld server, simply do:

```
./qwsv
```

The official QuakeWorld server manual lives at http://qwcentral.stomped.com.

2.11 Mods & Addons

One of the very cool things about the Quake games is that the authors made them easily extensible. End users can create their own levels, add new weapons or monsters, or even completely change the rules of the game.

Capture the Flag

This is my favorite variation of both Quake and Quake 2. Instead of just running around and killing everyone you meet (which definitely has its merits, don't get me wrong!), CTF is team-based and more strategic. Dave 'Zoid' Kirsch, also the maintainer of the Linux Quake ports, created this mod.

Everything you need to know about CTF can be found at http://captured.com/threewave/ You need ftp://ftp.cdrom.com/pub/quake/planetquake/threewave/ctf/client/3wctfc.zip to play. To install the client piece of CTF, simply create a directory called `ctf` in your Quake directory and unzip the `3wctfc.zip` file there. The accompanying `readme.txt` file is chock full of good information.

```
cd /usr/local/games/quake
mkdir ctf
cd ctf
unzip -L /wherever/you/put/it/3wctfc.zip
```

For information on running a CTF server, see the web pages mentioned above.

TOC

Mission Packs **

Activision released two add-on packs of extra levels for Quake, **Scourge of Armagon** and **Dissolution of Eternity**.

- **Mission Pack 1: The Scourge of Armagon** Assuming your CD is mounted on `/mnt/cdrom` and Quake is installed in `/usr/local/games/quake`:

  ```
  cd /usr/local/games/quake
  mkdir hipnotic
  cp /mnt/cdrom/hipnotic/pak0.pak hipnotic
  cp /mnt/cdrom/hipnotic/config.cfg hipnotic
  ```

  Play the mission pack like this:

  ```
  cd /usr/local/games/quake
  ./quake.x11 -game hipnotic
  ```

- **Mission Pack 2: Dissolution of Eternity** Installing the second mission pack is pretty much the same procedure as installing the first. Follow the directions for Mission Pack 1, but replace the word *hipnotic* with the word *rogue*, and skip the `config.cfg` step, as this file isn't included on the Mission Pack 2 CD.

Quake Tools

Anybody care to contribute some info about qcc, bsp and all that?

---

3. Quake II

To install Quake II on your Linux system, you'll need some flavor of the official Quake II distribution from id. This will be either the retail Windows CD-ROM that you bought at your favorite software store, or the demo version you downloaded from the net. See Download the Necessary Files for details on acquiring the demo version.

Alternatively, if you've already got Quake installed on a Windows machine, you can use the relevant files from that installation.

3.1 Prerequisites

You will need, as a bare minimum, the following:

- A Pentium 90 or better (133 recommended) computer
- 16 MB RAM (24 recommended)
- The Quake 2 CD-ROM **or** the demo version (q2-314-demo-x86.exe)
- Linux kernel version 2.0.24 or later
- libc 5.2.18 or later
- One of the following:
  - X11 server that supports the MITSM shared memory extension. 8 and 16 bit displays are supported. (for X renderer)
  - SVGAlib 1.2.10 or later (for SVGA and GL renderer)
- 25-400 megabytes free disk space (depending on how you install)
- Access to the root account of the machine you're installing on

Optional:

- A supported soundcard
- A 3Dfx Voodoo Graphics or Voodoo2 or Voodoo Rush 3D graphics accelerator card.
- 3Dfx glide libraries installed (for GL renderer)
- Mesa 2.6 or later (for GL renderer)

3.2 Installing Quake II

Download the Necessary Files**

All the necessary files for Linux Quake II are available at id Software's ftp site, ftp.idsoftware.com. This site can be quite busy at times,so you may want to use one of these mirror sites instead:

- ftp.cdrom.com/pub/idgames/idstuff (California, USA)
- ftp.gamesnet.net/idsoftware (California, USA)
- ftp.stomped.com/pub/mirror/idstuff (Minnesota, USA)
- mirrors.telepac.pt/pub/idgames (Lisbon, Portugal)
- download.netvision.net.il/pub/mirrors/idsoftware (Haifa, Israel)

TOC

The Quake II files mentioned in in this section are:

- Quake II Linux Binaries
  - libc5 tar.gz package ftp.idsoftware.com/idstuff/quake2/unix/quake2-3.19a-i386-unknown-linux2.0.tar.gz
  - glibc tar.gz package ftp.idsoftware.com/idstuff/quake2/unix/quake2-3.19a-glibc-i386-unknown-linux2.0.tar.gz
  - libc5 rpm package ftp.idsoftware.com/idstuff/quake2/unix/quake2-3.19a-2.i386.rpm
  - glibc rpm package ftp.idsoftware.com/idstuff/quake2/unix/quake2-3.19a-glibc-2.i386.rpm
- Quake II Demo Version for Windows ftp://ftp.idsoftware.com/idstuff/quake2/q2-314-demo-x86.exe
- Quake II Game Source ftp://ftp.idsoftware.com/idstuff/quake2/source/q2source-3.14.shar.Z
- Quake II Capture the Flag ftp://ftp.idsoftware.com/idstuff/quake2/ctf/102.zip

Other software mentioned:

- SVGAlib graphics library http://sunsite.unc.edu/pub/Linux/libs/graphics/svgalib-1.3.0.tar.gz
- SVGAlib libc5 binary http://sunsite.unc.edu/pub/Linux/libs/graphics/svgalib-1.3.0.libc5.bin.tar.gz
- Glide runtime libraries http://glide.xxedgexx.com/3DfxRPMS.html
- Mesa 3D graphics library http://www.ssec.wisc.edu/~brianp/Mesa.html
- unzip archive utility http://sunsite.unc.edu/pub/Linux/utils/compress/unzip-5.31.tar.gz

Create the Installation Directory

The first thing you'll need to do is decide where you want to install Quake II. Lots of folks like to put it in `/usr/games/quake`. Anal system administrator that I am, I choose to install anything that's not part of my Linux distribution under `/usr/local`. So for me, Quake II goes in `/usr/local/games/quake2`. If you choose to install somewhere else, please substitute the appropriate path wherever `/usr/local/games/quake2` is mentioned.

So go ahead and create the directory you'll install Quake II in, and cd to it. The rest of these instructions will assume that this is your current directory.

```
mkdir /usr/local/games/quake2
cd /usr/local/games/quake2
```

Installing from CD

Place your Quake II CD in your CD ROM drive, and mount it:

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

If your CDROM is typically mounted elsewhere, substitute its location for `/mnt/cdrom`. If you're not sure where your CDROM is mounted, please see the documentation for your particular distribution.

- **Full Install** The simplest install method is a "full" install, which involves copying the entire contents of your CD to your hard drive. This requires about 350 MB, and is accomplished by issuing the following commands:

TOC

```
cd /usr/local/games/quake2
cp -r /mnt/cdrom/Install/Data/* .
```

There'll be a bunch of unnecessary Windows files hanging around that you can safely delete:

```
rm -f /usr/local/quake2/*.dll
rm -f /usr/local/quake2/quake2.exe
rm -f /usr/local/quake2/baseq2/gamex386.dll
```

- **Medium Install** If 450 MB is too much space for you to devote to Quake II, you can skip installing the movie cutscenes and link to them on the CD-ROM instead. This will reduce the space requirements to about 200 MB:

```
cd /usr/local/games/quake2
mkdir baseq2
cp /mnt/cdrom/Install/Data/baseq2/pak0.pak baseq2
cp -r /mnt/cdrom/Install/Data/baseq2/players baseq2
ln -s /mnt/cdrom/Install/Data/baseq2/video baseq2/video
```

  Note that this doesn't mean you have to mount your Quake II CD every time to want to play Quake II. If the game can't load the videos, it just won't display them.

Windows to Linux install

If you have Quake II installed under Windows on a different machine, you can transfer the files in `quake2\baseq2\` to your Linux system via FTP or some other mechanism. Keep in mind that the filenames on your Linux system must be in lower case for Quake II to find them, so you may have to rename them after the transfer. Also note that it may be necessary to delete your Windows installation after you do this to remain in compliance with the terms of id's software license. It's not my fault if you do something illegal.

If your Windows and Linux systems are on the same machine, you have two options: copy the files from your Windows partition to your Linux partition, or link to the necessary files from Linux. Both options will work equally well. You'll just save a lot of disk space when you link instead of copy. As usual, replace `/win95/games/quake2` in the following examples with the correct path to your Windows partition and Quake II installation.

- If you want to copy the files from your Windows partition, do something like this:

```
cd /usr/local/games/quake2
cp -r /win95/games/quake2/baseq2 .
```

- To create links to your Windows Quake II files instead, do this:

```
cd /usr/local/games/quake2
ln -s /win95/games/quake2/baseq2 .
```

  This second method requires that the Windows partition you're linking to be writeable by users, which may not be appropriate for all systems. By making your Windows partition writeable, you are giving all users the opportunity to destroy your entire Windows installation. If that's ok with you, modify your `/etc/fstab` to

mount the Windows partition with the options *umask=002,gid=XXX*, where XXX is the group id number the "users" group. Look in `/etc/group` for this information. When `fstab` is updated, umount and re-mount the Windows partition and you're done.

TOC

You're done installing the Quake II data files. Move ahead to " Installing the Linux Binaries".

Installing the demo version **

id Software has a freely available demo version of Quake II at their ftp site. It's a 40 megabyte download. The demo includes all features of the full version, including multiplayer, but it only comes with three levels, so it may be difficult to find a server to play on.

See the section Download the Necessary Files above for the location of the Quake II demo. Download it and place it in your Quake II directory.

The demo distribution is a self-extracting zip file (it's self-extracting in other OS's anyway). You can extract it with the **unzip(1)** command, which should be included in most modern distributions. If you don't have unzip, you can download it from the location listed in the Download the Necessary Files section.

cd to your Quake II directory and extract the archive:

```
cd /usr/local/games/quake2
unzip q2-314-demo-x86.exe
```

Now we've got to delete some things and move some other things around:

```
rm -rf Splash Setup.exe
mv Install/Data/baseq2 .
mv Install/Data/DOCS docs
rm -rf Install
rm -f baseq2/gamex86.dll
```

The Quake II demo is now installed. You just need to add the Linux binaries.

3.3 Adding the Linux Binaries **

There are four Linux Quake II packages available for download:

- a libc5 tar.gz package
- a glibc tar.gz package
- a libc5 rpm package
- a glibc rpm package

Install just one of these packages. Each contains the same files, they're just linked against different libraries. Redhat 5.x users should choose the the glibc rpm package. Users of glibc based systems without rpm support should use the glibc tar package. The libc5 rpm is for Redhat distributions prior to 5.0 and other distributions that use the rpm package format. The libc5 tar.gz package is for Slackware and everyone else.

See the Download the Necessary Files section for the location of the Linux Quake II files.

TOC

Installing the RPM packages **

Installation of the rpm packages should be as simple as:

```
su root
rpm -Uvh quake2-xxxxx.i386.rpm
```

Rpm may complain that it can't find `libglide2x.so`. The Glide library is only necessary if you have a 3Dfx card and want to run Quake II in GL mode. If you don't plan to use the GL mode, you can override the glide dependency with the `--nodeps` option:

```
su root
rpm -Uvh quake2-xxxxx.i386.rpm --nodeps
```

Installing the tar.gz packages **

To install, just untar the file in your Quake II directory. Do it as root so the proper file permissions get set:

```
cd /usr/local/games/quake
su root
tar -xzf qwcl2.21-i386-unknown-linux2.0.tar.gz
```

3.4 Setting Permissions **

If you ran `rpm` or `tar` as root when installing the Quake II package on your system, the file permissions should be properly set already. The `quake2` executable was installed setuid root so that it can access the graphics devices on your system. For security, the `ref_*.so` rendering libraries are owned by root and writeable only by him. If root doesn't own the libraries, or they're world writeable, `quake2` will refuse to run.

If you plan to only run Quake II with the GL or X renderers, your `quake2` doesn't need to be setuid root. See Running X and GL games without setuid in the Tips and Tricks section below for information on running Quake II without root permissions.

Quake2.conf **

For security reasons, there is a `quake2.conf` file, which tells Quake II where to find the rendering libraries it needs (`ref_*.so`). It contains only one line, which should be the path to your Quake II installation. Quake II looks for this file in `/etc`. If you installed Quake II from an `.rpm` file, this file was installed for you. If you installed from a `.tar` package, you need to create it like so:

```
su root
cd /usr/local/games/quake2
pwd > /etc/quake2.conf
chmod 644 /etc/quake2.conf
```

TOC

3.5 The X Renderer

Quake II should be ready to run under X now. Give it a try:

```
cd /usr/local/games/quake2
./quake2 +set vid_ref softx
```

If all is well, after a pretty significant pause, a small Quake II window will appear with the first demo running in it. You should hear sound effects and possibly music, if the CD is mounted. If any of this fails to occur, please see section Troubleshooting for help.

3.6 The SVGAlib Renderer

You need SVGAlib installed and configured if you're going to use either the ref_soft or ref_gl renderers. (Quake II uses SVGAlib to process keyboard and mouse input, in case you're wondering why you'd need it for the GL renderer). SVGAlib comes with most modern distributions, and **must** be properly configured before Quake II will run correctly outside of X.

`libvga.config` is SVGAlib's configuration file. On most systems you'll find it in either `/etc` or `/etc/vga`. Make sure the mouse, monitor, and video card settings in this file are correct for your system. See the SVGAlib documentation for more details.

If you don't already have SVGAlib on your system, download it from the location mentioned in the files section above. If you have a RedHat 5.x or other glibc-based Linux distribution, see Glibc, RedHat 5.x, Debian 2 considerations in the Troubleshooting/FAQs section for important information about compiling libraries for Quake II. A precompiled libc5 SVGAlib binary is available at http://sunsite.unc.edu/pub/Linux/libs/graphics/svgalib-1.3.0.libc5.bin.tar.gz for those who don't want to deal with the hassle of compiling for libc5.

You should run Quake II from a virtual console when using the ref_soft or ref_gl renderers. It won't run from X unless you're root when you start it, and that's not advisable. So if you're in X, do a CTRL+ALT+F1, login and then:

```
cd /usr/local/games/quake2
./quake2
```

Running SVGA and GL games from X in the Tips & Tricks section below explains how to launch SVGA and GL Quake II from X without manually switching to a virtual console.

3.7 The OpenGL Renderer

Hardware-accelerated OpenGL Quake is Quake the way God intended it to be. There is no substitute, and once you've experienced it there's no going back.

To run Quake II in GL mode, you need a 3D card with the Voodoo, Voodoo2 or Voodoo Rush graphics chipset on it. There are specific issues to be dealt with if you have a Voodoo Rush card, and I won't go into them now because frankly, I wouldn't know what I was talking about. A future version of this HOWTO will cover Rush issues (If somebody wants to write about Voodoo Rush issues, I'll gladly include it here).

TOC

The SVGAlib, Glide, and Mesa libraries must all be installed and configured properly on your system for `quake` to work. The following sections will very briefly cover what you need to do to get them going.

Bernd Kreimeier's (bk@gamers.org) Linux 3Dfx HOWTO (http://www.gamers.org/dEngine/xf3D/howto/3Dfx-HOWTO.html) is good source for further information.

The 3dfx.glide.linux newsgroup on the 3dfx news server (news.3dfx.com) is another good source of information about the intersection of Linux, glide, Mesa and Quake.

SVGAlib

Quake II uses SVGAlib to get input from the mouse and keyboard, so you'll need to configure it as outlined in section SVGAlib Renderer section.

Glide

Glide is a library that provides an API for programming 3Dfx based cards. If you want the Mesa graphics library to use your 3Dfx card, you've gotta have it.

The latest version of glide can always be found at http://glide.xxedgexx.com/3DfxRPMS.html. Select the package(s) appropriate for your system, and install according to the instructions on the web page.

Note that unless you download the 3Dfx device driver package in addition to the Glide library, you will only be able to run Glide applications (like GLQuake) as root. Install the `/dev/3dfx` module and you can play GLQuake as a regular user.

Once you have glide installed, try out the test program that comes with it. Remember this program: it's a good way to reset your display if you ever have a glide application (like GLQuake) crash and leave your screen switched off. **NOTE: run this test from a VC, *not* X!** It's possible for the test app to lose mouse and keyboard focus in X, and then you'll have no way of shutting it down.

```
/usr/local/glide/bin/test3Dfx
```

Your screen should turn blue and prompt you to hit any key. After you press a key you should be returned to the prompt. 3dfx.glide.linux on 3dfx's news server (news.3dfx.com) is a great source of information for Linux glide-specific problems.

Mesa

Once glide's installed, you need to install Mesa, a free OpenGL implementation by Brian Paul (brianp@elastic.avid.com). Luckily, you won't have to look far, because Mesa 2.6 is included with the Quake II binaries. All you have to do is move it to the right place:

```
cd /usr/local/games/quake2
cp libMesaGL.so.2.6 /usr/local/lib
ldconfig
```

If you want to upgrade Mesa to a more recent version (Mesa 3.0 is the most recent version as of this writing), you can download the latest from ftp://iris.ssec.wisc.edu/pub/Mesa If you have a RedHat 5.x or other glibc-based Linux

TOC

distribution, see Glibc, RedHat 5.x, Debian 2 considerations in the Troubleshooting/FAQs section for important information about compiling libraries for Quake.

After you've built it according to the instructions, you will have to do two things:

- Remove your old Mesa installation. If you previously installed a `libMesaGL.so.2.6` as described above, you must remove it or Quake II may not use the new version.

```
cd /usr/local/lib/
rm -f libMesaGL.so.2*
```

- If the new Mesa has a major version number that's greater than 2, you need to create a link to it with the name `libMesaGL.so.2`:

```
cd /usr/local/lib/
ln -s /wherever/you/installed/it/libMesaGL.so.3.0 libMesaGL.so.2
ldconfig
```

Now switch to a VC (CTRL+ALT+F1) and start Quake II:

```
cd /usr/local/games/quake2
./quake2 +set vid_ref gl
```

lib3dfxgl.so **

With Quake II version 3.19, an alternative to the Mesa library is available. `lib3dfxgl.so` is a mini-GL driver optimized for Quake that provides slightly better framerates than Mesa. This is a port of a driver that 3Dfx developed for Quake under Windows, and apparently not all of its features work properly yet. So hopefully we can expect its performance to improve with time.

Like Mesa, `lib3dfxgl.so` requires Glide in order to access your 3Dfx card. The Quake II packages come with a script, `quake2.3dfxgl` for running Quake II with this library on glibc systems. The next paragraph explains how to run Quake II with `lib3dfxgl.so` on a libc5 system. On a glibc system, the `glqwcl` executable **must not be setuid**, nor should you run it as root. `glqwcl` will silently load Mesa rather than `lib3dfxgl.so` if it runs with root permissions. This non-root requirement implies that you have the `/dev/3dfx` driver installed.

On a libc5 system, the non-root requirements mentioned above apply, but you also need to create a symbolic link to `lib3dfxgl.so` called `libMesaGL.so.2` like so:

```
cd /usr/local/games/quake2
ln -sf lib3dfxgl.so libMesaGL.so.2
```

Then start Quake II from a script that tells `$LD_LIBRARY_PATH` to look in the current directory:

```
#!/bin/sh

LD_LIBRARY_PATH=".:$LD_LIBRARY_PATH" ./quake2 +set vid_ref gl $*
```

You can tell which driver is being loaded by looking for output like the following in your console as Quake II s

```
GL_VENDOR: 3Dfx Interactive Inc.
GL_RENDERER: 3Dfx Interactive Voodoo^2(tm)
GL_VERSION: 1.1
GL_EXTENSIONS: 3DFX_set_global_palette WGL_EXT_swap_control
GL_EXT_paletted_texture GL_EXT_shared_texture_palette GL_SGIS_multitexture
```

If `GL_VENDOR` says `Brian Paul` rather than `3Dfx Interactive Inc.`, that means Mesa is still being used rather than the miniport dirver.

3.8 The GLX Renderer **

`ref_glx.so` is linked against standard OpenGL libraries instead of Mesa. This allows Quake II to run on other 3D hardware that is supported by other OpenGL implementations. At this time, I dont' know of any OpenGL implementations that support hardware other than 3Dfx, but this renderer ensures that when they appear, we'll be able to play Quake II with them.

This is a GLX application, and as such, must be run from X.

You can use this client with Mesa/3Dfx if you install Mesa and Glide as explained in the previous section, then set the `$MESA_GLX_FX` environment variable to "fullscreen" before you run `quake2`:

```
export MESA_GLX_FX=fullscreen
./quake2 +set vid_ref glx +set _windowed_mouse 1
```

Why the `+set _windowed_mouse 1` option? Remember that this is an X application which happens to use your 3Dfx card. Even though the display takes up your entire screen, Quake II is stil running in a window. This means that if you're not *very* careful, you could move the mouse pointer outside the Quake II window, and Quake II will suddenly stop responding to mouse and keyboard input. `+set _windowed_mouse 1` avoids this problem by telling `quake2` to grab the mouse and not let it move outside its window.

3.9 Linux-Specific Command Line Options

This section will cover command line options that are specific to the Linux version of Quake II. There are plenty of other Quake II options, but they're beyond the scope of this HOWTO. Check out some of the sites listed in section General Quake Information for this kind of information.

These are actually cvars (client variables) that you can set in the Q2 console, but it makes the most sense to set them on the command line. Set them with `+set` on the command line, like:

```
./quake2 +set cd_dev /dev/hdc
```

cd_dev ***device***

Name of the CD-ROM device.

nocdaudio ***value***

Disable CD audio if *value* is nonzero

sndbits *num*

Set sound bit sample size. Default is 16.

sndspeed *num*

Set sound sample speed. Usual values are 8000, 11025, 22051 and 44100. If set to zero, causes the sound driver to attempt speeds in the following order: 11025, 22051, 44100, 8000.

sndchannels *num*

Indicates stereo or mono sound. Defaults to 2 (stereo). Use 1 for mono.

nostdout *value*

Don't do any output to stdout. Use this if you don't want all the console output dumped to your terminal.

3.10 Quake II Servers

Linux's strength as an internet server make it a perfect platform for running an internet Quake II server. This section will touch on the basics and Linux-specific aspects of starting up a Quake II server. More detailed information about running Quake II servers is available elsewhere (see section Other Sources of Information earlier in this document.

Listen Servers

You can start a Quake II "Listen" server from within the game via the *Multiplayer* menu. This allows you to host a game and participate in it at the same time.

To start a Listen server, start Quake II, bring up the Quake II menu with the ESC key, and select *Multiplayer*. It should be pretty self-explanatory from there.

Dedicated Servers

For a permanent, stand-alone Quake II server that needs to run without constant attention, using the Listen server is impractical. Quake II has a Dedicated server mode that is better suited to this type of use. A dedicated server is started from the command line and uses fewer system resources than a Listen server because it doesn't start the graphical client piece at all.

To start a dedicated server, use the command line option `+set dedicated 1`. You can set additional server parameters either on the command line or in a config file that you `+exec` on the command line. Your config file should reside in the `baseq2` directory.

A few common server options are listed below. To set options on the command line, do `+set fraglimit 30`. Options are set the same way in a config file, only you don't want the `+` before the `set`. Invoke your config file like this: `+exec server.cfg`.

fraglimit

Number of frags required before the map changes

timelimit

Time in minutes that must pass before the map changes

hostname

The name of your Quake II server. This is an arbitrary string and has nothing to do with your DNS hostname.

TOC

maxclients

The maximum number of players that can connect to the server at once.

For enough Quake II console and command line information to choke a horse, see Farenheit 176 (http://www.planetquake.com/f176).

Other Sources of Server Information

- The Q2 Server FAQ has a basic step-by-step guide to set up a Q2 Server under Linux: http://www.bluesnews.com/faqs/q2s-faq.html
- Grant Cornelius Reticulus Copernicus Sperry (flubber@xmission.com) has some basic Q2 server config files and startup scripts at http://www.atomicage.com:80/quake/server/server_cfg/.

3.11 Mods & Addons

Quake II modifications like Capture the Flag, Jailbreak, and Lithium II are very popular extensions of the original Quake II game. Some mods reside entirely on the server (Lithium), and some also require changes to your client (CTF). For server only mods, you just connect normally and play. Client-side mods require you to install additional files in your `quake2` directory before you can play.

Client Side Mods

Generally, installation of a client-side mod consists of just downloading the client package and upacking it in your Quake II directory, but you should refer to the mod's documentation for specific details. It may be necessary to download a Linux-specific package in addition to the main (Windows) client package. Also be aware that all mods may not be available for Linux.

Client-side mod packages usually contain a new `gamei386.so` file and one or more `.pak` files. Other new files may be included as well. These new files will be installed in a subdirectory below your Quake II directory. Use `+set game` *`mod-dir`* on the command line to run the mod. Rocket Arena 2, for example, gets installed in a directory called `arena`. To play RA2, your would start your client like so:

```
./quake2 +set game arena
```

Capture the Flag

Since this is by far the most popular variation of multiplayer Quake II, I've included specific instructions for installing this mod. Capture the Flag for Quake II is available from id's ftp site. Download it, then install like so:

```
cd /usr/local/games/quake2
mkdir ctf
cd ctf
unzip -L /wherever/you/put/it/q2ctf102.zip
```

Start Quake II with `+set game ctf` to play CTF.

Server Side Mods

Running a Quake II mod on a server isn't much different than running one on the client side. Generally you'll need to install `gamei386.so` and `server.cfg` files in a new subdirectory and then start your server like

```
./quake2 +set game XXXX +set dedicated 1 +exec server.cfg
```

Where XXXX above is the name of the mod's new subdirectory. The exact procedure will vary from mod to mod, of course. See the mod's documentation for specific details.

Game Source

The entire game, with the exception of the engine itself, resides in a shared library, `gamei386.so`. Quake II mods are created by changing the contents of this file. The C source is freely available (section Download the Necessary Files above) for anyone to download and modify.

After you've downloaded the source, here's how to get started with it:

```
cd /usr/local/games/quake2
mkdir mymod
cd mymod
gunzip /wherever/you/put/it/q2source-3.14.shar.Z
sh /wherever/you/put/it/q2source-3.14.shar
```

You'll be presented with a bunch of legalese that you must answer `yes` to, then the game source will be extracted. Building a new `gamei386.so` out of these sources is accomplished with a simple `make`. You can run Quake II with the newly compiled library like so:

```
cd /usr/local/games/quake2
./quake2 +set game mymod
```

Not too exciting yet, since what you just built is identical to the "stock" `gamei386.so`, but this should be good information for aspiring mod authors.

Mission Packs **

- **Mission Pack 1: The Reckoning** The Reckoning requires Quake II version 3.15 or later to run. You'll need at least 95 MB for a minimum installation. Another 90 MB are required if you want to install the video sequences as well. Assuming your CD is mounted on `/mnt/cdrom` and Quake II is installed in `/usr/local/games/quake2`:

  ```
  cd /usr/local/games/quake2
  cp -r /mnt/cdrom/Data/all/* xatrix/
  rm -f xatrix/gamex86.dll
  ```

  If you want to install the video sequences:

  ```
  cp -r /mnt/cdrom/Data/max/xatrix/video xatrix
  ```

TOC

Play The Reckoning like this:

```
cd /usr/local/games/quake2
./quake2 +set game xatrix
```

- **Mission Pack 2: Ground Zero** It requires Quake II version 3.17 or later to run. You'll need at least 120 MB for a minimum installation. Another 115 MB are required if you want to install the video sequences as well. Assuming your CD is mounted on `/mnt/cdrom` and Quake II is installed in `/usr/local/games/quake2`:

```
cd /usr/local/games/quake2
cp -r /mnt/cdrom/Data/all/* rogue/
rm -f rogue/gamex86.dll
```

If you want to install the video sequences:

```
cp -r /mnt/cdrom/Data/max/rogue/video rogue
```

Play Ground Zero like this:

```
cd /usr/local/games/quake2
./quake2 +set game rogue
```

---

4. Related Software ++

4.1 QStat

Qstat is a command line based program that returns the status of internet Quake, QuakeWorld, and Quake 2 servers created by Steve Jankowski mailto:steve@activesw.com.

Here's the feature summary from the QStat homepage:

- Supports Windows 95, NT, and most Unixes
- Comes with C source code and a binary for Windows
- Supports old Quake (NetQuake), QuakeWorld, Hexen II, and Quake II servers
- Can display all available statistics, including player info and server rules
- Output templates for automatic HTML generation
- Raw display mode for integration with HTML page generators
- Built-in host name cache
- Sort by ping time, game, or both
- More options than you can wiggle a mouse at

Qstat is a must-have tool if you're planning on doing any net play. A number of front-ends for qstat have been written as well. Some of them are listed later in this section.

You can get the latest version of qstat from the QStat Homepage (http://www.activesw.com/people/steve/qstat.html.

4.2 XQF

XQF is a graphical front-end to QStat that uses the GTK toolkit. This is the best QuakeWorld/Quake2 server browser that currently exists, and Roman Pozlevich (roma@botik.ru), is still cranking out revisions at the rate of about one per month.

If you're familiar with GameSpy for the Windows platform, this is the closest thing to it for Linux.

The XQF homepage is at http://www.linuxgames.com/xqf.

4.3 QuickSpy

QuickSpy is a text-based QuakeWorld server browser. It's another front-end to QStat and it works pretty well. If you don't run X and you don't have Quake II, this is a decent option. Beware though, it's no longer under development.

TOC

You can get QuickSpy at http://diana.ecs.soton.ac.uk/~rht96r/quake/quickspy/.

4.4 QPlug for Linux

Qplug is a Netscape plugin which retrieves and displays QuakeWorld and Quake II server information embedded in a web page. A Windows Qplug has been around for some time. The author, Olivier Debon (odebon@club-internet.fr) wrote the Linux version from scratch without ever having seen the Windows version.

QPlug for Linux can be got at http://www.geocities.com/TimesSquare/Labyrinth/5084/qplug.html.

4.5 qkHacklib

David Bucciarelli (tech.hmw@plus.it), author of the 3Dfx driver for Mesa, has written a library called qkHack, which tries to emulate all the SVGAlib/fxMesa functions used by Quake and Quake II. This would remove the need for SVGAlib when running glquake or Quake II with ref_gl. Other features from the qkHacklib README:

- You can dynamically switch between fullscreen rendering and the in window rendering just pressing the TAB key (you must start your X server in 16 bpp mode in order to use this feature)
- You can press Ctrl-C in the shell or kill the Quake process without problems
- you can enable/disable the mouse and keyboard 'grabbing' pressing F11/F12
- you can iconify and pause Quake pressing the F10 (it will not eat more CPU cycles). You can restart everything with a double click in the "****" icon
- you can run Quake with any Mesa driver (for example with the X11 driver but you must recompile the Mesa without the Voodoo driver)
- you can run Quake under any Linux box and get the hardware acclerated output on a SGI box (OK, this is a bit exotic and theoretical as feature but it is an example of how powerful can be an GLX/OpenGL application)

I've tried qkHacklib and it works great for Quake on my system. In Quake II, however, the mouse response becomes *really* slow. Others report complete success, though, so give it a try it it sounds like something you need.

David Bucciarelli's qkHacklib web page is at http://www-hmw.caribel.pisa.it/fxmesa/fxqkhack.html.

4.6 GiMd2Viewer

GiMd2Viewer is a Quake 2 model viewer written for Gtk and OpenGL by Lionel Ulmer (bbrox@mygale.org). It loads models and textures from either plain files or .PAK files. It will also animate the models (with frame interpolation).

This program is still under developement and I haven't tried it yet, but it sounds pretty nifty. Check it out

at http://www.mygale.org/~bbrox/GiMd2Viewer/.

4.7 QIPX

QIPX is a set of programs that allow Linux Quake clients (using TCP/IP) to connect with DOS Quake clients (using IPX). I guess this is useful if you're playing netquake on a LAN. QIPX is available at http://www.geocities.com/SiliconValley/Park/6083/qipx.html.

4.8 Ice

Ice is a Quake map editor for UN*X created by C.J. Beyer and John Watson. I haven't used this program, nor do I know what its development status is. The Ice homepage is at http://styx.phy.vanderbilt.edu/~ice/.

4.9 Q2getty

Q2getty is a hack of mingetty by Mike Gleason (mgleason@ncftp.com) that allows you to automatically run and respawn a program (like a Quake server) on a virtual console. This program is available in the files section at http://www.ncftpd.com/unixstuff/q2getty.html.

4.10 rcon

Rcon is a pair of tools that allow remote administration of a Quake II server using the RCON protocol. Michael Dwyer (michael_dwyer@mwiworks.com) is the author. Rcon 1.1 is available at http://sunsite.unc.edu/pub/Linux/games/quake/rcon-1.1.tar.gz.

4.11 qlog ++

Qlog is a GPL'ed QuakeWorld/Quake II server log parser that generates comprehensive player statistics. Craig Knudsen (cknudsen@radix.net) is the author. The qlog homepage is http://www.radix.net/~cknudsen/qlog/.

4.12 Cheapo **

Cheapo is a proxy that can be used to route QuakeWorld network traffic. Additionally, the proxy can modify the data and has features for enhancing gameplay. You can connect to cheapo as if it were a Quakeworld server, and then give it commands that forward you to a real server. The proxy can also be run on a firewall machine incapable of handling Quake traffic, so that machines inside the firewall can be used for playing. The Cheapo homepage is at http://www.saunalahti.fi/~softech/.

4.13 qgraph **

TOC

QGraph (Quake Graph) is a utility to help people in managing Quake's DeathMathes, turnments and Quakeworld games. QGraph is a program who connects (via Lan or the Internet) to a Quake, QuakeII, Quakeworld and Hexen2 Servers and shows you realtime data about the game running on that server. The QGraph homepage is at http://www.frag.com/qgraph.

---

5. Troubleshooting/FAQs

5.1 General

OS difference considerations

- **Case sensitivity** - In DOS and Windows, case is not important. `BASE1.TXT` is the same as `base1.txt`. Under Linux and other unices, case IS significant. `MOTD.TXT` and `motd.txt` are different files. This can cause problems with player models and skin files if they're installed with upper- or mixed-case filenames. `players/male/santa.PCX` needs to be renamed to `santa.pcx` in order to work in Linux. The `fixskins.sh` script included with quakeworld will convert all filenames in a directory to lowercase. It's reproduced below for your convenience:

```
#!/bin/sh
for x in *; do
   y=`echo $x | tr '[A-Z]' '[a-z]'`
   if [ $x != $y ]; then
      mv $x $y
   fi
done
```

- **Path delimiters** - DOS and Windows use the backslash "\" character to separate file pathname elements. In Unix, the backslash is an escape character. If you use file pathnames in your config files (or your mod code, or anywhere else, for that matter), be sure you're using "`/`" and not "\".
- **End of line characters** - Under DOS/Windows, each line of a text file ends with a carriage return character (CR) and a linefeed character (LF). Unix textfiles just have a linefeed at the end of each line. Using DOS/Win formatted textfiles in Unix can cause all kinds of mysterious Quake problems. Using the incorrectly formatted `quake2.conf` file from the Quake2 3.17 package, for example, gives the error "`LoadLibrary("ref_XXX.so") failed: No such file or directory`". LMCTF-TE reports a floating point exception. If you've got an problem you can't explain, try removing the CRs from your text files:

```
mv file.txt file.bak; tr -d '\r' < file.bak > file.txt
```

Glibc, RedHat 5.x, Debian 2 considerations **

The following applies the Quake I binaries (`squake`, `glquake`, and `quake.x11`) only. As of versions 2.30 and 3.19 respectively, QuakeWorld and Quake II are available in both libc5 and glibc versions.

The Quake executables were compiled with libc5. Newer Linux distributions like RedHat 5.1 and Debian 2.0 use the incompatible glibc as their default C library. If you're running Quake on a glibc system, there are a few things to watch out for:

- Both RedHat 5 and Debian 2 have libc5 compatibility packages that allow you to run libc5-based applications. Make sure you have these packages installed. Both distributions put the libc5-based libraries in `/usr/i486-linux-libc5/lib`.
- Make sure Quake uses the correct libraries. Create a script like the one below that points `$LD_LIBRARY_PATH` to

your compatibility libraries directory before it runs Quake.

```
#!/bin/sh
export LD_LIBRARY_PATH=/usr/i486-linux-libc5/lib
./quake2 +set vid_ref gl $*
```

- If you are going to compile a library like SVGAlib or Mesa for use with Quake, it must be compiled with libc5 and friends. Simply building a new library according its installation instructions will result in a library that is linked against your default library, glibc. You must ensure your new library is linked only with libc5 and libc5-based libraries so it will be compatible with Quake. See your distribution's documentation for information on linking to the non-default libraries.

My mouse doesn't work or seems to respond randomly.

- Is gpm running? gpm is a program that enables you to cut and paste with the mouse in virtual consoles. Many distributions enable it by default. It may be interfering with Quake. Check if it's running with the following command:

  ```
  ps aux | grep gpm
  ```

  If you get output like

  ```
  root      6115  0.0  0.4   832   316  ?  S   17:54   0:00 gpm -t PS/2
  ```

  then gpm is running and interfering with Quake. `gpm -k` (as root) ought to stop gpm. If it doesn't (`gpm -k` doesn't always work on my system), kill gpm with the command `killall gpm`. If you never use gpm, you may want to stop it from running at startup. See the documentation for your distribution for information on how to do this.
- Is your mouse defined properly in `libvga.config`? This file usually lives in `/etc` or `/etc/vga`. Open it up and look for a line like

  ```
  mouse Microsoft
  ```

  On my system, this is the first option in the file. Make sure the mouse type is appropriate for your hardware.

My Microsoft Intellimouse or Logitech MouseMan+ isn't working correctly.

SVGAlib, which handles mouse input for SVGA and GL Quake/QW/Q2, didn't directly support the Intellimouse until version 1.3.0. If you have a version of SVGAlib prior to 1.3.0, you should upgrade, then use mouse type `IntelliMouse` (for serial mice) or `IMPS2` (for PS/2 mice) in your `libvga.config` file.

My mouse is "laggy" and seems much slower than under Windows.

- For many people, just cranking up the value of `sensitivity` in the game console cures the problem. Setting `sensitivity` by hand in the console or in a `.cfg` file allows you to increase the mouse sensitivity more than the slider in the Options menu. `sensitivity 15`, for example.
- From Zoid's 1/7/98 .plan update:

  > If you are experience video 'lag' in the GL renderer (the frame rate feels like it's lagging behind your mouse movement) type "gl_finish 1" in the console. This forces update on a per frame basis.
- The latest version of SVGAlib (1.3.0) provides a slew of parameters in `libvga.config` that you can use to

customize the behavior of your mouse. With the proper settings it should be possible to make your mouse feel a way you want. On my system, just changing `mouse_accel_type` to `normal` (default is `power`) gave me t results I wanted. I haven't messed with the other settings, and I don't pretend to have a clue about what they all do.

- The little slider

I have a Voodoo2, and, when I try to run with the gl renderer, it reports that I don't have a Voodoo card installed.

There are different versions of Glide for Voodoo and Voodoo 2 cards. Be sure you downloaded the correct one for your system.

When I'm playing any of the Quake games under SVGAlib or GL and press CTRL-C, the game exits and sometimes leaves my console in an unusable state.

Sometimes when Quake/Quake II exits abnormally, it leaves my console unusable.

Yes. This bites. SVGAlib catches the CTRL-C and decides what to do with it instead of allowing Quake to handle it. I know of no way around this short of hacking SVGAlib.

If you run your Quake games from a script that resets the keyboard and terminal like the one below, you'll run less chance of ending up with a hosed terminal if this does happen, though.

```
#!/bin/sh
./quake2 $*
kbd_mode -a
reset
```

---

squake/quake2 fails to start and says "`svgalib: cannot get I/O permissions`"

The Quake executables must run as root, so you must either run them as root or make them setuid root. See the installation instructions in this document for details.

Sometimes after playing one of the Quake games in X, key repeat doesn't work any more.

For some reason, the X11 versions of Quake disable key repeat while they're running. If the program exits abnormally for some reason, key repeat never get turned back on. Do

```
xset r on
```

to reenable it.

Quake/Quake II says "/dev/dsp : device not configured"

Your sound hardware is not properly configured. You may simply need to do a `insmod sound`, or it may be necessary to rebuild your kernel. RedHat users may need to invoke the **sndconfig(8)** utility. See the documentation for your Linux distribution and/or the Linux Sound HOWTO for information on configuring your system's sound hardware.

GL Quake/Quake II run slower in Linux than in Windows. **

The Windows 3Dfx GL miniport is heavily optimized for the things Quake II does. Mesa on the other hand, is more general and less optimized As a result, Linux Quake II runs slower than under Windows. This isn't a limitation of Linux, but a limitation of the current drivers.

With the most recent releases of QuakeWorld and Quake II, the 3Dfx miniport mentioned above is available for Linux. While it still doesn't bring Linux Quake performance to par with Windows Quake, it's another step in that direction.

Additionally, for Pentium Pro and Pentium II users, there are some tweaks than can be done with memory buffering - the latest `/dev/3dfx` device driver has support for automatically setting this up for you. Enabling MTRRs can result in *significant* (10 fps on my system) GL Quake speedups. See http://glide.xxedgexx.com/MTRR.html for some more detailed information about this.

How can I start a server and log off, then come back to it later? **

screen(1) is a great utility for this sort of thing. It allows you to create many virtual screens in one tty and switch between them. Screen comes with most distributions. You can download it from ftp://prep.ai.mit.edu/pub/gnu or any GNU mirror.

Start screen by typing the command `screen`, then create a new screen window by pressing CTRL-A CTRL-C. You won't see much as you do these things, but be assured, something is happening.

Start a Quakeworld server:

```
/usr/local/games/quake/qwsv
```

Now open a new screen window with CTRL-A CTRL-C and start up a Quake II server:

```
/usr/local/games/quake2/quake2 +set dedicated 1
```

You can switch back and forth between your servers by pressing CTRL-A CTRL-N.

Press CTRL-A CTRL-D to detach from the screen program. Screen and your servers are still running, but they're no longer visible in your terminal window. You can logoff now and your processes will continue to run normally.

Use `screen -r` to re-attach to your previous screen process and access your servers again.

That's all there is to it. See the **screen(1)** man page for more detailed information.

5.2 Quake/QuakeWorld

Quake dies at startup with a segmentation fault.

This usually means your network setup isn't right. Try starting quake with the `-noudp` option and see if the error goes away. If that fixes it, check your `/etc/hosts` file and verify there's an entry for your machine in it. Use 127.0.0.1 for your IP address if you have a dialup account that gives you a different address each time you connect.

What's the difference between `glqwcl`, `glqwcl.3dfxgl`, and `glqwcl.glx`? ++

- **`glqwcl`** is the standard GL Quakeworld client you've seen in previous versions. It's linked against libMesaGL.so.2.
- **`glqwcl.3dfxgl`** is a script that runs `glqwcl` after preloading the 3Dfx GL miniport library, `lib3dfxgl.so`. Preloading the minport library causes its GL functions to get used instead of Mesa's. Since the GL minport is optimized for Quake, this is a good thing.
- **`glqwcl.glx`** is linked against standard OpenGL libriaries instead of Mesa. This will allow glquake to run on other 3D hardware that is supported by some other OpenGL implementation. This is an X application and so must be run from X.

When I run `glqwcl.glx` fullscreen from X, I can't use my mouse or keyboard. ++

Run `glqwcl.glx` with the `+_windowed_mouse 1` option. GLX Quakeworld is running in a window, even though it appears to take up your whole screen. If you move the mouse while the WM is in focus-follows-mouse mode, you're likely to move the pointer outside this window, and then Quake will stop responding to mouse and keyboard input. `+_windowed_mouse 1` makes Quakeworld grab the mouse exclusively.

5.3 Quake II

When I try to run Quake II with the GL renderer, it fails and says "LoadLibrary("ref_gl.so") failed: Unable to resolve symbol"

If immediately prior to the "Unable to resolve symbol" line, you have messages like "`can't resolve symbol 'fxMesaCreateContext'`", your Mesa library doesn't have glide support compiled in. See section The GL renderer in the Quake II installation section for information on installing Mesa and glide.

Quake II fails with the message `LoadLibrary("ref_XXX.so") failed: No such file or directory`

- `/etc/quake2.conf` doesn't have the correct path to your Quake II directory in it. This file should contain one line that is the directory Quake II lives in.
- If `/etc/quake2.conf` *does* contain the correct path, try removing the file and re-creating it by hand. Some versions of Quake II for Linux included an incorrectly formatted `quake2.conf` file.
- Do you have SVGAlib installed? Check `/lib`, `/usr/lib` and `/usr/local/lib` for a file called `libvga.so.1.X.X`, where the X's are some numbers. If nothing turns up, you need to get and install SVGAlib to run Quake II outside of X.
- If the renderer in question is `ref_gl.so`, Mesa may not be properly installed. Did you copy `libMesaGL.so.2.6` to a library directory like the installation instruactions told you to?
- If the renderer in question is `ref_gl.so`, did you install the glide libraries?

When I update the brightness while using the GL renderer, and hit "apply," nothing happens!

Type `vid_restart` in the console to make the changes take affect.

Note about the 3.17 distribution

As of this writing, the most recent Quake II version is 3.19. If for some reason, you're running version 3.17 instead, the following information may be helpful to you.

TOC

Two text files (`quake2.conf` and `fixperms.sh`) in the 3.17 distribution were inadvertently saved in MS-DOS C text-file format instead of the unix LF format. This means there's an extra carriage return character at the end of each these files and they're not going to behave right until you fix them.

We'll run them through **tr(1)** to strip out the CR's.

```
for i in fixperms.sh quake2.conf
do
mv $i $i.bak
tr -d '\r' < $i.bak > $i
done
```

When I run Quake II with `+set vid_ref glx` fullscreen from X, I can't use my mouse or keyboard. ++

Run GLX `quake2` with the `+set _windowed_mouse 1` option. GLX Quake2 is running in a window, even though it appears to take up your whole screen. If you move the mouse while the WM is in focus-follows-mouse mode, you're likely to move the pointer outside this window, and then Quake II will stop responding to mouse and keyboard input. `+set _windowed_mouse 1` makes Quake II grab the mouse exclusively.

Why can't I change to some of the SVGA modes that aree in the Quake II Video menu? **

SVGAlib probably doesn't know how to create the modes on your card. When Quake II starts up with the SVGA renderer (`ref_soft.so`), it prints a list of all the modes that SVGAlib tells it are available:

```
------- Loading ref_soft.so -------
Using RIVA 128 driver, 4096KB.
mode 320: 200 1075253220
mode 320: 240 1075253220
mode 320: 400 1075253220
mode 360: 480 1075253220
mode 640: 480 1075253220
mode 800: 600 1075253220
mode 1024: 768 1075253220
mode 1280: 1024 1075253220
```

These are the only modes you will be able to successfully switch to from the Video menu. If say, 512x384 isn't on the list, selecting it from the Video menu won't work.

SVGAlib does let you define new video modes for some chipsets in `libvga.config`, so you may be able create your own video mode this way. See the SVGAlib documentation for more detail on this topic.

---

6. Tips & Tricks

6.1 Running X and GL games without setuid

If you only run the X and GL versions of Quake, QuakeWorld or Quake II, you don't need to run them with root permissions. SVGA is the only mode that must be run as root. The X versions just need access to `/dev/dsp`, the sound device. The GL versions need access to the 3Dfx card as well as to `/dev/dsp`.

`/dev/dsp` needs to be readable and writeable by Quake. Most distributions give it 662 (`rw-rw--w-`) permissions by default. The simplest solution is to just `chmod 666 /dev/dsp`. On most systems, the ability to read from the sound device will not pose a significant security threat. If this approach is unacceptable for your system, create a group that owns `/dev/dsp` and make your Quake players members of that group.

You need the `/dev/3dfx` driver from Daryll Strauss' glide page (http://glide.xxedgexx.com/3DfxRPMS.html) in order run glide applications (like GLQuake) non-root. Download the `Device3Dfx.xxx.rpm` package and install according to the instructions on the web page. After you've installed the driver, make sure `/dev/3dfx` has 666 permissions (`chmod 666 /dev/3dfx`).

When `/dev/dsp` and `/dev/3dfx` are properly set up, you can remove the setuid bit from your Quake/QW/Q2 executables. Just do (as root) `chmod 0755 XXXXX`, where XXXXX is either `glquake`, `quake.x11`, or `quake2`.

If you've been playing as root prior to making these changes, many of your Quake files (like savegames) may be owned by root and inaccessible to a normal user, so remember to change the files' ownership before you attempt to play the game non-root.

6.2 Running SVGA and GL games from X **

The GLX Quakeworld and Quake II clients are native X applications, but since they use Mesa rather than the 3Dfx mini-driver, they're slower than the `lib3dfxgl.so` versions. For this reason, you may still favor this way of starting games from X over using the GLX clients.

This is based on a Linux Gazette 2 Cent Tip by Joey Hess (joey@kite.ml.org) The original is at http://www.ssc.com/lg/issue20/lg_tips20.html#squake

Yes, it's possible to run the Quake games from X if you're root, but such behavior is naughty, and you still run the risk of having Quake crash and leave the console unresponsive. With a little work you can make it possible for a regular user to run SVGA and GL Quake from X *AND* automatically switch back to X when the program is finished, regardless of whether it exited normally or not.

Note: when I say "Quake" in the text below, I really mean "quake, glquake squake, qwcl, glqwcl, qwcl.x11 or quake2".

- First, you'll need the **open(1)** package by Jon Tombs. This is a set of two very small programs that allow you to switch between virtual consoles and start programs on them. Download open from http://sunsite.unc.edu/pub/Linux/utils/console/. You shouldn't have to do much more than `make;make install` to compile and install it. Once it's installed, you need to make the `open` and `switchto` executables setuid root. So do this:

```
cd /usr/local/bin
```

```
chown root open switchto
chmod 4755 open switchto
```

- Next, save the following code to a file called `getvc.c`:

```
/* getvc.c
 * Prints the number of the current VC to stdout. Most of this code
 * was ripped from the open program, and this code is GPL'd
 *
 * Joey Hess, Fri Apr  4 14:58:50 EST 1997
 */

#include <sys/vt.h>
#include <fcntl.h>

main () {
  int fd = 0;
  struct vt_stat vt;

  if ((fd = open("/dev/console",O_WRONLY,0)) < 0) {
    perror("Failed to open /dev/console\n");
    return(2);
  }
  if (ioctl(fd, VT_GETSTATE, &vt) < 0) {
    perror("can't get VTstate\n");
    close(fd);
    return(4);
  }
  printf("%d\n",vt.v_active);
}

/* End of getvc.c */
```

Compile it and install it somewhere in your `$PATH`:

```
gcc getvc.c -o getvc
strip getvc
mv getvc /usr/local/bin
```

- Now create a script called `runvc`: your `$PATH`:

```
#!/bin/sh
# Run something on a VC, from X, and switch back to X when done.
# GPL Joey Hess, Thu, 10 Jul 1997 23:27:08 -0400
exec open -s -- sh -c "$* ; chvt `getvc`"
```

Make it executable and put it somewhere in your `$PATH`:

```
chmod 755 runvc
mv runvc /usr/local/bin
```

Now you can use the `runvc` command to start Quake. Continue to use whatever command line you usually use to start your game, but put `runvc` at the beginning:

```
runvc ./quake2 +set vid_ref gl +connect quake.foo.com
```

You'll automatically switch to a VC, run Quake and then switch back to X when it's done!

6.3 Keeping the mouse inside the window in X

From the Quake II `readme.linux` file:

> By default, the mouse will not be 'tied' to the Quake2 window. To cause Quake2 to grab the mouse, select 'Windowed Mouse' from the video menu, or type '_windowed_mouse 0' at the console. Do the reverse to release it. You can bind keys to grab and release the mouse in the console, like so: bind i "_windowed_mouse 1" bind o "_windowed_mouse 0" Then "i" will grab the mouse and "o" will release it.

6.4 3Dfx "tweak" settings work in Linux too

You may have seen or heard mention of ways to "tweak" Quake's OpenGL performance by setting various environment variables. These variables are equally valid under Linux. You just set them slightly differently. If a Windows/DOS "tweak" guide tells you to:

```
SET SST_GRXCLK=59
```

Under Linux, do it like this:

```
export SST_GRXCLK=59
```

6.5 The Poor Man's Server Browser

Joe S. (jszabo@eden.rutgers.edu) suggests:

```
a trick I do with qstat:
I make a file of my favorite servers, then do
qstat -f /C/quake2/file.txt | less
```

hit control z to suspend, then copy the ip address or hostname I want with gpm to a command line like:

```
./quake2 +connect expert.eqclans.com
```

Then I can use fg to go back to the list later...

TOC

6.6 Using `lib3dfxgl.so` for Quake I **

You can use the 3Dfx mini-driver (`lib3dfxgl.so`) from Quakeworld or Quake II with `glquake` too. Simply copy the `glqwcl.3dfxgl` or `quake2.3dfxgl` scripts to a new file, say `glquake.3dfxgl`. Then edit the `glquake.3dfxgl` script so it runs `glquake` instead of `glqwcl`. The same restrictions about running as root apply here, as do the differences between running under glibc and libc5. See the Quakeworld or Quake II sections for more information on `lib3dfxgl.so`.

---

7. Administrivia

7.1 New Versions of This Document

New versions of this document will be periodically posted to comp.os.linux.answers and rec.games.computer.quake.misc. They will also be uploaded to various WWW and FTP sites, including the LDP home page.

New versions of this document will be periodically posted to rec.games.computer.quake.misc and comp.os.linux.misc.

The latest version of the Linux Quake HOWTO can always be found at the following sites:

- http://www.linuxquake.com/
- http://webpages.mr.net/bobz/
- http://www.linuxgames.com/quake

7.2 Other Formats of This Document

This document is available in a variety of different formats. Far more than is necessary, actually. You can find them at http://www.linuxquake.com/howto/other-formats.

- ASCII text
- ASCII text with backspace-overstrikes , suitable for viewing with **man(1)**
- HTML all zipped up and ready to be dropped into your web site
- Postscript
- PRC for viewing on your Palm Pilot with Aportis Doc
- SGML
- LaTeX
- All of the above are also available in a single convenient package

7.3 Distribution Policy

Copyright (c) 1998, Bob Zimbinski, Brett A. Thomas and Mike Hallock. This document may be distributed under the terms set forth in the LDP license at sunsite.unc.edu/LDP/COPYRIGHT.html.

This HOWTO is free documentation; you can redistribute it and/or modify it under the terms of the LDP license. This document is distributed in the hope that it will be useful, but *without any warranty*; without even the implied warranty of *merchantability* or *fitness for a particular purpose*. Heck, even if this document *kills your dog*, it's not our fault.

See the LDP license for more details.

7.4 Revision History

Sections of this document that were updated in the last revision have a ** after the section heading. Sections updated in the revision prior to the last are marked with a ++.

Below is the full revision history of this document. It is not intended to be useful to anyone but the document authors.

```
$Log: Quake-HOWTO.sgml,v $
Revision 1.0.1.14  1998/10/05 19:54:17  bobz
Removed the long-dead LinQuake page from list of Linux Quake sites
Removed renaissoft.com link
Minor wording changes throughout
Added link to svgalib libc5 binary
Added info about q2 demo
Updated q2 installation info for the 3.19 release
moved section about 3.17 textfile problems to the troubleshooting section
Added part about lib3dfxgl.so to QW section
Added part about GLX client to QW section
Added part about lib3dfxgl.so to Q2 section
Added part about GLX client to Q2 section
Updated "slower than windows" FAQ
Removed obsolete TS tip about glibc & q2
Removed obsolete TS tip about playing w/CD mounted
Changed tip about running glquake with lib3dfxgl.so
Added instructions for Q2 mission pack 2
Added pointer to svgalib libc5 binary
Added FAQ about svgalib modes in Q2
Added Cheapo proxy to software section
Added screen FAQ
Added qgraph to  software sectoin
Decided I was lying when I promised a 3Dfx tweak section in the next update

Revision 1.0.1.13  1998/09/02 01:35:18  bobz
Fixed some broken url links
Restored <author> tag
added an acknowledgement
added a reminder to the non-root tip
added qlog listing

Revision 1.0.1.12  1998/08/30 21:35:23  bobz
Updated QuakeWorld install section for 2.30 release
Updated QW file list
Updated Quake 1.01 install info per email suggestion
Added QW 2.30 caveat to "Glibc considerations" section
Updated linux vs. windows speed section for new 3dfx miniport
Added glqwcl versions FAQ
Added troubleshooting tip about windows focus and glx
Added a sentence about /dev/3dfx permissions
Made runvc script usage clearer
Added tip about using mini-driver with Quake & QuakeII

Revision 1.0.1.11  1998/08/15 19:28:49  bobz
Added glibc topic in Troubleshooting/FAQ section
Moved non-setuid info to Tips & Tricks section
Moved rcs log into Revision Hist section
Changed distribution info a little until LDP wakes up
Changed references to quake.medina.net to www.linuxquake.com
Added linuxgames.com to list of places to get current version
Changed 3dfx Howto pointer
Moved explanation of change markings (** and ++) to intro section where they'll
```

be noticed
Mentioned zoid's GL driver port in the Win vs Linux FAQ
Fixed credit and URL in q2getty

Revision 1.0.1.10 1998/08/07 19:07:02 bobz
Changed XQF url
Added comments to the revision history section
Started marking changed sections
Added OS Considerations to troubleshooting section
Added rcon to Other Software section
Added ICE to Other Software section

Revision 1.0.1.9 1998/08/04 21:06:22 bobz
Updated Acknowledgements
Added discussion of Quake security
Added QIPX to "Related Software"
Added ts topic about -noudp

Revision 1.0.1.8 1998/08/03 22:09:28 bobz
Updated and restructured the q2 mods section
Minor change to gpm troubleshooting info
Changed sound ts info
Changed intellimouse ts info

Revision 1.0.1.7 1998/08/03 04:55:21 mikeh
Added "Playing" section of Q2 Mods & Addons section.
Added "Intelimouse" section of Troubleshooting section
Added RedHat sndconfig part to Troubleshooting section
Fixed a few spacing problems, and one error message

Revision 1.0.1.6 1998/08/03 02:01:13 bobz
Changed abstract.

Revision 1.0.1.5 1998/07/31 16:07:11 bobz
Added gpm -k mention in mouse troubleshooting
mentioned 'sensitivity' in mouse lag troubleshooting
Added qstat|less tip
Added "/dev/dsp not configured" to troubleshooting
Slightly reworded part of the mod compiling section
Added bit about running mods server side
corrected required version # in q2 mission pack 1
Updated acknowledgements section

Revision 1.0.1.4 1998/07/30 21:08:46 bobz
Fixed a problem with broken lines in <sect> tags

Revision 1.0.1.3 1998/07/30 17:40:29 bobz
Fixed broken link to Latex version.

Revision 1.0.1.2 1998/07/30 15:46:40 bobz
Test update.

Revision 1.0.1.1 1998/07/30 15:43:59 bobz
Changed version numbering scheme.

TOC

```
Revision 1.10  1998/07/30 13:51:24  bobz
Minor version reporting change

Revision 1.9  1998/07/29 20:42:03  bobz
Added some server links

Revision 1.8  1998/07/29 17:37:15  bobz
Added a Quake II model viewer to the Other Programs section

Revision 1.7  1998/07/29 16:02:23  bobz
Added some relative links in Other Formats to make it
more portable.

Revision 1.6  1998/07/29 15:18:12  bobz
Version and date stamps back to the way they were

Revision 1.5  1998/07/29 14:52:03  bobz
Added tags for automatic timestamping

Revision 1.4  1998/07/29 13:17:51  bobz
Changed captured.com's web address per webmaster's request

Revision 1.3  1998/07/29 02:11:02  bobz
Ooops again.  Made "Other formats" a <sect> instead of a <sect1>.

Revision 1.2  1998/07/29 01:55:41  bobz
Oops.  I deleted the doctype tag...

Revision 1.1  1998/07/29 01:48:34  bobz
Added Other Formats section.
Changed acknowledgments slightly
Added ftp.medina.net to list of ftp sites.

Revision 1.0 1998/07/28
First publicly released version

Revision 0.9 1998/07/25
Restructured,rewritten and expanded by Bob Zimbinski.

Revision 0.01 1998/06/16
First Pre-Release version v0.01by Brett A. Thomas and Mike Hallock.
```

RPM HOWTO (RPM at Idle)

Donnie Barnes, djb@redhat.com

v2.0, 8 April 1997

1. Introduction

2. Overview

3. General Information

- 3.1 Acquiring RPM
- 3.2 RPM Requirements

4. Using RPM

5. Now what can I *really* do with RPM?

6. Building RPMs

- 6.1 The rpmrc File
- 6.2 The Spec File
- 6.3 The Header
- 6.4 Prep
- 6.5 Build
- 6.6 Install
- 6.7 Optional pre and post Install/Uninstall Scripts
- 6.8 Files
- 6.9 Building It
- 6.10 Testing It
- 6.11 What to do with your new RPMs

- 6.12 What Now?

7. Multi-architectural RPM Building

- 7.1 Sample spec File
- 7.2 Optflags
- 7.3 Macros
- 7.4 Excluding Architectures from Packages
- 7.5 Finishing Up

8. Copyright Notice

---

1. Introduction

RPM is the **R**ed Hat **P**ackage **M**anager. While it does contain Red Hat in the name, it is completely intended to be an open packaging system available for anyone to use. It allows users to take source code for new software and package it into source and binary form such that binaries can be easily installed and tracked and source can be rebuilt easily. It also maintains a database of all packages and their files that can be used for verifying packages and querying for information about files and/or packages.

Red Hat Software encourages other distribution vendors to take the time to look at RPM and use it for their own distributions. RPM is quite flexible and easy to use, though it provides the base for a very extensive system. It is also completely open and available, though we would appreciate bug reports and fixes. Permission is granted to use and distribute RPM royalty free under the GPL.

More complete documentation is available on RPM in the book by Ed Bailey, *Maximum RPM*. That book is available for download or purchase at www.redhat.com.

2. Overview

First, let me state some of the philosophy behind RPM. One design goal was to allow the use of ``pristine'' sources. With RPP (our former packaging system of which *none* of RPM is derived), our source packages were the ``hacked'' sources that we built from. Theoretically, one could install a source RPP and then `make` it with no problems. But the sources were not the original ones, and there was no reference as to what changes we had to make to get it to build. One had to download the pristine sources separately. With RPM, you have the pristine sources along with a patch that we used to compile from. We see this as a big advantage. Why? Several reasons. For one, if a new version of a program comes out, you don't necessarily have to start from scratch to get it to compile under RHL. You can look at the patch to see what you *might* need to do. All the compile-in defaults are easily visible this way.

RPM is also designed to have powerful querying options. You can do searches through your entire database for packages or just certain files. You can also easily find out what package a file belongs to and where it came from. The RPM files themselves are compressed archives, but you can query individual packages easily and *quickly* because of a custom binary header added to the package with everything you could possibly need to know contained in uncompressed form. This allows for *fast* querying.

Another powerful feature is the ability to verify packages. If you are worried that you deleted an important file for some package, just verify it. You will be notified of any anomalies. At that point, you can reinstall the package if necessary. Any config files that you had are preserved as well.

We would like to thank the folks from the BOGUS distribution for many of their ideas and concepts that are included in RPM. While RPM was completely written by Red Hat Software, its operation is based on code written by BOGUS (PM and PMS).

3. General Information

3.1 Acquiring RPM

The best way to get RPM is to install Red Hat Linux. If you don't want to do that, you can still get and use RPM. It can be acquired from ftp.redhat.com.

3.2 RPM Requirements

The main requirement to run RPM is cpio 2.4.2 or greater. While this system is intended for use with Linux, it may very well be portable to other Unix systems. It has, in fact, been compiled on SunOS, Solaris, AIX, Irix, AmigaOS, and others. Be warned, the binary packages generated on a different type of Unix system will not be compatible.

Those are the minimal requirements to install RPMs. To build RPMs from source, you also need everything normally required to build a package, like `gcc`, `make`, etc.

4. Using RPM

In its simplest form, RPM can be used to install packages:

```
rpm -i foobar-1.0-1.i386.rpm
```

The next simplest command is to uninstall a package:

```
rpm -e foobar
```

One of the more complex but *highly* useful commands allows you to install packages via FTP. If you are connected to the net and want to install a new package, all you need to do is specify the file with a valid URL, like so:

```
rpm -i
ftp://ftp.pht.com/pub/linux/redhat/rh-2.0-beta/RPMS/foobar-1.0-1.i386.rpm
```

Please note, that RPM will now query and/or install via FTP.

While these are simple commands, rpm can be used in a multitude of ways as seen from the `Usage` message:

```
RPM version 2.3.9
Copyright (C) 1997 - Red Hat Software
This may be freely redistributed under the terms of the GNU Public License

usage: rpm {--help}
       rpm {--version}
       rpm {--initdb}   [--dbpath <dir>]
       rpm {--install -i} [-v] [--hash -h] [--percent] [--force] [--test]
                          [--replacepkgs] [--replacefiles] [--root <dir>]
                          [--excludedocs] [--includedocs] [--noscripts]
                          [--rcfile <file>] [--ignorearch] [--dbpath <dir>]
                          [--prefix <dir>] [--ignoreos] [--nodeps]
                          [--ftpproxy <host>] [--ftpport <port>]
                          file1.rpm ... fileN.rpm
       rpm {--upgrade -U} [-v] [--hash -h] [--percent] [--force] [--test]
                          [--oldpackage] [--root <dir>] [--noscripts]
                          [--excludedocs] [--includedocs] [--rcfile <file>]
                          [--ignorearch]  [--dbpath <dir>] [--prefix <dir>]
                          [--ftpproxy <host>] [--ftpport <port>]
                          [--ignoreos] [--nodeps] file1.rpm ... fileN.rpm
       rpm {--query -q} [-afpg] [-i] [-l] [-s] [-d] [-c] [-v] [-R]
                          [--scripts] [--root <dir>] [--rcfile <file>]
                          [--whatprovides] [--whatrequires] [--requires]
                          [--ftpuseport] [--ftpproxy <host>] [--ftpport <port>]
                          [--provides] [--dump] [--dbpath <dir>] [targets]
       rpm {--verify -V -y} [-afpg] [--root <dir>] [--rcfile <file>]
                          [--dbpath <dir>] [--nodeps] [--nofiles] [--noscripts]
                          [--nomd5] [targets]
       rpm {--setperms} [-afpg] [target]
       rpm {--setugids} [-afpg] [target]
```

```
rpm {--erase -e} [--root <dir>] [--noscripts] [--rcfile <file>]
                 [--dbpath <dir>] [--nodeps] [--allmatches]
                 package1 ... packageN
rpm {-b|t}[plciba] [-v] [--short-circuit] [--clean] [--rcfile  <file>]
                 [--sign] [--test] [--timecheck <s>] specfile
rpm {--rebuild} [--rcfile <file>] [-v] source1.rpm ... sourceN.rpm
rpm {--recompile} [--rcfile <file>] [-v] source1.rpm ... sourceN.rpm
rpm {--resign} [--rcfile <file>] package1 package2 ... packageN
rpm {--addsign} [--rcfile <file>] package1 package2 ... packageN
rpm {--checksig -K} [--nopgp] [--nomd5] [--rcfile <file>]
                    package1 ... packageN
rpm {--rebuilddb} [--rcfile <file>] [--dbpath <dir>]
rpm {--querytags}
```

You can find more details on what those options do in the RPM man page.

---

5. Now what can I *really* do with RPM?

RPM is a very useful tool and, as you can see, has several options. The best way to make sense of them is to look at some examples. I covered simple install/uninstall above, so here are some more examples:

- Let's say you delete some files by accident, but you aren't sure what you deleted. If you want to verify your entire system and see what might be missing, you would do:

  ```
  rpm -Va
  ```

- Let's say you run across a file that you don't recognize. To find out which package owns it, you would do:

  ```
  rpm -qf /usr/X11R6/bin/xjewel
  ```

  The output would be:

  ```
  xjewel-1.6-1
  ```

- You find a new koules RPM, but you don't know what it is. To find out some information on it, do:

  ```
  rpm -qpi koules-1.2-2.i386.rpm
  ```

  The output would be:

  ```
  Name        : koules                       Distribution: Red Hat Linux Colgate
  Version     : 1.2                                Vendor: Red Hat Software
  Release     : 2                                Build Date: Mon Sep 02 11:59:12 1996
  Install date: (none)                           Build Host: porky.redhat.com
  Group       : Games                            Source RPM: koules-1.2-2.src.rpm
  Size        : 614939
  Summary     : SVGAlib action game with multiplayer, network, and sound support
  Description :
  This arcade-style game is novel in conception and excellent in execution.
  No shooting, no blood, no guts, no gore.  The play is simple, but you
  still must develop skill to play.  This version uses SVGAlib to
  run on a graphics console.
  ```

- Now you want to see what files the koules RPM installs. You would do:

  ```
  rpm -qpl koules-1.2-2.i386.rpm
  ```

  The output is:

  ```
  /usr/doc/koules
  /usr/doc/koules/ANNOUNCE
  /usr/doc/koules/BUGS
  /usr/doc/koules/COMPILE.OS2
  /usr/doc/koules/COPYING
  /usr/doc/koules/Card
  /usr/doc/koules/ChangeLog
  /usr/doc/koules/INSTALLATION
  /usr/doc/koules/Icon.xpm
  /usr/doc/koules/Icon2.xpm
  /usr/doc/koules/Koules.FAQ
  /usr/doc/koules/Koules.xpm
  /usr/doc/koules/README
  /usr/doc/koules/TODO
  ```

```
/usr/games/koules
/usr/games/koules.svga
/usr/games/koules.tcl
/usr/man/man6/koules.svga.6
```

These are just several examples. More creative ones can be thought of really easy once you are familiar with RPM.

6. Building RPMs

Building RPMs is fairly easy to do, especially if you can get the software you are trying to package to build on its own.

The basic procedure to build an RPM is as follows:

- Make sure your `/etc/rpmrc` is setup for your system.
- Get the source code you are building the RPM for to build on your system.
- Make a patch of any changes you had to make to the sources to get them to build properly.
- Make a spec file for the package.
- Make sure everything is in its proper place.
- Build the package using RPM.

Under normal operation, RPM builds both binary and source packages.

6.1 The rpmrc File

Right now, the only configuration of RPM is available via the `/etc/rpmrc` file. An example one looks like:

```
require_vendor: 1
distribution: I roll my own!
require_distribution: 1
topdir: /usr/src/me
vendor: Mickiesoft
packager:  Mickeysoft Packaging Account <packages@mickiesoft.com>

optflags: i386 -O2 -m486 -fno-strength-reduce
optflags: alpha -O2
optflags: sparc -O2

signature: pgp
pgp_name: Mickeysoft Packaging Account
pgp_path: /home/packages/.pgp

tmppath: /usr/tmp
```

The `require_vendor` line causes RPM to require that it find a vendor line. This can come from the `/etc/rpmrc` or from the header of the spec file itself. To turn this off, change the number to `0`. The same holds true for the `require_distribution` and `require_group` lines.

The next line is the `distribution` line. You can define that here or later in the header of the spec file. When building for a particular distribution, it's a good idea to make sure this line is correct, even though it is not required. The `vendor` line works much the same way, but can be anything (ie. Joe's Software and Rock Music Emporium).

RPM also now has support for building packages on multiple architectures. The `rpmrc` file can hold an ``optflags'' variable for building things that require architecture specific flags when building. See later sections for how to use

TOC

this variable.

In addition to the above macros, there are several more. You can use:

```
rpm --showrc
```

to find out how your tags are set and what all the available flags are.

6.2 The Spec File

We'll begin with discussion of the spec file. Spec files are required to build a package. The spec file is a description of the software along with instructions on how to build it and a file list for all the binaries that get installed.

You'll want to name your spec file according to a standard convention. It should be the package name-dash-version number-dash-release number-dot-spec.

Here is a small spec file (vim-3.0-1.spec):

```
Summary: ejects ejectable media and controls auto ejection
Name: eject
Version: 1.4
Release: 3
Copyright: GPL
Group: Utilities/System
Source: sunsite.unc.edu:/pub/Linux/utils/disk-management/eject-1.4.tar.gz
Patch: eject-1.4-make.patch
Patch1: eject-1.4-jaz.patch
%description
This program allows the user to eject media that is autoejecting like
CD-ROMs, Jaz and Zip drives, and floppy drives on SPARC machines.

%prep
%setup
%patch -p1
%patch1 -p1

%build
make RPM_OPT_FLAGS="$RPM_OPT_FLAGS"

%install
install -s -m 755 -o 0 -g 0 eject /usr/bin/eject
install -m 644 -o 0 -g 0 eject.1 /usr/man/man1

%files
%doc README COPYING ChangeLog

/usr/bin/eject
/usr/man/man1/eject.1
```

6.3 The Header

TOC

The header has some standard fields in it that you need to fill in. There are a few caveats as well. The fields must be filled in as follows:

- `Summary:` This is a one line description of the package.
- `Name:` This must be the name string from the rpm filename you plan to use.
- `Version:` This must be the version string from the rpm filename you plan to use.
- `Release:` This is the release number for a package of the same version (ie. if we make a package and find it to be slightly broken and need to make it again, the next package would be release number 2).
- `Icon:` This is the name of the icon file for use by other high level installation tools (like Red Hat's ``glint''). It must be a gif and resides in the SOURCES directory.
- `Source:` This line points at the HOME location of the pristine source file. It is used if you ever want to get the source again or check for newer versions. Caveat: The filename in this line MUST match the filename you have on your own system (ie. don't download the source file and change its name). You can also specify more than one source file using lines like:

  ```
  Source0: blah-0.tar.gz
  Source1: blah-1.tar.gz
  Source2: fooblah.tar.gz
  ```

  These files would go in the `SOURCES` directory. (The directory structure is discussed in a later section, "The Source Directory Tree".)
- `Patch:` This is the place you can find the patch if you need to download it again. Caveat: The filename here must match the one you use when you make YOUR patch. You may also want to note that you can have multiple patch files much as you can have multiple sources.] You would have something like:

  ```
  Patch0: blah-0.patch
  Patch1: blah-1.patch
  Patch2: fooblah.patch
  ```

  These files would go in the `SOURCES` directory.
- `Copyright:` This line tells how a package is copyrighted. You should use something like GPL, BSD, MIT, public domain, distributable, or commercial.
- `BuildRoot:` This line allows you to specify a directory as the ``root'' for building and installing the new package. You can use this to help test your package before having it installed on your machine.
- `Group:` This line is used to tell high level installation programs (such as Red Hat's ``glint'') where to place this particular program in its hierarchical structure. The group tree currently looks something like this:

  ```
  Applications
      Communications
      Editors
          Emacs
      Engineering
      Spreadsheets
      Databases
      Graphics
      Networking
  ```

```
            Mail
            Math
            News
            Publishing
                TeX
        Base
            Kernel
        Utilities
            Archiving
            Console
            File
            System
            Terminal
            Text
        Daemons
        Documentation
        X11
            XFree86
                Servers
            Applications
                Graphics
                Networking
            Games
                Strategy
                Video
            Amusements
            Utilities
            Libraries
            Window Managers
        Libraries
        Networking
            Admin
            Daemons
            News
            Utilities
        Development
            Debuggers
            Libraries
                Libc
            Languages
                Fortran
                Tcl
            Building
            Version Control
            Tools
        Shells
        Games
```

- `%description` It's not really a header item, but should be described with the rest of the header. You need one description tag per package and/or subpackage. This is a multi-line field that should be used to give a comprehensive description of the package.

TOC

6.4 Prep

This is the second section in the spec file. It is used to get the sources ready to build. Here you need to do anything necessary to get the sources patched and setup like they need to be setup to do a `make`.

One thing to note: Each of these sections is really just a place to execute shell scripts. You could simply make an `sh` script and put it after the `%prep` tag to unpack and patch your sources. We have made macros to aid in this, however.

The first of these macros is the `%setup` macro. In its simplest form (no command line options), it simply unpacks the sources and `cd`'s into the source directory. It also takes the following options:

- `-n name` will set the name of the build directory to the listed `name`. The default is `$NAME-$VERSION`. Other possibilities include `$NAME`, `${NAME}${VERSION}`, or whatever the main tar file uses. (Please note that these ``$" variables are *not* real variables available within the spec file. They are really just used here in place of a sample name. You need to use the real name and version in your package, not a variable.)
- `-c` will create and cd to the named directory *before* doing the untar.
- `-b` # will untar Source# *before* cd'ing into the directory (and this makes no sense with `-c` so don't do it). This is only useful with multiple source files.
- `-a` # will untar Source# *after* cd'ing into the directory.
- `-T` This option overrides the default action of untarring the Source and requires a `-b 0` or `-a 0` to get the main source file untarred. You need this when there are secondary sources.
- `-D` Do *not* delete the directory before unpacking. This is only useful where you have more than one setup macro. It should *only* be used in setup macros *after* the first one (but never in the first one).

The next of the available macros is the `%patch` macro. This macro helps automate the process of applying patches to the sources. It takes several options, listed below:

- # will apply Patch# as the patch file.
- `-p` # specifies the number of directories to strip for the patch(1) command.
- `-P` The default action is to apply `Patch` (or `Patch0`). This flag inhibits the default action and will require a `0` to get the main source file untarred. This option is useful in a second (or later) `%patch` macro that required a different number than the first macro.
- You can also do `%patch`# instead of doing the real command: `%patch # -P`

That should be all the macros you need. After you have those right, you can also do any other setup you need to do via `sh` type scripting. Anything you include up until the `%build` macro (discussed in the next section) is executed via `sh`. Look at the example above for the types of things you might want to do here.

6.5 Build

There aren't really any macros for this section. You should just put any commands here that you would need to use to build the software once you had untarred the source, patched it, and cd'ed into the directory. This is just another set of commands passed to `sh`, so any legal `sh` commands can go here (including comments). **Your current working directory is reset in each of these sections to the toplevel of the source directory**, so keep that in mind. You can `cd` into subdirectories if necessary.

TOC

6.6 Install

There aren't really any macros here, either. You basically just want to put whatever commands here that are necessary to install. If you have `make install` available to you in the package you are building, put that here. If not, you can either patch the makefile for a `make install` and just do a `make install` here, or you can hand install them here with `sh` commands. You can consider your current directory to be the toplevel of the source directory.

6.7 Optional pre and post Install/Uninstall Scripts

You can put scripts in that get run before and after the installation and uninstallation of binary packages. A main reason for this is to do things like run `ldconfig` after installing or removing packages that contain shared libraries. The macros for each of the scripts is as follows:

- `%pre` is the macro to do pre-install scripts.
- `%post` is the macro to do post-install scripts.
- `%preun` is the macro to do pre-uninstall scripts.
- `%postun` is the macro to do post-uninstall scripts.

The contents of these sections should just be any `sh` style script, though you do *not* need the `#!/bin/sh`.

6.8 Files

This is the section where you *must* list the files for the binary package. RPM has no way to know what binaries get installed as a result of `make install`. There is *NO* way to do this. Some have suggested doing a `find` before and after the package install. With a multiuser system, this is unacceptable as other files may be created during a package building process that have nothing to do with the package itself.

There are some macros available to do some special things as well. They are listed and described here:

- `%doc` is used to mark documentation in the source package that you want installed in a binary install. The documents will be installed in `/usr/doc/$NAME-$VERSION-$RELEASE`. You can list multiple documents on the command line with this macro, or you can list them all separately using a macro for each of them.
- `%config` is used to mark configuration files in a package. This includes files like sendmail.cf, passwd, etc. If you later uninstall a package containing config files, any unchanged files will be removed and any changed files will get moved to their old name with a `.rpmsave` appended to the filename. You can list multiple files with this macro as well.
- `%dir` marks a single directory in a file list to be included as being owned by a package. By default, if you list a directory name *WITHOUT* a `%dir` macro, *EVERYTHING* in that directory is included in the file list and later installed as part of that package.
- `%files -f <filename>` will allow you to list your files in some arbitrary file within the build directory of the sources. This is nice in cases where you have a package that can build it's own filelist. You then just include that filelist here and you don't have to specifically list the files.

The biggest caveat in the file list is listing directories. If you list `/usr/bin` by accident, your binary package will contain *every* file in `/usr/bin` on your system.

6.9 Building It

The Source Directory Tree

The first thing you need is a properly configured build tree. This is configurable using the `/etc/rpmrc` file. Most people will just use `/usr/src`.

You may need to create the following directories to make a build tree:

- `BUILD` is the directory where all building occurs by RPM. You don't have to do your test building anywhere in particular, but this is where RPM will do it's building.
- `SOURCES` is the directory where you should put your original source tar files and your patches. This is where RPM will look by default.
- `SPECS` is the directory where all spec files should go.
- `RPMS` is where RPM will put all binary RPMs when built.
- `SRPMS` is where all source RPMs will be put.

Test Building

The first thing you'll probably want to to is get the source to build cleanly without using RPM. To do this, unpack the sources, and change the directory name to $NAME.orig. Then unpack the source again. Use this source to build from. Go into the source directory and follow the instructions to build it. If you have to edit things, you'll need a patch. Once you get it to build, clean the source directory. Make sure and remove any files that get made from a `configure` script. Then `cd` back out of the source directory to its parent. Then you'll do something like:

```
diff -uNr dirname.orig dirname > ../SOURCES/dirname-linux.patch
```

This will create a patch for you that you can use in your spec file. Note that the ``linux'' that you see in the patch name is just an identifier. You might want to use something more descriptive like ``config'' or ``bugs'' to describe *why* you had to make a patch. It's also a good idea to look at the patch file you are creating before using it to make sure no binaries were included by accident.

Generating the File List

Now that you have source that will build and you know how to do it, build it and install it. Look at the output of the install sequence and build your file list from that to use in the spec file. We usually build the spec file in parallel with all of these steps. You can create the initial one and fill in the easy parts, and then fill in the other steps as you go.

Building the Package with RPM

Once you have a spec file, you are ready to try and build your package. The most useful way to do it is with a command like the following:

```
rpm -ba foobar-1.0.spec
```

There are other options useful with the `-b` switch as well:

- `p` means just run the `prep` section of the specfile.
- `l` is a list check that does some checks on `%files`.

- `c` do a prep and compile. This is useful when you are unsure of whether your source will build at all. It seems useless because you might want to just keep playing with the source itself until it builds and then using RPM, but once you become accustomed to using RPM you will find instances when you will use it.
- `i` do a prep, compile, and install.
- `b` prep, compile, install, and build a binary package only.
- `a` build it all (both source and binary packages).

There are several modifiers to the `-b` switch. They are as follows:

- `--short-circuit` will skip straight to a specified stage (can only be used with c and i).
- `--clean` removes the build tree when done.
- `--keep-temps` will keep all the temp files and scripts that were made in /tmp. You can actually see what files were created in /tmp using the `-v` option.
- `--test` does not execute any real stages, but does keep-temp.

6.10 Testing It

Once you have a source and binary rpm for your package, you need to test it. The easiest and best way is to use a totally different machine from the one you are building on to test. After all, you've just done a lot of `make install`'s on your own machine, so it should be installed fairly well.

You can do an `rpm -u packagename` on the package to test, but that can be deceiving because in building the package, you did a `make install`. If you left something out of your file list, it will not get uninstalled. You'll then reinstall the binary package and your system will be complete again, but your rpm still isn't. Make sure and keep in mind that just because you do a `rpm -ba package`, most people installing your package will just be doing the `rpm -i package`. Make sure you don't do anything in the `build` or `install` sections that will need to be done when the binaries are installed by themselves.

6.11 What to do with your new RPMs

Once you've made your own RPM of something (assuming its something that hasn't already been RPM'ed), you can contribute your work to others (also assuming you RPM'ed something freely distributable). To do so, you'll want to upload it to ftp.redhat.com.

6.12 What Now?

Please see the above sections on Testing and What to do with new RPMs. We want all the RPMs available we can get, and we want them to be good RPMs. Please take the time to test them well, and then take the time to upload them for everyone's benefit. Also, *please* make sure you are only uploading *freely available software*. Commercial software and shareware should *not* be uploaded unless they have a copyright expressly stating that this is allowed. This includes Netscape software, ssh, pgp, etc.

---

7. Multi-architectural RPM Building

RPM can now be used to build packages for the Intel i386, the Digital Alpha running Linux, and the Sparc. It has been reported to work on SGI's and HP workstations as well. There are several features that make building packages on all platforms easy. The first of these is the ``optflags'' directive in the `/etc/rpmrc`. It can be used to set flags used when building software to architecture specific values. Another feature is the ``arch'' macros in the spec file. They can be used to do different things depending on the architecture you are building on. Another feature is the ``Exclude'' directive in the header.

7.1 Sample spec File

The following is part of the spec file for the ``fileutils'' package. It is setup to build on both the Alpha and the Intel.

```
Summary: GNU File Utilities
Name: fileutils
Version: 3.16
Release: 1
Copyright: GPL
Group: Utilities/File
Source0: prep.ai.mit.edu:/pub/gnu/fileutils-3.16.tar.gz
Source1: DIR_COLORS
Patch: fileutils-3.16-mktime.patch

%description
These are the GNU file management utilities.  It includes programs
to copy, move, list, etc, files.

The ls program in this package now incorporates color ls!

%prep
%setup

%ifarch alpha
%patch -p1
autoconf
%endif
%build
configure --prefix=/usr --exec-prefix=/
make CFLAGS="$RPM_OPT_FLAGS" LDFLAGS=-s
```

```
%install
rm -f /usr/info/fileutils*
make install
gzip -9nf /usr/info/fileutils*

.
.
.
```

7.2 Optflags

In this example, you see how the ``optflags'' directive is used from the `/etc/rpmrc`. Depending on which architecture you are building on, the proper value is given to `RPM_OPT_FLAGS`. You must patch the Makefile for your package to use this variable in place of the normal directives you might use (like `-m486` and `-O2`). You can get a better feel for what needs to be done by installing this source package and then unpacking the source and examine the Makefile. Then look at the patch for the Makefile and see what changes must be made.

7.3 Macros

The `%ifarch` macro is very important to all of this. Most times you will need to make a patch or two that is specific to one architecture only. In this case, RPM will allow you to apply that patch to just one architecture only.

In the above example, fileutils has a patch for 64 bit machines. Obviously, this should only be applied on the Alpha at the moment. So, we add an `%ifarch` macro around the 64 bit patch like so:

```
%ifarch axp
%patch1 -p1
%endif
```

This will insure that the patch is not applied on any architecture except the alpha.

7.4 Excluding Architectures from Packages

So that you can maintain source RPMs in one directory for all platforms, we have implemented the ability to ``exclude'' packages from being built on certain architectures. This is so you can still do things like

```
rpm --rebuild /usr/src/SRPMS/*.rpm
```

and have the right packages build. If you haven't yet ported an application to a certain platform, all you have to do is add a line like:

```
ExcludeArch: axp
```

to the header of the spec file of the source package. Then rebuild the package on the platform that it can build on. You'll then have a source package that builds on an Intel and can easily be skipped on an Alpha.

7.5 Finishing Up

Using RPM to make multi-architectural packages is usually easier to do than getting the package itself to build both places. As more of the hard packages get built this is getting much easier, however. As always, the best help when you get stuck building an RPM is to look a similar source package.

---

8. Copyright Notice

This document and its contents are copyright protected. Redistribution of this document is permitted as long as the content remains completely intact and unchanged. In other words, you may reformat and reprint or redistribute only.

The Linux Reading List HOWTO

by Eric S. Raymond

v1.6, 20 April 1999

This document lists the book I think are most valuable to a person trying to learn Unix (especially Linux) top to bottom.

1. Introduction

- 1.1 Purpose of this document
- 1.2 New versions of this document
- 1.3 Feedback and Corrections
- 1.4 Related Resources
- 1.5 Conventions Used In This Document

2. Books on Culture, History, and Pragmatics

3. Books on General Unix/Linux

- 3.1 Linux Installation and Administration
- 3.2 Using Unix & Linux
- 3.3 System Security

4. Books on Shell, Script, and Web Programming

5. Books on Text Formatting

- 5.1 Tex and LaTeX

TOC

6. Books on C and C++ Programming

- 6.1 C and C++
- 6.2 C System Call Interface

7. Books on Networking

8. Books on Unix Kernel Implementation

- 8.1 Ancestors of Linux
- 8.2 Linux
- 8.3 Relatives of Linux

9. Books on Intel processor architecture and programming

10. Books on PC-Class Hardware

11. Administrivia

- 11.1 Terms of Use
- 11.2 History

---

1. Introduction

1.1 Purpose of this document

This document lists what I consider to be the essential book-length references for learning Unix (especially Linux) and how to program under it.

1.2 New versions of this document

New versions of the Linux Reading List HOWTO will be periodically posted to comp.os.linux.answers. They will also be uploaded to various Linux WWW and FTP sites, including the LDP home page.

You can also view the latest version of this on the World Wide Web via the URL http://sunsite.unc.edu/LDP/HOWTO/Reading-List-HOWTO.html.

1.3 Feedback and Corrections

If you have questions or comments about this document (or just want to suggest a book that you think should be on it), please feel free to mail Eric S. Raymond, at esr@thyrsus.com. I welcome any suggestions or criticisms.

1.4 Related Resources

For on-line HOWTOs, magazines, and other non-book material, see the Linux Documentation Project home page.

Some years ago I wrote a less Linux-focused Unix bibliography that may still be of some interest and retains a certain amusement value. You can find the Loginataka at http://www.tuxedo.org/~esr/faqs/loginataka.html.

There's a collection of Web links to Linux book reviews called Opening Doors, Breaking Windows.

1.5 Conventions Used In This Document

Comments not in quotes below are either mine, or I have seen no reason to change them from those of Jim Haynes (previous maintainer of this document). Comments sent in by others are in quotes, and have the name of the commentator before them (JH is Jim Haynes).

"See" URLs attached to publishing information point directly into the publisher's web catalog and

typically take you to a page containing a cover shot, blurbs, and ordering information. Books that do have these lack them because the publisher is using frames and the catalog pages can't be bookmarked.

Topic listings go roughly from the outside in (culture to user-land programming to kernel programming to hardware). Within sections I have tried to list the most useful books first insofar as I am familiar with them. It's just an embarrassing coincidence that this lists one of my books first, honest! (Suggestions for a better organization cheerfully accepted.)

---

2. Books on Culture, History, and Pragmatics

The New Hacker's Dictionary (Third Edition)

Raymond, Eric S.; MIT Press; 1996; ISBN 0-262-68092-0; 547pp.
See http://www-mitpress.mit.edu/book-home.tcl?isbn=0262680920.

Um, er. A guide to Internet culture. Lots of people like it. HTML at the Jargon File Resource Page.

A Quarter Century of Unix

Salus, Peter H.; Addison-Wesley; 1994; ISBN 0-201-54777-5; 256pp.
See http://www.awl.com/cp/authors/salus/unix/unix.html

Linux is part of the Unix tradition. This book is an oral history of Unix -- how it originated, how it evolved, how it spread -- by the people who were there.

The Mythical Man Month (Anniversary Edition)

Brooks, Frederick P.; Addison-Wesley 1995 (ISBN 0-201-83595-9).
See http://heg-school.awl.com/cseng/authors/brooks/mmm-ae/mmm-ae.html.

The one book on software engineering that everyone should read.

Alan Cox: "This I'd recommend not for its technical value but for its application of common sense and reality to computing projects." JH: "Ah, yes. What if Linus had been given 200 programmers and had been told to produce Linux in 3 months!"

Bell System Technical Journal, July-August 1978, Vol. 57, No. 6, part 2

AT&T; 416 pp.

Many early papers on Unix, including Ritchie & Thompson, "The UNIX Time Sharing System"; Thompson, "UNIX Implementation"; Ritchie, "A Retrospective"; Bourne, "The UNIX Shell"...

3. Books on General Unix/Linux

3.1 Linux Installation and Administration

Linux Installation and Getting Started

Welsh, Matt; LDP; 1997. Available on the LDP home page, or directly at http://sunsite.unc.edu/LDP/gs.

How to bring up Linux. Explains a lot of Linux basics. Covers basic system administration.

Linux System Administrator's Guide

Wirzenius, Lars; LDP; 1997. Available on the LDP home page, or directly at http://sunsite.unc.edu/LDP/sag.

An excellent first book on how to maintain and administer a Linux system.

Essential System Administration (Second Edition)

Frisch, Aeleen; O'Reilly; 1995; ISBN 0-937175-80-3; 788 pp; $32.95.
See http://www.ora.com/catalog/esa2/noframes.html.

More in-depth coverage of normal system-administration tasks. Not Linux-specific but contains Linux material.

3.2 Using Unix & Linux

Linux in a Nutshell

Hekman, Jessica P. et al.; O'Reilly; ISBN 1-56592-167-4; 1997; 438 pp. $9.95.
See http://www.ora.com/catalog/linuxnut/noframes.html.

According to O'Reilly, "The Desktop Reference for Linux". For Linux users this obsoletes their "Unix In a Nutshell" which was SVr4/Solaris-oriented.

Running Linux (Second Edition)

Welsh, Matt, & Kaufman, Lar; O'Reilly; ISBN 1-56592-151-8; 1996; 650pp; $24.95.
See http://www.ora.com/catalog/runux2/noframes.html.

Everything you need in order to understand, install, and use the Linux operating system. Excellent beginner's book.

Hands-on-Linux

Sobell, Mark G.; Addison-Wesley; ISBN ISBN 0-201-32569-1; 1998; 1015 pp.

Just what the title says -- practical tutorials in basic Unix, shells, editors, mail programs, networking, Web tools, and utilities. Covers some system administration fundamentals. (This appears to be a repackaging of 1997's ``A Practical Guide to Linux'' from the same author, without Caldera OpenLinux Lite included.)

3.3 System Security

Practical Unix Security

Garfinkel, Simpson, and Spafford, Gene; O'Reilly Associates; ISBN 0-56592-148-8; 1991. See http://www.ora.com/catalog/puis/noframes.html.

Ronald P. Miller: "Some overlap with Essential System Admin., but all in all a solid book on security, especially for those aspiring to allow multiple-user, dial-up/net access to their Linux boxes."

Firewalls & Internet Security

Cheswick, William R. & Bellovin, Steven M.; Addison-Wesley; 1994; ISBN 0-201-63357-4; 320pp.

---

4. Books on Shell, Script, and Web Programming

Programming Perl (Second Edition)

Wall, Larry & Christiansen, Tom & Schwartz, Randal; O'Reilly; 1997; ISBN 0-56592-149-6; 644pp.
See http://www.ora.com/catalog/pperl2/noframes.html.

Shell (as a programming language for more than trivial scripting) is dead. Perl rules in its place. This is the second edition of the definitive Perl book -- vastly better organized than the first, and it covers Perl 5.

Emmanuel Pierre keeps a short list of Perl books.

Programming Python

Lutz, Mark; O'Reilly; 1997; ISBN 0-56592-197-6; 880pp.
See http://www.ora.com/catalog/python/noframes.html.

The next step beyond Perl. Python is beautifully designed, has better integration with C, and scales up better to large projects.

HTML: The Definitive Guide (2nd Edition)

Musciano, Chuck & Kennedy. Bill; O'Reilly; 1997; ISBN 0-56592-235-2; 552pp.
See http://www.ora.com/catalog/html2/noframes.html.

The best HTML tutorial/reference I have ever seen, and the only HTML book you need unless you want to do CGI.

The Unix Programming Environment

Kernighan, Brian, and Pike, Rob; Prentice-Hall; 1984; ISBN 0-13-937681-X; 1984.

A true classic -- possibly the best single-book exposition of the Unix philosophy.

5. Books on Text Formatting

5.1 Tex and LaTeX

The LaTeX Companion

Goossens, Michael & Mittlebach, Frank, & Samarin, Alexander; Addison-Wesley; 1994; ISBN 0-201-54199-8; 530pp.
See http://www.awl.com/cp/tlc.html.

`If you are one of those users who would like to know how LaTeX can be extended to create the nicest documents possible without becoming a (La)TeX guru, then this book is for you' --- from the Preface. Bruce Thompson adds: "A very nice book providing a lot of information about the new extensions to LaTeX, provides a large number of examples showing precisely how your document's layout can be manipulated"

LaTeX: A Document Preparation System (Second Edition)

Lamport, Leslie; Addison-Wesley; 1994; ISBN 0-201-52983-1; 256pp.
See http://heg-school.awl.com/cseng/authors/lamport/latex/latex.html.

Bruce Thompson: "The ultimate reference on LaTeX 2.09 by its author. A new edition covering LaTeX2e (the version included in the current TeX/LaTeX distribution) is in preparation. LaTeX 2.09 is fully supported by LaTeX2e. A must for anyone wanting to use LaTeX. Provides a gentle introduction to document preparation and the various tools that LaTeX provides for producing professional quality documents. Lots of examples."

The TeXbook, Volume A of Computers and Typesetting; Knuth, Donald A.

Addison-Wesley; 1986, ISBN 0-201-13448; 496pp. See http://www.awl.com/cp/TeXbook.html

Bruce Thompson: "The definitive user's guide and complete reference manual for TeX. Probably not needed for casual LaTeX use, but a fascinating book nonetheless." I'll strengthen that by adding that this book is not for the faint of heart.

The METAFONT book, Volume C of Computers and Typesetting

Knuth, Donald A.; Addison-Wesley; 1986; 0-201-13444-6, 1986; 384pp.
See http://www.awl.com/cp/METAFONTbook.html

Bruce Thompson: "The definitive user's guide and reference manual for METAFONT, the companion program to TeX for designing fonts. An excellent work if you're planning to design your own fonts for use in TeX and LaTeX. METAFONT is included with the normal TeX/LaTeX distribution." This book is *definitely* not for the faint of heart.

6. Books on C and C++ Programming

6.1 C and C++

The C Programming Language (Second Edition)

Kernighan, Brian W.; Ritchie, Dennis M; Prentice-Hall; 1988; ISBN 0-13-110362-8, 272pp.

The improved second edition, covering ANSI C, of the original classic C book coauthored by C's designer, "K&R". Still the best!

Who's Afraid of C++?

Heller, Steve; Academic Press; 1996; ISBN 0-12-339097; 508pp.

The best introductory book on C++ I have seen.

6.2 C System Call Interface

POSIX Programmer's Guide: Writing Portable Unix Programs

Lewine, Donald; O'Reilly; 1992; ISBN 0-937175-73-0; 607pp.
See http://www.ora.com/catalog/posix/noframes.html.

An excellent programmer's reference on the POSIX.1 standard. I like this one better than JH's choice.

The Posix.1 Standard: A Programmer's Guide

Zlotnick, Fred; Benjamin, Cummings; 1991; ISBN 0-8053-9605-5; 379pp.; $35.95 (USA).

JH: "When I complained about the lack of Section 2 man pages in Linux, somebody told me just to get a POSIX book, because that's what Linux does. I like this book because I'm not a professional programmer and the author gives copious explanations and examples."

Advanced Programming in The Unix Environment

Stevens, Richard; 1992; ISBN 0-201-56317; Addison-Wesley

A book on general Unix programming that is every bit as good as Stevens's classic on network programming.

Linux Application Development

Michael K. Johnson, Erik W. Troan; 1998; ISBN 0201308215; Addison-Wesley.

The best reference to the C API of Linux.

7. Books on Networking

Unix Network Programming

Stevens, W. Richard; Prentice Hall; 1990; ISBN 0-13-949876-1; 772 pp.; $54 (USA).

Everything you might want to know about the subject, and some things you probably didn't want to know (really, XNS!?). Generally regarded as definitive on the basics. Two more volumes are planned.

Linux Network Administrator's Guide

Kirch, Olaf; O'Reilly; 1995; ISBN 1-56592-087-2; 335pp.
See http://www.ora.com/catalog/linag/noframes.html.

A practical guide to Linux's TCP/IP and related services. Accessible on the Web at the Linux Documentation Project page, or directly at http://sunsite.unc.edu/LDP/LDP/nag/nag.html.

TCP/IP Network Adminstration

Hunt, Craig; O'Reilly Associates, ISBN 0-937175-82-X; 1992; 472pp.
See http://www.ora.com/catalog/tcp2/noframes.html.

Less Linux-specific than the Kirch book. Features deeper coverage of the TCP/IP core, including routing and BGP.

DNS and BIND (Second Edition)

Albitz, Paul, and Liu, Cricket; O'Reilly; 1996; ISBN 1-56592-236-0; 1992; 438pp; $32.95.
See http://www.ora.com/catalog/dns2/noframes.html.

In-depth coverage of DNS, useful for people running complicated multiple-subnet installations. Covers BIND library programming.

Sendmail (Second Edition)

Costales, Bryan & Allman, Eric; O'Reilly; ISBN 1-56592-222-0; 1997; 1050 pp; $32.95
See http://www.ora.com/catalog/sendmail2/noframes.html.

An exhaustive (and exhausting) guide to Linux's and Unix's default mail-transfer agent.

8. Books on Unix Kernel Implementation

8.1 Ancestors of Linux

The Design of the Unix Operating System

Bach, Maurice J.; Prentice-Hall; ISBN 0-13-201799-7; 470pp.; $60 (USA).

The book that got Linus started.

Operating Systems, Design and Implementation;

Tanenbaum, Andrew S.; Prentice-Hall; 1987.

Alan Cox (one of the core kernel people) likes this book. Tanenbaum designed Minix, which is the system Linus bootstrapped Linux up from.

8.2 Linux

The Linux Kernel book

Rémy Card, Èric Dumas, Franck Mével; John Wiley and Sons; 1998; ISBN 0-471-98141-9. $100 (AUS).

(Translated from the French language edition of "Programmation Linux 2.0"; same authors; 1997; Éditions Eyrolles; Paris, France.)

A very interesting and informative description of the operation of the kernel that fills in the gap between the POSIX interface and "The Design of the Unix Operating System" and the Linux source code. A good understanding of the design and operation of a Unix OS is a pre-requisite, but this book is an excellent help to going beyond that general understanding into actual work.

The primary author is one of the core developers for the ext2 filesystem, and the Linux Kernel book shows a firm grasp of the matter and clear explanations and structure. It's surprisingly readable for something working at such a low level. The book does seem to have suffered a little in the translation to English -- there are a few typos and grammatical mistakes, but it's quite readable. (The code example files are charmingly still named in French.)

Network protocol implementations are not covered.

The book's current to Linux 2.0.35 and foreshadows 2.1 and 2.2.

Linux Kernel Hackers' Guide

Johnson, Michael K.

Accessible on the Web at the Linux Documentation Project page, or directly at http://www.redhat.com:8080/HyperNews/get/khg.html.

LINUX Kernel Internals (Second Edition)

Beck, Michael & Bohme, Harold & Mirko, Dziadzka & Kunitz, Ulrich & Magnus, Robert & Verworner, Dick; Addison Wesley; 1998; ISBN:0-201-33143-8; 480.
See http://heg-school.awl.com/cseng/authors/beck.m/linux/linux.html.

A guide to Linux kernel programming; covers 2.0.

8.3 Relatives of Linux

The Design and Implementation of the 4.4BSD Unix Operating System

McKusick, Marshall Kirk, Bostic, Keith, Karels, Michael J., and Quarterman, John S.; Addison-Wesley; 1996; ISBN 0-201-54979-4; 608pp.
See http://heg-school.awl.com/cseng/authors/mckusick/4.4bsd/4.4bsd.html.

The successor to a classic book on the implementation of the 4.3 BSD kernel, which influenced Linux's design (especially near sockets and networking). This book covers the 4.4BSD base of BSD/OS, FreeBSD, and NetBSD.

Porting Unix to the 386; Jolitz, William F., and Jolitz, Lynne G.

Dr. Dobb's Journal; Jan 1991-July 1992.

---

9. Books on Intel processor architecture and programming

80386 Programmer's Reference Manual

Intel Corp.; ISBN 1-55512-022-9; 1986;

Part I. Applications Programming, data types, memory model, instruction set. Part II. Systems Programming, architecture, memory management, protection, multitasking, I/O, exceptions and interrupts, initialization, coprocessing and multiprocessing. Part III. Compatibility (with earlier x86 machines). Part IV. Instruction Set.

80386 System Software Writer's Guide

Intel Corp.; ISBN 1-55512-023-7; 1987.

This explains the 386 features for operating system writers. It includes a chapter on Unix implementation. A lot of the 80386 architecture seems to have been designed with Multics in mind; the features are not used by DOS or by Unix.

Programming the 80386

Crawford, John H & Gelsinger, Patrick P.; Sybex; ISBN 0-89588-381-3; 774pp.; $26.95 (USA).

This is the book the Jolitzes used when they ported BSD to the 386 architecture.

Pentium Processor User's Manual: Volume 3, Architecture and Programming Manual

Intel Corp.; 1993; ISBN 1-55512-195-0;

Pretty much the Pentium version of the 80386 Programmer's manual listed above.

10. Books on PC-Class Hardware

Note: these books are four or five years old and possibly out of date. I don't really grok hardware...

80386 Hardware Reference Manual

Intel Corp.; 1986; ISBN 1-55512-024-5;

Pin connections, timing, waveforms, block diagrams, voltages, all that kind of stuff.

The Indispensable PC Hardware Book

Messmer, Hans-Peter; Addison-Wesley; 1993; ISBN 0-201-62424-9; 1000 pp.

JH: "Covers the more recent stuff like EIDE and PCI."

11. Administrivia

11.1 Terms of Use

This document is copyright 1997 by Eric S. Raymond. You may use, disseminate, and reproduce it freely, provided you:

- Do not omit or alter this copyright notice.
- Do not omit or alter or omit the version number and date.
- Do not omit or alter the document's pointer to the current WWW version.
- Clearly mark any condensed, altered or versions as such.

These restrictions are intended to protect potential readers from stale or mangled versions. If you think you have a good case for an exception, ask me.

11.2 History

This was originally a mini-HOWTO maintained by Jim Haynes. I have changed the emphasis somewhat, trying to make it more a standalone document and less reliant on the various USENET bibliographic postings. The unattributed mini-reviews are mine rather than his.

Root RAID HOWTO cookbook

Michael A. Robinton, michael@bzs.org

v1.07, 25 March 1998

This document provides a cookbook for creating a root mounted raid filesystem and companion fallback rescue system using linux initrd. There are complete step-by-step instruction for both raid1 and raid5 md0 devices. Each step is accompanied by an explanation of it's purpose. Included with this revision is a generic ***linuxrc*** *initrd file which may be configured with a single three line /etc/raidboot.conf file for raid1 and raid5 configurations.*

1. Introduction

- 1.1 Where to get Up-to-date copies of this document.
- 1.2 Bugs
- 1.3 Acknowledgements
- 1.4 Copyright Notice

2. What you need BEFORE YOU START

- 2.1 Required Packages
- 2.2 Other similar implementations.
- 2.3 Documentation -- Recommended Reading
- 2.4 RAID resources

3. Quick Start for ROOT RAID

4. initrd‐ Cookbook for root mounted RAID

- 4.1 Security Reminder
- 4.2 Build the Kernel and Raid Tools
- 4.3 Build the *initrd* Rescue and Boot filesystem

- 4.4 Start the STEP by STEP instructions
- 4.5 Install the distribution - Slackware Specific
- 4.6 Install linux **pthreads**
- 4.7 Install Raid Tools
- 4.8 Remove un-needed directories and files from new filesystem.
- 4.9 Create /dev/md*x*
- 4.10 Create a bare filesystem suitable for *initrd*
- 4.11 Making 'initrd' boot the RAID device - linuxrc
- 4.12 Modifying the rc-scripts for SHUTDOWN
- 4.13 Configuring RAIDBOOT - raidboot.conf
- 4.14 Kernel 'loadlin and lilo' variables for RESCUE and RAID

5. Configuring the Production RAID system.

- 5.1 System specs.
- 5.2 Partitioning the hard drives.

6. Building the RAID file system.

- 6.1 /etc/raid5.conf
- 6.2 /etc/raid1.conf
- 6.3 Step by Step procedures for building production RAID file system.

7. One last thought.

8. Appendix A. - Bohumil Chalupa's md0 shutdown

9. Appendix B. - Sample SHUTDOWN scripts

- 9.1 Slackware - /etc/rc.d/rc.6
- 9.2 Debian bo - /etc/init.d/halt and /etc/init.d/reboot

TOC

10. Appendix C. - other setup files

- 10.1 linuxrc
- 10.2 loadlin -- linux.bat file - boot.par
- 10.3 linuxthreads Makefile.diff
- 10.4 raid1.conf
- 10.5 raid5.conf
- 10.6 raidboot.conf
- 10.7 rc.raidown

11. Appendix D. - obsolete linuxrc and shutdown scripts

- 11.1 Obsolete working - linuxrc
- 11.2 Obsolete working - shutdown scripts

12. Appendix E. - Gadi's raid stop patch for the linux kernel

13. Appendix F. - rc.raidown

14. Appendix G. - linuxrc theory of operation

---

1. Introduction

The reader is assumed to be familiar with the various types of raid implementations, their advantages and drawbacks. This is not a tutorial, just a set of instructions on how to implement root mounted raid on a linux system. All of the information necessary to become familiar with linux raid is listed here directly or by reference, please read it before send e-mail questions.

1.1 Where to get Up-to-date copies of this document.

Root-RAID-HOWTO

Available in LaTeX (for DVI and PostScript), plain text, and HTML.

sunsite.unc.edu/mdw/HOWTO/

Available in SGML and HTML.

ftp.bizsystems.com/pub/raid/

1.2 Bugs

As of this writing, the problem of stopping a root mounted RAID device has not yet been solved in a satisfactory way. A work-around proposed by Ed Welbon and implemented by Bohumil Chalupa is incorporated into this document which eliminates the need for a long ckraid at each boot for raid1 and raid5 devices. Without the workaround, it is necessary to **ckraid** the **md** device each time the system is re-booted. On a large array this can cause a severe availability performance degradation. On my 6 gig RAID1 device running on a Pentium 166 with 128 megs of ram, it takes well over half an hour to ckraid :-(after each re-boot. It takes over an hour on my 13 gig RAID5 array with a 20mb/sec scsi adaptor.

The workaround stores the status of the array at shutdown on the **real** boot device and compares it to a reference status placed there when the system is first built. If the status's match at reboot, the superblock on the array is rebuilt on the next boot, otherwise the operator is notified of the status error and the rescue system is left running with all the raid tools available.

Rebuilding the superblock causes the system to ignore that the array was powered down without mdstop by marking all the drives as **OK**, as if nothing happened. This only works if all the drives are OK at shutdown. If the array was operating with a bad drive, the operator must remove the bad drive prior to restarting the md device or the data can be corrupted.

None of this applies to raid0 which does not have to be mdstopped before shutdown.

Final proposed solutions to this problem include a **finalrd** similar to **initrd**, and **mdrootstop** which writes the **clean** flags to the array during shutdown when it is mounted read only. I am sure there are

TOC

others.

In the mean time, the problem has been by-passed for now Please let me know when this problem is solved more cleanly!!!

1.3 Acknowledgements

The writings and e-mail from the following individuals helped to make this document possible. Many of the ideas were *stolen* from the helpful work of others, I have just tried to put it all in **COOKBOOK** form so that it is straightforward to use. My thanks to:

- Linas Vepstas *for the RAID howto that explained most of this to me.*
- Gadi Oxman *for answering my dumb 'newbie' questions.*
- Ed Welbon *for the execellent* ***initrd.md*** *package that inspired me to write this.*
- Bohumil Chalupa *for implementing the re-boot 'workaround' that allows* **root-mounted-raid** *to work in a production environment.*
- and many others who contributed to this work in one way or another.

1.4 Copyright Notice

This document is GNU copyleft by Michael Robinton michael@bzs.org.

Permission to use, copy, distribute this document for any purpose is hereby granted, provided that the author's / editor's name and this notice appear in all copies and/or supporting documents; and that an unmodified version of this document is made freely available. This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY, either expressed or implied. While every effort has been taken to ensure the accuracy of the information documented herein, the author / editor / maintainer assumes NO RESPONSIBILITY for any errors, or for any damages, direct or consequential, as a result of the use of the information documented herein.

---

2. What you need BEFORE YOU START

The packages you need and the documentation that answers the most common questions about setting up and running raid are listed below. Please review them throughly.

2.1 Required Packages

You need to obtain the most recent versions of these packages.

- a linux kernel that supports raid, initrd and /dev/loopx

  I used linux-2.0.33 from sunsite
- raid145-971022-2.0.31 patch adds support for raid1/4/5
- raidtools-pre3-0.42 tools to create and maintain raid devices (documentation too).
- Gadi's raid stop patch in Appendix E.
- linuxthreads-0.71 required threads package. Use ftp, browser doesn't work ftp.inria.fr/INRIA/Projects/cristal/Xavier.Leroy
- A Linux distribution, ready to install.

  I used Slackware-3.4

Helpful but not required

- raidboot-0.01.tar.gz pre-built raid rescue/boot system.

The detailed instructions in this document are based on the above packages. If the packages have been updated or you use a different linux distribution, you may have to modify the procedures you find here.

The patches, tool assortment, etc... may vary with 2.1 kernels. Please check the most recent documentation at:

ftp.kernel.org/pub/linux/daemons/raid/

2.2 Other similar implementations.

I chose to include in the kernel all of the pieces necessary to run from boot without loading any modules. My kernel image is a little over 300k compressed.

Take a look at Ed Welbon's **initrd.md.tar.gz** for another way to make a bootable raid device. He uses loadable modules. A look at his concise scripts will show you how it is done if you need a very small kernel with modules.

http://www.realtime.net/~welbon/initrd.md.tar.gz

2.3 Documentation -- Recommended Reading

TOC

Please read:

/usr/src/linux/Documentation/initrd.txt

as well as the documentation and man pages that accompany the raidtools set. In particular, read **man mdadd** as well as the **QuickStart.RAID** document included in the raidtools package.

You may also wish to review:

- BootPrompt-HOWTO
- **man lilo**
- **man lilo.conf**

2.4 RAID resources

- sunsite.unc.edu/mdw/HOWTO/mini/Software-RAID
- www.ssc.com/lg/issue17/raid.html
- linas.org/linux/raid.html
- ftp.kernel.org/pub/linux/daemons/raid/
- www.realtime.net/~welbon/initrd.md.tar.gz
- luthien.nuclecu.unam.mx/~miguel/raid/

Mailing lists can be joined at:

- majordomo@nuclecu.unam.mx *send a message to* **subscribe raiddev**

  send mail to: raiddev@nuclecu.unam.mx
- majordomo@vger.rutgers.edu *send a message to* **subscribe linux-raid**

  send mail to: linux-raid@vger.rutgers.edu *(this seems to be the most active list)*

---

3. Quick Start for ROOT RAID

If you don't want to try and build and debug the rescue system, you can get a generic one created from Slackware-3.4 from:

ftp.bizsystems.com/pub/raid/raidboot-0.01.tar.gz

Perform the following steps:

- Compile the raid enabled kernel with built in support for your disk subsystem
- Test that the raid array will configure and mount correctly
- Build your OS on the raid system
- Correct the entries in **fstab** to show **/dev/md0** as the root device. Make sure that the partition(s) you use for booting are included in **fstab**.
- Modify your shutdown halt and reboot script(s) (mine is /etc/rc.d/rc.6) as shown in Modifying the rc-scripts for SHUTDOWN
- Copy the following from you development filesystem to the rescue system AND the new raid system

```
        cd /root/raidboot
        mkdir mnt
        gzip -d rescue.clean
        losetup /dev/loop0  rescue.clean
        mount /dev/loop0    mnt

copy these files

        cp -p /etc/*         mnt/etc
        cp -p /etc/rc.d/*    mnt/etc/rc.d
                {or as appropriate for your system}
        cp -a /lib/modules/* mnt/lib/modules
```

Correct the entries in **fstab** to show **/dev/md0** as the root device. Make sure that the partition(s) you use for booting is included in **fstab**.

Create **/etc/raidboot.conf** which describes the raid boot configuration. This file may **NOT** contain comments in the first three lines, after that it doesn't matter.

raidboot.conf

```
        /dev/sda1 /dev/sda2
        raidboot
        raid5.conf
# comments may only be placed 'after' the three
# configuration lines.
#
```

```
# This is 'raidboot.conf'
#
# line one, the partition(s) containing the 'initrd' raid-rescue system
#       It is not necessary to boot from these partitions, however,
#       since the rescue system will not fit on floppy, it is necessary
#       to know which partitions are to be used to load the rescue system
#
# line two, the path to the raidboot config information
#       Where the shutdown status, etc... is located at boot time
#       It does NOT include the mount point information, only 'path'
#       /mntpoint/'path'
#
# line -3-, name of the raid configuration file
#       Current raid configuration file i.e. raid1.conf, raid5.conf
```

A few more things to do and the raid systems is ready to boot.

Create rc.raidown, as described in Appendix F, and copy it to /etc/rc.d on the rescue, development, and raid system. Unmount the rescue system and zip it.

```
umount mnt
losetup -d /dev/loop0
mv rescue.clean rescue
gzip rescue
```

Copy the rescue file to the raidboot partitions.

```
cp rescue.gz /mnt_point(1)/raidboot
cp rescue.gz /mnt_point(2)/raidboot
```

Activate the raid array.

```
mdadd -ar
```

Save the **good** reference status to the raidboot partition

```
cat /proc/mdstat | grep md0 > /mnt_point(1)/raidboot/raidgood.ref
cat /proc/mdstat | grep md0 > /mnt_point(1)/raidboot/raidgood.ref
```

Lastly, configure the boot program as outlined in Boot Time Configuration Parameters and reboot your system onto the raid array.

4. initrd‐ Cookbook for root mounted RAID

This is the procedure to make an 'initrd' ramdisk with rescue tools for raid.

Specifically, this document referrs to RAID1 and RAID5 implementations.

4.1 Security Reminder

The rescue file system may be used stand alone. Should your raid array fail to mount, you are left with the rescue system mounted and running. TAKE THE APPROPRIATE SECURITY PRECAUTIONS!!!

4.2 Build the Kernel and Raid Tools

The first thing that must be done is to patch and build your kernel and become familiar with the raid tools. Make sure and include Gadi's raid stop patch in Appendix E. Configure, mount and test your raid device(s). The details of how to do this are included in the **raidtools** package and briefly reviewed later in this document.

4.3 Build the *initrd* Rescue and Boot filesystem

I used the **Slackware-3.4** distribution to build both the Rescue/Boot filesystem and the filesystem for the production machine. Any linux distribution should work fine. If you use a different distribution, review the Slackware specific portion of this procedure and modify it to suit your needs.

I use loadlin to boot the kernel image and ramdisk from a dos partition simply because there are oddball devices in my system that have dos configuration software. Lilo will work just as well and a small linux partition can be used instead containing only the raid/boot files and the **lilo** record.

For the raid boot/rescue system, I chose to create a minimum ramdisk system using the Slackware 'setup' script followed by installing the 'linuxthreads' package and 'raidtools' over the clean Slackware installation on my ramdisk. I used the ***identical*** procedure to build the production system. So the rescue and production systems are very similar.

This installation process gives me a 'bare' system (save a copy of the file) to which I overlay

```
/lib/modules/2.x.x......
/etc .... with a modified fstab, mdtab, raidX.conf, raidboot.conf
/etc/rc.d
/dev/md*
```

from my current system to customize it for the particular kernel and machine that it is/will-be running on.

This makes the boot/rescue system the same system that is running on the root mounted raid device, just skinnyed down a bit, while allowing the library, etc... revisions to always be current.

4.4 Start the STEP by STEP instructions

TOC

From the root home directory (/root):

```
cd /root
mkdir raidboot
cd raidboot
```

Create a mountpoints to work on

```
mkdir mnt
mkdir mnt2
```

Make a file large enough to do the file system install. This will be a lot larger than the final rescue file system. I chose 24 megs since 16 megs is not large enough

```
dd if=/dev/zero of=build bs=1024k count=24
```

associate the file with a loop device and generate an ext2 file system on the file

```
losetup /dev/loop0 build
mke2fs -v -m0 -L initrd /dev/loop0
mount /dev/loop0 mnt
```

4.5 Install the distribution - Slackware Specific

...skip Slackware Specific stuff and go to next section.

Now that an empty filesystem is created and mounted, run "setup".

```
Specify          /root/raidboot/mnt
```

as the **'target'**. The source is whatever you normally install from. Select the packages you wish to install and proceed but **DO NOT** configure.

Choose 'EXPERT' prompting mode.

I chose 'A', 'AP, and 'N' installing only the minimum to run the system plus an editor I am familiar with (vi, jed, joe) that is reasonably compact.

```
lqqqqqqqq SELECTING PACKAGES FROM SERIES A (BASE LINUX SYSTEM) qqqqqqqk
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x   [X] aaa_base  Basic filesystem, shell, and utils - REQUIRED    x x
x x   [X] bash      GNU bash-1.14.7 shell - REQUIRED                 x x
x x   [X] devs      Device files found in /dev - REQUIRED            x x
x x   [X] etc       System config files & utilities - REQUIRED       x x
x x   [X] shadow    Shadow password suite - REQUIRED                 x x
x x   [ ] ide       Linux 2.0.30 no SCSI (YOU NEED 1 KERNEL)         x x
x x   [ ] scsi      Linux 2.0.30 with SCSI (YOU NEED 1 KERNEL)       x x
x x   [ ] modules   Modular Linux device drivers                     x x
```

TOC

```
x x    [ ] scsimods  Loadable SCSI device drivers                    x x
x x    [X] hdsetup   Slackware setup scripts - REQUIRED              x x
x x    [ ] lilo      Boots Linux (not UMSDOS), DOS, OS/2, etc.       x x
x x    [ ] bsdlpr    BSD lpr - printer spooling system               x x
x x    [ ] loadlin   Boots Linux (UMSDOS too!) from MS-DOS           x x
x x    [ ] pnp       Plug'n'Play configuration tool                  x x
x x    [ ] umsprogs  Utilities needed to use the UMSDOS filesystem   x x
x x    [X] sysvinit  System V-like INIT programs - REQUIRED          x x
x x    [X] bin       GNU fileutils 3.12, elvis, etc. - REQUIRED      x x
x x    [X] ldso      Dynamic linker/loader - REQUIRED                x x
x x    [ ] ibcs2     Runs SCO/SysVr4 binaries                        x x
x x    [X] less      A text pager utility - REQUIRED                 x x
x x    [ ] pcmcia    PCMCIA card services support                    x x
x x    [ ] getty     Getty_ps 2.0.7e - OPTIONAL                      x x
x x    [X] gzip      The GNU zip compression - REQUIRED              x x
x x    [X] ps        Displays process info - REQUIRED                x x
x x    [X] aoutlibs  a.out shared libs - RECOMMENDED                 x x
x x    [X] elflibs   The ELF shared C libraries - REQUIRED           x x
x x    [X] util      Util-linux utilities - REQUIRED                 x x
x x    [ ] minicom   Serial transfer and modem comm package          x x
x x    [ ] cpio      The GNU cpio backup/archiving utility           x x
x x    [X] e2fsbn    Utilities for the ext2 file system              x x
x x    [X] find      GNU findutils 4.1                               x x
x x    [X] grep      GNU grep 2.0                                    x x
x x    [ ] kbd       Change keyboard mappings                        x x
x x    [X] gpm       Cut and paste text with your mouse              x x
x x    [X] sh_utils  GNU sh-utils 1.16 - REQUIRED                    x x
x x    [X] sysklogd  Logs system and kernel messages                 x x
x x    [X] tar       GNU tar 1.12 - REQUIRED                         x x
x x    [ ] tcsh      Extended C shell version 6.07                   x x
x x    [X] txtutils  GNU textutils-1.22 - REQUIRED                   x x
x x    [ ] zoneinfo  Configures your time zone                       x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
```

From the 'AP series, I use only 'JOE', and editor I like, and 'MC' a small and useful file management tool. You choose the utilities you will need on your system.

```
lqqqqqqqq SELECTING PACKAGES FROM SERIES AP (APPLICATIONS) qqqqqqqqk
x x      [ ] ispell    The International version of ispell          x x
x x      [ ] jove      Jonathan's Own Version of Emacs text editor  x x
x x      [ ] manpgs    More man pages (online documentation)        x x
x x      [ ] diff      GNU diffutils                                x x
x x      [ ] sudo      Allow special users limited root access      x x
x x      [ ] ghostscr  GNU Ghostscript version 3.33                 x x
x x      [ ] gsfonts1  Ghostscript fonts (part one)                 x x
x x      [ ] gsfonts2  Ghostscript fonts (part two)                 x x
x x      [ ] gsfonts3  Ghostscript fonts (part three)               x x
x x      [ ] jed       JED programmer's editor                      x x
x x      [X] joe       joe text editor, version 2.8                 x x
x x      [ ] jpeg      JPEG image compression utilities             x x
```

TOC

```
x x     [ ] bc        GNU bc - arbitrary precision math language   x x
x x     [ ] workbone  a text-based audio CD player                 x x
x x     [X] mc        The Midnight Commander file manager          x x
x x     [ ] mt_st     mt ported from BSD - controls tape drive     x x
x x     [ ] groff     GNU troff document formatting system         x x
x x     [ ] quota     User disk quota utilities                    x x
x x     [ ] sc        The 'sc' spreadsheet                         x x
x x     [ ] texinfo   GNU texinfo documentation system             x x
x x     [ ] vim       Improved vi clone                            x x
x x     [ ] ash       A small /bin/sh type shell - 62K             x x
x x     [ ] zsh       Zsh - a custom *nix shell                    x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
```

From the 'N' package I only loaded TCPIP. This isn't really necessary, but is very handy and allows access to the network while working on a repair or update with the root raid array dismounted. TCPIP also contains 'biff' which is used by some of the applications in 'A'. If you don't install 'N' you might want to install the biff package anyway.

```
lqqqq SELECTING PACKAGES FROM SERIES N (NETWORK/NEWS/MAIL/UUCP) qqqqqk
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x    [ ] apache    Apache WWW (HTTP) server                      x x
x x    [ ] procmail  Mail delivery/filtering utility               x x
x x    [ ] dip       Handles SLIP/CSLIP connections                x x
x x    [ ] ppp       Point-to-point protocol                       x x
x x    [ ] mailx     The mailx mailer                              x x
x x    [X] tcpip     TCP/IP networking programs                    x x
x x    [ ] bind      Berkeley Internet Name Domain server          x x
x x    [ ] rdist     Remote file distribution utility              x x
x x    [ ] lynx      Text-based World Wide Web browser             x x
x x    [ ] uucp      Taylor UUCP 1.06.1 with HDB && Taylor configs x x
x x    [ ] elm       Menu-driven user mail program                 x x
x x    [ ] pine      Pine menu-driven mail program                 x x
x x    [ ] sendmail  The sendmail mail transport agent             x x
x x    [ ] metamail  Metamail multimedia mail extensions           x x
x x    [ ] smailcfg  Extra configuration files for sendmail        x x
x x    [ ] cnews     Spools and transmits Usenet news              x x
x x    [ ] inn       InterNetNews news transport system            x x
x x    [ ] tin       The 'tin' news reader (local or NNTP)         x x
x x    [ ] trn       'trn' for /var/spool/news                     x x
x x    [ ] trn-nntp  'trn' for NNTP (install 1 'trn' maximum)      x x
x x    [ ] nn-spool  'nn' for /var/spool/news                      x x
x x    [ ] nn-nntp   'nn' for NNTP (install 1 'nn' maximum)        x x
x x    [ ] netpipes  Network pipe utilities                        x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
```

With the installation complete, say no to everything else (no to all configuration requests) and exit the script.

4.6 Install linux pthreads

Now you must install the 'linuxthreads-0.71' library. I have included this diff for the linuxthreads Makefile rather than explain the details of the installation by hand. Save the original Makefile, apply the diff and then:

```
        cd /usr/src/linuxthreads-0.71
  patch
        make
        make install

-------------------diff Makefile.old  Makefile.raid-----------------
2a3,13
> # If you are building "linuxthreads" for installation on a mount
> # point which is not the "root" partition, redefine 'BUILDIR' to
> # the mount point to use as the "root" directory
> # You may wish to do this if you are building an 'initial ram disk'
> # such as used with bootable root raid devices.
> # REQUIRES ldconfig version 1.9.5 or better
> # do ldconfig -v to check
> #
> BUILDIR=/root/raidboot/mnt
> #BUILDIR=
>
81,82c92,93
<       install pthread.h $(INCLUDEDIR)/pthread.h
<       install semaphore.h $(INCLUDEDIR)/semaphore.h
---
>       install pthread.h $(BUILDIR)$(INCLUDEDIR)/pthread.h
>       install semaphore.h $(BUILDIR)$(INCLUDEDIR)/semaphore.h
84c95
<       test -f /usr/include/sched.h || install sched.h $(INCLUDEDIR)/sched.h
---
>       test -f $(BUILDIR)/usr/include/sched.h || install sched.h
$(BUILDIR)$(INCLUDEDIR)/sched.h
86,89c97,103
<       install $(LIB) $(LIBDIR)/$(LIB)
<       install $(SHLIB) $(SHAREDLIBDIR)/$(SHLIB)
<       rm -f $(LIBDIR)/$(SHLIB0)
<       ln -s $(SHAREDLIBDIR)/$(SHLIB) $(LIBDIR)/$(SHLIB0)
---
>       install $(LIB) $(BUILDIR)$(LIBDIR)/$(LIB)
>       install $(SHLIB) $(BUILDIR)$(SHAREDLIBDIR)/$(SHLIB)
>       rm -f $(BUILDIR)$(LIBDIR)/$(SHLIB0)
>       ln -s $(SHAREDLIBDIR)/$(SHLIB) $(BUILDIR)$(LIBDIR)/$(SHLIB0)
> ifneq ($(BUILDIR),)
>       ldconfig -r ${BUILDIR} -n $(SHAREDLIBDIR)
> else
91c105,106
<       cd man; $(MAKE) MANDIR=$(MANDIR) install
---
> endif
>       cd man; $(MAKE) MANDIR=$(BUILDIR)$(MANDIR) install
```

TOC

4.7 Install Raid Tools

The next step is the installation of the raid tools. raidtools-0.42

You must run the "configure" script to point the Makefile at the build directory for the ramdisk files

```
cd /usr/src/raidtools-0.42
configure --sbindir=/root/raidboot/mnt/sbin --prefix=/root/raidboot/mnt/usr
make
make install
```

Now!! the Makefile for install is not quite right so do the following to clean up. This will be fixed in future releases so that the re-linking will not be necessary.

Fix the make install error

The file links specified in the Makefile at 'LINKS' must be removed and re-linked to operate properly.

```
cd /root/raidboot/mnt/sbin
ln -fs mdadd mdrun
ln -fs mdadd mdstop
```

4.8 Remove un-needed directories and files from new filesystem.

Delete the following directories from filesystem (CAUTION DON'T DELETE FROM YOUR RUNNING SYSTEM) it's easy to do, guess how I found out!!!

```
cd /root/raidboot/mnt
rm -r home/ftp/*
rm -r lost+found
rm -r usr/doc
rm -r usr/info
rm -r usr/local/man
rm -r usr/man
rm -r usr/openwin
rm -r usr/share/locale
rm -r usr/X*
rm -r var/man
rm -r var/log/packages
rm -r var/log/setup
rm -r var/log/disk_contents
```

4.9 Create /dev/md*x*

The last step simply copies the /dev/md* devices from the current file system onto the rescue file system. You could create these with mknode.

```
cp -a /dev/md* /root/raidboot/mnt/dev
```

4.10 Create a bare filesystem suitable for *initrd*

Now you have a clean re-useable filesystem ready for customization. Once customized, this file system can be used for rescue should the raid device(s) become corrupted and the raid tools needed to fix them. It will also be used to boot and root-mount the raid device by adding the linuxrc file which will be discussed next.

Copy the file system to a smaller device for the initrd file, 16 megs should be large enough.

Create the smaller file system and mount it

```
cd /root/raidboot
dd if=/dev/zero of=bare.fs bs=1024k count=16
```

associate the file with a loop device and generate a ext2 file system on the file

```
losetup /dev/loop1 bare.fs
mke2fs -v -m0 -L initrd /dev/loop1
mount /dev/loop1 mnt2
```

Copy the 'build' file system to 'bare.fs'

```
cp -a mnt/* mnt2
```

Save the 'bare.fs' system before customization so later update is easy. The 'build' file system is no longer needed and may be deleted.

```
cd /root/raidboot
umount mnt
umount mnt2
losetup -d /dev/loop0
losetup -d /dev/loop1
rm build
cp bare.fs rescue
gzip -9 bare.fs
```

Create the BOOT/RESCUE *initrd* filesystem

Now copy the system dependent items that match the kernel from the development platform, or you can manually modify the files in the rescue file system to match your target system.

```
losetup /dev/loop0 rescue
mount /dev/loop0 mnt
```

Make sure your etc directory is clean of `*~`, core and log files. The next 2 commands creates some warning messages, ignore them.

```
cp -dp /etc/* mnt/etc
cp -dp /etc/rc.d/* mnt/etc/rc.d
```

```
        mkdir  mnt/lib/modules
        cp -a  /lib/modules/2.x.x mnt/lib/modules <--- your current 2.x.x
```

Edit the following files to correct them for your rescue system.

```
        cd mnt

Non-network
        etc/fstab
        etc/mdtab       should work OK
Network
        etc/hosts
        etc/resolv.conf
        etc/hosts.equiv         and related files
        etc/rc.d/rc.inet1       correct ip#, mask, gateway, etc...
        etc/rc.d/rc.S           remove entire section on file system status
                from:
                        # Test to see if the root partition isread-only
                to but not including:
                        # remove /etc/mtab* so that mount will .....
                                This avoids the annoying warning that
                                the ramdisk is mounted rw.
        etc/rc.d/rc.xxxxx       others as required, see later on in this doc
        root/.rhosts            if present
        home/xxxx/xxxx          others as required

    WARNING:    The above procedure moves your password and shadow
                files onto the rescue disk!!!!!

    WARNING:    You may not wish to do this for security reasons.
```

Create any directories for mounting /dev/disk... as may be required that are unique to your system. These are the mountpoints for booting the system (boot partition and backup boot partition). My system boot from dos using **loadlin**, however linux partition(s) and lilo will work fine. My system uses:

```
        cd /root/raidboot/mnt           <--- initrd root
        mkdir dosa                      dos partition mount point
        mkdir dosb                      dos mirror mount point
```

The rescue file system is complete!

You will note upon examination of the files in the rescue file system, that there are still many files that could be deleted. I have not done this since it would overly complicate this procedure and most raid systems have adequate disk and memory. If you wish to skinny down the file system, go to it!

4.11 Making 'initrd' boot the RAID device - linuxrc

To make the rescue disk boot the raid device, you need only copy the executable script file:

linuxrc

to the root of the device.

The theory of operation for this **linuxrc** file is discussed in Appendix G, linuxrc theory of operation.

A very simple and much easier to understand (working) linuxrc is included in Appendix D, *obsolete linuxrc and shutdown scripts*. Copy the following text to **linuxrc** and save in your development area.

```
 -------------------- linuxrc ----------------------
#!/bin/sh
# ver 1.13 3-6-98
#
################# BEGIN 'linuxrc' ##################
#                DEFINE FUNCTIONS                  #
#####################################################
# Define 'Fault' function in the event something
# goes wrong during the execution of 'linuxrc'
#
FaultExit () {
# correct fstab to show '/dev/ram0' for rescue system
    /bin/cat /etc/fstab | {
    while read Line
    do
        if [ -z "$( echo ${Line} | /usr/bin/grep md0 )" ]; then
            echo ${Line}
        else
            echo "/dev/ram0 / ext2 defaults 1 1"
        fi
    done
    } > /etc/tmp.$$
    /bin/mv /etc/tmp.$$ /etc/fstab
#       point root at /dev/ram0 (the rescue system)
        echo 0x100>/proc/sys/kernel/real-root-dev
        /bin/umount /proc
        exit
}

# Define 'Warning' procdure to print banner on boot terminal
#
Warning () {
    echo '**********************************'
    echo -e " $*"
    echo '**********************************'
}

# Define 'SplitKernelArg' to help extract 'Raid' related kernel arguments
SplitKernelArg () { eval $1='$( IFS=,; echo $2)' }

#Define 'SplitConfArgs' to help extract system configuration arguments
SplitConfArgs () {
    RaidBootType=$1
    RaidBootDevice=$2
    RaidConfigPath=$3
```

TOC

```
}
##########################################################
################### MAIN linuxrc #######################
##########################################################
# mount the proc file system
/bin/mount /proc

# Get the boot partition and configuration location from command line
CMDLINE=`/bin/cat /proc/cmdline`
for Parameter in $CMDLINE; do
    Parameter=$( IFS='='; echo ${Parameter} )
    case $Parameter in
        Raid*) SplitKernelArg $Parameter;;
    esac
done

# check for 'required raid boot'
if [ -z "${Raid_Conf}" ]; then
    Warning Kernel command line \'Raid_Conf\' missing
    FaultExit
fi
SplitConfArgs $Raid_Conf

# tmp mount the boot partition
/bin/mount -t ${RaidBootType} ${RaidBootDevice} /mnt

# get etc files from primary raid system
pushd /etc

# this will un-tar into 'etc' (see rc.6)
if [ ! -f /mnt/${RaidConfigPath}/raidboot.etc ]; then
# bad news, this file should be here
    Warning required file \'raidboot.etc\' \
    missing from ${RaidBootDevice}/${RaidConfigPath} \\n \
    \\tUsing rescue system defaults
else
    /bin/tar -xf /mnt/${RaidConfigPath}/raidboot.etc
fi
# get 'real' raidboot device for this boot
# status path, and name of raidX.conf
if [ ! -f /mnt/${RaidConfigPath}/raidboot.cfg ]; then
# bad news, this file should be here
    Warning required file 'raidboot.cfg' \
    missing from ${RaidBootDevice}/${RaidConfigPath}\\n \
    \\tUsing rescue system defaults
# Get the first raidX.conf file name in $RArg1
    RaidBootDevs=$RaidBootDevice
    RaidStatusPath=$RaidConfigPath
    for RaidConfigEtc in $( ls raid*.conf )
    do break; done
```

```
else
    {
    read RaidBootDevs
    read RaidStatusPath
    read RaidConfigEtc
    } < /mnt/${RaidConfigPath}/raidboot.cfg

fi
popd
/bin/umount /mnt

# Set a flag in case the raid status file is not found
#
RAIDOWN="raidboot.ro not found"
RAIDREF="raidgood.ref not found"
echo "Reading md0 shutdown status."

# search for raid shutdown status
for Device in ${RaidBootDevs}
do
#   these filesystem types should be in 'fstab' since
#   the partitions must be mounted for a clean raid shutdown
    /bin/mount ${Device} /mnt
    if [ -f /mnt/${RaidStatusPath}/raidboot.ro ]; then
        RAIDOWN=`/bin/cat /mnt/${RaidStatusPath}/raidboot.ro`
        RAIDREF=`/bin/cat /mnt/${RaidStatusPath}/raidgood.ref`
        /bin/umount /mnt
        break
    fi
    /bin/umount /mnt
done
# Test for a clean shutdown with array matching reference
if [ "${RAIDOWN}" != "${RAIDREF}" ]; then
    Warning shutdown ERROR ${RAIDOWN}
    FaultExit
fi

# The raid array is clean, remove shutdown status files
for Device in ${RaidBootDevs}
do
    /bin/mount ${Device} /mnt
    /bin/rm -f /mnt/${RaidStatusPath}/raidboot.ro
    /bin/umount /mnt
done

# Write a clean superblock on all raid devices

echo "write clean superblocks"
/sbin/mkraid -f --only-superblock /etc/${RaidConfigEtc}
```

```
# Activate raid array(s)
if [ -z "$Raid_ALT" ]; then
    /sbin/mdadd -ar
else
    /sbin/mdadd $Raid_ALT
fi

#  If there are errors - BAIL OUT and leave rescue running
if [ $? -ne 0 ]; then
   Warning some RAID device has errors
   FaultExit
fi

# Everything is fine, let the kernel mount /dev/md0
# tell the kernel to switch to /dev/md0 as the /root device
# The 0x900 value is the device number calculated by:
#  256*major_device_number + minor_device number
echo "/dev/md0 mounted on root"
echo 0x900>/proc/sys/kernel/real-root-dev
# umount /proc to deallocate initrd device ram space
/bin/umount /proc
exit
#------------------ end linuxrc ----------------------
```

Add 'linuxrc' to initrd boot device

```
        cd /root/raidboot
        chmod 777 linuxrc
        cp -p linuxrc mnt
```

4.12 Modifying the rc-scripts for SHUTDOWN

To complete the installation, modify the rc scripts to save the md status to the real root device when shutdown occurs.

```
In slackware this is rc.0 -> rc.6
In debian 'bo' this is in both 'halt' and 'reboot'

If you implement this in another distribution, please e-mail
the instructions and sample files so they can be included here.
```

I have modified Bohumil Chalupa's raid stop work-around slightly. His original solution is presented in Appendix A.

Since there are no linux partitions left on the production system except **md0**, the boot partitions are used to store the **raidOK readonly** status. I chose to write a file to each of the duplicate boot partitions containing the status of the md array at shutdown and signifying that the md device has been remounted RO. This allows the system to be fail safe when any of the hard drives die.

The shutdown script is modified to call rc.raidown which saves the necessary information to successfully reboot

and mount the raid device. Examples of shutdown scripts for various linux distributions are shown in Append

To capture the raid array shutdown status insert a call to rc.raidown after any **case** statements (if present) but before the actual shutdown (kills, status saves, etc...) begins and before the file systems are dismounted.

```
############ Save raid boot and status info ##############
#
  if [ -x /etc/rc.d/rc.raidown ]; then
    /etc/rc.d/rc.raidown
  fi
################## end raid boot #########################
```

After all the file systems are dismounted (the root file system 'will not' dismount) but before any powerfail status check add:

```
################ for raid arrays #########################
# Stop all known raid arrays (except root which won't stop)
  if [ -x /sbin/mdstop ]; then
    echo "Stopping raid"
    /sbin/mdstop -a
  fi
##########################################################
```

This will cleanly stop all raid devices except root. Root status is passed to the next boot in **raidstat.ro**.

Copy the rc file to your new raid array, the rescue file system that is still mounted on **/root/raidboot/mnt** and the development system if it is on the same machine.

Modify rescue **etc/fstab** as needed and make sure rescue **mdtab** is correct.

Now copy the rescue disk to your dos partition and everything should be ready to boot the raid device as root.

```
        umount mnt
        losetup -d /dev/loop0
        gzip -9 rescue
```

Copy rescue.gz to your boot partitions.

All that remains is to creat the configuration file **raidboot.conf** and test the new file system by rebooting.

4.13 Configuring RAIDBOOT - raidboot.conf

The comments following the example configuration file explain each of the three lines. This example file is for a 4 drive raid5 scsii array with duplicate boot partitions on drives sda1 and sdb1. Put the paramaters descriptive of your file systems here instead.

```
  /dev/sda1 /dev/sdb1
  linux
  raid5.conf
# comments may only be placed 'after' the three
# configuration lines.
#
```

TOC

```
# This is 'raidboot.conf'
#
# line one, the partition(s) containing the 'initrd' raid-rescue system
#       It is not necessary to boot from these partitions, however,
#       since the rescue system will not fit on floppy, it is necessary
#       to know which partitions are to be used to load the rescue system
#
# line two, the path to the raidboot config information
#       Where the shutdown status, etc... is located at boot time
#       It does NOT include the mount point information, only 'path'
#       /mntpoint/'path'
#
# line -3-, name of the raid configuration file
#       Current raid configuration file i.e. raid1.conf, raid5.conf
```

4.14 Kernel 'loadlin and lilo' variables for RESCUE and RAID

There are two kernel variables for the RESCUE and RAID system, only the first need be specified.

- Raid_Conf=msdos,/dev/sda1,raidboot

  This variable points to raid boot device and configuration file. For floppy rescue boot, you may want to specify this on the kernel command line or in the loadlin or lilo boot file

  format: `'filesystem-type,device,path-to-config-from-mountpoint'`

- Raid_ALT=-r,-p5,/dev/md0,/dev/sda3 /dev/sdb3 /dev/sdc3 /dev/sdd3

  Alternate mdadd parameters necessary when booting with non-redundant raid array. These are the comma separated command line parameters for **mdadd**. Unless they are needed to start a failed/non-redundant array, COMMENT OUT OR SPECIFY WITH A 'NULL'.

  i.e. Raid_ALT=

Either of these parameters may be specified in the lilo or loadlin boot parameter file or on the loadlin kernel command line. Care must be taken that the maximum line length is not exceeded, however, if the command line is used (128 characters).

When booting with **lilo**, the parameters are included in the lilo config file in the form:

```
append="Raid_Conf=msdos,/dev/sda1,raidboot"
append="Raid_ALT=-r,-p5,/dev/md0,/dev/sda3 /dev/sdb3 /dev/sdc3 /dev/sdd3"
```

See **man lilo.conf** for more detailed information.

Since I have some hardware that requires DOS configuration utilities, I have a small dos partition on the system. Therefore, I used loadlin to boot the raid5 system from the dos partition with a mirror (copy) on the companion disk. An identical method is used for the raid1 system. The example below uses loadlin, but the procedure is very similar for lilo.

My dos root system contains a small editor among the utilities so I can modify the boot parameters of loadlin if necessary, allowing me to reboot the linux system on my swap disk while testing.

The dos system contains this tree for linux

```
c:\raidboot.bat
c:\raidboot\loadlin.exe
c:\raidboot\zimage
c:\raidboot\rescue.gz
c:\raidboot\raidboot.cfg
c:\raidboot\raidboot.etc
c:\raidboot\raidgood.ref
c:\raidboot\raidstat.ro (only at shutdown)
```

linux.bat contains:

```
---------------------- linux.bat ---------------------------
echo "Start the LOADLIN process:"
c:\raidboot\loadlin @c:\raidboot\boot.par
-------------------- end linux.bat -------------------------
```

boot.par contains:

```
# loadlin boot parameter file
#
# version 1.02 3-6-98

# linux kernel image
c:\linux\zimage

# target root device
root=/dev/md0
#root=/dev/ram0
#root=/dev/sdc5

# mount root device as 'ro'
ro

# size of ram disk
ramdisk_size=16384

# initrd file name
initrd=c:\raidboot\rescue.gz
#noinitrd

# memory ends here
mem=131072k

# points to raid boot device, configuration file
# for floppy rescue boot, you may want to specify
# this on the command line instead of here
# format 'filesystem-type,device,path-to-config-frm_mntpnt'
Raid_Conf=msdos,/dev/sda1,raidboot

# Alternate mdadd parameters
```

```
        # necessary when boot with non-redundant raid
        # otherwise, COMMENT OUT OR SPECIFY 'NULL'
        #Raid_ALT=-r,-p5,/dev/md0,/dev/sda3 /dev/sdb3 /dev/sdc3 /dev/sdd3

        # ethernet devices
        ether=10,0x300,eth0

***** >> NOTE!! the only difference between forcing the rescue system to
            run and the raid device mounting, is the loadlin parameter

                root=/dev/ram0          for the rescue system
                root=/dev/md0           for RAID

                With root=/dev/ram0 the RAID device will not mount
                and the rescue system will run unconditionally.
```

If the RAID array fails, the rescue system is left mounted and running.

---

5. Configuring the Production RAID system.

5.1 System specs. Two systems with identical motherboards were configured.

```
                                  Raid-1          Raid-5
Motherboard:    Iwill P55TU     dual ide        adaptec scsi
Processor:      Intel P200
Disks:                          2ea  7 gig      4 ea Segate 4.2 gig
                                Maxtors         wide scsii
```

The disk drives are designated by linux as 'sda' through 'sdd' on the raid5 system and 'hda' and 'hdc' on the raid1 system.

5.2 Partitioning the hard drives.

Since testing a large root mountable RAID array is difficult because of the ckraid re-boot problem, I re-partitioned my swap space to include a smaller RAID partition for testing purposes, sda6,sdb6,sdc6,sdd6, and a small root and /usr/src partition pair for developing and testing the raid kernel and tools. You may find this helpful.

```
        <bf/DEVELOPMENT SYSTEM - RAID5/
   Device       System          Size    Purpose

  /dev/sda1     dos boot        16 meg  boot partition
* /dev/sda2     extended        130 meg (see below)
  /dev/sda3     linux native    4 gig   primary raid5-1
----------------------sda2-------------------------------
* /dev/sda5     linux swap      113 meg SWAP space
* /dev/sda6     linux native    16 meg  test raid5-1
==============================================================
  /dev/sdb1     dos boot        16 meg  boot partition duplicate
* /dev/sdb2     extended        130 meg (see below)
  /dev/sdb3     linux native    4 gig   primary raid5-2
----------------------sdb2-------------------------------
* /dev/sdb5     linux swap      113 meg SWAP space
* /dev/sdb6     linux native    16 meg  test raid5-2
==============================================================
* /dev/sdc2     extended        146 meg (see below)
```

TOC

```
  /dev/sdc3     linux native    4 gig   primary raid5-3
----------------------sdc2-------------------------------
* /dev/sdc5     linux swap      130 meg development root partition
* /dev/sdc6     linux native    16 meg  test raid5-3
==============================================================
* /dev/sdd2     extended        146 meg (see below)
  /dev/sdd3     linux native    4 gig   primary raid5-4
----------------------sdd2-------------------------------
* /dev/sdd5     linux swap      130 meg development /usr/src
* /dev/sdd6     linux native    16 meg  test raid5-4


        <bf/DEVELOPMENT SYSTEM - RAID1/
   Device       System          Size    Purpose

  /dev/hda1     dos             16meg   boot partition
* /dev/hda2     extended        126m    (see below)
  /dev/hda3     linux           126m    development root partition
  /dev/hda4     linux           6+gig   raid1-1
----------------------hda2-------------------------------
* /dev/hda5     linux            26m    test raid1-1
* /dev/hda6     linux swap      100m
==============================================================

  /dev/hdc1     is simply an exact copy of hda1 so the
                partion can be made active if hda fails
* /dev/hdc2     extended        126m    (see below)
  /dev/hdc3     linux           126m    development /usr/src
  /dev/hdc4     linux           6+gig   raid1-2
----------------------hdc2-------------------------------
* /dev/hdc5     linux            26m    test raid1-2
* /dev/hdc6     linux swap      100m
```

The sdx2 and hdx3 partitions were switched to 'swap' after developing this utility. I could have done it on another machine, however, the libraries and kernels are all about a year or more out of date on my other linux boxes and I preferred to build it on the target machine.

The partitioning scheme was chosen so that in the event that any one of the drives fails catastrophically, the system will continue to run and be bootable with minimum effort and NO data loss.

- If any single hard drive fails, the boot will abort, and the rescue system will run. Examination of the screen message or /dos*x*/raidboot/raidstat.ro will tell the operator the status of the failed array.
- If sda1 (raid5) or hda1 (raid1) fails, the dos backup boot partition must be made 'active' and the bios must recognize the new partition as the boot device or it must be physically be moved to the *x*da position. Alternatively, the system could be booted from a floppy disk using the initrd image on the remaining backup boot drive. The raid system can then be made active again by issuing:

```
"/sbin/mkraid /etc/raid<it/x/.conf -f --only-superbloc
```

to rebuild the remaining superblock(s).

- Once this is done, then

```
mdadd -ar
```

- Examine the status of the array to verify that everything is OK then replace the good array reference with the current status until the failed disk can be repaired or replaced.

```
cat /proc/mdstat | grep md0 > /dosx/raidboot/raidgood.ref

shutdown -r now
```

to do a clean reboot, and the system is up again.

---

6. Building the RAID file system.

This description is for my RAID systems described in the system specs. Your system may have a different RAID architecture, so modify as appropriate. Please read the man pages and QuickStart.RAID that come with the raidtools-0.42

6.1 /etc/raid5.conf

```
# raid-5 configuration
raiddev                 /dev/md0
raid-level              5
nr-raid-disks           4
chunk-size              32

# Parity placement algorithm
parity-algorithm        left-symmetric

# Spare disks for hot reconstruction
#nr-spare-disks         0

device                  /dev/sda3
raid-disk               0

device                  /dev/sdb3
raid-disk               1

device                  /dev/sdc3
raid-disk               2

device                  /dev/sdd3
raid-disk               3
```

6.2 /etc/raid1.conf

```
# raid-1 configuration
raiddev                 /dev/md0
raid-level              1
nr-raid-disks           2
nr-spare-disks          0

device                  /dev/hda4
```

```
raid-disk                0

device                   /dev/hdc4
raid-disk                1
```

6.3 Step by Step procedures for building production RAID file system.

For my RAID5 system I did a complete install of:

```
Slackware-3.4            any current distribution should work OK
linuxthreads-0.71
raidtools-0.42
linux-2.0.33 with raid145 patch and Gadi's patch
```

Create and format the raid device.

```
mkraid /etc/raid5.conf
mdcreate raid5 /dev/md0 /dev/sda3 /dev/sdb3 /dev/sdc3 /dev/sdd3
mdadd -ar
mke2fs /dev/md0
mkdir /md
mount -t ext2 /dev/md0 /md
```

Create the reference files that reboot will use, this may be different on your system.

```
cat /proc/mdstat | grep md0 > /dosa/raidboot/raidgood.ref
cat /proc/mdstat | grep md0 > /dosb/raidboot/raidgood.ref
```

Use Slackware-3.4 or another distribution to build your OS

```
setup
```

Specify '/md' as the target, and the source whatever your normally use. Select and install the disksets of interest except for the kernel. Configure the system, but skip the section on lilo and kernel booting. Exit setup.

Install 'pthreads'

```
cd /usr/src/linuxthreads-0.71
```

edit the Makefile and specify

```
BUILDIR=/md

make
make install
```

Install 'raidtools'

```
cd /usr/src/raidtools-0.42
```

```
        configure --sbindir=/md/sbin --prefix=/md/usr
```

fix the raidtools make install error

```
        cd /md/sbin
        rm mdrun
        rm mdstop
        ln -s mdadd mdrun
        ln -s mdadd mdstop
```

Create /dev/mdx

```
        cp -a /dev/md* /md/dev
```

Add the system configuration from the current system (ignore errors).

```
        cp -dp /etc/* mnt/etc
        cp -dp /etc/rc.d/* mnt/etc/rc.d         (include the new rc.6)
        mkdir  mnt/lib/modules
        cp -a  /lib/modules/2.x.x mnt/lib/modules <--- your current 2.x.x
```

Edit the following files to correct them for your file system

```
        cd /md

Non-network
        etc/fstab       correct for real root and raid devices.
        etc/mdtab       should work OK
Network
        etc/hosts
        etc/resolv.conf
        etc/hosts.equiv         and related files
        etc/rc.d/rc.inet1       correct ip#, mask, gateway, etc...
        etc/rc.d/rc.S           remove entire section on file system status
                from:
                        # Test to see if the root partition isread-only
                to but not including:
                        # remove /etc/mtab* so that mount will .....
                                This avoids the annoying warning that
                                the ramdisk is mounted rw.
        etc/rc.d/rc.xxxxx       others as required
        root/.rhosts            if present
        home/xxxx/xxxx          others as required

    WARNING:    The above procedure moves your password and shadow
                files onto the new file system!!!!!

    WARNING:    You may not wish to do this for security reasons.
```

Create any directories for mounting /dev/disk... as may be required that are unique to your system. Mine need:

```
cd /md          <--- new file system root
mkdir dosa                      dos partition mount point
mkdir dosb                      dos mirror mount point
```

The new file system is complete. Make sure and save the md reference status to the 'real' root device and you are ready to boot.

mount the dos partitions on dosa and dosb

```
cat /proc/mdstat | grep md0 > /dosa/raidboot/raidgood.ref
cat /proc/mdstat | grep md0 > /dosb/raidboot/raidgood.ref

mdstop /dev/md0
```

7. One last thought.

Remember that an expert is someone who knows at least 1% more than you do about a subject. Bear this in mind when you e-mail me for help. I'll try, but I've only done this once for raid1 and once for raid5!

Michael Robinton Michael@bzs.org

8. Appendix A. - Bohumil Chalupa's md0 shutdown

Bohumil Chalupa's post to the linux raid list on the work around for the raid1 + 5 mdstop problem. His solution does not address the possibility of the raid device being corrupt at shutdown. So I have added a simple status comparison to a good reference status at boot. This allows the operator to intervene if something is wrong with a disk in the array. The description of this is in the main body of this document.

```
> From: Bohumil Chalupa <bochal@apollo.karlov.mff.cuni.cz>
>
> I can now boot initrd and use linuxrc to start the RAID1 array,
> then successfully switch root to /dev/md0.
>
> I don't know, however, any way how to cleanly _stop_ the array.

Well. I have to answer myself :-)

> Date: Mon, 29 Dec 1997 02:21:38 -0600 (CST)
> From: Edward Welbon <welbon@bga.com>
> Subject: Re: dismounting root raid device
>
> For md devices other than raid0, there is probably state that needs to
> be saved that is only known once all writes have completed.  Such state
> of course can't be saved to root once it is mounted readonly.  In that
> case, you would have to be able to mount a writeable filesystem "X"
> on the readonly root and be able to write to "X" (I recall doing this
> during "rescue" operations, but not as an automated procedure).
>
> The filesystem "X" would presumably be a boot device from which the raid
> (during linuxrc exection via initrd) would pickup it's initial state from.
> Fortunately raid0 isn't required to write out any state (though it would
> be pleasant to be able to write the check sums to mdtab after an mdstop).
> Eventually, I will fiddle with this but it doesn't seem difficult though
> the "devil" is always in the "details".

Yes, that's it.
I had this idea in mind for some time already, but had no time to try it.
Yesterday I did, and it works.

With my RAID1 (mirror), I don't save any checksums or raid superblock data.
I only save an information on the "real" boot partition, that the root md
volume was remounted readonly during shutdown. Then, during boot, the
linuxrc script runs mkraid --only-superblock  when it finds this
information; otherwise, it runs ckraid.
This means, that the raid superblock information is not updated during
shutdown; it's updated at the boot time.
```

It is not very clean, I'm afraid, :-(but it works.

I'm using Slackware and initrd.md by Edward Welbon to boot the root raid device.
As far as I remember now, the only modified files are
mkdisk and linuxrc, and /etc/rc.d/rc.6 shutdown script.
And lilo.conf, of course.

I'm appending the important parts.

Bohumil Chalupa

```
--------------- my.linuxrc follows -----------------
#!/bin/sh
# we need /proc
/bin/mount /proc
# start up the md0 device. let the /etc/rc.d scripts get the rest of them
# we should do as little as possible here
# ____________________________________________
# root raid1 shutdown test & recreation
# /start must be created on the rd image in my.mkdisk
echo "preparing md0: mounting /start"
/bin/mount /dev/sda2 /start -t ext2
echo "reading saved md0 state from /start"
if [ -f /start/root.raid.ok ]; then
 echo "raid ok, modyfying superblock"
 rm /start/root.raid.ok
 /sbin/mkraid /etc/raid1.conf -f --only-superblock
else
 echo "raid not clean, runing ckraid --fix"
 /sbin/ckraid --fix /etc/raid1.conf
fi
echo "unmounting /start"
/bin/umount /start
# ____________________________________________
#
echo "adding md0 for root file system"
/sbin/mdadd /dev/md0 /dev/sda1 /dev/sdb1
echo "starting md0"
/sbin/mdrun -p1 /dev/md0
# tell kernel we want to switch to /dev/md0 as root device, the 0x900 value
# is arrived at via 256*major_device_number + minor_device number.
echo "setting real-root-dev"
/bin/echo 0x900>/proc/sys/kernel/real-root-dev
#  unmount /proc so that the ram disk can be deallocated.
echo "unmounting /proc"
/bin/umount /proc
/bin/echo "We are hopefully ready to mount /dev/md0 (major 9, minor 0) as
root"
```

```
exit
--------------- end of my.linuxrc ----------------------------------


----------- extract from /etc/rc.d/rc.6 follows -----------------
  # Turn off swap, then unmount local file systems.
  echo "Turning off swap."
  swapoff -a
  echo "Unmounting local file systems."
  umount -a -tnonfs
  # Don't remount UMSDOS root volumes:
  if [ ! "`mount | head -1 | cut -d ' ' -f 5`" = "umsdos" ]; then
    mount -n -o remount,ro /
  fi

  # Save raid state
  echo "Saving RAID state"
  /bin/mount -n /dev/sda2 /start -t ext2
  touch /start/root.raid.ok
  /bin/umount -n /start

-------------- end of excerpt from rc.6 ------------------------


----------------- part of my.mkdisk follows ---------------------
#
#  now we have the filesystem ready to be populated, we need to
#  get a few important directories.  I had endless trouble till
#  I created a pristine mtab.  In my case, it is convenient that
#  /etc/mdtab is copied over, this way I can activate md with
#  a simple "/sbin/mdadd -ar" in linuxrc.
#
cp -a $ROOT/etc $MOUNTPNT 2>cp.stderr 1>cp.stdout
rm -rf $MOUNTPNT/etc/mtab
rm -rf $MOUNTPNT/etc/ppp*
rm -rf $MOUNTPNT/etc/termcap
rm -rf $MOUNTPNT/etc/sendmail*
rm -rf $MOUNTPNT/etc/rc.d
rm -rf $MOUNTPNT/etc/dos*
cp -a $ROOT/sbin $ROOT/dev $ROOT/lib $ROOT/bin $MOUNTPNT 2>>cp.stderr
1>>cp.stdout
# ______________________________________________________________________
#  RAID: will need mkraid and ckraid
cp -a $ROOT/usr/sbin/mkraid $ROOT/usr/sbin/ckraid $MOUNTPNT/sbin
2>>cp.stderr 1>>cp.stdout
# ----------------------------------------------------------------------
#  it seems that init wont come out to play unless it has utmp.   this can
#  probably be pruned back alot.  no telling what the real bug was 8-).
#
```

```
mkdir $MOUNTPNT/var $MOUNTPNT/var/log $MOUNTPNT/var/run $MOUNTPNT/initrd
touch $MOUNTPNT/var/run/utmp $MOUNTPNT/etc/mtab
chmod a+r $MOUNTPNT/var/run/utmp $MOUNTPNT/etc/mtab
ln -s /var/run/utmp $MOUNTPNT/var/log/utmp
ln -s /var/log/utmp $MOUNTPNT/etc/utmp
ls -lstrd $MOUNTPNT/etc/utmp $MOUNTPNT/var/log/utmp $MOUNTPNT/var/run/utmp
#
#  since I wanted to change the mount point, I needed this though
#  I suppose that I could have done a "mkdir /proc" in linuxrc.
#
mkdir $MOUNTPNT/proc
chmod 555 $MOUNTPNT/proc
#
#  -------------------------------------------------------
#  we'll mount the real boot device to /start temporarily
#  to check the root raid state saved at shutdown time
#
mkdir $MOUNTPNT/start
#  ---------------------------------------------------------
#
#  need linuxrc  (it is, after all, the point of this exercise).
#
if [ -x ./my.linuxrc ]; then
  cp -a ./my.linuxrc $MOUNTPNT/linuxrc
  chmod 777 $MOUNTPNT/linuxrc
else
   ln -s /bin/sh $MOUNTPNT/linuxrc
fi
#
----------------- part of my.mkdisk ends -----------------
```

---

9. Appendix B. - Sample SHUTDOWN scripts

- Slackware
- Debian

9.1 Slackware - /etc/rc.d/rc.6

```
#! /bin/sh
#
# rc.6          This file is executed by init when it goes into runlevel
#               0 (halt) or runlevel 6 (reboot). It kills all processes,
#               unmounts file systems and then either halts or reboots.
#
# Version:      @(#)/etc/rc.d/rc.6      1.50    1994-01-15
#
# Author:       Miquel van Smoorenburg <miquels@drinkel.nl.mugnet.org>
# Modified by:  Patrick J. Volkerding, <volkerdi@ftp.cdrom.com>
#
# Modified by:  Michael A. Robinton < michael@bizsystems.com >
#               to add call to rc.raidown
  # Set the path.
  PATH=/sbin:/etc:/bin:/usr/bin

  # Set linefeed mode to avoid staircase effect.
  stty onlcr

  echo "Running shutdown script $0:"

  # Find out how we were called.
  case "$0" in
        *0)
                message="The system is halted."
                command="halt"
                ;;
        *6)
                message="Rebooting."
                command=reboot
                ;;
        *)
                echo "$0: call me as \"rc.0\" or \"rc.6\" please!"
                exit 1
                ;;
```

TOC

```
  esac

############ Save raid boot and status info ##############
#
if [ -x /etc/rc.d/rc.raidown ]; then
   /etc/rc.d/rc.raidown
fi
################## end raid boot #########################

  # Kill all processes.
  # INIT is supposed to handle this entirely now, but this didn't always
  # work correctly without this second pass at killing off the processes.
  # Since INIT already notified the user that processes were being killed,
  # we'll avoid echoing this info this time around.
  if [ "$1" != "fast" ]; then # shutdown did not already kill all processes
    killall5 -15
    killall5 -9
  fi

  # Try to turn off quota and accounting.
  if [ -x /usr/sbin/quotaoff ]
  then
        echo "Turning off quota."
        /usr/sbin/quotaoff -a
  fi
  if [ -x /sbin/accton ]
  then
        echo "Turning off accounting."
        /sbin/accton
  fi

  # Before unmounting file systems write a reboot or halt record to wtmp.
  $command -w

  # Save localtime
  [ -e /usr/lib/zoneinfo/localtime ] && cp /usr/lib/zoneinfo/localtime /etc

  # Asynchronously unmount any remote filesystems:
  echo "Unmounting remote filesystems."
  umount -a -tnfs &

  # Turn off swap, then unmount local file systems.
  echo "Turning off swap."
  swapoff -a
  echo "Unmounting local file systems."
  umount -a -tnonfs
  # Don't remount UMSDOS root volumes:
  if [ ! "`mount | head -1 | cut -d ' ' -f 5`" = "umsdos" ]; then
```

TOC

```
    mount -n -o remount,ro /
  fi

################ for raid arrays ########################
# Stop all known raid arrays (except root which won't stop)
if [ -x /sbin/mdstop ]; then
  echo "Stopping raid"
  /sbin/mdstop -a
fi
##########################################################

  # See if this is a powerfail situation.
  if [ -f /etc/powerstatus ]; then
    echo "Turning off UPS, bye."
    /sbin/powerd -q
    exit 1
  fi

  # Now halt or reboot.
  echo "$message"
  [ ! -f /etc/fastboot ] && echo "On the next boot fsck will be FORCED."
  $command -f
############### end rc.6 ################################
```

9.2 Debian bo - /etc/init.d/halt and /etc/init.d/reboot

The modifications shown here for Debian bo halt and reboot files are NOT TESTED. When you test this, please e-mail me so I can remove this comment.

/etc/init.d/halt

```
#! /bin/sh
#
# halt          The commands in this script are executed as the last
#               step in runlevel 0, ie halt.
#
# Version:      @(#)halt  1.10  26-Apr-1997  miquels@cistron.nl
#

PATH=/sbin:/bin:/usr/sbin:/usr/bin

############ Save raid boot and status info #############
#
if [ -x /etc/rc.d/rc.raidown ]; then
   /etc/rc.d/rc.raidown
fi
################## end raid boot ########################
```

TOC

```
# Kill all processes.
echo -n "Sending all processes the TERM signal... "
killall5 -15
echo "done."
sleep 5
echo -n "Sending all processes the KILL signal... "
killall5 -9
echo "done."

# Write a reboot record to /var/log/wtmp.
halt -w

# Save the random seed between reboots.
/etc/init.d/urandom stop

echo -n "Deactivating swap... "
swapoff -a
echo "done."

echo -n "Unmounting file systems... "
umount -a
echo "done."

mount -n -o remount,ro /

################ for raid arrays #########################
# Stop all known raid arrays (except root which won't stop)
if [ -x /sbin/mdstop ]; then
  echo "Stopping raid"
  /sbin/mdstop -a
fi
##########################################################

# See if we need to cut the power.
if [ -x /etc/init.d/ups-monitor ]
then
        /etc/init.d/ups-monitor poweroff
fi

halt -d -f
############# end halt ####################
```

/etc/init.d/reboot

```
#! /bin/sh
#
```

TOC

```
# reboot        The commands in this script are executed as the last
#               step in runlevel 6, ie reboot.
#
# Version:      @(#)reboot  1.9  02-Feb-1997  miquels@cistron.nl
#

PATH=/sbin:/bin:/usr/sbin:/usr/bin

############ Save raid boot and status info ##############
#
if [ -x /etc/rc.d/rc.raidown ]; then
   /etc/rc.d/rc.raidown
fi
################## end raid boot #########################

# Kill all processes.
echo -n "Sending all processes the TERM signal... "
killall5 -15
echo "done."
sleep 5
echo -n "Sending all processes the KILL signal... "
killall5 -9
echo "done."

# Write a reboot record to /var/log/wtmp.
halt -w

# Save the random seed between reboots.
/etc/init.d/urandom stop

echo -n "Deactivating swap... "
swapoff -a
echo "done."

echo -n "Unmounting file systems... "
umount -a
echo "done."

mount -n -o remount,ro /

################ for raid arrays #########################
# Stop all known raid arrays (except root which won't stop)
if [ -x /sbin/mdstop ]; then
  echo "Stopping raid"
  /sbin/mdstop -a
fi
##########################################################
```

```
echo -n "Rebooting... "
reboot -d -f -i
```

10. Appendix C. - other setup files

10.1 linuxrc linuxrc file

10.2 loadlin -- linux.bat file - boot.par linux.bat file - boot.par

10.3 linuxthreads Makefile.diff linuxthreads Makefile.diff

10.4 raid1.conf raid1.conf

10.5 raid5.conf raid5.conf

10.6 raidboot.conf raidboot.conf

10.7 rc.raidown rc.raidown

11. Appendix D. - obsolete linuxrc and shutdown scripts

11.1 Obsolete working - linuxrc

This linuxrc file works fine with the shutdown procedure in the next subsection.

```
 ---------------------- linuxrc --------------------
#!/bin/sh
# ver 1.07 2-12-98
# linuxrc - for raid1 using small dos partition and loadlin
#

# mount the proc file system
/bin/mount /proc

# This may vary for your system.
# Mount the dos partitions, try both
# in case one disk is dead
/bin/mount /dosa
/bin/mount /dosc

# Set a flag in case the raid status file is not found
# then check both drives for the status file
RAIDOWN="raidstat.ro not found"
/bin/echo "Reading md0 shutdown status."
if [ -f /dosa/raidboot/raidstat.ro ]; then
  RAIDOWN=`/bin/cat /dosa/raidboot/raidstat.ro`
  RAIDREF=`/bin/cat /dosc/raidboot/raidgood.ref`
else
  if [ -f /dosc/raidboot/raidstat.ro ]; then
    RAIDOWN=`/bin/cat /dosc/raidboot/raidstat.ro`
    RAIDREF=`/bin/cat /dosc/raidboot/raidgood.ref`
  fi
fi

# Test for a clean shutdown with all disks operational
if [ "${RAIDOWN} != ${RAIDREF}" ]; then
  echo "ERROR ${RAIDOWN}"
#  Use the next 2 lines to BAIL OUT and leave rescue running
   /bin/echo 0x100>/proc/sys/kernel/real-root-dev
   exit                 # leaving the error files in dosa/raidboot,etc...
fi

# The raid array is clean, proceed by removing
# status file and writing a clean superblock
/bin/rm /dosa/raidboot/raidstat.ro
```

```
/bin/rm /dosc/raidboot/raidstat.ro
/sbin/mkraid /etc/raid1.conf -f --only-superblock

/bin/umount /dosa
/bin/umount /dosc

# Mount raid array
echo "Mounting md0, root filesystem"
/sbin/mdadd -ar

#  If there are errors - BAIL OUT and leave rescue running
if [ $? -ne 0 ]; then
   echo "RAID device has errors"
#  Use the next 3 lines to BAIL OUT
   /bin/rm /etc/mtab            # remove bad mtab
   /bin/echo 0x100>/proc/sys/kernel/real-root-dev
   exit
fi

# else tell the kernel to switch to /dev/md0 as the /root device
# The 0x900 value the device number calculated by:
#  256*major_device_number + minor_device number
/bin/echo 0x900>/proc/sys/kernel/real-root-dev

# umount /proc to deallocate initrd device ram space
/bin/umount /proc
/bin/echo "/dev/md0 mounted as root"
exit
#------------------ end linuxrc ----------------------
```

11.2 Obsolete working - shutdown scripts

This shutdown procedure works fine with the preceeding **linuxrc**

To capture the raid array shutdown status, just before the file systems are dismounted insert:

```
RAIDSTATUS=`/bin/cat /proc/mdstat | /usr/bin/grep md0`
```

After all the file systems are dismounted (the root file system 'will not' dismount) add:

```
# root device remains mounted RO
# mount dos file systems RW
mount -n -o remount,ro /
echo "Writing RAID read-only boot FLAG(s)."
mount -n /dosa
mount -n /dosc
# create raid mounted RO flag in duplicate
# containing the shutdown status of the raid array
echo ${RAIDSTATUS} > /dosa/raidboot/raidstat.ro
echo ${RAIDSTATUS} > /dosc/raidboot/raidstat.ro
```

```
        umount -n /dosa
        umount -n /dosc

        # Stop all the raid arrays (except root)
        echo "Stopping raid"
        mdstop -a
```

This will cleanly stop all raid devices except root. Root status is passed to the next boot in **raidstat.ro**.

The complete shutdown script from my old raid1 Slackware system follows, I have switched raid1 to the new procedure with the /etc/raidboot.conf file.

```
#! /bin/sh
#
# rc.6          This file is executed by init when it goes into runlevel
#               0 (halt) or runlevel 6 (reboot). It kills all processes,
#               unmounts file systems and then either halts or reboots.
#
# Version:      @(#)/etc/rc.d/rc.6      1.50    1994-01-15
#
# Author:       Miquel van Smoorenburg <miquels@drinkel.nl.mugnet.org>
# Modified by:  Patrick J. Volkerding, <volkerdi@ftp.cdrom.com>
# Modified by:  Michael A. Robinton, <michael@bzs.org> for RAID shutdown

  # Set the path.
  PATH=/sbin:/etc:/bin:/usr/bin

  # Set linefeed mode to avoid staircase effect.
  stty onlcr

  echo "Running shutdown script $0:"

  # Find out how we were called.
  case "$0" in
        *0)
                message="The system is halted."
                command="halt"
                ;;
        *6)
                message="Rebooting."
                command=reboot
                ;;
        *)
                echo "$0: call me as \"rc.0\" or \"rc.6\" please!"
                exit 1
                ;;
  esac

  # Kill all processes.
  # INIT is supposed to handle this entirely now, but this didn't always
```

```
# work correctly without this second pass at killing off the processes.
# Since INIT already notified the user that processes were being killed,
# we'll avoid echoing this info this time around.
if [ "$1" != "fast" ]; then # shutdown did not already kill all processes
  killall5 -15
  killall5 -9
fi

# Try to turn off quota and accounting.
if [ -x /usr/sbin/quotaoff ]
then
      echo "Turning off quota."
      /usr/sbin/quotaoff -a
fi
if [ -x /sbin/accton ]
then
      echo "Turning off accounting."
      /sbin/accton
fi

# Before unmounting file systems write a reboot or halt record to wtmp.
$command -w

# Save localtime
[ -e /usr/lib/zoneinfo/localtime ] && cp /usr/lib/zoneinfo/localtime /etc

# Asynchronously unmount any remote filesystems:
echo "Unmounting remote filesystems."
umount -a -tnfs &

# you must have issued
# 'cat /proc/mdstat | grep md0 > {your boot vol}/raidboot/raidgood.ref'
# before linuxrc will execute properly with this info
RAIDSTATUS=`/bin/cat /proc/mdstat | /usr/bin/grep md0 # capture raid status`

# Turn off swap, then unmount local file systems.
# clearing mdtab as well
echo "Turning off swap."
swapoff -a
echo "Unmounting local file systems."
umount -a -tnonfs

# Don't remount UMSDOS root volumes:
if [ ! "`mount | head -1 | cut -d ' ' -f 5`" = "umsdos" ]; then
  mount -n -o remount,ro /
fi

# root device remains mounted
# mount dos file systems RW
echo "Writing RAID read-only boot FLAG(s)."
```

```
mount -n /dosa
mount -n /dosc
# create raid mounted RO flag in duplicate
# containing the shutdown status of the raid array
echo ${RAIDSTATUS} > /dosa/raidboot/raidstat.ro
echo ${RAIDSTATUS} > /dosc/raidboot/raidstat.ro

umount -n /dosa
umount -n /dosc

# Stop all the raid arrays (except root)
echo "Stopping raid"
mdstop -a

# See if this is a powerfail situation.
if [ -f /etc/power_is_failing ]; then
  echo "Turning off UPS, bye."
  /sbin/powerd -q
  exit 1
fi

# Now halt or reboot.
echo "$message"
[ ! -f /etc/fastboot ] && echo "On the next boot fsck will be FORCED."
$command -f
```

12. Appendix E. - Gadi's raid stop patch for the linux kernel

```
--- linux/drivers/block/md.c.old        Fri Nov 21 13:37:11 1997
+++ linux/drivers/block/md.c    Sat Dec  6 13:34:28 1997
@@ -622,8 +622,13 @@
       return do_md_run (minor, (int) arg);

     case STOP_MD:
-      return do_md_stop (minor, inode);
-
+      err = do_md_stop(minor, inode);
+      if (err) {
+        printk("md: enabling auto mdstop for %s\n",
kdevname(inode->i_rdev));
+        md_dev[minor].auto_mdstop = 1;
+      }
+      return err;
+
     case BLKGETSIZE:   /* Return device size */
     if  (!arg)  return -EINVAL;
     err=verify_area (VERIFY_WRITE, (long *) arg, sizeof(long));
@@ -692,6 +697,10 @@

   sync_dev (inode->i_rdev);
   md_dev[minor].busy--;
+  if (!md_dev[minor].busy && md_dev[minor].auto_mdstop) {
+       do_md_stop(minor, inode);
+       md_dev[minor].auto_mdstop = 0;
+  }
 }

 static int md_read (struct inode *inode, struct file *file,
--- linux/include/linux/md.h~   Fri Nov 21 13:29:14 1997
+++ linux/include/linux/md.h    Fri Nov 21 13:29:14 1997
@@ -260,6 +260,7 @@
   int                 repartition;
   int                 busy;
   int                 nb_dev;
+  int                 auto_mdstop;
```

```
  void                    *private;
};
```

13. Appendix F. - rc.raidown

Copy the following text into the script file **rc.raidown** and save it in **/etc/rc.d**.

```
#! /bin/sh
#
# rc.raidown    This file is executed by init when it goes into runlevel
#               0 (halt) or runlevel 6 (reboot). It saves the status of
#               a root mounted raid array for subsequent re-boot
#
# Version:      1.08    3-25-98 Michael A. Robinton < michael@bizsystems.com >
#
############ Save raid boot and status info ##############
if [ -f /etc/raidboot.conf ]
then
  {
  read RaidBootDevs
  read RaidStatusPath
  read RaidConfigEtc
  } < /etc/raidboot.conf

# you must have issued
#       cat /proc/mdstat | grep md0 >
#               {your boot vol mnt(s)}/{RaidStatusPath}/raidgood.ref
# before linuxrc will execute properly with this info
#
#       capture raid status
  RAIDSTATUS=`/bin/cat /proc/mdstat | /usr/bin/grep md0`
  mkdir /tmp/raid$$
  echo "Writing RAID read-only boot FLAG(s)."
  for Device in ${RaidBootDevs}
  do
# get mount point for raid boot device or use tmp
    RBmount=$( cat /proc/mounts | /usr/bin/grep ${Device} )
    if [ -n ${RBmounts} ]; then
      RBmount=$( echo ${RBmount} | cut -f 2 -d ' ' )
    else
      RBmount="/tmp/raid$$"
      mount ${Device} ${RBmount}
    fi
  if [ -d ${RBmount}/${RaidStatusPath} ]; then
# Create raid mounted RO flag = shutdown status of raid array
    echo ${RAIDSTATUS} > ${RBmount}/${RaidStatusPath}/raidboot.ro
# Don't propagate 'fstab' from ramdisk
    if [ -f /linuxrc ]; then
      FSTAB=
```

```
    else
      FSTAB=fstab
    fi
    pushd /etc
# Save etc files for rescue system
    /bin/tar --ignore-failed-read \
        -cf ${RBmount}/${RaidStatusPath}/raidboot.etc \
        raid*.conf mdtab* ${FSTAB} lilo.conf
    popd
# Create new raidboot.cfg
    {
    /bin/echo ${RaidBootDevs}
    /bin/echo ${RaidStatusPath}
    /bin/echo ${RaidConfigEtc}
    } > ${RBmount}/${RaidStatusPath}/raidboot.cfg
    /bin/umount ${RBmount}
  fi
  done
  rmdir /tmp/raid$$
  echo "Raid boot armed"
fi
################## end raid boot #########################
```

14. Appendix G. - linuxrc theory of operation

This is the complex form of the linuxrc file for root mounted raid. It must be processed with 'bash' or another shell that recognizes shell functions.

The advantage is that it is generic and is not dependent on startup files and parameters located in the **initrd** image.

A **Raid_Conf** parameter passed to **linuxrc** by the kernel at boot from lilo or loadlin contains a pointer to the boot devices and location the of initial 2 raidboot files needed by **linuxrc** (*raidboot.etc and raidboot.cfg placed by the shutdown script*).

raidboot.etc containing the 'tar'ed files:

```
raid*
mdtab*
fstab
lilo.conf                ( if applicable )
```

from the primary system that are transferred to the initrd **/etc**etc directory at startup. With care, this file may be edited if necessary when your system 'really' crashes.

raidboot.cfg contains the name of the boot partition in use and applicable backup(s) as well as the path to the rest of the raid start up file used by **linuxrc**. This file is normally created by the shutdown file and may be created manually if necessary.

raidboot.cfg is of the form, 3 lines - no comments

```
/dev/bootdev1 /dev/bootdev2 [/dev/bootdev3 ... and so on]
raid-status/path
name_of_raidX.conf_file
```

the **raid-status/path** does not include the name of the mountpoint

the **raidX.conf** filename is that one found in /etc and normally used for **ckraid** and **mkraid**.

The following additional files reside on the permanent raid boot partitions. This is usually the same as above, but in emergency situations may be loaded from anywhere they are available, such as a floppy boot disk.

- **raidgood.ref** created by the command cat /proc/mdstat | grep md0 > /{raid_status_path}/raidgood.ref

  See the shutdown scripts for saving this file and the next

- **raidstat.ro** created at each shutdown by the shutdown rc file, saving the exit status of the raid array.

The Linux SCSI programming HOWTO

Heiko Eißfeldt heiko@colossus.escape.de

v1.5, 7 May 1996

This document deals with programming the Linux generic SCSI interface.

1. What's New?

2. Introduction

3. What Is The Generic SCSI Interface?

4. What Are The Requirements To Use It?

- 4.1 Kernel Configuration
- 4.2 Device Files
- 4.3 Device Mapping

5. Programmers Guide

6. Overview Of Device Programming

7. Opening The Device

8. The Header Structure

TOC

9. Inquiry Command Example

10. The Sense Buffer

11. Example Using Sense Buffer

12. Ioctl Functions

13. Driver Defaults

- 13.1 Transfer Lengths
- 13.2 Timeout And Retry Values

14. Obtaining The Scsi Specifications

15. Related Information Sources

- 15.1 HOWTOs and FAQs
- 15.2 Mailing list
- 15.3 Example code

16. Other useful stuff

- 16.1 Device driver writer helpers
- 16.2 Utilities

17. Other SCSI Access Interfaces

18. Final Comments

19. Acknowledgments

Appendix

20. Appendix

21. Error handling

- 21.1 Error status decoding
- 21.2 Status codes
- 21.3 SCSI Sense Keys
- 21.4 Host codes
- 21.5 Driver codes

22. Additional sense codes and additional sense code qualifiers

- 22.1 ASC and ASCQ in lexical order
- 22.2 ASC and ASCQ in numerical order

23. A SCSI command code quick reference

24. Example programs

---

1. What's New?

Newer kernels have changed the interface a bit. This affects a section formerly entitled 'rescanning the devices'. Now it is possible to add/remove SCSI devices on the fly.

Since kernel 1.3.98 some important header files have been moved/split (sg.h and scsi.h).

Some stupid bugs have been replaced by newer ones.

2. Introduction

This document is a guide to the installation and programming of the Linux generic SCSI interface.

It covers kernel prerequisites, device mappings, and basic interaction with devices. Some simple C programming examples are included. General knowledge of the SCSI command set is required; for more information on the SCSI standard and related information, see the appendix to this document.

Note the plain text version of this document lacks cross references (they show up as ``").

3. What Is The Generic SCSI Interface?

The generic SCSI interface has been implemented to provide general SCSI access to (possibly exotic) pieces of SCSI hardware. It was developed by Lawrence Foard (`entropy@world.std.com`) and sponsored by Killy Corporation (see the comments in `scsi/sg.h`).

The interface makes special device handling possible from user level applications (i.e. outside the kernel). Thus, kernel driver development, which is more risky and difficult to debug, is not necessary.

However, if you don't program the driver properly it is possible to hang the SCSI bus, the driver, or the kernel. Therefore, it is important to properly program the generic driver and to first back up all files to avoid losing data. Another useful thing to do before running your programs is to issue a `sync` command to ensure that any buffers are flushed to disk, minimizing data loss if the system hangs.

Another advantage of the generic driver is that as long as the interface itself does not change, all applications are independent of new kernel development. In comparison, other low-level kernel drivers have to be synchronized with other internal kernel changes.

Typically, the generic driver is used to communicate with new SCSI hardware devices that require special user applications to be written to take advantage of their features (e.g. scanners, printers, CD-ROM jukeboxes). The generic interface allows these to be written quickly.

4. What Are The Requirements To Use It?

4.1 Kernel Configuration

You must have a supported SCSI controller, obviously. Furthermore, your kernel must have controller support as well as generic support compiled in. Configuring the Linux kernel (via `make config` under /usr/src/linux) typically looks like the following:

```
 ...
*
* SCSI support
*
SCSI support? (CONFIG_SCSI) [n] y
*
* SCSI support type (disk, tape, CDrom)
*
 ...
Scsi generic support (CONFIG_CHR_DEV_SG) [n] y
*
* SCSI low-level drivers
*
 ...
```

If available, modules can of course be build instead.

4.2 Device Files

The generic SCSI driver uses its own device files, separate from those used by the other SCSI device drivers. They can be generated using the `MAKEDEV` script, typically found in the `/dev` directory. Running `MAKEDEV sg` produces these files:

```
crw-------   1 root     system    21,   0 Aug 20 20:09 /dev/sga
crw-------   1 root     system    21,   1 Aug 20 20:09 /dev/sgb
crw-------   1 root     system    21,   2 Aug 20 20:09 /dev/sgc
crw-------   1 root     system    21,   3 Aug 20 20:09 /dev/sgd
crw-------   1 root     system    21,   4 Aug 20 20:09 /dev/sge
crw-------   1 root     system    21,   5 Aug 20 20:09 /dev/sgf
crw-------   1 root     system    21,   6 Aug 20 20:09 /dev/sgg
crw-------   1 root     system    21,   7 Aug 20 20:09 /dev/sgh
                                    |     |
```

```
                                        major,   minor device numbers
```

TOC

Note that these are character devices for raw access. On some systems these devices may be called `/dev/{sg0,sg1,...}`, depending on your installation, so adjust the following examples accordingly.

4.3 Device Mapping

These device files are dynamically mapped to SCSI id/LUNs on your SCSI bus (LUN = logical unit). The mapping allocates devices consecutively for each LUN of each device on each SCSI bus found at time of the SCSI scan, beginning at the lower LUNs/ids/buses. It starts with the first SCSI controller and continues without interruption with all following controllers. This is currently done in the initialisation of the SCSI driver.

For example, assuming you had three SCSI devices hooked up with ids 1, 3, and 5 on the first SCSI bus (each having one LUN), then the following mapping would be in effect:

```
/dev/sga -> SCSI id 1
/dev/sgb -> SCSI id 3
/dev/sgc -> SCSI id 5
```

If you now add a new device with id 4, then the mapping (after the next rescan) will be:

```
/dev/sga -> SCSI id 1
/dev/sgb -> SCSI id 3
/dev/sgc -> SCSI id 4
/dev/sgd -> SCSI id 5
```

Notice the change for id 5 -- the corresponding device is no longer mapped to `/dev/sgc` but is now under `/dev/sgd`.

Luckily newer kernels allow for changing this order.

Dynamically insert and remove SCSI devices

If a newer kernel and the `/proc` file system is running, a non-busy device can be removed and installed 'on the fly'.

To remove a SCSI device:

```
echo "scsi remove-single-device a b c d" > /proc/scsi/scsi
```

and similar, to add a SCSI device, do

```
echo "scsi add-single-device a b c d" > /proc/scsi/scsi
```

where

```
a == hostadapter id (first one being 0)
b == SCSI channel on hostadapter (first one being 0)
c == ID
d == LUN (first one being 0)
```

So in order to swap the `/dev/sgc` and `/dev/sgd` mappings from the previous example, we could do

```
echo "scsi remove-single-device 0 0 4 0" > /proc/scsi/scsi
echo "scsi remove-single-device 0 0 5 0" > /proc/scsi/scsi
echo "scsi add-single-device 0 0 5 0" > /proc/scsi/scsi
echo "scsi add-single-device 0 0 4 0" > /proc/scsi/scsi
```

since generic devices are mapped in the order of their insertion.

When adding more devices to the scsi bus keep in mind there are limited spare entries for new devices. The memory has been allocated at boot time and has room for 2 more devices.

---

5. Programmers Guide

The following sections are for programmers who want to use the generic SCSI interface in their own applications. An example will be given showing how to access a SCSI device with the INQUIRY and the TESTUNITREADY commands.

When using these code examples, note the following:

- the location of the header files `sg.h` and `scsi.h` has changed in kernel version 1.3.98. Now these files are located at `/usr/src/linux/include/scsi`, which is hopefully linked to `/usr/include/scsi`. Previously they were in `/usr/src/linux/drivers/scsi`. We assume a newer kernel in the following text.
- the generic SCSI interface was extended in kernel version 1.1.68; the examples require at least this version. But please avoid kernel version 1.1.77 up to 1.1.89 and 1.3.52 upto 1.3.56 since they had a broken generic scsi interface.
- the constant DEVICE in the header section describing the accessed device should be set according to your available devices (see section sec-header.

6. Overview Of Device Programming

The header file `include/scsi/sg.h` contains a description of the interface (this is based on kernel version 1.3.98):

```
struct sg_header
 {
  int pack_len;
                  /* length of incoming packet (including header) */
  int reply_len;   /* maximum length of expected reply */
  int pack_id;     /* id number of packet */
  int result;      /* 0==ok, otherwise refer to errno codes */
  unsigned int twelve_byte:1;
               /* Force 12 byte command length for group 6 & 7 commands  */
  unsigned int other_flags:31;                  /* for future use */
  unsigned char sense_buffer[16]; /* used only by reads */
  /* command follows then data for command */
 };
```

This structure describes how a SCSI command is to be processed and has room to hold the results of the execution of the command. The individual structure components will be discussed later in section sec-header.

The general way of exchanging data with the generic driver is as follows: to send a command to an opened generic device, `write()` a block containing these three parts to it:

```
struct sg_header
SCSI command
data to be sent with the command
```

To obtain the result of a command, `read()` a block with this (similar) block structure:

```
struct sg_header
data coming from the device
```

This is a general overview of the process. The following sections describe each of the steps in more detail.

NOTE: Up to recent kernel versions, it is necessary to block the SIGINT signal between the `write()` and the corresponding `read()` call (i.e. via `sigprocmask()`). A return after the `write()` part without any `read()` to fetch the results will block on subsequent accesses. This signal blocking has not yet been included in the example code. So better do not issue SIGINT (a la ˆC) when running these examples.

7. Opening The Device

A generic device has to be opened for read and write access:

```
int fd = open (device_name, O_RDWR);
```

(This is the case even for a read-only hardware device such as a cdrom drive).

We have to perform a `write` to send the command and a `read` to get back any results. In the case of an error the return code is negative (see section sec-errorhandling for a complete list).

8. The Header Structure

The header structure `struct sg_header` serves as a controlling layer between the application and the kernel driver. We now discuss its components in detail.

int pack_len

defines the size of the block written to the driver. This is defined within the kernel for internal use.

int reply_len

defines the size of the block to be accepted at reply. This is defined from the application side.

int pack_id

This field helps to assign replies to requests. The application can supply a unique id for each request. Suppose you have written several commands (say 4) to one device. They may work in parallel, one being the fastest. When getting replies via 4 reads, the replies do not have to have the order of the requests. To identify the correct reply for a given request one can use the `pack_id` field. Typically its value is incremented after each request (and wraps eventually). The maximum amount of outstanding requests is limited by the kernel to SG_MAX_QUEUE (eg 4).

int result

the result code of a `read` or `write` call. This is (sometimes) defined from the generic driver (kernel) side. It is safe to set it to null before the `write` call. These codes are defined in `errno.h` (0 meaning no error).

unsigned int twelve_byte:1

This field is necessary only when using non-standard vendor specific commands (in the range 0xc0 - 0xff). When these commands have a command length of 12 bytes instead of 10, this field has to be set to one before the write call. Other command lengths are not supported. This is defined from the application side.

unsigned char sense_buffer[16]

This buffer is set after a command is completed (after a `read()` call) and contains the SCSI sense code. Some command results have to be read from here (e.g. for `TESTUNITREADY`). Usually it contains just zero bytes. The value in this field is set by the generic driver (kernel) side.

The following example function interfaces directly with the generic kernel driver. It defines the header structure, sends the command via `write`, gets the result via `read` and does some (limited) error checking. The sense buffer data is available in the output buffer (unless a NULL pointer has been given, in which case it's in the input buffer). We will use it in the examples which follow.

Note: Set the value of `DEVICE` to your device descriptor.

```
#define DEVICE "/dev/sgc"

/* Example program to demonstrate the generic SCSI interface */
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <fcntl.h>
#include <errno.h>
#include <scsi/sg.h>
```

TOC

```
#define SCSI_OFF sizeof(struct sg_header)
static unsigned char cmd[SCSI_OFF + 18];      /* SCSI command buffer */
int fd;                                  /* SCSI device/file descriptor */

/* process a complete SCSI cmd. Use the generic SCSI interface. */
static int handle_SCSI_cmd(unsigned cmd_len,         /* command length */
                           unsigned in_size,         /* input data size */
                           unsigned char *i_buff,    /* input buffer */
                           unsigned out_size,        /* output data size */
                           unsigned char *o_buff     /* output buffer */
                           )
{
    int status = 0;
    struct sg_header *sg_hd;

    /* safety checks */
    if (!cmd_len) return -1;            /* need a cmd_len != 0 */
    if (!i_buff) return -1;             /* need an input buffer != NULL */
#ifdef SG_BIG_BUFF
    if (SCSI_OFF + cmd_len + in_size > SG_BIG_BUFF) return -1;
    if (SCSI_OFF + out_size > SG_BIG_BUFF) return -1;
#else
    if (SCSI_OFF + cmd_len + in_size > 4096) return -1;
    if (SCSI_OFF + out_size > 4096) return -1;
#endif

    if (!o_buff) out_size = 0;      /* no output buffer, no output size */

    /* generic SCSI device header construction */
    sg_hd = (struct sg_header *) i_buff;
    sg_hd->reply_len   = SCSI_OFF + out_size;
    sg_hd->twelve_byte = cmd_len == 12;
    sg_hd->result = 0;
#if     0
    sg_hd->pack_len    = SCSI_OFF + cmd_len + in_size; /* not necessary */
    sg_hd->pack_id;     /* not used */
    sg_hd->other_flags; /* not used */
#endif

    /* send command */
    status = write( fd, i_buff, SCSI_OFF + cmd_len + in_size );
    if ( status < 0 || status != SCSI_OFF + cmd_len + in_size ||
                       sg_hd->result ) {
        /* some error happened */
        fprintf( stderr, "write(generic) result = 0x%x cmd = 0x%x\n",
                    sg_hd->result, i_buff[SCSI_OFF] );
        perror("");
        return status;
    }
```

TOC

```
    if (!o_buff) o_buff = i_buff;       /* buffer pointer check */

    /* retrieve result */
    status = read( fd, o_buff, SCSI_OFF + out_size);
    if ( status < 0 || status != SCSI_OFF + out_size || sg_hd->result ) {
        /* some error happened */
        fprintf( stderr, "read(generic) status = 0x%x, result = 0x%x, "
                         "cmd = 0x%x\n",
                         status, sg_hd->result, o_buff[SCSI_OFF] );
        fprintf( stderr, "read(generic) sense "
                "%x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x\n",
                sg_hd->sense_buffer[0],         sg_hd->sense_buffer[1],
                sg_hd->sense_buffer[2],         sg_hd->sense_buffer[3],
                sg_hd->sense_buffer[4],         sg_hd->sense_buffer[5],
                sg_hd->sense_buffer[6],         sg_hd->sense_buffer[7],
                sg_hd->sense_buffer[8],         sg_hd->sense_buffer[9],
                sg_hd->sense_buffer[10],        sg_hd->sense_buffer[11],
                sg_hd->sense_buffer[12],        sg_hd->sense_buffer[13],
                sg_hd->sense_buffer[14],        sg_hd->sense_buffer[15]);
        if (status < 0)
            perror("");
    }
    /* Look if we got what we expected to get */
    if (status == SCSI_OFF + out_size) status = 0; /* got them all */

    return status;  /* 0 means no error */
}
```

While this may look somewhat complex at first appearance, most of the code is for error checking and reporting (which is useful even after the code is working).

`Handle_SCSI_cmd` has a generalized form for all SCSI commands types, falling into each of these categories:

```
       Data Mode                | Example Command
==================================================
neither input nor output data | test unit ready
 no input data, output data   | inquiry, read
 input data, no output data   | mode select, write
   input data, output data    | mode sense
```

9. Inquiry Command Example

One of the most basic SCSI commands is the `INQUIRY` command, used to identify the type and make of the device. Here is the definition from the SCSI-2 specification (for details refer to the SCSI-2 standard).

Table 44: INQUIRY Command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code (12h)							
1	Logical Unit Number			Reserved				EVPD
2	Page Code							
3	Reserved							
4	Allocation Length							
5	Control							

The output data are as follows:

Table 45: Standard INQUIRY Data Format

Bit / Byte	7	6	5	4	3	2	1	0
0	Peripheral Qualifier			Peripheral Device Type				
1	RMB	Device-Type Modifier						
2	ISO Version			ECMA Version		ANSI-Approved Version		
3	AENC	TrmIOP		Reserved		Response Data Format		
4	Additional Length (n-4)							
5	Reserved							
6	Reserved							
7	RelAdr	WBus32	WBus16	Sync	Linked	Reserved	CmdQue	SftRe
8	(MSB)							
- -	---	Vendor Identification						---
15								(LSB)

```
| 16  | (MSB)
|- - -+---                        Product Identification
| 31  |                                                          (LSB) |
|-----+-------------------------------------------------------------------|
| 32  | (MSB)                                                             |
|- - -+---                        Product Revision Level                ---|
| 35  |                                                          (LSB) |
|-----+-------------------------------------------------------------------|
| 36  |                                                                   |
|- - -+---                        Vendor Specific                       ---|
| 55  |                                                                   |
|-----+-------------------------------------------------------------------|
| 56  |                                                                   |
|- - -+---                        Reserved                              ---|
| 95  |                                                                   |
|=====+===================================================================|
|     |                       Vendor-Specific Parameters                  |
|=====+===================================================================|
| 96  |                                                                   |
|- - -+---                        Vendor Specific                       ---|
| n   |                                                                   |
+=====================================================================+
```

The next example uses the low-level function `handle_SCSI_cmd` to perform the Inquiry SCSI command.

We first append the command block to the generic header, then call `handle_SCSI_cmd`. Note that the output buffer size argument for the `handle_SCSI_cmd` call excludes the generic header size. After command completion the output buffer contains the requested data, unless an error occurred.

```
#define INQUIRY_CMD     0x12
#define INQUIRY_CMDLEN  6
#define INQUIRY_REPLY_LEN 96
#define INQUIRY_VENDOR  8       /* Offset in reply data to vendor name */

/* request vendor brand and model */
static unsigned char *Inquiry ( void )
{
  unsigned char Inqbuffer[ SCSI_OFF + INQUIRY_REPLY_LEN ];
  unsigned char cmdblk [ INQUIRY_CMDLEN ] =
      { INQUIRY_CMD,  /* command */
                  0,  /* lun/reserved */
                  0,  /* page code */
                  0,  /* reserved */
  INQUIRY_REPLY_LEN,  /* allocation length */
                  0 };/* reserved/flag/link */

  memcpy( cmd + SCSI_OFF, cmdblk, sizeof(cmdblk) );

  /*
   * +------------------+
   * | struct sg_header | <- cmd
   * +------------------+
   * | copy of cmdblk   | <- cmd + SCSI_OFF
   * +------------------+
```

```
     */

    if (handle_SCSI_cmd(sizeof(cmdblk), 0, cmd,
                        sizeof(Inqbuffer) - SCSI_OFF, Inqbuffer )) {
        fprintf( stderr, "Inquiry failed\n" );
        exit(2);
    }
    return (Inqbuffer + SCSI_OFF);
}
```

The example above follows this structure. The Inquiry function copies its command block behind the generic header (given by `SCSI_OFF`). Input data is not present for this command. `Handle_SCSI_cmd` will define the header structure. We can now implement the function `main` to complete this working example program.

```
void main( void )
{
  fd = open(DEVICE, O_RDWR);
  if (fd < 0) {
    fprintf( stderr, "Need read/write permissions for "DEVICE".\n" );
    exit(1);
  }

  /* print some fields of the Inquiry result */
  printf( "%s\n", Inquiry() + INQUIRY_VENDOR );
}
```

We first open the device, check for errors, and then call the higher level subroutine. Then we print the results in human readable format including the vendor, product, and revision.

Note: There is more information in the Inquiry result than this little program gives. You may want to extend the program to give device type, ANSI version etc. The device type is of special importance, since it determines the mandatory and optional command sets for this device. If you don't want to program it yourself, you may want to use the scsiinfo program from Eric Youngdale, which requests nearly all information about an SCSI device. Look at tsx-11.mit.edu in pub/Linux/ALPHA/scsi.

---

10. The Sense Buffer

Commands with no output data can give status information via the sense buffer (which is part of the header structure). Sense data is available when the previous command has terminated with a CHECK CONDITION status. In this case the kernel automatically retrieves the sense data via a REQUEST SENSE command. Its structure is:

```
+=====-========-========-========-========-========-========-========-========+
|  Bit|   7    |   6    |   5    |   4    |   3    |   2    |   1    |   0    |
|Byte |        |        |        |        |        |        |        |        |
|=====+========+=================================================================|
| 0   | Valid  |                  Error Code (70h or 71h)                      |
|-----+-------------------------------------------------------------------------|
| 1   |                          Segment Number                                 |
|-----+-------------------------------------------------------------------------|
| 2   |Filemark|  EOM   |  ILI   |Reserved|         Sense Key                 |
|-----+-------------------------------------------------------------------------|
| 3   | (MSB)                                                                   |
|- - -+---                     Information                                  ---|
| 6   |                                                                  (LSB)  |
|-----+-------------------------------------------------------------------------|
| 7   |                          Additional Sense Length (n-7)                  |
|-----+-------------------------------------------------------------------------|
| 8   | (MSB)                                                                   |
|- - -+---                     Command-Specific Information                 ---|
| 11  |                                                                  (LSB)  |
|-----+-------------------------------------------------------------------------|
| 12  |                          Additional Sense Code                          |
|-----+-------------------------------------------------------------------------|
| 13  |                          Additional Sense Code Qualifier                |
|-----+-------------------------------------------------------------------------|
| 14  |                          Field Replaceable Unit Code                    |
|-----+-------------------------------------------------------------------------|
| 15  |  SKSV  |                                                                |
|- - -+------------            Sense-Key Specific                           ---|
| 17  |                                                                         |
|-----+-------------------------------------------------------------------------|
| 18  |                                                                         |
|- - -+---                     Additional Sense Bytes                       ---|
| n   |                                                                         |
+=========================================================================+
```

Note: The most useful fields are Sense Key (see section sec-sensekeys), Additional Sense Code and Additional Sense Code Qualifier (see section sec-sensecodes). The latter two are used combined as a pair.

11. Example Using Sense Buffer

Here we will use the TEST UNIT READY command to check whether media is loaded into our device. The header declarations and function `handle_SCSI_cmd` from the inquiry example will be needed as well.

Table 73: TEST UNIT READY Command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code (00h)							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							
5	Control							

Here is the function which implements it:

```
#define TESTUNITREADY_CMD 0
#define TESTUNITREADY_CMDLEN 6

#define ADD_SENSECODE 12
#define ADD_SC_QUALIFIER 13
#define NO_MEDIA_SC 0x3a
#define NO_MEDIA_SCQ 0x00

int TestForMedium ( void )
{
  /* request READY status */
  static unsigned char cmdblk [TESTUNITREADY_CMDLEN] = {
      TESTUNITREADY_CMD, /* command */
                      0, /* lun/reserved */
                      0, /* reserved */
                      0, /* reserved */
                      0, /* reserved */
                      0};/* control */

  memcpy( cmd + SCSI_OFF, cmdblk, sizeof(cmdblk) );

  /*
   * +------------------+
   * | struct sg_header | <- cmd
   * +------------------+
```

```
     * | copy of cmdblk   | <- cmd + SCSI_OFF
     * +------------------+
     */

    if (handle_SCSI_cmd(sizeof(cmdblk), 0, cmd,
                            0, NULL)) {
        fprintf (stderr, "Test unit ready failed\n");
        exit(2);
    }

    return
     *(((struct sg_header*)cmd)->sense_buffer +ADD_SENSECODE) !=
                                                    NO_MEDIA_SC ||
     *(((struct sg_header*)cmd)->sense_buffer +ADD_SC_QUALIFIER) !=
                                                    NO_MEDIA_SCQ;
}
```

Combined with this `main` function we can do the check.

```
void main( void )
{
  fd = open(DEVICE, O_RDWR);
  if (fd < 0) {
    fprintf( stderr, "Need read/write permissions for "DEVICE".\n" );
    exit(1);
  }

  /* look if medium is loaded */

  if (!TestForMedium()) {
    printf("device is unloaded\n");
  } else {
    printf("device is loaded\n");
  }
}
```

The file `generic_demo.c` from the appendix contains both examples.

12. Ioctl Functions

There are two ioctl functions available:

- `ioctl(fd, SG_SET_TIMEOUT, &Timeout);` sets the timeout value to `Timeout` * 10 milliseconds. `Timeout` has to be declared as int.
- `ioctl(fd, SG_GET_TIMEOUT, &Timeout);` gets the current timeout value. `Timeout` has to be declared as int.

13. Driver Defaults

13.1 Transfer Lengths

Currently (at least up to kernel version 1.1.68) input and output sizes have to be less than or equal than 4096 bytes unless the kernel has been compiled with `SG_BIG_BUFF` defined, if which case it is limited to `SG_BIG_BUFF` (e.g. 32768) bytes. These sizes include the generic header as well as the command block on input. `SG_BIG_BUFF` can be safely increased upto (131072 - 512). To take advantage of this, a new kernel has to be compiled and booted, of course.

13.2 Timeout And Retry Values

The default timeout value is set to one minute (`Timeout` = 6000). It can be changed through an ioctl call (see section sec-ioctl). The default number of retries is one.

14. Obtaining The Scsi Specifications

There are standards entitled SCSI-1 and SCSI-2 (and possibly soon SCSI-3). The standards are mostly upward compatible.

The SCSI-1 standard is (in the author's opinion) mostly obsolete, and SCSI-2 is the most widely used. SCSI-3 is very new and very expensive. These standardized command sets specify mandatory and optional commands for SCSI manufacturers and should be preferred over the vendor specific command extensions which are not standardized and for which programming information is seldom available. Of course sometimes there is no alternative to these extensions.

Electronic copies of the latest drafts are available via anonymous ftp from:

- ftp.cs.tulane.edu:pub/scsi
- ftp.symbios.com:/pub/standards
- ftp.cs.uni-sb.de:/pub/misc/doc/scsi

(I got my SCSI specification from the Yggdrasil Linux CD-ROM in the directory /usr/doc/scsi-2 and /usr/doc/scsi-1).

The SCSI FAQ also lists the following sources of printed information:

The SCSI specification: Available from:

```
        Global Engineering Documents
        15 Inverness Way East
        Englewood Co  80112-5704
        (800) 854-7179
          SCSI-1: X3.131-1986
          SCSI-2: X3.131-199x
          SCSI-3 X3T9.2/91-010R4 Working Draft

(Global Engineering Documentation in Irvine, CA (714)261-1455??)

SCSI-1: Doc \# X3.131-1986 from ANSI, 1430 Broadway, NY, NY 10018

IN-DEPTH EXPLORATION OF SCSI can be obtained from
Solution Technology, Attn: SCSI Publications, POB 104, Boulder Creek,
CA 95006, (408)338-4285, FAX (408)338-4374

THE SCSI ENCYLOPEDIA and the SCSI BENCH REFERENCE can be obtained from
ENDL Publishing, 14426 Black Walnut Ct., Saratoga, CA 95090,
(408)867-6642, FAX (408)867-2115

SCSI: UNDERSTANDING THE SMALL COMPUTER SYSTEM INTERFACE was published
by Prentice-Hall, ISBN 0-13-796855-8
```

15. Related Information Sources

15.1 HOWTOs and FAQs

The Linux **SCSI-HOWTO** by Drew Eckhardt covers all supported SCSI controllers as well as device specific questions. A lot of troubleshooting hints are given. It is available from sunsite.unc.edu in /pub/Linux/docs/LDP and its mirror sites.

General questions about SCSI are answered in the **SCSI-FAQ** from the newsgroup Comp.Periphs.Scsi (available on tsx-11 in pub/linux/ALPHA/scsi and mirror sites).

15.2 Mailing list

There is a **mailing list** for bug reports and questions regarding SCSI development under Linux. To join, send email to `majordomo@vger.rutgers.edu` with the line `subscribe linux-scsi` in the body of the message. Messages should be posted to `linux-scsi@vger.rutgers.edu`. Help text can be requested by sending the message line "help" to `majordomo@vger.rutgers.edu`.

15.3 Example code

sunsite.unc.edu: apps/graphics/hpscanpbm-0.3a.tar.gz

This package handles a HP scanjet scanner through the generic interface.

tsx-11.mit.edu: BETA/cdrom/private/mkisofs/cdwrite-1.3.tar.gz

The cdwrite package uses the generic interface to write a cd image to a cd writer.

sunsite.unc.edu: apps/sound/cds/cdda2wav*.src.tar.gz

A shameless plug for my own application, which copies audio cd tracks into wav files.

16. Other useful stuff

Things that may come in handy. I don't have no idea if there are newer or better versions around. Feedback is welcome.

16.1 Device driver writer helpers

These documents can be found at the sunsite.unc.edu ftp server and its mirrors.

/pub/Linux/docs/kernel/kernel-hackers-guide

The LDP kernel hackers guide. May be a bit outdated, but covers the most fundamental things.

/pub/Linux/docs/kernel/drivers.doc.z

This document covers writing character drivers.

/pub/Linux/docs/kernel/tutorial.doc.z

Tutorial on writing a character device driver with code.

/pub/Linux/docs/kernel/scsi.paper.tar.gz

A Latex document describing howto write a SCSI driver.

/pub/Linux/docs/hardware/DEVICES

A list of device majors and minors used by Linux.

16.2 Utilities

tsx-11.mit.edu: ALPHA/scsi/scsiinfo*.tar.gz

Program to query a scsi device for operating parameters, defect lists, etc. An X-based interface is available which requires you have Tk/Tcl/wish installed. With the X-based interface you can easily alter the settings on the drive.

tsx-11.mit.edu: ALPHA/kdebug

A gdb extension for kernel debugging.

17. Other SCSI Access Interfaces

In Linux there is also another SCSI access method via SCSI_IOCTL_SEND_COMMAND ioctl calls, which is deprecated. Special tools like 'scsiinfo' utilize it.

There are some other similar interfaces in use in the un*x world, but not available for Linux:

1. CAM (Common Access Method) developed by Future Domain and other SCSI vendors. Linux has little support for a SCSI CAM system yet (mainly for booting from hard disk). CAM even supports target mode, so one could disguise ones computer as a peripheral hardware device (e.g. for a small SCSI net).
2. ASPI (Advanced SCSI Programming Interface) developed by Adaptec. This is the de facto standard for MS-DOS machines.

There are other application interfaces from SCO(TM), NeXT(TM), Silicon Graphics(TM) and SUN(TM) as well.

18. Final Comments

The generic SCSI interface bridges the gap between user applications and specific devices. But rather than bloating a lot of programs with similar sets of low-level functions, it would be more desirable to have a shared library with a generalized set of low-level functions for a particular purpose. The main goal should be to have independent layers of interfaces. A good design would separate an application into low-level and hardware independent routines. The low-level routines could be put into a shared library and made available for all applications. Here, standardized interfaces should be followed as much as possible before making new ones.

By now you should know more than I do about the Linux generic SCSI interface. So you can start developing powerful applications for the benefit of the global Linux community now...

19. Acknowledgments

Special thanks go to Jeff Tranter for proofreading and enhancing the text considerably as well as to Carlos Puchol for useful comments. Drew Eckhardt's and Eric Youngdale's help on my first (dumb) questions about the use of this interface has been appreciated.

20. Appendix

21. Error handling

The functions `open`, `ioctl`, `write` and `read` can report errors. In this case their return value is -1 and the global variable errno is set to the error number. The errno values are defined in `/usr/include/errno.h`. Possible values are:

```
Function | Error         | Description
=========|===============|=============================================
open     | ENXIO         | not a valid device
         | EACCES        | access mode is not read/write (O_RDWR)
         | EBUSY         | device was requested for nonblocking access,
         |               | but is busy now.
         | ERESTARTSYS   | this indicates an internal error. Try to
         |               | make it reproducible and inform the SCSI
         |               | channel (for details on bug reporting
         |               | see Drew Eckhardts SCSI-HOWTO).
ioctl    | ENXIO         | not a valid device
read     | EAGAIN        | the device would block. Try again later.
         | ERESTARTSYS   | this indicates an internal error. Try to
         |               | make it reproducible and inform the SCSI
         |               | channel (for details on bug reporting
         |               | see Drew Eckhardts SCSI-HOWTO).
write    | EIO           | the length is too small (smaller than the
         |               | generic header struct). Caution: Currently
         |               | there is no overlength checking.
         | EAGAIN        | the device would block. Try again later.
         | ENOMEM        | memory required for this request could not be
         |               | allocated. Try later again unless you
         |               | exceeded the maximum transfer size (see above)
select   |               | none
close    |               | none
```

For read/write positive return values indicate as usual the amount of bytes that have been successfully transferred. This should equal the amount you requested.

21.1 Error status decoding

Furthermore a detailed reporting is done via the kernels `hd_status` and the devices `sense_buffer` (see section sec-sensebuff) both from the generic header structure.

The meaning of `hd_status` can be found in `drivers/scsi/scsi.h`: This `unsigned int` is composed out of different parts:

```
  lsb   |    ...    |    ...    | msb
=======|===========|===========|============
status | sense key | host code | driver byte
```

These macros from `drivers/scsi/scsi.h` are available, but unfortunately cannot be easily used due to weird header file interdependencies. This has to be cleaned.

```
          Macro          | Description
```

```
=======================|=============================================
status_byte(hd_status) | The SCSI device status. See section Status codes
msg_byte(hd_status)    | From the device. See section SCSI sense keys
host_byte(hd_status)   | From the kernel. See section Hostcodes
driver_byte(hd_status) | From the kernel. See section midlevel codes
```

21.2 Status codes

The following status codes from the SCSI device (defined in `scsi/scsi.h`) are available.

```
Value | Symbol
======|====================
0x00  | GOOD
0x01  | CHECK_CONDITION
0x02  | CONDITION_GOOD
0x04  | BUSY
0x08  | INTERMEDIATE_GOOD
0x0a  | INTERMEDIATE_C_GOOD
0x0c  | RESERVATION_CONFLICT
```

Note that these symbol values have been **shifted right once**. When the status is CHECK_CONDITION, the sense data in the sense buffer is valid (check especially the additional sense code and additional sense code qualifier).

These values carry the meaning from the SCSI-2 specification:

```
                           Table 27: Status Byte Code
+=================================-==============================+
|       Bits of Status Byte       |  Status                      |
|  7   6   5   4   3   2   1   0  |                              |
|---------------------------------+------------------------------|
|  R   R   0   0   0   0   0   R  |  GOOD                        |
|  R   R   0   0   0   0   1   R  |  CHECK CONDITION             |
|  R   R   0   0   0   1   0   R  |  CONDITION MET               |
|  R   R   0   0   1   0   0   R  |  BUSY                        |
|  R   R   0   1   0   0   0   R  |  INTERMEDIATE                |
|  R   R   0   1   0   1   0   R  |  INTERMEDIATE-CONDITION MET  |
|  R   R   0   1   1   0   0   R  |  RESERVATION CONFLICT        |
|  R   R   1   0   0   0   1   R  |  COMMAND TERMINATED          |
|  R   R   1   0   1   0   0   R  |  QUEUE FULL                  |
|                                 |                              |
|       All Other Codes           |  Reserved                    |
|-----------------------------------------------------------------|
|  Key: R = Reserved bit                                          |
+=================================================================+

A definition of the status byte codes is given below.

GOOD.  This status indicates that the target has successfully completed the
command.

CHECK CONDITION.  This status indicates that a contingent allegiance condition
has occurred (see 6.6).
```

TOC

```
CONDITION MET.  This status or INTERMEDIATE-CONDITION MET is returned when
the requested operation is satisfied (see the SEARCH DATA and PRE-FETCH
commands).

BUSY.  This status indicates that the target is busy.  This status shall be
returned whenever a target is unable to accept a command from an otherwise
acceptable initiator (i.e., no reservation conflicts).  The recommended
initiator recovery action is to issue the command again at a later time.

INTERMEDIATE.  This status or INTERMEDIATE-CONDITION MET shall be returned for
every successfully completed command in a series of linked commands (except
the last command), unless the command is terminated with CHECK CONDITION,
RESERVATION CONFLICT, or COMMAND TERMINATED status.  If INTERMEDIATE or
INTERMEDIATE-CONDITION MET status is not returned, the series of linked
commands is terminated and the I/O process is ended.

INTERMEDIATE-CONDITION MET.  This status is the combination of the CONDITION
MET and INTERMEDIATE statuses.

RESERVATION CONFLICT.  This status shall be returned whenever an initiator
attempts to access a logical unit or an extent within a logical unit that is
reserved with a conflicting reservation type for another SCSI device (see the
RESERVE and RESERVE UNIT commands).  The recommended initiator recovery action
is to issue the command again at a later time.

COMMAND TERMINATED.  This status shall be returned whenever the target
terminates the current I/O process after receiving a TERMINATE I/O PROCESS
message (see 5.6.22).  This status also indicates that a contingent allegiance
condition has occurred (see 6.6).

QUEUE FULL.  This status shall be implemented if tagged queuing is
implemented.  This status is returned when a SIMPLE QUEUE TAG, ORDERED QUEUE
TAG, or HEAD OF QUEUE TAG message is received and the command queue is full.
The I/O process is not placed in the command queue.
```

21.3 SCSI Sense Keys

These kernel symbols (from `scsi/scsi.h`) are predefined:

```
Value | Symbol
======|================
0x00  | NO_SENSE
0x01  | RECOVERED_ERROR
0x02  | NOT_READY
0x03  | MEDIUM_ERROR
0x04  | HARDWARE_ERROR
0x05  | ILLEGAL_REQUEST
0x06  | UNIT_ATTENTION
0x07  | DATA_PROTECT
0x08  | BLANK_CHECK
0x0a  | COPY_ABORTED
0x0b  | ABORTED_COMMAND
```

```
0x0d  | VOLUME_OVERFLOW
0x0e  | MISCOMPARE
```

A verbatim list from the SCSI-2 doc follows (from section 7.2.14.3):

Table 69: Sense Key (0h-7h) Descriptions

Sense Key	Description
0h	NO SENSE. Indicates that there is no specific sense key information to be reported for the designated logical unit. This would be the case for a successful command or a command that received CHECK CONDITION or COMMAND TERMINATED status because one of the filemark, EOM, or ILI bits is set to one.
1h	RECOVERED ERROR. Indicates that the last command completed successfully with some recovery action performed by the target. Details may be determinable by examining the additional sense bytes and the information field. When multiple recovered errors occur during one command, the choice of which error to report (first, last, most severe, etc.) is device specific.
2h	NOT READY. Indicates that the logical unit addressed cannot be accessed. Operator intervention may be required to correct this condition.
3h	MEDIUM ERROR. Indicates that the command terminated with a non-recovered error condition that was probably caused by a flaw in the medium or an error in the recorded data. This sense key may also be returned if the target is unable to distinguish between a flaw in the medium and a specific hardware failure (sense key 4h).
4h	HARDWARE ERROR. Indicates that the target detected a non-recoverable hardware failure (for example, controller failure, device failure, parity error, etc.) while performing the command or during a self test.
5h	ILLEGAL REQUEST. Indicates that there was an illegal parameter in the command descriptor block or in the additional parameters supplied as data for some commands (FORMAT UNIT, SEARCH DATA, etc.). If the target detects an invalid parameter in the command descriptor block, then it shall terminate the command without altering the medium. If the target detects an invalid parameter in the additional parameters supplied as data, then the target may have already altered the medium. This sense key may also indicate that an invalid IDENTIFY message was received (5.6.7).
6h	UNIT ATTENTION. Indicates that the removable medium may have been changed or the target has been reset. See 6.9 for more detailed information about the unit attention condition.
7h	DATA PROTECT. Indicates that a command that reads or writes the medium was attempted on a block that is protected from this

TOC

```
|        |  operation.  The read or write operation is not performed.
+=========================================================================+
```

Table 70: Sense Key (8h-Fh) Descriptions

```
+========-=========================================================================+
| Sense  |  Description                                                            |
|  Key   |                                                                         |
|--------+-------------------------------------------------------------------------|
|   8h   |  BLANK CHECK.  Indicates that a write-once device or a sequential-      |
|        |  access device encountered blank medium or format-defined end-of-       |
|        |  data indication while reading or a write-once device encountered a|
|        |  non-blank medium while writing.                                        |
|--------+-------------------------------------------------------------------------|
|   9h   |  Vendor Specific.  This sense key is available for reporting vendor|
|        |  specific conditions.                                                   |
|--------+-------------------------------------------------------------------------|
|   Ah   |  COPY ABORTED.  Indicates a COPY, COMPARE, or COPY AND VERIFY           |
|        |  command was aborted due to an error condition on the source            |
|        |  device, the destination device, or both.  (See 7.2.3.2 for            |
|        |  additional information about this sense key.)                          |
|--------+-------------------------------------------------------------------------|
|   Bh   |  ABORTED COMMAND.  Indicates that the target aborted the command.       |
|        |  The initiator may be able to recover by trying the command again.      |
|--------+-------------------------------------------------------------------------|
|   Ch   |  EQUAL.  Indicates a SEARCH DATA command has satisfied an equal         |
|        |  comparison.                                                            |
|--------+-------------------------------------------------------------------------|
|   Dh   |  VOLUME OVERFLOW.  Indicates that a buffered peripheral device has      |
|        |  reached the end-of-partition and data may remain in the buffer         |
|        |  that has not been written to the medium.  A RECOVER BUFFERED DATA      |
|        |  command(s) may be issued to read the unwritten data from the           |
|        |  buffer.                                                                |
|--------+-------------------------------------------------------------------------|
|   Eh   |  MISCOMPARE.  Indicates that the source data did not match the data|
|        |  read from the medium.                                                  |
|--------+-------------------------------------------------------------------------|
|   Fh   |  RESERVED.                                                              |
+=========================================================================+
```

21.4 Host codes

The following host codes are defined in `drivers/scsi/scsi.h`. They are set by the kernel driver.

```
Value | Symbol          | Description
======|=================|=========================================
0x00  | DID_OK          | No error
0x01  | DID_NO_CONNECT  | Couldn't connect before timeout period
0x02  | DID_BUS_BUSY    | BUS stayed busy through time out period
0x03  | DID_TIME_OUT    | TIMED OUT for other reason
0x04  | DID_BAD_TARGET  | BAD target
0x05  | DID_ABORT       | Told to abort for some other reason
0x06  | DID_PARITY      | Parity error
0x07  | DID_ERROR       | internal error
```

```
0x08  | DID_RESET      | Reset by somebody
0x09  | DID_BAD_INTR   | Got an interrupt we weren't expecting
```

TOC

21.5 Driver codes

The midlevel driver categorizes the returned status from the lowlevel driver based on the sense key from the device. It suggests some actions to be taken such as retry, abort or remap. The routine scsi_done from scsi.c does a very differentiated handling based on host_byte(), status_byte(), msg_byte() and the suggestion. It then sets the driver byte to show what it has done. The driver byte is composed out of two nibbles: the driver status and the suggestion. Each half is composed from the below values being 'or'ed together (found in scsi.h).

```
Value | Symbol         | Description of Driver status
======|================|=========================================
0x00  | DRIVER_OK      | No error
0x01  | DRIVER_BUSY    | not used
0x02  | DRIVER_SOFT    | not used
0x03  | DRIVER_MEDIA   | not used
0x04  | DRIVER_ERROR   | internal driver error
0x05  | DRIVER_INVALID | finished (DID_BAD_TARGET or DID_ABORT)
0x06  | DRIVER_TIMEOUT | finished with timeout
0x07  | DRIVER_HARD    | finished with fatal error
0x08  | DRIVER_SENSE   | had sense information available

Value | Symbol         | Description of suggestion
======|================|=========================================
0x10  | SUGGEST_RETRY  | retry the SCSI request
0x20  | SUGGEST_ABORT  | abort the request
0x30  | SUGGEST_REMAP  | remap the block (not yet implemented)
0x40  | SUGGEST_DIE    | let the kernel panic
0x80  | SUGGEST_SENSE  | get sense information from the device
0xff  | SUGGEST_IS_OK  | nothing to be done
```

22. Additional sense codes and additional sense code qualifiers

When the status of the executed SCSI command is CHECK_CONDITION, sense data is available in the sense buffer. The additional sense code and additional sense code qualifier are contained in that buffer.

From the SCSI-2 specification I include two tables. The first is in lexical, the second in numerical order.

22.1 ASC and ASCQ in lexical order

The following table list gives a list of descriptions and device types they apply to.

```
+=========================================================================+
|           D - DIRECT ACCESS DEVICE                                      |
|           .T - SEQUENTIAL ACCESS DEVICE                                 |
|           . L - PRINTER DEVICE                                          |
|           .  P - PROCESSOR DEVICE                                       |
|           .  .W - WRITE ONCE READ MULTIPLE DEVICE                       |
|           .  . R - READ ONLY (CD-ROM) DEVICE                            |
|           .  .  S - SCANNER DEVICE                                      |
|           .  .  .O - OPTICAL MEMORY DEVICE                              |
|           .  .  . M - MEDIA CHANGER DEVICE                              |
|           .  .  .  C - COMMUNICATION DEVICE                             |
|           .  .  .  .                                                    |
| ASC ASCQ  DTLPWRSOMC  DESCRIPTION                                       |
| --- ----              -------------------------------------------------- |
| 13h  00h  D   W  O    ADDRESS MARK NOT FOUND FOR DATA FIELD             |
| 12h  00h  D   W  O    ADDRESS MARK NOT FOUND FOR ID FIELD               |
| 00h  11h       R      AUDIO PLAY OPERATION IN PROGRESS                  |
| 00h  12h       R      AUDIO PLAY OPERATION PAUSED                       |
| 00h  14h       R      AUDIO PLAY OPERATION STOPPED DUE TO ERROR         |
| 00h  13h       R      AUDIO PLAY OPERATION SUCCESSFULLY COMPLETED       |
| 00h  04h   T    S     BEGINNING-OF-PARTITION/MEDIUM DETECTED            |
| 14h  04h   T          BLOCK SEQUENCE ERROR                              |
| 30h  02h  DT  WR O    CANNOT READ MEDIUM - INCOMPATIBLE FORMAT          |
| 30h  01h  DT  WR O    CANNOT READ MEDIUM - UNKNOWN FORMAT               |
| 52h  00h   T          CARTRIDGE FAULT                                   |
| 3Fh  02h  DTLPWRSOMC  CHANGED OPERATING DEFINITION                      |
| 11h  06h      WR O    CIRC UNRECOVERED ERROR                            |
| 30h  03h  DT          CLEANING CARTRIDGE INSTALLED                      |
| 4Ah  00h  DTLPWRSOMC  COMMAND PHASE ERROR                               |
| 2Ch  00h  DTLPWRSOMC  COMMAND SEQUENCE ERROR                            |
| 2Fh  00h  DTLPWRSOMC  COMMANDS CLEARED BY ANOTHER INITIATOR             |
| 2Bh  00h  DTLPWRSO C  COPY CANNOT EXECUTE SINCE HOST CANNOT DISCONNECT  |
| 41h  00h  D           DATA PATH FAILURE (SHOULD USE 40 NN)              |
| 4Bh  00h  DTLPWRSOMC  DATA PHASE ERROR                                  |
| 11h  07h      W  O    DATA RESYCHRONIZATION ERROR                       |
| 16h  00h  D   W  O    DATA SYNCHRONIZATION MARK ERROR                   |
| 19h  00h  D      O    DEFECT LIST ERROR                                 |
```

TOC

19h	03h	D O	DEFECT LIST ERROR IN GROWN LIST
19h	02h	D O	DEFECT LIST ERROR IN PRIMARY LIST
19h	01h	D O	DEFECT LIST NOT AVAILABLE
1Ch	00h	D O	DEFECT LIST NOT FOUND
32h	01h	D W O	DEFECT LIST UPDATE FAILURE
40h	NNh	DTLPWRSOMC	DIAGNOSTIC FAILURE ON COMPONENT NN (80H-FFH)
63h	00h	R	END OF USER AREA ENCOUNTERED ON THIS TRACK
00h	05h	T S	END-OF-DATA DETECTED
14h	03h	T	END-OF-DATA NOT FOUND
00h	02h	T S	END-OF-PARTITION/MEDIUM DETECTED
51h	00h	T O	ERASE FAILURE
0Ah	00h	DTLPWRSOMC	ERROR LOG OVERFLOW
11h	02h	DT W SO	ERROR TOO LONG TO CORRECT
03h	02h	T	EXCESSIVE WRITE ERRORS
3Bh	07h	L	FAILED TO SENSE BOTTOM-OF-FORM
3Bh	06h	L	FAILED TO SENSE TOP-OF-FORM
00h	01h	T	FILEMARK DETECTED
14h	02h	T	FILEMARK OR SETMARK NOT FOUND
09h	02h	WR O	FOCUS SERVO FAILURE
31h	01h	D L O	FORMAT COMMAND FAILED
58h	00h	O	GENERATION DOES NOT EXIST

Table 71: (continued)

ASC	ASCQ	DTLPWRSOMC	DESCRIPTION
1Ch	02h	D O	GROWN DEFECT LIST NOT FOUND
00h	06h	DTLPWRSOMC	I/O PROCESS TERMINATED
10h	00h	D W O	ID CRC OR ECC ERROR
22h	00h	D	ILLEGAL FUNCTION (SHOULD USE 20 00, 24 00, OR 26 00)
64h	00h	R	ILLEGAL MODE FOR THIS TRACK
28h	01h	M	IMPORT OR EXPORT ELEMENT ACCESSED
30h	00h	DT WR OM	INCOMPATIBLE MEDIUM INSTALLED
11h	08h	T	INCOMPLETE BLOCK READ
48h	00h	DTLPWRSOMC	INITIATOR DETECTED ERROR MESSAGE RECEIVED
3Fh	03h	DTLPWRSOMC	INQUIRY DATA HAS CHANGED
44h	00h	DTLPWRSOMC	INTERNAL TARGET FAILURE
3Dh	00h	DTLPWRSOMC	INVALID BITS IN IDENTIFY MESSAGE
2Ch	02h	S	INVALID COMBINATION OF WINDOWS SPECIFIED
20h	00h	DTLPWRSOMC	INVALID COMMAND OPERATION CODE
21h	01h	M	INVALID ELEMENT ADDRESS
24h	00h	DTLPWRSOMC	INVALID FIELD IN CDB
26h	00h	DTLPWRSOMC	INVALID FIELD IN PARAMETER LIST
49h	00h	DTLPWRSOMC	INVALID MESSAGE ERROR
11h	05h	WR O	L-EC UNCORRECTABLE ERROR
60h	00h	S	LAMP FAILURE
5Bh	02h	DTLPWRSOM	LOG COUNTER AT MAXIMUM
5Bh	00h	DTLPWRSOM	LOG EXCEPTION
5Bh	03h	DTLPWRSOM	LOG LIST CODES EXHAUSTED
2Ah	02h	DTL WRSOMC	LOG PARAMETERS CHANGED
21h	00h	DT WR OM	LOGICAL BLOCK ADDRESS OUT OF RANGE
08h	00h	DTL WRSOMC	LOGICAL UNIT COMMUNICATION FAILURE
08h	02h	DTL WRSOMC	LOGICAL UNIT COMMUNICATION PARITY ERROR

TOC

ASC	ASCQ	DTLPWRSOMC	DESCRIPTION
08h	01h	DTL WRSOMC	LOGICAL UNIT COMMUNICATION TIME-OUT
4Ch	00h	DTLPWRSOMC	LOGICAL UNIT FAILED SELF-CONFIGURATION
3Eh	00h	DTLPWRSOMC	LOGICAL UNIT HAS NOT SELF-CONFIGURED YET
04h	01h	DTLPWRSOMC	LOGICAL UNIT IS IN PROCESS OF BECOMING READY
04h	00h	DTLPWRSOMC	LOGICAL UNIT NOT READY, CAUSE NOT REPORTABLE
04h	04h	DTL O	LOGICAL UNIT NOT READY, FORMAT IN PROGRESS
04h	02h	DTLPWRSOMC	LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED
04h	03h	DTLPWRSOMC	LOGICAL UNIT NOT READY, MANUAL INTERVENTION REQUIRED
25h	00h	DTLPWRSOMC	LOGICAL UNIT NOT SUPPORTED
15h	01h	DTL WRSOM	MECHANICAL POSITIONING ERROR
53h	00h	DTL WRSOM	MEDIA LOAD OR EJECT FAILED
3Bh	0Dh	M	MEDIUM DESTINATION ELEMENT FULL
31h	00h	DT W O	MEDIUM FORMAT CORRUPTED
3Ah	00h	DTL WRSOM	MEDIUM NOT PRESENT
53h	02h	DT WR OM	MEDIUM REMOVAL PREVENTED
3Bh	0Eh	M	MEDIUM SOURCE ELEMENT EMPTY
43h	00h	DTLPWRSOMC	MESSAGE ERROR
3Fh	01h	DTLPWRSOMC	MICROCODE HAS BEEN CHANGED
1Dh	00h	D W O	MISCOMPARE DURING VERIFY OPERATION
11h	0Ah	DT O	MISCORRECTED ERROR
2Ah	01h	DTL WRSOMC	MODE PARAMETERS CHANGED
07h	00h	DTL WRSOM	MULTIPLE PERIPHERAL DEVICES SELECTED
11h	03h	DT W SO	MULTIPLE READ ERRORS
00h	00h	DTLPWRSOMC	NO ADDITIONAL SENSE INFORMATION
00h	15h	R	NO CURRENT AUDIO STATUS TO RETURN
32h	00h	D W O	NO DEFECT SPARE LOCATION AVAILABLE
11h	09h	T	NO GAP FOUND
01h	00h	D W O	NO INDEX/SECTOR SIGNAL
06h	00h	D WR OM	NO REFERENCE POSITION FOUND

Table 71: (continued)

ASC	ASCQ	DTLPWRSOMC	DESCRIPTION
02h	00h	D WR OM	NO SEEK COMPLETE
03h	01h	T	NO WRITE CURRENT
28h	00h	DTLPWRSOMC	NOT READY TO READY TRANSITION, MEDIUM MAY HAVE CHANGED
5Ah	01h	DT WR OM	OPERATOR MEDIUM REMOVAL REQUEST
5Ah	00h	DTLPWRSOM	OPERATOR REQUEST OR STATE CHANGE INPUT (UNSPECIFIED)
5Ah	03h	DT W O	OPERATOR SELECTED WRITE PERMIT
5Ah	02h	DT W O	OPERATOR SELECTED WRITE PROTECT
61h	02h	S	OUT OF FOCUS
4Eh	00h	DTLPWRSOMC	OVERLAPPED COMMANDS ATTEMPTED
2Dh	00h	T	OVERWRITE ERROR ON UPDATE IN PLACE
3Bh	05h	L	PAPER JAM
1Ah	00h	DTLPWRSOMC	PARAMETER LIST LENGTH ERROR
26h	01h	DTLPWRSOMC	PARAMETER NOT SUPPORTED
26h	02h	DTLPWRSOMC	PARAMETER VALUE INVALID
2Ah	00h	DTL WRSOMC	PARAMETERS CHANGED
03h	00h	DTL W SO	PERIPHERAL DEVICE WRITE FAULT
50h	02h	T	POSITION ERROR RELATED TO TIMING
3Bh	0Ch	S	POSITION PAST BEGINNING OF MEDIUM

TOC

3Bh	0Bh	S	POSITION PAST END OF MEDIUM
15h	02h	DT WR O	POSITIONING ERROR DETECTED BY READ OF MEDIUM
29h	00h	DTLPWRSOMC	POWER ON, RESET, OR BUS DEVICE RESET OCCURRED
42h	00h	D	POWER-ON OR SELF-TEST FAILURE (SHOULD USE 40 NN)
1Ch	01h	D O	PRIMARY DEFECT LIST NOT FOUND
40h	00h	D	RAM FAILURE (SHOULD USE 40 NN)
15h	00h	DTL WRSOM	RANDOM POSITIONING ERROR
3Bh	0Ah	S	READ PAST BEGINNING OF MEDIUM
3Bh	09h	S	READ PAST END OF MEDIUM
11h	01h	DT W SO	READ RETRIES EXHAUSTED
14h	01h	DT WR O	RECORD NOT FOUND
14h	00h	DTL WRSO	RECORDED ENTITY NOT FOUND
18h	02h	D WR O	RECOVERED DATA - DATA AUTO-REALLOCATED
18h	05h	D WR O	RECOVERED DATA - RECOMMEND REASSIGNMENT
18h	06h	D WR O	RECOVERED DATA - RECOMMEND REWRITE
17h	05h	D WR O	RECOVERED DATA USING PREVIOUS SECTOR ID
18h	03h	R	RECOVERED DATA WITH CIRC
18h	01h	D WR O	RECOVERED DATA WITH ERROR CORRECTION & RETRIES APPLIED
18h	00h	DT WR O	RECOVERED DATA WITH ERROR CORRECTION APPLIED
18h	04h	R	RECOVERED DATA WITH L-EC
17h	03h	DT WR O	RECOVERED DATA WITH NEGATIVE HEAD OFFSET
17h	00h	DT WRSO	RECOVERED DATA WITH NO ERROR CORRECTION APPLIED
17h	02h	DT WR O	RECOVERED DATA WITH POSITIVE HEAD OFFSET
17h	01h	DT WRSO	RECOVERED DATA WITH RETRIES
17h	04h	WR O	RECOVERED DATA WITH RETRIES AND/OR CIRC APPLIED
17h	06h	D W O	RECOVERED DATA WITHOUT ECC - DATA AUTO-REALLOCATED
17h	07h	D W O	RECOVERED DATA WITHOUT ECC - RECOMMEND REASSIGNMENT
17h	08h	D W O	RECOVERED DATA WITHOUT ECC - RECOMMEND REWRITE
1Eh	00h	D W O	RECOVERED ID WITH ECC CORRECTION
3Bh	08h	T	REPOSITION ERROR
36h	00h	L	RIBBON, INK, OR TONER FAILURE
37h	00h	DTL WRSOMC	ROUNDED PARAMETER
5Ch	00h	D O	RPL STATUS CHANGE
39h	00h	DTL WRSOMC	SAVING PARAMETERS NOT SUPPORTED
62h	00h	S	SCAN HEAD POSITIONING ERROR
47h	00h	DTLPWRSOMC	SCSI PARITY ERROR
54h	00h	P	SCSI TO HOST SYSTEM INTERFACE FAILURE
45h	00h	DTLPWRSOMC	SELECT OR RESELECT FAILURE

Table 71: (concluded)

ASC	ASCQ	DTLPWRSOMC	DESCRIPTION
3Bh	00h	TL	SEQUENTIAL POSITIONING ERROR
00h	03h	T	SETMARK DETECTED
3Bh	04h	L	SLEW FAILURE
09h	03h	WR O	SPINDLE SERVO FAILURE
5Ch	02h	D O	SPINDLES NOT SYNCHRONIZED
5Ch	01h	D O	SPINDLES SYNCHRONIZED
1Bh	00h	DTLPWRSOMC	SYNCHRONOUS DATA TRANSFER ERROR
55h	00h	P	SYSTEM RESOURCE FAILURE
33h	00h	T	TAPE LENGTH ERROR
3Bh	03h	L	TAPE OR ELECTRONIC VERTICAL FORMS UNIT NOT READY

TOC

```
| 3Bh  01h   T          TAPE POSITION ERROR AT BEGINNING-OF-MEDIUM                |
| 3Bh  02h   T          TAPE POSITION ERROR AT END-OF-MEDIUM                      |
| 3Fh  00h  DTLPWRSOMC  TARGET OPERATING CONDITIONS HAVE CHANGED                  |
| 5Bh  01h  DTLPWRSOM   THRESHOLD CONDITION MET                                   |
| 26h  03h  DTLPWRSOMC  THRESHOLD PARAMETERS NOT SUPPORTED                        |
| 2Ch  01h        S     TOO MANY WINDOWS SPECIFIED                                |
| 09h  00h  DT  WR O    TRACK FOLLOWING ERROR                                     |
| 09h  01h      WR O    TRACKING SERVO FAILURE                                    |
| 61h  01h        S     UNABLE TO ACQUIRE VIDEO                                   |
| 57h  00h       R      UNABLE TO RECOVER TABLE-OF-CONTENTS                       |
| 53h  01h   T          UNLOAD TAPE FAILURE                                       |
| 11h  00h  DT  WRSO    UNRECOVERED READ ERROR                                    |
| 11h  04h  D   W  O    UNRECOVERED READ ERROR - AUTO REALLOCATE FAILED           |
| 11h  0Bh  D   W  O    UNRECOVERED READ ERROR - RECOMMEND REASSIGNMENT           |
| 11h  0Ch  D   W  O    UNRECOVERED READ ERROR - RECOMMEND REWRITE THE DATA       |
| 46h  00h  DTLPWRSOMC  UNSUCCESSFUL SOFT RESET                                   |
| 59h  00h         O    UPDATED BLOCK READ                                        |
| 61h  00h        S     VIDEO ACQUISITION ERROR                                   |
| 50h  00h   T          WRITE APPEND ERROR                                        |
| 50h  01h   T          WRITE APPEND POSITION ERROR                               |
| 0Ch  00h   T    S     WRITE ERROR                                               |
| 0Ch  02h  D   W  O    WRITE ERROR - AUTO REALLOCATION FAILED                    |
| 0Ch  01h  D   W  O    WRITE ERROR RECOVERED WITH AUTO REALLOCATION              |
| 27h  00h  DT  W  O    WRITE PROTECTED                                           |
|                                                                                 |
| 80h  XXh     \                                                                  |
| THROUGH       >       VENDOR SPECIFIC.                                          |
| FFh  XX      /                                                                  |
|                                                                                 |
| XXh  80h     \                                                                  |
| THROUGH       >       VENDOR SPECIFIC QUALIFICATION OF STANDARD ASC.            |
| XXh  FFh     /                                                                  |
|                      ALL CODES NOT SHOWN ARE RESERVED.                          |
|---------------------------------------------------------------------------------|
```

22.2 ASC and ASCQ in numerical order

Table 364: ASC and ASCQ Assignments

```
+=================================================================================+
|           D - DIRECT ACCESS DEVICE                                              |
|           .T - SEQUENTIAL ACCESS DEVICE                                         |
|           . L - PRINTER DEVICE                                                  |
|           .  P - PROCESSOR DEVICE                                               |
|           .  .W - WRITE ONCE READ MULTIPLE DEVICE                               |
|           .  . R - READ ONLY (CD-ROM) DEVICE                                    |
|           .  .  S - SCANNER DEVICE                                              |
|           .  .  .O - OPTICAL MEMORY DEVICE                                      |
|           .  .  . M - MEDIA CHANGER DEVICE                                      |
|           .  .  .  C - COMMUNICATION DEVICE                                     |
|           .  .  .  .                                                            |
| ASC ASCQ  DTLPWRSOMC  DESCRIPTION                                               |
```

TOC

ASC	ASCQ	DTLPWRSOMC	DESCRIPTION
00	00	DTLPWRSOMC	NO ADDITIONAL SENSE INFORMATION
00	01	T	FILEMARK DETECTED
00	02	T S	END-OF-PARTITION/MEDIUM DETECTED
00	03	T	SETMARK DETECTED
00	04	T S	BEGINNING-OF-PARTITION/MEDIUM DETECTED
00	05	T S	END-OF-DATA DETECTED
00	06	DTLPWRSOMC	I/O PROCESS TERMINATED
00	11	R	AUDIO PLAY OPERATION IN PROGRESS
00	12	R	AUDIO PLAY OPERATION PAUSED
00	13	R	AUDIO PLAY OPERATION SUCCESSFULLY COMPLETED
00	14	R	AUDIO PLAY OPERATION STOPPED DUE TO ERROR
00	15	R	NO CURRENT AUDIO STATUS TO RETURN
01	00	DW O	NO INDEX/SECTOR SIGNAL
02	00	DWR OM	NO SEEK COMPLETE
03	00	DTL W SO	PERIPHERAL DEVICE WRITE FAULT
03	01	T	NO WRITE CURRENT
03	02	T	EXCESSIVE WRITE ERRORS
04	00	DTLPWRSOMC	LOGICAL UNIT NOT READY, CAUSE NOT REPORTABLE
04	01	DTLPWRSOMC	LOGICAL UNIT IS IN PROCESS OF BECOMING READY
04	02	DTLPWRSOMC	LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED
04	03	DTLPWRSOMC	LOGICAL UNIT NOT READY, MANUAL INTERVENTION REQUIRED
04	04	DTL O	LOGICAL UNIT NOT READY, FORMAT IN PROGRESS
05	00	DTL WRSOMC	LOGICAL UNIT DOES NOT RESPOND TO SELECTION
06	00	DWR OM NO	REFERENCE POSITION FOUND
07	00	DTL WRSOM	MULTIPLE PERIPHERAL DEVICES SELECTED
08	00	DTL WRSOMC	LOGICAL UNIT COMMUNICATION FAILURE
08	01	DTL WRSOMC	LOGICAL UNIT COMMUNICATION TIME-OUT
08	02	DTL WRSOMC	LOGICAL UNIT COMMUNICATION PARITY ERROR
09	00	DT WR O	TRACK FOLLOWING ERROR
09	01	WR O	TRA CKING SERVO FAILURE
09	02	WR O	FOC US SERVO FAILURE
09	03	WR O	SPI NDLE SERVO FAILURE

Table 364: (continued)

```
D - DIRECT ACCESS DEVICE
.T - SEQUENTIAL ACCESS DEVICE
. L - PRINTER DEVICE
.  P - PROCESSOR DEVICE
.  .W - WRITE ONCE READ MULTIPLE DEVICE
.  . R - READ ONLY (CD-ROM) DEVICE
.  .  S - SCANNER DEVICE
.  .  .O - OPTICAL MEMORY DEVICE
.  .  . M - MEDIA CHANGER DEVICE
.  .  .  C - COMMUNICATION DEVICE
.  .  .  .
```

ASC	ASCQ	DTLPWRSOMC	DESCRIPTION
0A	00	DTLPWRSOMC	ERROR LOG OVERFLOW
0B	00		
0C	00	T S	WRITE ERROR
0C	01	D W O	WRITE ERROR RECOVERED WITH AUTO REALLOCATION

0C	02	D	W	O	WRITE ERROR - AUTO REALLOCATION FAILED
0D	00				
0E	00				
0F	00				
10	00	D	W	O	ID CRC OR ECC ERROR
11	00	DT	WRSO		UNRECOVERED READ ERROR
11	01	DT	W SO		READ RETRIES EXHAUSTED
11	02	DT	W SO		ERROR TOO LONG TO CORRECT
11	03	DT	W SO		MULTIPLE READ ERRORS
11	04	D	W	O	UNRECOVERED READ ERROR - AUTO REALLOCATE FAILED
11	05		WR	O	L-EC UNCORRECTABLE ERROR
11	06		WR	O	CIRC UNRECOVERED ERROR
11	07		W	O	DATA RESYCHRONIZATION ERROR
11	08	T			INCOMPLETE BLOCK READ
11	09	T			NO GAP FOUND
11	0A	DT		O	MISCORRECTED ERROR
11	0B	D	W	O	UNRECOVERED READ ERROR - RECOMMEND REASSIGNMENT
11	0C	D	W	O	UNRECOVERED READ ERROR - RECOMMEND REWRITE THE DATA
12	00	D	W	O	ADDRESS MARK NOT FOUND FOR ID FIELD
13	00	D	W	O	ADDRESS MARK NOT FOUND FOR DATA FIELD
14	00	DTL	WRSO		RECORDED ENTITY NOT FOUND
14	01	DT	WR	O	RECORD NOT FOUND
14	02	T			FILEMARK OR SETMARK NOT FOUND
14	03	T			END-OF-DATA NOT FOUND
14	04	T			BLOCK SEQUENCE ERROR
15	00	DTL	WRSOM		RANDOM POSITIONING ERROR
15	01	DTL	WRSOM		MECHANICAL POSITIONING ERROR
15	02	DT	WR	O	POSITIONING ERROR DETECTED BY READ OF MEDIUM
16	00	DW		O	DATA SYNCHRONIZATION MARK ERROR
17	00	DT	WRSO		RECOVERED DATA WITH NO ERROR CORRECTION APPLIED
17	01	DT	WRSO		RECOVERED DATA WITH RETRIES
17	02	DT	WR	O	RECOVERED DATA WITH POSITIVE HEAD OFFSET
17	03	DT	WR	O	RECOVERED DATA WITH NEGATIVE HEAD OFFSET
17	04		WR	O	RECOVERED DATA WITH RETRIES AND/OR CIRC APPLIED
17	05	D	WR	O	RECOVERED DATA USING PREVIOUS SECTOR ID
17	06	D	W	O	RECOVERED DATA WITHOUT ECC - DATA AUTO-REALLOCATED
17	07	D	W	O	RECOVERED DATA WITHOUT ECC - RECOMMEND REASSIGNMENT
17	08	D	W	O	RECOVERED DATA WITHOUT ECC - RECOMMEND REWRITE
18	00	DT	WR	O	RECOVERED DATA WITH ERROR CORRECTION APPLIED
18	01	D	WR	O	RECOVERED DATA WITH ERROR CORRECTION & RETRIES APPLIED
18	02	D	WR	O	RECOVERED DATA - DATA AUTO-REALLOCATED
18	03		R		RECOVERED DATA WITH CIRC
18	04		R		RECOVERED DATA WITH LEC
18	05	D	WR	O	RECOVERED DATA - RECOMMEND REASSIGNMENT
18	06	D	WR	O	RECOVERED DATA - RECOMMEND REWRITE

Table 364: (continued)

```
D - DIRECT ACCESS DEVICE
.T - SEQUENTIAL ACCESS DEVICE
. L - PRINTER DEVICE
.  P - PROCESSOR DEVICE
```

```
|            .   .W - WRITE ONCE READ MULTIPLE DEVICE
|            .   . R - READ ONLY (CD-ROM) DEVICE
|            .   .  S - SCANNER DEVICE
|            .   .  .O - OPTICAL MEMORY DEVICE
|            .   .  . M - MEDIA CHANGER DEVICE
|            .   .  .  C - COMMUNICATION DEVICE
|            .   .  .  .
| ASC ASCQ  DTLPWRSOMC  DESCRIPTION
| --- ----              -------------------------------------------------------
|  19  00   D      O    DEFECT LIST ERROR
|  19  01   D      O    DEFECT LIST NOT AVAILABLE
|  19  02   D      O    DEFECT LIST ERROR IN PRIMARY LIST
|  19  03   D      O    DEFECT LIST ERROR IN GROWN LIST
|  1A  00   DTLPWRSOMC  PARAMETER LIST LENGTH ERROR
|  1B  00   DTLPWRSOMC  SYNCHRONOUS DATA TRANSFER ERROR
|  1C  00   D      O    DEFECT LIST NOT FOUND
|  1C  01   D      O    PRIMARY DEFECT LIST NOT FOUND
|  1C  02   D      O    GROWN DEFECT LIST NOT FOUND
|  1D  00   D   W  O    MISCOMPARE DURING VERIFY OPERATION
|  1E  00   D   W  O    RECOVERED ID WITH ECC
|  1F  00
|  20  00   DTLPWRSOMC  INVALID COMMAND OPERATION CODE
|  21  00   DT  WR OM   LOGICAL BLOCK ADDRESS OUT OF RANGE
|  21  01           M   INVALID ELEMENT ADDRESS
|  22  00   D           ILLEGAL FUNCTION (SHOULD USE 20 00, 24 00, OR 26 00)
|  23  00
|  24  00   DTLPWRSOMC  INVALID FIELD IN CDB
|  25  00   DTLPWRSOMC  LOGICAL UNIT NOT SUPPORTED
|  26  00   DTLPWRSOMC  INVALID FIELD IN PARAMETER LIST
|  26  01   DTLPWRSOMC  PARAMETER NOT SUPPORTED
|  26  02   DTLPWRSOMC  PARAMETER VALUE INVALID
|  26  03   DTLPWRSOMC  THRESHOLD PARAMETERS NOT SUPPORTED
|  27  00   DT  W  O    WRITE PROTECTED
|  28  00   DTLPWRSOMC  NOT READY TO READY TRANSITION(MEDIUM MAY HAVE CHANGED)
|  28  01           M   IMPORT OR EXPORT ELEMENT ACCESSED
|  29  00   DTLPWRSOMC  POWER ON, RESET, OR BUS DEVICE RESET OCCURRED
|  2A  00   DTL WRSOMC  PARAMETERS CHANGED
|  2A  01   DTL WRSOMC  MODE PARAMETERS CHANGED
|  2A  02   DTL WRSOMC  LOG PARAMETERS CHANGED
|  2B  00   DTLPWRSO C  COPY CANNOT EXECUTE SINCE HOST CANNOT DISCONNECT
|  2C  00   DTLPWRSOMC  COMMAND SEQUENCE ERROR
|  2C  01         S     TOO MANY WINDOWS SPECIFIED
|  2C  02         S     INVALID COMBINATION OF WINDOWS SPECIFIED
|  2D  00    T          OVERWRITE ERROR ON UPDATE IN PLACE
|  2E  00
|  2F  00   DTLPWRSOMC  COMMANDS CLEARED BY ANOTHER INITIATOR
|  30  00   DT  WR OM   INCOMPATIBLE MEDIUM INSTALLED
|  30  01   DT  WR O    CANNOT READ MEDIUM - UNKNOWN FORMAT
|  30  02   DT  WR O    CANNOT READ MEDIUM - INCOMPATIBLE FORMAT
|  30  03   DT          CLEANING CARTRIDGE INSTALLED
|  31  00   DT  W  O    MEDIUM FORMAT CORRUPTED
|  31  01   D L    O    FORMAT COMMAND FAILED
|  32  00   D   W  O    NO DEFECT SPARE LOCATION AVAILABLE
|  32  01   D   W  O    DEFECT LIST UPDATE FAILURE
```

TOC

```
|  33  00    T          TAPE LENGTH ERROR
|  34  00
|  35  00
|  36  00     L         RIBBON, INK, OR TONER FAILURE
+==============================================================================+
```

Table 364: (continued)

```
+==============================================================================+
|           D - DIRECT ACCESS DEVICE                                           |
|           .T - SEQUENTIAL ACCESS DEVICE                                      |
|           . L - PRINTER DEVICE                                               |
|           .  P - PROCESSOR DEVICE                                            |
|           .  .W - WRITE ONCE READ MULTIPLE DEVICE                            |
|           .  . R - READ ONLY (CD-ROM) DEVICE                                 |
|           .  .  S - SCANNER DEVICE                                           |
|           .  .  .O - OPTICAL MEMORY DEVICE                                   |
|           .  .  . M - MEDIA CHANGER DEVICE                                   |
|           .  .  .  C - COMMUNICATION DEVICE                                  |
|           .  .  .  .                                                         |
| ASC ASCQ  DTLPWRSOMC  DESCRIPTION                                            |
| --- ----              -------------------------------------------------------|
|  37  00   DTL WRSOMC  ROUNDED PARAMETER                                      |
|  38  00                                                                      |
|  39  00   DTL WRSOMC  SAVING PARAMETERS NOT SUPPORTED                        |
|  3A  00   DTL WRSOM   MEDIUM NOT PRESENT                                     |
|  3B  00    TL         SEQUENTIAL POSITIONING ERROR                           |
|  3B  01    T          TAPE POSITION ERROR AT BEGINNING-OF-MEDIUM             |
|  3B  02    T          TAPE POSITION ERROR AT END-OF-MEDIUM                   |
|  3B  03     L         TAPE OR ELECTRONIC VERTICAL FORMS UNIT NOT READY       |
|  3B  04     L         SLEW FAILURE                                           |
|  3B  05     L         PAPER JAM                                              |
|  3B  06     L         FAILED TO SENSE TOP-OF-FORM                            |
|  3B  07     L         FAILED TO SENSE BOTTOM-OF-FORM                         |
|  3B  08    T          REPOSITION ERROR                                       |
|  3B  09         S     READ PAST END OF MEDIUM                                |
|  3B  0A         S     READ PAST BEGINNING OF MEDIUM                          |
|  3B  0B         S     POSITION PAST END OF MEDIUM                            |
|  3B  0C         S     POSITION PAST BEGINNING OF MEDIUM                      |
|  3B  0D           M   MEDIUM DESTINATION ELEMENT FULL                        |
|  3B  0E           M   MEDIUM SOURCE ELEMENT EMPTY                            |
|  3C  00                                                                      |
|  3D  00   DTLPWRSOMC  INVALID BITS IN IDENTIFY MESSAGE                       |
|  3E  00   DTLPWRSOMC  LOGICAL UNIT HAS NOT SELF-CONFIGURED YET               |
|  3F  00   DTLPWRSOMC  TARGET OPERATING CONDITIONS HAVE CHANGED               |
|  3F  01   DTLPWRSOMC  MICROCODE HAS BEEN CHANGED                             |
|  3F  02   DTLPWRSOMC  CHANGED OPERATING DEFINITION                           |
|  3F  03   DTLPWRSOMC  INQUIRY DATA HAS CHANGED                               |
|  40  00   D           RAM FAILURE (SHOULD USE 40 NN)                         |
|  40  NN   DTLPWRSOMC  DIAGNOSTIC FAILURE ON COMPONENT NN (80H-FFH)           |
|  41  00   D           DATA PATH FAILURE (SHOULD USE 40 NN)                   |
|  42  00   D           POWER-ON OR SELF-TEST FAILURE (SHOULD USE 40 NN)       |
|  43  00   DTLPWRSOMC  MESSAGE ERROR                                          |
|  44  00   DTLPWRSOMC  INTERNAL TARGET FAILURE                                |
|  45  00   DTLPWRSOMC  SELECT OR RESELECT FAILURE                             |
```

TOC

46	00	DTLPWRSOMC	UNSUCCESSFUL SOFT RESET
47	00	DTLPWRSOMC	SCSI PARITY ERROR
48	00	DTLPWRSOMC	INITIATOR DETECTED ERROR MESSAGE RECEIVED
49	00	DTLPWRSOMC	INVALID MESSAGE ERROR
4A	00	DTLPWRSOMC	COMMAND PHASE ERROR
4B	00	DTLPWRSOMC	DATA PHASE ERROR
4C	00	DTLPWRSOMC	LOGICAL UNIT FAILED SELF-CONFIGURATION
4D	00		
4E	00	DTLPWRSOMC	OVERLAPPED COMMANDS ATTEMPTED
4F	00		
50	00	T	WRITE APPEND ERROR
50	01	T	WRITE APPEND POSITION ERROR
50	02	T	POSITION ERROR RELATED TO TIMING
51	00	T O	ERASE FAILURE
52	00	T	CARTRIDGE FAULT

Table 364: (continued)

```
D - DIRECT ACCESS DEVICE
.T - SEQUENTIAL ACCESS DEVICE
. L - PRINTER DEVICE
.  P - PROCESSOR DEVICE
.  .W - WRITE ONCE READ MULTIPLE DEVICE
.  . R - READ ONLY (CD-ROM) DEVICE
.  .  S - SCANNER DEVICE
.  .  .O - OPTICAL MEMORY DEVICE
.  .  . M - MEDIA CHANGER DEVICE
.  .  .  C - COMMUNICATION DEVICE
.  .  .  .
```

ASC	ASCQ	DTLPWRSOMC	DESCRIPTION
53	00	DTL WRSOM	MEDIA LOAD OR EJECT FAILED
53	01	T	UNLOAD TAPE FAILURE
53	02	DT WR OM	MEDIUM REMOVAL PREVENTED
54	00	P	SCSI TO HOST SYSTEM INTERFACE FAILURE
55	00	P	SYSTEM RESOURCE FAILURE
56	00		
57	00	R	UNABLE TO RECOVER TABLE-OF-CONTENTS
58	00	O	GENERATION DOES NOT EXIST
59	00	O	UPDATED BLOCK READ
5A	00	DTLPWRSOM	OPERATOR REQUEST OR STATE CHANGE INPUT (UNSPECIFIED)
5A	01	DT WR OM	OPERATOR MEDIUM REMOVAL REQUEST
5A	02	DT W O	OPERATOR SELECTED WRITE PROTECT
5A	03	DT W O	OPERATOR SELECTED WRITE PERMIT
5B	00	DTLPWRSOM	LOG EXCEPTION
5B	01	DTLPWRSOM	THRESHOLD CONDITION MET
5B	02	DTLPWRSOM	LOG COUNTER AT MAXIMUM
5B	03	DTLPWRSOM	LOG LIST CODES EXHAUSTED
5C	00	D O	RPL STATUS CHANGE
5C	01	D O	SPINDLES SYNCHRONIZED
5C	02	D O	SPINDLES NOT SYNCHRONIZED
5D	00		

TOC

5E	00		
5F	00		
60	00	S	LAMP FAILURE
61	00	S	VIDEO ACQUISITION ERROR
61	01	S	UNABLE TO ACQUIRE VIDEO
61	02	S	OUT OF FOCUS
62	00	S	SCAN HEAD POSITIONING ERROR
63	00	R	END OF USER AREA ENCOUNTERED ON THIS TRACK
64	00	R	ILLEGAL MODE FOR THIS TRACK
65	00		
66	00		
67	00		
68	00		
69	00		
6A	00		
6B	00		
6C	00		
6D	00		
6E	00		
6F	00		

Table 364: (concluded)

```
D - DIRECT ACCESS DEVICE
.T - SEQUENTIAL ACCESS DEVICE
. L - PRINTER DEVICE
.  P - PROCESSOR DEVICE
.  .W - WRITE ONCE READ MULTIPLE DEVICE
.  . R - READ ONLY (CD-ROM) DEVICE
.  .  S - SCANNER DEVICE
.  .  .O - OPTICAL MEMORY DEVICE
.  .  . M - MEDIA CHANGER DEVICE
.  .  .  C - COMMUNICATION DEVICE
.  .  .  .
```

ASC	ASCQ	DTLPWRSOMC	DESCRIPTION
70	00		
71	00		
72	00		
73	00		
74	00		
75	00		
76	00		
77	00		
78	00		
79	00		
7A	00		
7B	00		
7C	00		
7D	00		
7E	00		
7F	00		

TOC

```
|  80  xxh \
|   THROUGH >  VENDOR SPECIFIC.
|  FF  xxh /
|
|  xxh 80 \
|  THROUGH >  VENDOR SPECIFIC QUALIFICATION OF STANDARD ASC.
|  xxh FF /
|              ALL CODES NOT SHOWN OR BLANK ARE RESERVED.
+==========================================================================+
```

23. A SCSI command code quick reference

Table 365 is a numerical order listing of the command operation codes.

Table 365: SCSI-2 Operation Codes

```
+=============================================================================+
|           D - DIRECT ACCESS DEVICE                        Device Column Key  |
|           .T - SEQUENTIAL ACCESS DEVICE                   M = Mandatory      |
|           . L - PRINTER DEVICE                            O = Optional       |
|           .  P - PROCESSOR DEVICE                         V = Vendor Specific|
|           .  .W - WRITE ONCE READ MULTIPLE DEVICE         R = Reserved       |
|           .  . R - READ ONLY (CD-ROM) DEVICE                                 |
|           .  .  S - SCANNER DEVICE                                           |
|           .  .  .O - OPTICAL MEMORY DEVICE                                   |
|           .  .  . M - MEDIA CHANGER DEVICE                                   |
|           .  .  .  C - COMMUNICATION DEVICE                                  |
|           .  .  .  .                                                         |
|        OP DTLPWRSOMC Description                                             |
|----------+----------+-------------------------------------------------------|
|        00 MMMMMMMMMM TEST UNIT READY                                         |
|        01  M         REWIND                                                  |
|        01 O V OO OO  REZERO UNIT                                             |
|        02 VVVVVV  V                                                          |
|        03 MMMMMMMMMM REQUEST SENSE                                           |
|        04   O        FORMAT                                                  |
|        04 M      O   FORMAT UNIT                                             |
|        05 VMVVVV  V  READ BLOCK LIMITS                                       |
|        06 VVVVVV  V                                                          |
|        07         O  INITIALIZE ELEMENT STATUS                               |
|        07 OVV O  OV  REASSIGN BLOCKS                                         |
|        08          M GET MESSAGE(06)                                         |
|        08 OMV OO OV  READ(06)                                                |
|        08    O       RECEIVE                                                 |
|        09 VVVVVV  V                                                          |
|        0A   M        PRINT                                                   |
|        0A          M SEND MESSAGE(06)                                        |
|        0A    M       SEND(06)                                                |
|        0A OM  O  OV  WRITE(06)                                               |
|        0B O   OO OV  SEEK(06)                                                |
|        0B   O        SLEW AND PRINT                                          |
|        0C VVVVVV  V                                                          |
|        0D VVVVVV  V                                                          |
|        0E VVVVVV  V                                                          |
|        0F VOVVVV  V  READ REVERSE                                            |
|        10   O O      SYNCHRONIZE BUFFER                                      |
|        10 VM VVV     WRITE FILEMARKS                                         |
|        11 VMVVVV     SPACE                                                   |
|        12 MMMMMMMMMM INQUIRY                                                 |
```

```
|        13 VOVVVV     VERIFY(06)
|        14 VOOVVV     RECOVER BUFFERED DATA
|        15 OMO OOOOOO MODE SELECT(06)
|        16 M   MM MO  RESERVE
|        16  MM   M    RESERVE UNIT
|        17 M   MM MO  RELEASE
|        17  MM   M    RELEASE UNIT
|        18 OOOOOOOO   COPY
|        19 VMVVVV     ERASE
|        1A OMO OOOOOO MODE SENSE(06)
|        1B  O         LOAD UNLOAD
|        1B       O    SCAN
|        1B   O        STOP PRINT
|        1B O   OO O   STOP START UNIT
+==============================================================================+
```

Table 365: (continued)

```
+==============================================================================+
|           D - DIRECT ACCESS DEVICE                        Device Column Key  |
|           .T - SEQUENTIAL ACCESS DEVICE                   M = Mandatory      |
|           . L - PRINTER DEVICE                            O = Optional       |
|           .  P - PROCESSOR DEVICE                         V = Vendor Specific|
|           .  .W - WRITE ONCE READ MULTIPLE DEVICE         R = Reserved       |
|           .  . R - READ ONLY (CD-ROM) DEVICE                                 |
|           .  .  S - SCANNER DEVICE                                           |
|           .  .  .O - OPTICAL MEMORY DEVICE                                   |
|           .  .  . M - MEDIA CHANGER DEVICE                                   |
|           .  .  .  C - COMMUNICATION DEVICE                                  |
|           .  .  .  .                                                         |
|        OP DTLPWRSOMC Description                                             |
|----------+----------+-------------------------------------------------------|
|        1C OOOOOOOOOO RECEIVE DIAGNOSTIC RESULTS                              |
|        1D MMMMMMMMMM SEND DIAGNOSTIC                                         |
|        1E OO  OO OO  PREVENT ALLOW MEDIUM REMOVAL                           |
|        1F                                                                    |
|        20 V   VV V                                                           |
|        21 V   VV V                                                           |
|        22 V   VV V                                                           |
|        23 V   VV V                                                           |
|        24 V   VVM    SET WINDOW                                             |
|        25       O    GET WINDOW                                              |
|        25 M   M  M   READ CAPACITY                                           |
|        25      M     READ CD-ROM CAPACITY                                   |
|        26 V   VV                                                             |
|        27 V   VV                                                             |
|        28          O GET MESSAGE(10)                                         |
|        28 M   MMMM   READ(10)                                                |
|        29 V   VV O   READ GENERATION                                         |
|        2A          O SEND MESSAGE(10)                                        |
|        2A       O    SEND(10)                                                |
|        2A M   M  M   WRITE(10)                                               |
|        2B  O         LOCATE                                                  |
|        2B         O  POSITION TO ELEMENT                                     |
|        2B O   OO O   SEEK(10)                                                |
```

```
|        2C V      O   ERASE(10)
|        2D V   O  O   READ UPDATED BLOCK
|        2E O   O  O   WRITE AND VERIFY(10)
|        2F O   OO O   VERIFY(10)
|        30 O   OO O   SEARCH DATA HIGH(10)
|        31       O    OBJECT POSITION
|        31 O   OO O   SEARCH DATA EQUAL(10)
|        32 O   OO O   SEARCH DATA LOW(10)
|        33 O   OO O   SET LIMITS(10)
|        34       O    GET DATA BUFFER STATUS
|        34 O   OO O   PRE-FETCH
|        34  O         READ POSITION
|        35 O   OO O   SYNCHRONIZE CACHE
|        36 O   OO O   LOCK UNLOCK CACHE
|        37 O      O   READ DEFECT DATA(10)
|        38     O  O   MEDIUM SCAN
|        39 OOOOOOOO   COMPARE
|        3A OOOOOOOO   COPY AND VERIFY
|        3B OOOOOOOOOO WRITE BUFFER
|        3C OOOOOOOOOO READ BUFFER
|        3D     O  O   UPDATE BLOCK
|        3E O   OO O   READ LONG
|        3F O   O  O   WRITE LONG
+==============================================================================+
```

Table 365: (continued)

```
+==============================================================================+
|           D - DIRECT ACCESS DEVICE                        Device Column Key  |
|           .T - SEQUENTIAL ACCESS DEVICE                   M = Mandatory      |
|           . L - PRINTER DEVICE                            O = Optional       |
|           .  P - PROCESSOR DEVICE                         V = Vendor Specific|
|           .  .W - WRITE ONCE READ MULTIPLE DEVICE         R = Reserved       |
|           .  . R - READ ONLY (CD-ROM) DEVICE                                 |
|           .  .  S - SCANNER DEVICE                                           |
|           .  .  .O - OPTICAL MEMORY DEVICE                                   |
|           .  .  . M - MEDIA CHANGER DEVICE                                   |
|           .  .  .  C - COMMUNICATION DEVICE                                  |
|           .  .  .  .                                                         |
|        OP DTLPWRSOMC Description                                             |
|----------+----------+-------------------------------------------------------|
|        40 OOOOOOOOOO CHANGE DEFINITION                                       |
|        41 O          WRITE SAME                                              |
|        42      O     READ SUB-CHANNEL                                        |
|        43      O     READ TOC                                                |
|        44      O     READ HEADER                                             |
|        45      O     PLAY AUDIO(10)                                          |
|        46                                                                    |
|        47      O     PLAY AUDIO MSF                                          |
|        48      O     PLAY AUDIO TRACK INDEX                                  |
|        49      O     PLAY TRACK RELATIVE(10)                                 |
|        4A                                                                    |
|        4B      O     PAUSE RESUME                                            |
|        4C OOOOOOOOOO LOG SELECT                                              |
```

```
|        4D OOOOOOOOOO LOG SENSE
|        4E
|        4F
|        50
|        51
|        52
|        53
|        54
|        55 OOO OOOOOO MODE SELECT(10)
|        56
|        57
|        58
|        59
|        5A OOO OOOOOO MODE SENSE(10)
|        5B
|        5C
|        5D
|        5E
|        5F
+==============================================================================+
```

Table 365: (concluded)

```
+==============================================================================+
|           D - DIRECT ACCESS DEVICE                        Device Column Key  |
|           .T - SEQUENTIAL ACCESS DEVICE                   M = Mandatory      |
|           . L - PRINTER DEVICE                            O = Optional       |
|           .  P - PROCESSOR DEVICE                         V = Vendor Specific|
|           .  .W - WRITE ONCE READ MULTIPLE DEVICE         R = Reserved       |
|           .  . R - READ ONLY (CD-ROM) DEVICE                                 |
|           .  .  S - SCANNER DEVICE                                           |
|           .  .  .O - OPTICAL MEMORY DEVICE                                   |
|           .  .  . M - MEDIA CHANGER DEVICE                                   |
|           .  .  .  C - COMMUNICATION DEVICE                                  |
|           .  .  .  .                                                         |
|        OP DTLPWRSOMC Description                                             |
|----------+----------+-------------------------------------------------------|
|        A0                                                                    |
|        A1                                                                    |
|        A2                                                                    |
|        A3                                                                    |
|        A4                                                                    |
|        A5         M  MOVE MEDIUM                                             |
|        A5      O     PLAY AUDIO(12)                                          |
|        A6         O  EXCHANGE MEDIUM                                         |
|        A7                                                                    |
|        A8          O GET MESSAGE(12)                                         |
|        A8     OO O   READ(12)                                                |
|        A9      O     PLAY TRACK RELATIVE(12)                                 |
|        AA          O SEND MESSAGE(12)                                        |
|        AA     O  O   WRITE(12)                                               |
|        AB                                                                    |
|        AC        O   ERASE(12)                                               |
|        AD                                                                    |
|        AE     O  O   WRITE AND VERIFY(12)                                    |
```

TOC

```
|         AF     OO O   VERIFY(12)
|         B0     OO O   SEARCH DATA HIGH(12)
|         B1     OO O   SEARCH DATA EQUAL(12)
|         B2     OO O   SEARCH DATA LOW(12)
|         B3     OO O   SET LIMITS(12)                                    |
|         B4                                                              |
|         B5                                                              |
|         B5         O  REQUEST VOLUME ELEMENT ADDRESS                    |
|         B6                                                              |
|         B6         O  SEND VOLUME TAG                                   |
|         B7        O   READ DEFECT DATA(12)                              |
|         B8                                                              |
|         B8         O  READ ELEMENT STATUS                               |
|         B9                                                              |
|         BA                                                              |
|         BB                                                              |
|         BC                                                              |
|         BD                                                              |
|         BE                                                              |
|         BF                                                              |
+=========================================================================+
```

24. Example programs

Here is the C example program, which requests manufacturer/model and reports if a medium is loaded in the device.

```
#define DEVICE "/dev/sgc"
/* Example program to demonstrate the generic SCSI interface */
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <fcntl.h>
#include <errno.h>
#include <scsi/sg.h>

#define SCSI_OFF sizeof(struct sg_header)
static unsigned char cmd[SCSI_OFF + 18];      /* SCSI command buffer */
int fd;                                     /* SCSI device/file descriptor */

/* process a complete scsi cmd. Use the generic scsi interface. */
static int handle_scsi_cmd(unsigned cmd_len,         /* command length */
                           unsigned in_size,         /* input data size */
                           unsigned char *i_buff,    /* input buffer */
                           unsigned out_size,        /* output data size */
                           unsigned char *o_buff     /* output buffer */
                           )
{
    int status = 0;
    struct sg_header *sg_hd;

    /* safety checks */
    if (!cmd_len) return -1;            /* need a cmd_len != 0 */
    if (!i_buff) return -1;             /* need an input buffer != NULL */
#ifdef SG_BIG_BUFF
    if (SCSI_OFF + cmd_len + in_size > SG_BIG_BUFF) return -1;
    if (SCSI_OFF + out_size > SG_BIG_BUFF) return -1;
#else
    if (SCSI_OFF + cmd_len + in_size > 4096) return -1;
    if (SCSI_OFF + out_size > 4096) return -1;
#endif

    if (!o_buff) out_size = 0;

    /* generic scsi device header construction */
    sg_hd = (struct sg_header *) i_buff;
    sg_hd->reply_len   = SCSI_OFF + out_size;
    sg_hd->twelve_byte = cmd_len == 12;
    sg_hd->result = 0;
#if     0
```

```
    sg_hd->pack_len    = SCSI_OFF + cmd_len + in_size; /* not necessary */
    sg_hd->pack_id;     /* not used */
    sg_hd->other_flags; /* not used */
#endif

    /* send command */
    status = write( fd, i_buff, SCSI_OFF + cmd_len + in_size );
    if ( status < 0 || status != SCSI_OFF + cmd_len + in_size ||
                       sg_hd->result ) {
        /* some error happened */
        fprintf( stderr, "write(generic) result = 0x%x cmd = 0x%x\n",
                     sg_hd->result, i_buff[SCSI_OFF] );
        perror("");
        return status;
    }

    if (!o_buff) o_buff = i_buff;       /* buffer pointer check */

    /* retrieve result */
    status = read( fd, o_buff, SCSI_OFF + out_size);
    if ( status < 0 || status != SCSI_OFF + out_size || sg_hd->result ) {
        /* some error happened */
        fprintf( stderr, "read(generic) result = 0x%x cmd = 0x%x\n",
                sg_hd->result, o_buff[SCSI_OFF] );
        fprintf( stderr, "read(generic) sense "
                "%x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x\n",
                sg_hd->sense_buffer[0],         sg_hd->sense_buffer[1],
                sg_hd->sense_buffer[2],         sg_hd->sense_buffer[3],
                sg_hd->sense_buffer[4],         sg_hd->sense_buffer[5],
                sg_hd->sense_buffer[6],         sg_hd->sense_buffer[7],
                sg_hd->sense_buffer[8],         sg_hd->sense_buffer[9],
                sg_hd->sense_buffer[10],        sg_hd->sense_buffer[11],
                sg_hd->sense_buffer[12],        sg_hd->sense_buffer[13],
                sg_hd->sense_buffer[14],        sg_hd->sense_buffer[15]);
        if (status < 0)
            perror("");
    }
    /* Look if we got what we expected to get */
    if (status == SCSI_OFF + out_size) status = 0; /* got them all */

    return status;  /* 0 means no error */
}

#define INQUIRY_CMD     0x12
#define INQUIRY_CMDLEN  6
#define INQUIRY_REPLY_LEN 96
#define INQUIRY_VENDOR  8       /* Offset in reply data to vendor name */

/* request vendor brand and model */
static unsigned char *Inquiry ( void )
{
```

```
    unsigned char Inqbuffer[ SCSI_OFF + INQUIRY_REPLY_LEN ];
    unsigned char cmdblk [ INQUIRY_CMDLEN ] =
        { INQUIRY_CMD,  /* command */
                    0,  /* lun/reserved */
                    0,  /* page code */
                    0,  /* reserved */
    INQUIRY_REPLY_LEN,  /* allocation length */
                    0 };/* reserved/flag/link */

    memcpy( cmd + SCSI_OFF, cmdblk, sizeof(cmdblk) );

    /*
     * +------------------+
     * | struct sg_header | <- cmd
     * +------------------+
     * | copy of cmdblk   | <- cmd + SCSI_OFF
     * +------------------+
     */

    if (handle_scsi_cmd(sizeof(cmdblk), 0, cmd,
                        sizeof(Inqbuffer) - SCSI_OFF, Inqbuffer )) {
        fprintf( stderr, "Inquiry failed\n" );
        exit(2);
    }
    return (Inqbuffer + SCSI_OFF);
  }

  #define TESTUNITREADY_CMD 0
  #define TESTUNITREADY_CMDLEN 6

  #define ADD_SENSECODE 12
  #define ADD_SC_QUALIFIER 13
  #define NO_MEDIA_SC 0x3a
  #define NO_MEDIA_SCQ 0x00
  int TestForMedium ( void )
  {
    /* request READY status */
    static unsigned char cmdblk [TESTUNITREADY_CMDLEN] = {
        TESTUNITREADY_CMD, /* command */
                        0, /* lun/reserved */
                        0, /* reserved */
                        0, /* reserved */
                        0, /* reserved */
                        0};/* reserved */

    memcpy( cmd + SCSI_OFF, cmdblk, sizeof(cmdblk) );

    /*
     * +------------------+
     * | struct sg_header | <- cmd
     * +------------------+
```

TOC

```
     * | copy of cmdblk   | <- cmd + SCSI_OFF
     * +------------------+
     */

    if (handle_scsi_cmd(sizeof(cmdblk), 0, cmd,
                                0, NULL)) {
        fprintf (stderr, "Test unit ready failed\n");
        exit(2);
    }

    return
     *(((struct sg_header*)cmd)->sense_buffer +ADD_SENSECODE) !=
                                                        NO_MEDIA_SC ||
     *(((struct sg_header*)cmd)->sense_buffer +ADD_SC_QUALIFIER) !=
                                                        NO_MEDIA_SCQ;
}

void main( void )
{
  fd = open(DEVICE, O_RDWR);
  if (fd < 0) {
    fprintf( stderr, "Need read/write permissions for "DEVICE".\n" );
    exit(1);
  }

  /* print some fields of the Inquiry result */
  printf( "%s\n", Inquiry() + INQUIRY_VENDOR );

  /* look if medium is loaded */
  if (!TestForMedium()) {
    printf("device is unloaded\n");
  } else {
    printf("device is loaded\n");
  }
}
```

SRM Firmware Howto

David Mosberger

v0.5, 17 August 1996

This document describes how to boot Linux/Alpha using the SRM firmware, which is the firmware normally used to boot DEC Unix. Generally, it is preferable to use MILO instead of aboot since MILO is perfectly adapted to the needs of Linux. However, MILO is not always available for a particular system and MILO does not presently have the ability to boot over the network. In either case, using the SRM console may be the right solution.

Unless you're interested in technical details, you may want to skip right to Section aboot.

1. How Does SRM Boot an OS?

- 1.1 Loading The Secondary Bootstrap Loader

2. The Raw Loader

3. The aboot Loader

- 3.1 Getting and Building aboot
- 3.2 Floppy Installation
- 3.3 Harddisk Installation
- 3.4 CD-ROM Installation
- 3.5 Building the Linux Kernel
- 3.6 Booting Linux
- 3.7 Booting Over the Network

4. Sharing a Disk With DEC Unix

- 4.1 Partitioning the disk
- 4.2 Installing `aboot`

1. How Does SRM Boot an OS?

All versions of SRM can boot from SCSI disks and the versions for recent platforms, such as the Noname or AlphaStations can boot from floppy disks as well. Network booting via `bootp` is supported. Note that older SRM versions (notably the one for the Jensen) *cannot* boot from floppy disks. Also, booting from IDE disk drives is unsupported.

Booting Linux with SRM is a two step process: first, SRM loads and transfers control to the secondary bootstrap loader. Then the secondary bootstrap loader sets up the environment for Linux, reads the kernel image from a disk filesystem and finally transfers control to Linux.

Currently, there are two secondary bootstrap loaders for Linux: the *raw* loader that comes with the Linux kernel and `aboot` which is distributed separately. These two loaders are described in more detail below.

1.1 Loading The Secondary Bootstrap Loader

SRM knows nothing about filesystems or disk-partitions. It simply expects that the secondary bootstrap loader occupies a consecutive range of physical disk sector, starting from a given offset. The information on the size of the secondary bootstrap loader and the offset of its first disk sector is stored in the first 512 byte sector. Specifically, the long integer at offset 480 stores the *size* of the secondary bootstrap loader (in 512-byte blocks) and the long at offset 488 gives the *sector number* at which the secondary bootstrap loader starts. The first sector also stores a flag-word at offset 496 which is always 0 and a checksum at offset 504. The checksum is simply the sum of the first 63 long integers in the first sector.

If the checksum in the first sector is correct, SRM goes ahead and reads the *size* sectors starting from the sector given in the *sector number* field and places them in *virtual* memory at address `0x20000000`. If the reading completes successfully, SRM performs a jump to address `0x20000000`.

2. The Raw Loader

The sources for this loader can be found in directory

```
linux/arch/alpha/boot
```

of the Linux kernel source distribution. It loads the Linux kernel by reading `START_SIZE` bytes starting at disk offset `BOOT_SIZE+512` (also in bytes). The constants `START_SIZE` and `BOOT_SIZE` are defined in `linux/include/asm-alpha/system.h`. `START_SIZE` must be at least as big as the kernel image (i.e., the size of the `.text`, `.data`, and `.bss` segments). Similarly, `BOOT_SIZE` must be at least as big as the image of the raw bootstrap loader. Both constants should be an integer multiple of the sector size, which is 512 bytes. The default values are currently 2MB for `START_SIZE` and 16KB for `BOOT_SIZE`. Note that if you want to boot from a 1.44MB floppy disk, you have to reduce `START_SIZE` to 1400KB and make sure that the kernel you want to boot is no bigger than that.

To build a raw loader, simply type `make rawboot` in `/usr/src/linux`. This should produce the following files in `arch/alpha/boot`:

`tools/lxboot`:

The first sector on the disk. It contains the offset and size of the next file in the format described above.

`tools/bootlx`:

The raw boot loader that will load the file below.

`vmlinux.nh`:

The raw kernel image consisting of the `.text`, `.data`, and `.bss` segments of the object file in `/usr/src/linux/vmlinux`. The extension `.nh` indicates that this file has no object-file header.

The concatenation of these three files should be written to the disk from which you want to boot. For example, to boot from a floppy, insert an empty floppy disk in, say, `/dev/fd0` and then type:

```
cat tools/lxboot tools/bootlx vmlinux >/dev/fd0
```

You can then shutdown the system and boot from the floppy by issueing the command `boot dva0`.

3. The aboot Loader

When using the SRM firmware, `aboot` is the preferred way of booting Linux. It supports:

- direct booting from various filesystems (`ext2`, `ISO9660`, and `UFS`, the DEC Unix filesystem)
- booting of executable object files (both ELF and ECOFF)
- booting compressed kernels
- network booting (using bootp)
- partition tables in DEC Unix format (which is compatible with BSD Unix partition tables)
- interactive booting and default configurations for SRM consoles that cannot pass long option strings

3.1 Getting and Building aboot

The latest sources for `aboot` are available in this ftp directory. The description in this manual applies to `aboot` version 0.5 or newer.

Once you downloaded and extracted the latest tar file, take a look at the `README` and `INSTALL` files for installation hints. In particular, be sure to adjust the variables in `Makefile` and in `include/config.h` to match your environment. Normally, you won't need to change anything when building under Linux, but it is always a good idea to double check. If you're satisfied with the configuration, simply type `make` to build it (if you're not building under Linux, be advised that `aboot` requires GNU `make`).

After running `make`, the `aboot` directory should contain the following files:

aboot

This is the actual `aboot` executable (either an ECOFF or ELF object file).

bootlx

Same as above, but it contains only the text, data and bss segments---that is, this file is not an object file.

sdisklabel/writeboot

Utility to install `aboot` on a hard disk.

tools/e2writeboot

Utility to install `aboot` on an ext2 filesystem (usually used for floppies only).

tools/isomarkboot

Utility to install `aboot` on a iso9660 filesystem (used by CD-ROM distributors).

tools/abootconf

Utility to configure an installed `aboot`.

TOC

3.2 Floppy Installation

The bootloader can be installed on a floppy using the `e2writeboot` command (note: this can't be done on a Jensen since its firmware does *not* support booting from floppy). This command requires that the disk is not overly fragmented as it needs to find enough contiguous file blocks to store the entire `aboot` image (currently about 90KB). If `e2writeboot` fails because of this, reformat the floppy and try again (e.g., with `fdformat(1)`). For example, the following steps install `aboot` on floppy disk assuming the floppy is in drive `/dev/fd0`:

```
fdformat /dev/fd0
mke2fs /dev/fd0
e2writeboot /dev/fd0 bootlx
```

3.3 Harddisk Installation

Since the `e2writeboot` command may fail on highly fragmented disks and since reformatting a harddisk is not without pain, it is generally safer to install `aboot` on a harddisk using the `swriteboot` command. `swriteboot` requires that the first few sectors are reserved for booting purposes. We suggest that the disk be partitioned such that the first partition starts at an offset of 2048 sectors. This leaves 1MB of space for storing `aboot`. On a properly partitioned disk, it is then possible to install `aboot` as follows (assuming the disk is `/dev/sda`):

```
swriteboot /dev/sda bootlx
```

On a Jensen, you will want to leave some more space, since you need to write a kernel to this place, too---2MB should be sufficient when using compressed kernels. Use `swriteboot` as described in Section booting to write `bootlx` together with the Linux kernel.

3.4 CD-ROM Installation

To make a CD-ROM bootable by SRM, simply build `aboot` as described above. Then, make sure that the `bootlx` file is present on the iso9660 filesystem (e.g., copy `bootlx` to the directory that is the filesystem master, then run `mkisofs` on that directory). After that, all that remains to be done is to mark the filesystem as SRM bootable. This is achieved with a command of the form:

```
isomarkboot filesystem bootlx
```

The command above assumes that `filesystem` is a file containing the iso9660 filesystem and that `bootlx` has been copied into the root directory of that filesystem. That's it!

3.5 Building the Linux Kernel

A bootable Linux kernel can be built with the following steps. During the `make config`, be sure to answer "yes" to the question whether you want to boot the kernel via SRM.

```
cd /usr/src/linux
make config
make dep
```

```
make boot
```

The last command will build the file `arch/alpha/boot/vmlinux.gz` which can then be copied to the disk from which you want to boot from. In our floppy disk example above, this would entail:

```
mount /dev/fd0 /mnt
cp arch/alpha/boot/vmlinux.gz /mnt
umount /mnt
```

3.6 Booting Linux

With the SRM firmware and `aboot` installed, Linux is generally booted with a command of the form:

`boot` *`devicename`* `-fi` *`filename`* `-fl` *`flags`*

The *filename* and *flags* arguments are optional. If they are not specified, SRM uses the default values stored in environment variables `BOOT_OSFILE` and `BOOT_OSFLAGS`. The syntax and meaning of these two arguments is described in more detail below.

Boot Filename

The filename argument takes the form:

[*n*/]*filename*

n is a single digit in the range 1..8 that gives the partition number from which to boot from. *filename* is the path of the file you want boot. For example to boot from the second partition of SCSI device 6, you would enter:

```
boot dka600 -file 2/vmlinux.gz
```

Or to boot from floppy drive 0, you'd enter:

```
boot dva0 -file vmlinux.gz
```

If a disk has no partition table , `aboot` pretends the disk contains one `ext2` partition starting at the first diskblock. This allows booting from floppy disks.

As a special case, partition number 0 is used to request booting from a disk that does not (yet) contain a file system. When specifying "partition" number 0, `aboot` assumes that the Linux kernel is stored right behind the `aboot` image. Such a layout can be achieved with the `swriteboot` command. For example, to setup a filesystem-less boot from `/dev/sda`, one could use the command:

```
swriteboot /dev/sda bootlx vmlinux.gz
```

Booting a system in this way is not normally necessary. The reason this feature exists is to make it possible to get Linux installed on a systems that can't boot from a floppy disk (e.g., the Jensen).

Boot Flags

A number of bootflags can be specified. The syntax is:

```
-flags "options..."
```

TOC

Where "options..." is any combination the following options (separated by blanks). There are many more bootoptions, depending on what drivers your kernel has installed. The options listed below are therefore just examples to illustrate the general idea:

load_ramdisk=1

Copy root file system from a (floppy) disk to the RAM disk before starting the system. The RAM disk will be used in lieu of the root device. This is useful to bootstrap Linux on a system with only one floppy drive.

floppy=*str*

Sets floppy configuration to *str*.

root=*dev*

Select device *dev* as the root-file system. The device can be specified as a major/minor hex number (e.g., 0x802 for /dev/sda2) or one of a few canonical names (e.g., `/dev/fd0`, `/dev/sda2`).

single

Boot system in single user mode.

kgdb

Enable kernel-gdb (works only if `CONFIG_KGDB` is enabled; a second Alpha system needs to be connected over the serial port in order to make this work)

Some SRM implementations (e.g., the one for the Jensen) are handicapped and allow only short option strings (e.g., at most 8 characters). In such a case, `aboot` can be booted with the single-character boot flag "i". With this flag, `aboot` will prompt the user to interacively enter a boot option string of up to 256 characters. For example:

```
boot dka0 -fl i
aboot> 3/vmlinux.gz root=/dev/sda3 single
```

Since booting in that manner quickly becomes tedious, `aboot` allows to define short-hands for frequently used commandlines. In particular, a single digit option (0-9) requests that `aboot` uses the corresponding option string stored in file `/etc/aboot.conf`. A sample `aboot.conf` is shown below:

```
#
# aboot default configurations
#
0:3/vmlinux.gz root=/dev/sda3
1:3/vmlinux.gz root=/dev/sda3 single
2:3/vmlinux.new.gz root=/dev/sda3
3:3/vmlinux root=/dev/sda3
8:- root=/dev/sda3            # fs-less boot of raw kernel
9:0/vmlinux.gz root=/dev/sda3 # fs-less boot of (compressed) ECOFF kernel
-
```

With this configuration file, the command

```
boot dka0 -fl 1
```

corresponds exactly to the boot command shown above. It is quite easy to forget what number corresponds to what option string. To alleviate this problem, boot with option "h" and `aboot` will print the contents of `/etc/aboot.conf` before issueing the prompt for the full option string.

TOC

Finally, whenever `aboot` prompts for an option string, it is possible to enter one of the single character flags ("h", or "0"-"9") to get the same effect as if that flag had been specified in the boot command line. For example could boot with flag "i" and then type "h" (followed by return) to remind yourself of the contents of `/etc/aboot.conf`

Selecting the Partition of /etc/aboot.conf

When installed on a harddisk, `aboot` needs to know what partition to search for the `/etc/aboot.conf` file. A newly compiled `aboot` will search the *second* partition (e.g., `/dev/sda2`). Since it would be inconvenient to have to recompile `aboot` just to change the partition number, `abootconf` allows to directly modify an installed `aboot`. Specifically, if you want to change `aboot` to use the *third* partition on disk `/dev/sda`, you'd use the command:

```
abootconf /dev/sda 3
```

You can verify the current setting by simply omitting the partition number. That is: `abootconf /dev/sda` will print the currently selected partition number. Note that `aboot` does have to be installed already for this command to succeed. Also, when installing a new `aboot`, the partition number will fall back to the default (i.e., it will be necessary to rerun `abootconf`).

Since `aboot` version 0.5, it is also possible to select the `aboot.conf` partition via the boot command line. This can be done with a command line of the form *a*`:`*b* where *a* is the partition that holds `/etc/aboot.conf` and *b* is a single-letter option as described above (`0-9`, `i`, or `h`). For example, if you type `boot -fl "3:h" dka100` the system boots from SCSI ID 1, loads `/etc/aboot.conf` from the third partition, prints its contents on the screen and waits for you to enter the boot options.

3.7 Booting Over the Network

Two prelimenary steps are necessary before Linux can be booted via a network. First, you need to set the SRM environment variables to enable booting via the bootp protocol and second you need to setup another machine as the your boot server. Please refer to the SRM documentation that came with your machine for information on how to enable bootp. Setting up the boot server is obviously dependent on what operating system that machine is running, but typically it involves starting the program `bootpd` in the background after configuring the `/etc/bootptab` file. The `bootptab` file has one entry describing each client that is allowed to boot from the server. For example, if you want to boot the machine `myhost.cs.arizona.edu`, then an entry of the following form would be needed:

```
myhost.cs.arizona.edu:\
        :hd=/remote/:bf=vmlinux.bootp:\
        :ht=ethernet:ha=08012B1C51F8:hn:vm=rfc1048:\
        :ip=192.12.69.254:bs=auto:
```

This entry assumes that the machine's Ethernet address is `08012B1C51F8` and that its IP address is 192.12.69.254. The Ethernet address can be found with the `show device` command of the SRM console or, if Linux is running, with the `ifconfig` command. The entry also defines that if the client does not specify otherwise, the file that will be booted is `vmlinux.bootp` in directory `/remote`. For more information on configuring `bootpd`, please refer to its man page.

Next, build `aboot` with with the command `make netboot`. Make sure the kernel that you want to boot has been built already. By default, the `aboot` `Makefile` uses the kernel in

TOC

`/usr/src/linux/arch/alpha/boot/vmlinux.gz` (edit the `Makefile` if you want to use a differe path). The result of `make netboot` is a file called `vmlinux.bootp` which contains `aboot` *and* the Linu kernel, ready for network booting.

Finally, copy `vmlinux.bootp` to the bootsever's directory. In the example above, you'd copy it into `/remote/vmlinux.bootp`. Next, power up the client machine and boot it, specifying the Ethernet adapter as the boot device. Typically, SRM calls the first Ethernet adapter `ewa0`, so to boot from that device, you'd use the command:

```
boot ewa0
```

The `-fi` and `-fl` options can be used as usual. In particular, you can ask `aboot` to prompt for Linux kernel arguments by specifying the option `-fl i`.

---

4. Sharing a Disk With DEC Unix

Unfortunately, DEC Unix doesn't know anything about Linux, so sharing a single disk between the two OSes is not entirely trivial. However, it is not a difficult task if you heed the tips in this section. The section assumes you are using `aboot` version 0.5 or newer.

4.1 Partitioning the disk

First and foremost: *never* use any of the Linux partitioning programs (`minlabel` or `fdisk`) on a disk that is also used by DEC Unix. The Linux `minlabel` program uses the same partition table format as DEC Unix `disklabel`, but there are some incompatibilities in the data that `minlabel` fills in, so DEC Unix will simply refuse to accept a partition table generated by `minlabel`. To setup a Linux `ext2` partition under DEC Unix, you'll have to change the disktab entry for your disk. For the purpose of this discussion, let's assume that you have an rz26 disk (a common 1GB drive) on which you want to install Linux. The disktab entry under DEC Unix v3.2 looks like this (see file `/etc/disktab`):

```
rz26|RZ26|DEC RZ26 Winchester:\
        :ty=winchester:dt=SCSI:ns#57:nt#14:nc#2570:\
        :oa#0:pa#131072:ba#8192:fa#1024:\
        :ob#131072:pb#262144:bb#8192:fb#1024:\
        :oc#0:pc#2050860:bc#8192:fc#1024:\
        :od#393216:pd#552548:bd#8192:fd#1024:\
        :oe#945764:pe#552548:be#8192:fe#1024:\
        :of#1498312:pf#552548:bf#8192:ff#1024:\
        :og#393216:pg#819200:bg#8192:fg#1024:\
        :oh#1212416:ph#838444:bh#8192:fh#1024:
```

The interesting fields here are `o`*?*, and `p`*?*, where *?* is a letter in the range `a-h` (first through 8-th partition). The `o` value gives the starting offset of the partition (in sectors) and the `p` value gives the size of the partition (also in sectors). See `disktab(4)` for more info. Note that DEC Unix likes to define overlapping partitions. For the entry above, the partition layout looks like this (you can verify this by adding up the various `o` and `p` values):

```
  a     b         d           e           f
|---|-------|-----------|-----------|-----------|

                          c
|-----------------------------------------------|

                    g                   h
            |-----------------|-----------------|
```

TOC

DEC Unix insists that partition `a` starts at offset 0 and that partition `c` spans the entire disk. Other than that, you can setup the partition table any way you like.

Let's suppose you have DEC Unix using partition `g` and want to install Linux on partition `h` with partition `b` being a (largish) swap partition. To get this layout without destroying the existing DEC Unix partition, you need to set the partition types explicitly. You can do this by adding a `t` field for each partition. In our case, we add the following line to the above disktab entry.

```
:ta=unused:tb=swap:tg=4.2BSD:th=resrvd8:
```

Now why do we mark partition `h` as "reservd8" instead of "ext2"? Well, DEC Unix doesn't know about Linux. It so happens that partition type "ext2" corresponds to a numeric value of 8, and DEC Unix uses the string "reservd8" for that value. Thus, in DEC Unix speak, "reservd8" means "ext2". OK, this was the hard part. Now we just need to install the updated disktab entry on the disk. Let's assume the disk has SCSI id 5. In this case, we'd do:

```
disklabel -rw /dev/rrz5c rz26
```

You can verify that everything is all right by reading back the disklabel with `disklabel -r /dev/rrz5c`. At this point, you may want to reboot DEC Unix and make sure the existing DEC Unix partition is still alive and well. If that is the case, you can shut down the machine and start with the Linux installation. Be sure to skip the disk partitioning step during the install. Since we already installed a good partition table, you should be able to proceed and select the 8th partition as the Linux root partition and the 2nd partition as the swap partition. If the disk is, say, the second SCSI disk in the machine, then the device name for these partitions would be `/dev/sdb8` and `/dev/sdb2`, respectively (note that Linux uses letters to name the drives and numbers to name the partitions, which is exactly reversed from what DEC Unix does; the Linux scheme makes more sense, of course ;-).

4.2 Installing `aboot`

First big caveat: with the SRM firmware, you can boot one and only one operating system per disk. For this reason, it is generally best to have at least two SCSI disks in a machine that you want to dualboot between Linux and DEC Unix. Of course, you could also boot Linux from a floppy if speed doesn't matter or over the network, if you have a `bootp`-capable server. But in this section we assume you want to boot Linux from a disk that contains one or more DEC Unix partitions.

Second big caveat: installing `aboot` on a disk shared with DEC Unix renders the first and third partition unusable (since those *must* have a starting offset of 0). For this reason, we recommend that you change the size of partition `a` to something that is just big enough to hold `aboot` (1MB should be plenty).

Once these two caveats are taken care of, installing `aboot` is almost as easy as usual: since partition `a` and `c` will overlap with `aboot`, we need to tell `swriteboot` that this is indeed OK. We can do this under Linux with a command line of the following form (again, assuming we're trying to install `aboot` on the second SCSI disk):

```
swriteboot -f1 -f3 /dev/sdb bootlx
```

TOC

The `-f1` means that we want to force writing `bootlx` even though it overlaps with partition 1. The corresponding applies for partition 3.

This is it. You should now be able to shutdown the system and boot Linux from the harddisk. In our example, the SRM command line to do this would be:

```
boot dka5 -fi 8/vmlinux.gz -fl root=/dev/sdb8
```

Serbian HOWTO

Zoltan Csala, zolika@sezampro.PRVO-SKLONI-OVO.yu

v1.0, November 1998

Cilj ovog dokumenta je da opise kako da podesite Linux i razne aplikacije za Linux prema standardima srpskog jezika. Tu spadaju: raspored slova na tastaturi, fontovi, velicina papira i drugo. Nadamo se da ce ovaj dokument koristiti i drugi korisnici Linuksa kojima je blizak srpski jezik.

1. Uvod

2. Prikazivanje nasih slova na ekranu

- 2.1 Skup znakova "ISO-8859-2"
- 2.2 Konzolni rezim rada
- 2.3 X-Windows graficko okruzenje

3. Podesavanje tastature

- 3.1 Konzolni rezim rada
- 3.2 X-Windows graficko okruzenje

4. Prilagodjavanje nekih aplikacija nasim slovima

- 4.1 Shell-ovi
- 4.2 Ostali programi

TOC

5. Razna podesavanja

- 5.1 Vremenska zona
- 5.2 Format papira A4
- 5.3 Format tekstualnih datoteka za druge operativne sisteme

6. Podesavanje stampanja

- 6.1 Stampaci koji podrzavaju ISO-8859-2 kodnu stranicu
- 6.2 Stampaci koji podrzavaju neki drugi 8-bitni kodni raspored
- 6.3 PostScript stampaci

7. Podrska za lokalna podesavanja u libc 5.4.x i navise

8. Korisni saveti za programiranje sa X11

9. Vazne reference i FTP sajtovi

- 9.1 Ostale reference od znacaja
- 9.2 Linux u Jugoslaviji
- 9.3 FTP sajtovi
- 9.4 USENET diskusione liste

10. Epilog

- 10.1 Posveta
- 10.2 Logicno pitanje... i logican odgovor
- 10.3 Zahvalnice
- 10.4 Izjava o odricanju od odgovornosti
- 10.5 Autorska prava
- 10.6 A note to English readers

---

1. Uvod

Gotovo svi korisnici bilo kog operativnog sistema van granica Sjedinjenih Americkih Drzava imaju dva problema. Prvi je u nacinu saopstavanja kompjuteru da imate tastaturu koja sadrzi i druge znake osim americkih. Drugi problem je - kako podesiti kompjuter da prikaze specijalne znake iz abecede (ili azbuke) jezika kojim govorite. Da stvari budu jos gore, neke aplikacije ce vas smatrati izuzetkom ako niste Amerikanac i zahtevace specijalne opcije ili podesavanja "environment" promenljivih.

Pod Linuxom mozete da promenite nacin na koji vas kompjuter interpretira ono sto otkucate na tastaturi. To radite komandom `loadkeys` i `xmodmap`. Komanda `loadkeys` ce modifikovati tastaturu u konzolnom nacinu rada, dok ce `xmodmap` uraditi sve sto je potrebno kada se nalazite u X11 grafickom okruzenju.

Da biste prikazali nasa slova treba da saopstite aplikacijama da koristite ISO-8859-2 (takodje poznat i kao Latin-2) skup znakova. Ovo nije uvek potrebno, ali nekoliko kljucnih aplikacija zahteva posebnu paznju.

Ovaj HOWTO je pisan sa ciljem da korisnicima koji razumeju srpski jezik objasni kako da izvedu potrebna podesavanja. Ako budete imali problema i posle citanja ovog dokumenta, mozete da procitate *German HOWTO*, *Linux Keyboard and Console HOWTO* ili *ISO 8859-1 National Character Set HOWTO*. Mnogi saveti u ovom tekstu preuzeti su od tamo. Takodje pogledajte odeljak Vazne reference i FTP sajtovi za linkove ka tim dokumentima. Mozete poslati i email na moju elektronsku adresu, a ja cu pokusati da vam odgovorim kad mi vreme dozvoli.

Najveci problem je da su poruke o greskama, meniji i dokumentacija gotovo svih aplikacija na engleskom jeziku. Postoji GNU projekt kome je cilj pokusaj resavanja ovog problema. Da biste videli o cemu se radi, preuzmite datoteku `ABOUT-NLS` ili paket `gettext-0.10.tar.gz` (ili bilo koju noviju verziju) sa vama najblizeg FTP sajta koji ima kopiju `prep.ai.mit.edu`. Dokumentacija u paketu `gettext` opisuje kako da koristite prevode u vasim programima.

2. Prikazivanje nasih slova na ekranu

2.1 Skup znakova "ISO-8859-2"

ISO-8859-2 ili "Latin-2" je clan porodice osmobitnih kodnih rasporeda ISO 8859. Taj skup kodnih rasporeda pokriva alfabete kojima se pise u Evropi, Severnoj i Juznoj Americi, Africi i nekim zemljama Azije. Skup kodnih rasporeda je napravilo evropsko udruzenje proizodjaca racunara (European Computer Manufacturer's Association, ECMA), a potvrdio ga je kao medjunarodni standard organizacija za standardizaciju ISO sa sedistem u Zenevi.

Kodni raspored "Latin 2" koristi sve potrebne znake koji se koriste za pisanje na albanskom, ceskom, engleskom, finskom, hrvatskom, irskom, madjarskom, nemackom, poljskom, rumunskom, slovackom, slovenackom, srpskom (latinica) i luzickosrpskom jeziku.

Neki proizvodjaci racunara ili operativnih sistema, kao sto su IBM, Apple i Microsoft, upotrebljavaju svoje kodne rasporede za slova u gorepomenutim jezicima. Ti kodni rasporedi su po pravilu nepodudarni kako medjusobno, tako i sa ISO standardom.

2.2 Konzolni rezim rada

Skup znakova za konzolni rezim rada mozete naci u paketu kbd koji je napisao Andrijes Bouver [Andries Bouwer] sa univerziteta u Ajndhovenu [Eindhoven], Holandija. U `Slackware` distribuciji pomenuti paket naci cete kao deo veceg paketa keytbls. Mozete ga instalirati (ako vec to niste ucinili) kao i ostale pakete, naredbom `pkgtool` ili `installpkg`. Kada je rec o distribuciji `RedHat`, na raspolaganju vam je RPM paket `kbd-x.xx-x`. Instalira se na poznati nacin, komandom `rpm -Uvv kbd-x.xx-x` (opciju `vv` navodite ako zelite da vidite sta se desava, sto bih vam toplo preporucio).

Po instalaciji paketa, potreban skup znakova za prikaz nasih slova dobijate komandom `setfont` na sledeci nacin:

```
setfont lat2-16
```

Vise o paketu `kbd` naci cete u poglavlju 3, Podesavanje tastature.

2.3 X-Windows graficko okruzenje

Najkraci postupak za dobijanje nasih slova na ekranu pod X-Windowsima bio bi:

1. Preuzmite fontove sa slovenackog FTP sajta
   `ftp://ftp.arnes.si/software/unix/Latin-2-fonts/`
2. Prijavite se na sistem kao `root`.

TOC

3. Kreirajte odgovarajuce poddirektorijume u direktorijumu `/usr/X11R6/lib/X11/fonts/`, primer:

   ```
   /usr/X11R6/lib/X11/fonts/ISO-8859-2/100dpi
   /usr/X11R6/lib/X11/fonts/ISO-8859-2/75dpi
   ```

   Raspakujte fontove (`tar -xvf ...`) u odgovarajuce poddirektorijume.

4. X server zahteva fontove u odgovarajucem, prevedenom formatu. Pribavljene fontove prevedite komandom `bdftopcf`, koja je deo paketa `XFree86-X.X.X-XX` (RedHat):

   ```
   for FILE in *.bdf
   do
       bdftopcf $FILE -o `basename $FILE .bdf`.pcf
   done
   ```

5. Kompresujte tako dobijene fontove komandom `gzip`:

   ```
   gzip *.pcf
   ```

6. U svakom poddirektorijumu X server ocekuje datoteku `fonts.dir` koja preslikava imena datoteka sa fontovima u kanonicka imena fontova (X Logical Font Description, XLFD). Datoteku kreirajte sa

   ```
   mkfontdir
   ```

   koju morate ponoviti u svakom poddirektorijumu direktorijuma `ISO-8859-2`.

7. Ostaje vam jos samo da X server "date do znanja" da ste instalirali nove fontove. To cinite nizom komandi `xset`:

   ```
   xset +fp /usr/X11R6/lib/X11/fonts/ISO-8859-2/75dpi/
   xset +fp /usr/X11R6/lib/X11/fonts/ISO-8859-2/100dpi/
   xset fp rehash
   ```

   Ako je sve islo kao sto treba, naredba

   ```
   xlsfonts -fn "*-iso8859-2"
   ```

   mora prikazati nazive novih fontova.

8. Komanda `xset` vazi samo dok ne izadjete iz X Windows-a. Da biste promene u putu za pretrazivanje fontova ucinili trajnim, morate da promenite konfiguracionu datoteku XFree86 servera. Ona se nalazi u

   ```
   /usr/X11/lib/X11/Xconfig  (Slackware) ili
   /etc/X11/XF86Config  (RedHat)
   ```

   Potrazite u njoj redove koji pocinju sa `FontPath` i dodajte:

```
FontPath "/usr/X11R6/lib/X11/fonts/ISO-8859-2/75dpi"
FontPath "/usr/X11R6/lib/X11/fonts/ISO-8859-2/100dpi"
```

Navodnici oko puta do direktorijuma se *moraju* otkucati.

To bi bilo sve. Od sada X-Windows ima "infrastrukturu" za prikazivanje nasih slova.

3. Podesavanje tastature

3.1 Konzolni rezim rada

skup znakova za konzolni rezim rada i tabele za preslikavanje tastature mozete naci u paketu kbd Andriesa Brouwera sa univerziteta u Ajndhovenu, Holandija. Autor paketa je takodje napisao i Keyboard HOWTO (u pripremi na srpskom jeziku), gde mozete dobiti dodatna obavestenja o ovoj temi. U Slackware distribuciji mozete naci pomenuti paket pod imenom `keytbls`, dok je kod RedHat distribucije u pitanju `kbd-X.XX-X.i386.rpm`. Paketi se instaliraju komandama `pkgtool` ili `rpm`, respektivno.

Paket `kbd` sadrzi raspored za srpsku tastaturu, po ugledu na postojece pisace masine kod nas. Ovaj paket ce raditi ako prethodno pripremite i odgovarajuce fontove po ISO-8859-2 kodnom rasporedu. Ukoliko koristite neki drugi kodni raspored na dirkama tastature, pogledajte datoteke u `/usr/lib/kbd/keytables` direktorijumu (RedHat) ili `/usr/lib/kbd direktorijumu` (Slackware) i napravite ono sto vam je potrebno.

Ceo postupak izbora ekranskg fonta i rasporeda na tastaturi sastoji se u sledecem:

```
# Biramo font za ekran visine 16 piksela
setfont lat2-16 ( ili iso02.f16, ako nemate prvi fajl)
# Srpska latinicna tastatura
loadkeys sr
# Aktivacija tastature
echo -e -n "\\033(K" > /dev/console
```

Gornje redove mozete dodati u datoteku `/etc/rc.d/rc.local`, kako bi se izvrsili prilikom svakog pokretanja Linuxa.

3.2 X-Windows graficko okruzenje

U X-Windows grafickom okruzenju tastaturu podesavamo za nasa slova komandom `xmodmap`. Datoteku sa rasporedom tastera koji se koristi na pisacim masinama za srpski jezik mozete naci na

`ftp://www.linux.org.yu/sr/utils/Xmodmap.sr.pc102`

Upotreba je jednostavna:

```
xmodmap Xmodmap.sl.pc102
```

Ako hocete da promenu ucinite trajnom, prekopirajte datoteku u `~/.Xmodmap` (samo za vas nalog na Linuxu) ili u `/etc/X11/xinit/Xmodmap` (raspored tastera ce vaziti za sve korisnike). Ova promena ce se aktivirati od momenta novog pokretanja X-Windowsa.

4. Prilagodjavanje nekih aplikacija nasim slovima

Prilagodjavanja tastature i ekrana, na zalost, najcesce nisu dovoljna da bi pojedini programi korektno postupali sa nasim slovima. Zato su potrebna dodatna podesavanja. Da bi neki od vaznih programa umeli da prepoznaju nasa slova, potrebno je uraditi sledece:

4.1 Shell-ovi

bash:

Bourne shell (i Bourne-Again-Shell) ce ispravno razumeti srpska podesavanja ako u datoteku `~/.inputrc` ubacite sledece redove:

```
set meta-flag on
set convert-meta off
set output-meta on
```

Mozda ne bi bilo lose da ove redove dodate u `/etc/skel`, jer se tamo nalaze datoteke koje se kopiraju u `$HOME` direktorijum svakog novog korisnika.

(t)csh:

Unos nasih znakova moguc je ako dodamo sledece redove u `~/.cshrc` ili (samo za vas) ili `/etc/csh.cshrc` (za sve korisnike):

```
stty pass8
setenv LC_CTYPE iso_8859_2
setenv LANG C
```

4.2 Ostali programi

elm:

U datoteku `~/.elm/elmrc` dodajte sledece:

```
charset = iso-8859-2
displaycharset = iso-8859-2
textencoding = 8bit
```

Medjutim, nije garantovano da ce ovo raditi u svim verzijama elm-a.

emacs:

Dodajte sledece redove u vasu `~/.emacs` datoteku ili u sistemsku inicijaliza- cionu datoteku

(verovatno `/usr/lib/emacs/site-lisp/default.el` ili `/usr/share/emacs/site-lisp/default.el`):

```
(standard-display-european t)
(require 'iso-syntax)
(set-input-mode (car (current-input-mode))
        (nth 1 (current-input-mode))
        0)
```

groff:

Komandu navedite kao

```
groff -Tlatin2 <vasa_groff_datoteka>
```

Nemojte zaboraviti da ovo izmenite u `/etc/man.config` da biste dobili nasa slova u `man` stranicama, kada one jednog dana budu na raspolaganju na srpskom jeziku. Takodje, nemojte uklanjati `-mandoc` prekidac.

joe:

Kucajte komandu kao

```
joe -asis
```

ili dodajte sledece u vasu `~/.joerc` datoteku:

```
-asis
```

Crtica (-) MORA biti u prvoj koloni datoteke.

kermit:

Ovo je najbolje sto mozete dobiti, ali nije bas sasvim zadovoljavajuce. Stavite sledece u vasu `~/.kermrc` datoteku:

```
set terminal bytesize 8
set command bytesize 8
set file bytesize 8
set file character-set latin2-iso
set transfer character-set latin2-iso
set terminal character-set latin2-iso
```

less:

Postavite sledecu promenljivu:

```
LESSCHARSET=latin1
```

ls:

Kucajte komandu kao

```
ls -N
```

ili, ako to ne radi, kao

```
ls --8bit
```

man:

Vidite deo o `groff` komandi.

metamail:

Postavite sledecu promenljivu:

```
MM_CHARSET-ISO-8859-2
```

nn:

Stavite sledece u `~/.nn/init` datoteku:

```
set data-bits 8
```

pine:

Stavite sledecu definiciju u `~/.pinerc` datoteku:

```
character-set=ISO-8859-2
```

Ovo mozete uraditi i iz "Setup"-a, opcija "Config".

rlogin:

Navedite komandu kao

```
rlogin -8 neki.sajt.yu
```

telnet:

Stavite po jednu liniju u vasu `~/.telnetrc` datoteku za **svaki** host kome zelite da pristupite koristeci telnet:

```
<ime hosta> set outbinary true
```

Na primer:

```
moje.malo.mesto.yu      set outbinary true
moj.ISP.yu              set outbinary true
```

tin:

Stavite sledece definicije u vasu `~/.tin/headers` datoteku:

```
Mime-Version: 1.0
Content-Type: text/plain; charset=iso-8859-2
Content-Transfer-Encoding: 8bit
```

Sada mozete da saljete poruke u diskusione grupe koristeci nasa slova.

---

5. Razna podesavanja

5.1 Vremenska zona

Jugoslavija se nalazi u centralno-evropskoj vremenskoj zoni (CET ili MET) koja tokom zime odgovara grinickom vremenu uvecanom za 1 sat (GMT+1). Vremensku zonu na Linux sistemu mozete postaviti pravljenjem simbolicke veze:

```
ln -sf /usr/lib/zoneinfo/MET /etc/localtime
```

ili

```
ln -sf /usr/lib/zoneinfo/Europe/Belgrade /etc/localtime
```

Ovim, takodje, prelazite na letnje racunanje vremena (GMT+2).

Sistemsko vreme mozete da uskladite sa CMOS casovnikom zadavanjem komande `clock` dok ste prijavljeni kao `root` korisnik. Ako je vas CMOS casovnik podesen na GMT vreme (poznato i kao UTC vreme - standard svih pravih UNIX sistema), onda mozete koristiti

```
clock -u -s
```

U suprotnom, za uskladjivanje lokalnog vremena koristite

```
clock -s
```

5.2 Format papira A4

- dvips: Promenite datoteku `/usr/lib/texmf/dvips/config.ps` ili `~/.dvips`.
- ghostscript: Dodajte opciju `-sPAPERSIZE=a4` u komandnoj liniji kojom pozivate program.
- ghostview: Dodajte sledecu liniju u `~/.Xresources`:

  ```
  Ghostview.pageMedia: A4
  ```

- xdvi: Dodajte sledecu liniju u `~/.Xresources`:

  ```
  XDvi.paper: A4
  ```

5.3 Format tekstualnih datoteka za druge operativne sisteme

Koristeci paket `recode` mozete prevesti datoteke koje sadrze tekst u ISO-8859-2 rasporedu u neki drugi raspored, na primer Microsoft CP-1250, IBM Codepage 852 ili neki treci. Upotreba je jednostavna:

```
recode cp852:latin2 <ime_datoteke>
```

Paket je dostupan kao `recode-3.4.tar.gz` (ili noviji) na svim sajtovima koji drze kopiju sajta `prep.ai.mit.edu`.

6. Podesavanje stampanja

Kod stampanja mozete naici na razne situacije:

- Stampac neposredno podrzava ISO Latin 2 raspored (idealno!)
- Stampac podrzava jedan od ostalih 8-bitnih rasporeda sa nasim slovima (na pr. kodni raspored 852), ali i 7-bitni raspored JUS I.B1.002 (bolje)
- Stampac razume PostScript (dobro)
- Stampac ne razume nista drugo osim ASCII znakova (lose!)

6.1 Stampaci koji podrzavaju ISO-8859-2 kodnu stranicu

Neki od novijih stampaca neposredno podrzavaju ISO-8859-2 kodni raspored. Medju njima su:

- Hewlett-Packard LaserJet 4 Plus i 4M Plus
- Hewlett-Packard LaserJet 5L
- Epson FX-2170
- Epson LQ-870
- Fujitsu DL700
- Fujitsu DL1150/1250
- Fujitsu DL3700/3800
- Fujitsu DL6400/6600

Sve sto treba da uradite je da stampacu naredite da koristi odgovarajucu kodnu stranicu. Komandna sekvenca za stampace Hewlett-Packard je:

```
\033(@N\033(s0p12h0s0b4099T
```

gde je `\033` oktalni kod znaka `Escape`.

Jedan od nacina da pomocu datoteke `/etc/printcap` prilagodimo stampac da stampa po tom rasporedu opisan je u sledecem odeljku.

6.2 Stampaci koji podrzavaju neki drugi 8-bitni kodni raspored

Ako stampac podrzava neki drugi kodni raspored s nasim slovima, ipak mozemo pomocu njega stampati tekstove po ISO-8859-2 rasporedu, ako datoteku `/etc/printcap` prilagodimo na odgovarajuci nacin.

Postupak cemo objasniti na primeru stampaca marke Epson, koji ima znake po YUSCII rasporedu na mesto svedske abecede.

```
lp|ascii|epson|Epson LQ-850:\
        :lp=/dev/lp1:\
        :sd=/usr/spool/lpd/epson:\
        :lf=/usr/spool/lpd/ERRORLOG:\
        :mx#0:\
        :sh:
jus|Epson LQ-850 sa znakovima JUS I.B1.002:\
        :lp=/dev/null:\
        :sd=/usr/spool/lpd/jus:\
        :if=/usr/spool/lpd/jus_filter:\
        :lf=/usr/spool/lpd/ERRORLOG:\
        :mx#0:\
        :sh
latin2|text|Epson LQ-850 sa znakovima po ISO Latin 2:\
        :lp=/dev/null:\
        :sd=/usr/spool/lpd/latin2:\
        :if=/usr/spool/lpd/latin2_filter:\
        :lf=/usr/spool/lpd/ERRORLOG:\
        :mx#0:\
        :sh:
```

U pitanju su tri reda za stampanje. Prva je americki (ASCII) izbor znakova. Drugi koristi `/var/spool/lpd/jus_epson`:

```
#!/bin/sh
/usr/spool/lpd/jus_epson | lpr -Pascii
```

Potrebno je da datoteku filtrirate kroz filter `/var/spool/lpd/jus_epson` i da je postavite u prvi red za stampanje. Filter `jus_epson` ne radi nista drugo osim sto bira svedski kodni raspored, prepise datoteku sa ulaza na izlaz i na kraju opet bira americki raspored. To je jedini deo programa koji zavisi od vrste stampaca:

```
#!/bin/sh
# Stampanje dokumenta kodiranih po JUS I.B1.002 na stampac Epson LQ
#
# Biramo srpski (svedski) raspored
/bin/echo "\033R\005\c"
# Ispisujemo dokument
cat
# Biramo ASCII raspored
/bin/echo "\033R\000\c"
```

Treci red (queue) je modifikacija drugog. Najpre se dokument pretvara iz Latin 2 u JUS kodni raspored, zatim se koristi prethodno opisani filter da doda na pocetku i na kraju potrebne kodove za stampac, i na kraju se datoteka salje u prvi red za stampanje.

```
#!/bin/sh
/usr/local/bin/recode --force latin2:yu | \
/usr/spool/lpd/jus_epson | lpr -Pascii
```

Ovde pretpostavljamo da posedujemo program `recode` u `/usr/local/bin` direktorijumu.

U ovu svrhu morate u direktorijumu `/usr/spool/lpd` (ili `/var/spool/lpd`) kreirati poddirektorijume `/usr/spool/lpd/epson, /usr/spool/lpd/jus` i `/usr/spool/lpd/latin2`.

Stampanje bi trebalo da ide lako:

1. Bez eksplicitne izbora reda, ili sa izborima `-Pepson` ili `-Pascii`, datoteka se stampa kao ASCII:
   ```
   lpr <datoteka>
   ```
2. Izborom reda `-Pjus` datoteka se stampa kao dokument po JUS I.B1.002 rasporedu:
   ```
   lpr -Pjus <datoteka>
   ```
3. Izborom reda `-Platin2` ili `-Ptext` datoteka se stampa po ISO-8859-2 rasporedu:
   ```
   lpr -Platin2 <datoteka>
   ```

6.3 PostScript stampaci

Datoteku mozete lako odstampati ako koristite jedan od filtera za PostScript stampace koji podrzavaju ISO-8859-2 raspored. Takvi programi su, na primer, `a2ps` i GNU `enscript`. Poslednji podrzava ISO-8859-2 raspored, dok je podrska za prvi program u izradi.

Enscript za Red Hat distribucije mozete naci na http://rufus.w3.org/linux/RPM "skladistu" RPM paketa.

GNU enscript

Ako niste instalirali GNU enscript, mozete to da ucinite po uputstvima koja prate program. Osim njega, potreban vam je i neki font u `Type 1` obliku koji sadrzi nasa slova. Dobar izbor bi bio IBM Courier, koji se distribuira uz X Windows sisteme ili sa programom `ghostscript`. Ako font postoji, ali ne znate tacno gde se nalazi, mozete ga pronaci komandom

```
find / -name cour.pf[ab]
```

U licnu (`~/.enscriptrc`) ili sistemsku konfiguracionu datoteku (`/usr/local/etc/enscript.cfg` ili `/usr/loca/etc/enscriptsite.cfg`) dodajte put do datoteke sa fontovima. Osim toga, GNU enscript zahteva u svakom direktorijumu sa fontovima datoteku `font.map`, koja sadrzi preslikavanje izmedju fontova i datoteka u kojima se fontovi nalaze - slicno kao `fonts.dir`. Datoteku `font.map` mozete da kreirate naredbom `mkafmmap`.

Primer konfiguracione datoteke `enscriptsite.cfg`:

```
...
#  Gde su PostScript fontovi?
AFMPath: /usr/lib/X11/fonts/Type1:/usr/local/lib/ghostscript/fonts
...
# Podrazumevani nacin kodiranja:
DefaultEncoding: latin2
...
# Podrazumevani format papira:
DefaultMedia: A4
...
# lpd bira red za stampanje prekidacem -P...
QueueParam: -P
...
```

```
# Za stampanje koristimo naredbu lpr, a ne lp
Spooler: lp
```

Posto ste sve namestili, jednostavno posaljite datoteku na stampanje komandom

```
enscript <datoteka>
```

7. Podrska za lokalna podesavanja u libc 5.4.x i navise

Podrska lokalnim podesavanjima je poboljsana u `libc 5.4.x` i navise. Mozete izbeci mnoga pojedinacna podesavanja programa opisana u poglavlju 4, Prilagodjavanje nekih aplikacija nasim slovima ako je vas sistem pripremljen za lokalnu podrsku. Debian distribucija podrzava lokalna podesavanja ako instalirate paket `wg15-locale`. Procitajte *Locales mini-HOWTO* ako zelite da podesite lokalnu podrsku na ne-Debian sistemima sa `libc 5.4.x`. RedHat distribucija takodje podrzava lokalna podesavanja od verzije 5.0 navise ako instalirate paket `glibc-X.X.X-XX-i386.rpm`. Sistemi sa GNU libc 2 (libc 6.x) takodje podrzavaju lokalna podesavanja.

Ako vasa Linux distribucija ne podrzava srpska latinicna lokalna podesavanja, posluzite se sledecim trikovima:

1. Najpre proverite da li vas sistem podrzava slovenacka ili hrvatska lokalna podesavanja. Ako je to tacno, predjite na korak 3.

2. Nadjite na Internetu tekstualnu datoteku `sl_SI` ili `hr_HR` u kojoj su opisana odgovarajuca podesavanja za hrvatski, odnosno slovenacki jezik.

3. Prijavite se na sistem kao `root`.

4. Predjite u direktorijum `/usr/share/i18n/locales` ako koristite RedHat distribuciju, ili odgovarajuci direktorijum na Slackware, Debian ili Caldera distribuciji.

5. `% cp sl_SI` (ili `hr_HR`) `/usr/share/i18n/locales/sr_YU`

6. `% cd /usr/share/i18n/locales`

7. Koristeci vas omiljeni editor, promenite sledece linije u datoteci `sr_YU` (navedeni brojevi ne moraju nuzno biti tacni, vec su dati radi lakse orijentacije u datoteci):

```
4:    % Serbian Language Locale for Yugoslavia
6:    % Address:
7:    %
8:    % Contact:
9:    % Email:
10:   % Tel:
12:   % Language: sr
13:   % Territory: YU
2117: yesexpr "<<(><d><D><y><Y><)/>><*><.>"
2118: noexpr  "<<(><n><N><)/>><*><.>"
2122: int_curr_symbol           "<Y><U><N><SP>"
2123: currency_symbol           "<d><i><n>"
2148:        "<u><t><o>";"<s><r><e>";/
```

```
2161:        "<j><u><l>";"<a><v><g>";/
2171:        "<a><v><g><u><s><t>";/
2177: d_fmt  "<%><d><-><%><m><-><%><Y><.>"
```

8. Izvrsite komandu

```
localedef -c -i sr_YU -f ISO-8859-2 sr_YU
```

Ako sve protekne u redu, trebalo bi da imate srpsku latinicnu lokalnu podrsku. Da biste je aktivirali, trebalo bi da podesite promenljivu iz okruzenja:

```
LC_ALL=sr_YU
```

Takodje, mozete pokusati sa `sr_YU.ISO_8859_2` ako `sr_YU` ne radi. Da biste proverili da li lokalna podrska radi, otkucajte

```
date
```

Trebalo bi da dobijete datum sa nasim skracenicama imena dana u nedelji i meseca. Pomenute promenljive postavljaju sve kategorije lokalne podrske. Mozete postaviti pojedinacnu kategoriju koristeci ime kategorije kao promenljivu. Lokalne kategorije su:

```
Lokalna kategorija      Primena
------------------      -------
LC_COLLATE              Redosled sortiranja slova u stringu.
LC_CTYPE                Klasifikacija i pretvaranje znakova.
LC_MESSAGES             Prevod odgovora "da" i "ne".
LC_MONETARY             Novcani format.
LC_NUMERIC              Format ne-monetarnih numerickih vrednosti.
LC_TIME                 Format datuma i vremena.
LC_ALL                  Postavlja sve gorenavedene kategorije (i ima
                        prvenstvo u odnosu na sve).
LANG                    Postavlja sve kategorije, ali moze da bude
                        nadjacana bilo kojom pojedinacnom lokalnom
                        kategorijom.
```

Neki programi, kao sto su `bash` ili GNU `emacs` ipak zahtevaju posebna podesavanja, kao sto je opisano u poglavlju 4, Prilagodjavanje nekih aplikacija nasim slovima, ali vecina bi trebalo da radi bez dodatnog "nadzora". Programi kao sto su `nvi` koji ranije nisu radili sa 8-bitnim znacima sada bi trebalo da rade. Lokalna podrska bi trebalo da postane prisutnija sa pojavom distribucija koje su zasnovane na GNU libc 2 biblioteci. Vodite racuna da, mada Red Hat Linux 5.0 sadrzi GNU libc 2, lokalna podrska ne radi. Da biste je aktivirali, treba da izvrsite sledeci skript (zanemarite upozorenja):

```
#!/bin/sh
localedef -c -i en_DK -f ISO-8859-1 en_DK
localedef -c -i sv_SE -f ISO-8859-1 sv_SE
localedef -c -i fi_FI -f ISO-8859-1 fi_FI
localedef -c -i sv_FI -f ISO-8859-1 sv_FI
localedef -c -i ro_RO -f ISO-8859-1 ro_RO
localedef -c -i pt_PT -f ISO-8859-1 pt_PT
localedef -c -i no_NO -f ISO-8859-1 no_NO
localedef -c -i nl_NL -f ISO-8859-1 nl_NL
localedef -c -i fr_BE -f ISO-8859-1 fr_BE
localedef -c -i nl_BE -f ISO-8859-1 nl_BE
localedef -c -i da_DK -f ISO-8859-1 da_DK
```

```
localedef -c -i kl_GL -f ISO-8859-1 kl_GL
localedef -c -i it_IT -f ISO-8859-1 it_IT
localedef -c -i is_IS -f ISO-8859-1 is_IS
localedef -c -i fr_LU -f ISO-8859-1 fr_LU
localedef -c -i fr_FR -f ISO-8859-1 fr_FR
localedef -c -i de_DE -f ISO-8859-1 de_DE
localedef -c -i de_CH -f ISO-8859-1 de_CH
localedef -c -i fr_CH -f ISO-8859-1 fr_CH
localedef -c -i en_CA -f ISO-8859-1 en_CA
localedef -c -i fr_CA -f ISO-8859-1 fr_CA
localedef -c -i fo_FO -f ISO-8859-1 fo_FO
localedef -c -i et_EE -f ISO-8859-1 et_EE
localedef -c -i es_ES -f ISO-8859-1 es_ES
localedef -c -i en_US -f ISO-8859-1 en_US
localedef -c -i en_GB -f ISO-8859-1 en_GB
localedef -c -i en_IE -f ISO-8859-1 en_IE
localedef -c -i de_LU -f ISO-8859-1 de_LU
localedef -c -i de_BE -f ISO-8859-1 de_BE
localedef -c -i de_AT -f ISO-8859-1 de_AT
localedef -c -i sl_SI -f ISO-8859-2 sl_SI
localedef -c -i ru_RU -f ISO-8859-5 ru_RU
localedef -c -i pl_PL -f ISO-8859-2 pl_PL
localedef -c -i lv_LV -f BALTIC lv_LV
localedef -c -i lt_LT -f BALTIC lt_LT
localedef -c -i iw_IL -f ISO-8859-8 iw_IL
localedef -c -i hu_HU -f ISO-8859-2 hu_HU
localedef -c -i hr_HR -f ISO-8859-2 hr_HR
localedef -c -i gr_GR -f ISO-8859-7 gr_GR
```

---

8. Korisni saveti za programiranje sa X11

Prikazivanje 8-bitnih znakova je vrlo lako. Koristite ih isto kao sto biste koristili 7-bitni ASCII skup znakova. Medjutim, da bi aplikacija prihvatila unos 8-bitnih znakova predstavlja totalno drugu pricu.

Ako koristite na pr. Xt okruzenje i Motif, treba da dodate samo jednu liniju u vas program. **Prvi** poziv Xt biblioteci treba da bude procedura `XtSetLanguageProc`. Na primer:

```
int main (int argc, char** argv)
{
        ...
        XtSetLanguageProc (NULL, NULL, NULL);
        top = XtAppInitialize ( ... );
        ...
}
```

Posle ovog vas program ce automatski da uzima u obzir `LC_CTYPE` promenljivu i tumaci tastere prema `Compose` tabelama u `/usr/lib/X11/locale/` direktorijumu. Ovo bi trebalo da radi za gotovo sve evropske jezike i u celini je prenosivo. Kako XFree86 visejezicna podrska postaje bolja, tako ce vas program biti korisniji i na Bliskom i Srednjem istoku.

Ovaj metod unosa podrzavaju Xt, Xlib i Motif v1.2 (i noviji). Prema podacima kojima raspolazem, Xaw biblioteka ga samo delimicno podrzava. Ako znate nesto vise u vezi sa ovim, voleo bih da mi napisete koji red putem elektronske poste :) .

Ovo poglavlje je sastavljeno na osnovu obimnije diskusije u delu *Programming for internationalization* Majkla Gsvinda [Michael Gschwind]. Pogledajte poglavlje 9, Reference i FTP sajtovi za pokazivac na ovaj dokument.

9. Vazne reference i FTP sajtovi

9.1 Ostale reference od znacaja

Ovaj dokument bi trebalo da je na raspolaganju na svim racunarima koji drze kopiju sajta `sunsite.unc.edu` i `tsx-11.mit.edu`. Srpski sajt mozete naci na adresi http://www.linux.org.yu/sr/ .

German HOWTO (na nemackom) napisao Winfred Truemper. Mnogi nacionalni HOWTO dokumenti (finski, poljski, slovenacki) su raspolozivi na odgovarajucim jezicima.

Linux Keyboard and Console HOWTO napisao Andries Brouwer.

Locales mini-HOWTO napisao Peeter Joot.

Dokumenti *ISO 8859-1 National Character Set FAQ* i *Programming for Internationalization* (i mnogo vise) koje je napisao Michael Gschwind raspolozivi su na njegovoj prezentaciji http://www.visivie.tuwien.ac.at/mike/i18n.html

9.2 Linux u Jugoslaviji

Harald T. Alvestrand iz Norveske vodi evidenciju o korisnicima Linuxa po drzavama. Spisak srpskih (ili jugoslovenskih) korisnika je na

`http://domen.uninett.no:29659/bycountry/YU.html`

Ako niste medju njima, prijavite se - odmah!

`http://domen.uninett.no:29659/`

Od leta 1997. u Beogradu je organizovano Udruzenje korisnika Linuksa u Jugoslaviji (LUGY - Linux Users Group of Yugoslavia). Udruzenje ima svoju prezentaciju na

`http://www.linux.org.yu`

Jugoslovenska grupa LUGY ima, kao sto i dolikuje, svoju diskusionu listu! Na nju se mozete prijaviti ako na adresu majordomo@beocity.com posaljete poruku `subscribe linux` u telu poruke. U momentu dok ovo citate, moguce je da je pretplata na diskusionu listu prebacena na "pravo" mesto, tj. na majordomo@linux.org.yu, no, to ostavljam vama da proverite.

9.3 FTP sajtovi

U vreme pisanja ovog dokumenta, autoru su bila poznata dva FTP sajta posvecena Linuksu i srpskim podesavanjima:

`ftp://ftp.emi.yu/pub/Linux`

http://www.linux.org.yu/sr/utils/

9.4 USENET diskusione liste

Kada je o USENET-u rec, postoji diskusiona grupa yu.os.unix u kojoj se, pored ostalog, raspravlja i o Linuksu. Jedan od news servera, preko koga mozete da citate clanke, jeste i:

news.beotel.net

---

10. Epilog

10.1 Posveta

Ovaj dokument je posvecen uspomeni na Milenu Rogulj ("teta-Milenu"), dragu prijateljicu koju mi je smrt prerano otela.

10.2 Logicno pitanje... i logican odgovor

Deo ovog dokumenta govori o namestanju nasih slova na Linuksu, ali ovde ih nema. Kako to?

Pretpostavlja se da je ovaj dokument prvi u nizu dokumenata na srpskom jeziku koga ce procitati novopeceni korisnik Linuksa. S obzirom da, kao takav, ne raspolaze znanjem koje je potrebno da namesti nasa slova, ovaj dokument ih ne koristi jer bi nas novopeceni korisnik video samo "kuke i kvake".

10.3 Zahvalnice

Hvala:

- Svim pretplatnicima srpske "linux" diskusione liste na korisnim predlozima (linux@beocity.com),
- Matu Velsu, autoru *Linux Installation and Getting Started*,
- Ajenu Dzeksonu, autoru *Linux frequently asked questions with answers*, i posebno
- Linusu Torvaldsu i GNU-u koji su nam dali Linuks.

10.4 Izjava o odricanju od odgovornosti

Ovaj dokument je "takav kakav je". Ulozeno je puno truda da bude sto bolji, ali ove informacije koristite na sopstvenu odgovornost. Ni u kom slucaju autor ovog teksta ne preuzima odgovornost za bilo kakve stete, posredne ili neposredne, nastale koriscenjem saveta iz dokumenta.

Suvisle primedbe su dobrodosle. Slobodno mi posaljite predloge za poboljsanja, kao i uocene nepravilnosti. Argumentovana (i jos jednom: ARGUMENTOVANA!) kritika je dobrodosla. Pohvale ce biti sacuvane na posebnoj disketi. Sve ostalo ce bez milosti zavrsiti u `/dev/null`.

10.5 Autorska prava

Ako nije drugacije napomenuto, autorska prava na Linux *HOWTO* dokumente poseduju njihovi autori. *HOWTO* dokumenti mogu se reprodukovati i distribuirati u celini ili delimicno, na fizickom ili elektronskom mediju, sve dok je ovo obavestenje prisutno na svim primercima. Komercijalna distribucija

dozvoljena je i podrzava se; medjutim, autor ovog dokumenta bi o takvom distribuiranju zeleo da bu obavesten.

Svi prevodi, izvedeni ili sazeti radovi koji ukljucuju Linux *HOWTO* dokumente moraju biti pod ovom licencom. Dakle, nije vam dozvoljeno da napravite rad izveden iz *HOWTO*-a i da namecete dodatna ogranicenja na njegovu distribuciju. Izuzeci od ovog pravila moguci su pod odredjenim uslovima; kontaktirajte Linux *HOWTO* koordinatora na nize navedenoj adresi.

Ukratko, mi, prevodioci *HOWTO* dokumenata na srpski zelimo da ohrabrimo sirenje ovih informacija kroz sto je moguce vise kanala. Medjutim, zelimo da zadrzimo autorska prava na *HOWTO* dokumente i zeleli bismo da budemo obavesteni o bilo kakvim namerama u vezi sa redistribuicijom svih srpskih *HOWTO* dokumenata.

Ako imate pitanja, kontaktirajte Tima Bajnuma [Tim Bynum], Linux *HOWTO* koordinatora, na linux-howto@sunsite.unc.edu.

10.6 A note to English readers

This document is a copyrighted work. Its use in part or as whole, for non-profit or for-profit purposes, and its dissemination in any form, either printed or electronic is explicitelly allowed, as long as the above copyright notice is retained on all copies. Its suggested citation is

> Serbian-HOWTO Copyright (C) 1998 by Zoltan Csala
> http://www.linux.org.yu/sr/howto/Serbian-HOWTO

Single copies of any document citing this guide would be much appreciated.

The information in this document is provided in belief of being accurate and useful, however without any warranty, expressed or implied, including, but not limited to, the implied warranties of merchantibility or fitness for a particular purpose. The entire risk of using the information goes with the user. The author can not be liable for any possible damage, direct or indirect, arising from the use of information provided in this document.

The Linux Serial HOWTO

David S.Lawyer bf347@lafn.org original by Greg Hankins

v2.01, August 1999

This document describes serial port features other than those which should be covered by Modem-HOWTO, PPP-HOWTO, Serial-Programming-HOWTO, or Text-Terminal-HOWTO. It lists info on multiport serial cards. It contains technical info about the serial port itself in more detail than found in the above HOWTOs and should be best for troubleshooting when the problem is the serial port itself. If you are dealing with a Modem, PPP (used for Internet access on a phone line), or a Text-Terminal, those HOWTOs should be consulted first.

1. Introduction

- 1.1 Copyright, Disclaimer, & Credits
- 1.2 Release Notes
- 1.3 New Versions of this Serial-HOWTO
- 1.4 Related HOWTO's re the Serial Port
- 1.5 Feedback
- 1.6 What is a Serial Port?

2. How the Hardware Transfers Bytes

- 2.1 Transmitting
- 2.2 Receiving
- 2.3 The Large Serial Buffers

3. Serial Port Basics

- 3.1 What is a Serial Port ?
- 3.2 IO Address & IRQ

- 3.3 Names: ttyS0, ttyS1, etc.
- 3.4 Interrupts
- 3.5 Data Flow (Speeds)
- 3.6 Flow Control
- 3.7 Data Flow Path; Buffers
- 3.8 Complex Flow Control Example
- 3.9 Serial Software: Device Driver Module

4. Is the Serial Port Obsolete?

- 4.1 Introduction
- 4.2 EIA-232 Cable Is Low Speed & Short Distance
- 4.3 Inefficient Interface to the Computer

5. Multiport Serial Boards/Cards/Adapters

- 5.1 Intro to Multiport Serial
- 5.2 Making "devices" in the /dev directory
- 5.3 Standard PC Serial Boards
- 5.4 Dumb Multiport Serial Boards (with 8250/16450/16550A UART's)
- 5.5 Intelligent Multiport Serial Boards

6. Configuring the Serial Port

- 6.1 Configuring Overview
- 6.2 Common mistakes made re low-level configuring
- 6.3 I/O Address & IRQ: Boot-time messages
- 6.4 What is the current IO address and IRQ of my Serial Port ?
- 6.5 Choosing Serial IRQs
- 6.6 Choosing Addresses --Video card conflict with ttyS3
- 6.7 Set IO Address & IRQ in the hardware (mostly for PnP)
- 6.8 Giving the IRQ and IO Address to Setserial
- 6.9 High-level Configuring: stty, etc.

TOC

7. Serial Port Devices /dev/ttyS2, etc.

- 7.1 Serial Port Device Names & Numbers
- 7.2 Link ttySN to /dev/modem ?
- 7.3 Notes For Multiport Boards
- 7.4 Creating Devices In the /dev directory

8. Interesting Programs You Should Know About

- 8.1 Serial Monitoring/Diagnostics Programs
- 8.2 Changing Interrupt Priority
- 8.3 What is Setserial ?
- 8.4 Stty
- 8.5 What is isapnp ?

9. Speed (Flow Rate)

- 9.1 Can't Set a High Enough Speed
- 9.2 Higher Serial Throughput

10. Communications Programs And Utilities

- 10.1 List of Software
- 10.2 kermit and zmodem

11. Serial Tips And Miscellany

- 11.1 Line Drivers
- 11.2 Known Defective Hardware
- 11.3 What Are Lock Files ?

12. Troubleshooting

- 12.1 Serial Electrical Test Equipment
- 12.2 Serial Monitoring/Diagnostics

- 12.3 The following subsections are in both the Serial and Modem HOWTOs:
- 12.4 My Serial Port is Physically There but Can't be Found
- 12.5 Slow: Text appears on the screen slowly after long delays
- 12.6 The Startup Screen Show Wrong IRQs for the Serial Ports.
- 12.7 "Cannot open /dev/ttyS?: Permission denied"
- 12.8 "Operation not supported by device" (error message) for ttySx
- 12.9 "Cannot create lockfile. Sorry" (error message)
- 12.10 Software which may help

13. Interrupt Problem Details

- 13.1 Symptoms of Mis-set or Conflicting Interrupts
- 13.2 Mis-set Interrupts
- 13.3 Interrupt Conflicts
- 13.4 Resolving Interrupt Problems

14. What Are UARTs? How Do They Affect Performance?

- 14.1 Introduction to UARTS
- 14.2 Two Types of UARTs
- 14.3 FIFOs
- 14.4 UART Model Numbers

15. Pinout and Signals

- 15.1 Pinout
- 15.2 Signals May Have No Fixed Meaning
- 15.3 Cabling Between Serial Ports
- 15.4 RTS/CTS and DTR/DSR Flow Control
- 15.5 Preventing a Port From Opening

16. Voltage Waveshapes

- 16.1 Voltage for a Bit
- 16.2 Voltage Sequence for a Byte
- 16.3 Parity Explained
- 16.4 Forming a Byte (Framing)
- 16.5 How "Asynchronous" is Synchronized

17. Other Serial Devices (not async EIA-232)

- 17.1 Successors to EIA-232
- 17.2 EIA-422-A (balanced) and EIA-423-A (unbalanced)
- 17.3 EIA-485
- 17.4 EIA-530
- 17.5 EIA-612/613
- 17.6 The Universal Serial Bus (USB)
- 17.7 Synchronization & Synchronous

18. Other Sources of Information

- 18.1 Books
- 18.2 Serial Software
- 18.3 Linux Documents
- 18.4 Usenet newsgroups:
- 18.5 Serial Mailing List
- 18.6 Internet

---

1. Introduction

This HOWTO covers basic info on the Serial Port and multiport serial cards. Information specific to modems and text-terminals has been moved to Modem-HOWTO and Text-Terminal-HOWTO. Info on getty has been also moved to these HOWTOs since mgetty and uugetty are best for modems while agetty is best for text-terminals. If you are dealing with a modem, text terminal, or printer, then you may not need to consult this HOWTO. But if you are using the serial port for some other device, using a multiport serial card, trouble-shooting the serial port itself, or want to understand more technical details of the serial port, then you may want to use this HOWTO as well as some of the other HOWTOs. (See Related HOWTO's) This HOWTO lists info on various multiport serial cards since they may be used for either modems or text-terminals. This HOWTO addresses Linux running on Intel x86 hardware, although it might be valid for other architectures.

1.1 Copyright, Disclaimer, & Credits

Copyright

Copyright (c) 1993-1997 by Greg Hankins, 1998-1999 by David S. Lawyer. Please freely copy and distribute (sell or give away) this document. For corrections and minor changes contact the maintainer. Otherwise you may create derivative works and distribute them provided you:

1. Discuss it with the maintainer (if there is one). 2. Put the derivative work at an appropriate Internet site for free downloading and supply the url to both LDP and the HOWTO maintainer. 3. License the work in the spirit of this license or use GPL. 4. Give due credit to previous authors and major contributors.

Disclaimer

While I haven't intentionally tried to mislead you, there are likely a number of errors in this document. Please let me know about them. Since this is free documentation, it should be obvious that I cannot be held legally responsible for any errors.

Credits

Most of the original Serial-HOWTO was written by Greg Hankins. `gregh@cc.gatech.edu` He also rewrote many contributions by others in order to maintain continuity in the writing style and flow. He wrote: "Thanks to everyone who has contributed or commented, the list of people has gotten too long to list (somewhere over one hundred). Special thanks to Ted T'so for answering questions about the serial drivers. Approximately half of v2.00 was from Greg Hankins HOWTO and the other half is by David Lawyer. Ted T'so has continued to be helpful.

TOC

1.2 Release Notes

2.00 was a major revision which has removed info on Terminals and Modems from the old Serial-HOWTO and put such info into:

- Text-Terminal-HOWTO
- Modem-HOWTO

2.01: Added info on Plug-and-Play from Modem-HOWTO and more. Info on setserial and stty has been updated. I still haven't checked out all the info on multiport cards to see if it's up-to-date. The fact that this HOWTO was pieced together from various sources has resulted in a certain lack of integration. This may be improved on in future versions.

1.3 New Versions of this Serial-HOWTO

New versions of the Serial-HOWTO will be available to browse and/or download at LDP mirror sites. For a list of mirror sites see: http://metalab.unc.edu/LDP/mirrors.html. Various formats are available. If you only want to quickly check the date of the latest version look at: http://metalab.unc.edu/LDP/HOWTO/Serial-HOWTO.html.

1.4 Related HOWTO's re the Serial Port

Modems, Text-Terminals, some printers, and other peripherals often use the serial port. Get these HOWTOs from the nearest mirror site as explained above.

- `Modem-HOWTO` is about installing and configuring modems
- `Printing-HOWTO` has info on using a serial printer
- `Serial-Programming-HOWTO` helps you write C programs (or parts of them) that read and write to the serial port and/or check/set its state. A new version is expected soon.
- `Text-Terminal-HOWTO` is about how they work, how to install configure, and repair them.

1.5 Feedback

Please send me any questions, comments, suggestions, or additional material. I'm always eager to hear about what you think about this HOWTO. I'm also always on the lookout for improvements! Tell me exactly what you don't understand, or what could be clearer. You can reach me via email at `mailto:bf347@lafn.org (David Lawyer)`.

1.6 What is a Serial Port?

The conventional serial port (not the newer USB port, or HSSI port) is a very old I/O port. Almost all PC's have them. But Macs (Apple Computer) after mid 1998 (with colored cases) only have the USB port. The common specification is RS-232 (or EIA-232). The connector for the serial port is often seen as

one or two 9-pin connectors (in some cases 25-pin) on the back of a PC. But the serial port is is more than just that. It includes the associated electronics which must produce signals conforming to the EIA-232 specification. See Voltage Waveshapes. One pin is used to send out data bytes and another to receive data bytes. Another pin is a common signal ground. The other "useful" pins are used mainly for signalling purposes with a steady negative voltage meaning "off" and a steady positive voltage meaning "on".

The UART (Universal Asynchronous Receiver-Transmitter) chip does most of the work. Today, the functionality of this chip is usually built into another chip. See What Are UARTs? These have improved over time and old models (several years old) are now obsolete.

The serial port was originally designed for connecting modems but it's used to connect many other devices also such as mice, text-terminals, some printers, etc. to a computer. You just plug these devices into the serial port using the correct cable. Many internal modem cards have a built-in serial port so when you install one inside your PC it's as if you just installed another serial port in your PC.

---

2. How the Hardware Transfers Bytes

2.1 Transmitting

Transmitting is sending bytes out of the serial port away from the computer. The first explanation given here will be grossly oversimplified. Then more detail will be added in later explanations. When the computer wants to send a byte out the serial port the CPU sends the byte on the bus inside the computer to the I/O address of the serial port. The serial port takes the byte, and sends it out one bit at a time (a serial bit-stream) on the transmit pin of the serial connector. For what a bit (and byte) look like electrically see Voltage Waveshapes. Everything in this section is about transmission of bytes (and not about reception which uses another set of hardware inside the serial port).

Here's a replay of the above in a little more detail (but still very incomplete). Most of the work at the serial port is done by the UART chip (or the like). To transmit a byte, the serial device driver program (running on the CPU) sends a byte to the serial port"s I/O address. This byte gets into a 1-byte "transmit shift register" in the serial port. From this shift register bits are taken from the byte one-by-one and sent out bit-by-bit on the serial line. Then when the last bit has been sent the shift register needs another byte to send it could ask the CPU to send it another byte. Thus would be simple but it would likely introduce delays since the CPU might not be able to get the byte immediately. After all, the CPU is usually doing other things besides just handling the serial port.

A way to eliminate delays of getting a byte into the shift register is to arrange things so that the CPU gets the byte before the shift register needs it and stores it in a serial port buffer (in hardware). Then when the shift register has sent out its byte and needs a new byte immediately, the serial port hardware just transfers the next byte from its own buffer to the shift register. No need to call the CPU to do this.

The size of this serial port buffer was originally only one byte, but today it is usually 16 bytes (more in higher priced serial ports). Now there is still the problem of keeping this buffer sufficiently supplied with bytes so that when the shift register needs a byte to transmit it will always find one there (unless there are no more bytes to send). This is done by contacting the CPU using an interrupt.

For the old fashioned one-byte buffer it works like this. When the shift register grabs the byte out of the buffer and the butter needs another byte, it sends an interrupt to the CPU by putting a voltage on a dedicated wire on the computer bus. Unless the CPU is doing something very important, the interrupt forces it to stop what it was doing and start running a program which will supply another byte to the port's buffer. Thus the purpose of this buffer is to keep a ready supply of bytes (waiting to be sent) queued in hardware so that there will be no gaps in transmission of bytes out the serial port.

Once the CPU gets the interrupt, it will know who sent the interrupt since there is a dedicated interrupt wire for each serial port (unless interrupts are shared). Then the CPU will start running the serial device driver which checks registers at I/0 addresses to find out what has happened. It finds out that the serial's transmit buffer is empty and waiting for another byte. So if there are more bytes to send, it sends the next byte to the serial port's I/0 address. This next byte should arrive when the previous byte is still in the

TOC

transmit shift register and still being transmitted bit-by-bit.

In review, when a byte has been fully transmitted out the transmit wire of the serial port and the shift register is now empty the following 3 things happen almost simultaneously:

1. Another new byte is moved from the transmit buffer into the transmit shift register
2. The transmission of this new byte (bit-by-bit) begins
3. Another interrupt is issued to tell the device driver to send another byte to the now empty transmit buffer

We say that the serial port is interrupt driven. Each time the serial port issues an interrupt, the CPU sends it another byte. Once a byte has been sent to the transmit buffer by the CPU, then the CPU is free to pursue some other activity until it gets the next interrupt. The serial port transmits bits at a fixed rate which is selected by the user (or an application program). It's sometimes called the baud rate. The serial port also adds extra bits to each byte (start, stop and perhaps paritey bits) so there are often 10 bits sent per byte. At a rate (also called speed) of 19,200 bits per second (bps), there are thus 1,920 bytes/sec (and also 1,920 interrupts/sec).

Doing all this is a lot of work for the CPU. This is true for many reasons. First, just sending one 8-bit byte at a time over a 32-bit data bus (or even 64-bit) is not a very efficient use of bus width. Also, there is a lot of overhead in handing each interrupt. When the interrupt is received, the device driver only knows that something caused an interrupt at the serial port but doesn't know that it's because a character has been sent. The device driver has to make various checks to find out what happened. The same interrupt could mean that a character was received, one of the control lines changed state, etc.

One improvement has been the enlargement of the buffer size of the serial port. Most serial ports today have 16-byte buffers instead of just 1-byte buffers. This means that when the CPU gets an interrupt it gives the serial port up to 16 new bytes to transmit. This is fewer interrupts to service but data must still be transferred one byte at a time over a wide bus. The 16-byte buffer is acutally a FIFO (First In First Out) queue. For a repetition of some of the above plus a few more details see FIFOs.

2.2 Receiving

Receiving bytes by a serial port is similar to sending them only it's in the opposite direction. It's also interrupt driven. For the obsolete type of serial port with 1-byte buffers, when a byte is fully received it goes into the 1-byte receive buffer. Then the port gives the CPU an interrupt to tell it to pick up that byte so that the serial port will have room for storing the byte which is currently being received. For newer serial ports with 16-byte buffers, this interrupt (to fetch the bytes) may be sent after 14 bytes are in the receive buffer. The CPU then stops what it was doing, runs the interrupt service routine, and picks up 1 to 16 bytes from the port. For an interrupt sent when the 14th byte has been received, there could be 16 bytes to get if 2 more bytes have arrived since the interrupt. But if 3 more bytes should arrive (instead of 2), then the 16 byte buffer will overrun.

2.3 The Large Serial Buffers

TOC

We've talked about small (1 or 16 byte) serial port hardware buffers but there are also much larger buffers in main memory. When the CPU takes bytes (1 to 16) out of the receive buffer of the hardware, it puts them into a huge (say 8K-byte) receive buffer in main memory. Then a program that is getting bytes from the serial port takes the bytes it's receiving out of that large buffer (using a "read" statement in the program). A similar situation exists for bytes that are to be transmitted. When the CPU needs to fetch some bytes to be transmitted it takes them out of a large (8K-byte) transmit buffer in main memory and puts them into the small transmit buffer (1 or 16 bytes) in the hardware.

---

3. Serial Port Basics

You don't have to understand the basics to use the serial port But understanding it may help to determine what is wrong if you run into problems. This section not only presents new topics but also repeats some of what was said in the previous section How the Hardware Transfers Bytes but in greater detail.

3.1 What is a Serial Port ?

Intro to Serial

The serial port is an I/O (Input/Output) device.

An I/O device is just a way to get data into and out of a computer. There are many types of I/O devices such as serial ports, parallel ports, disk drive controllers, ethernet boards, universal serial buses, etc. Most PC's have one or two serial ports. Each has a 9-pin connector (sometimes 25-pin) on the back of the computer. Computer programs can send data (bytes) to the transmit pin (output) and receive bytes from the receive pin (input). The other pins are for control purposes and ground.

The serial port is much more than just a connector. It converts the data from parallel to serial and changes the electrical representation of the data. Inside the computer, data bits flow in parallel (using many wires at the same time). Serial flow is a stream of bits over a single wire (such as on the transmit or receive pin of the serial connector). For the serial port to create such a flow, it must convert data from parallel (inside the computer) to serial on the transmit pin (and conversely).

Most of the electronics of the serial port is found in a computer chip (or a section of a chip) known as a UART. For more details on UARTs see the section What Are UARTs? How Do They Affect Performance?. But you may want to finish this section first so that you will hopefully understand how the UART fits into the overall scheme of things.

Pins and Wires

Old PC's used 25 pin connectors but only about 9 pins were actually used so today most connectors are only 9-pin. Each of the 9 pins usually connects to a wire. Besides the two wires used for transmitting and receiving data, another pin (wire) is signal ground. The voltage on any wire is measured with respect to this ground. Thus the minimum number of wires to use for 2-way transmission of data is 3. Except that it has been known to work with no signal ground wire but with degraded performance and sometimes with errors.

There are still more wires which are for control purposes (signalling) only and not for sending bytes. All of these signals could have been shared on a single wire, but instead, there is a separate dedicated wire for every type of signal. Some (or all) of these control wires are called "modem control lines". Modem control wires are either in the asserted state (on) of +12 volts or in the negated state (off) of -12 volts.

TOC

One of these wires is to signal the computer to stop sending bytes out the serial port cable. Conversely another wire signals the device attached to the serial port to stop sending bytes to the computer. If the attached device is a modem, other wires may tell the modem to hang up the telephone line or tell the computer that a connection has been made or that the telephone line is ringing (someone is attempting to call in). See section Pinout and Signals for more details.

RS-232 or EIA-232, etc.

The serial port (not the USB) is usually a RS-232-C, EIA-232-D, or EIA-232-E. These three are almost the same thing. The original RS (Recommended Standard) prefix became EIA (Electronics Industries Association) and later EIA/TIA after EIA merged with TIA (Telecommunications Industries Association). The EIA-232 spec provides also for synchronous (sync) communication but the hardware to support sync is almost always missing on PC's. The RS designation is obsolete but is still widely used. EIA will be used in this howto. Some documents use the full EIA/TIA designation. For info on other (non-EIA-232) serial ports see the section Other Serial Devices (not async EIA-232)

3.2 IO Address & IRQ

Since the computer needs to communicate with each serial port, the operating system must know that each serial port exists and where it is (its I/O address). It also needs to know which wire (IRQ number) the serial port must use to request service from the computer's CPU. It requests service by sending an interrupt on this wire. Thus every serial port device must store in its non-volatile memory both its I/O address and its Interrupt ReQuest number: IRQ. See Interrupts. For the PCI bus it doesn't work exactly this way since the PCI bus has its own system of interrupts. But since the PCI-aware BIOS sets up chips to map these PCI interrupts to IRQs, it seemingly behaves just as described above except that sharing of interrupts is allowed (2 or more devices may use the same IRQ number).

I/O addresses are not the same as memory addresses. When an I/O addresses is put onto the computer's address bus, another wire is energized. This both tells main memory to ignore the address and tells all devices which have I/O addresses (such as the serial port) to listen to the address to see if it matches the device's. If the address matches, then the I/O device reads the data on the data bus.

3.3 Names: ttyS0, ttyS1, etc.

The serial ports are named ttyS0, ttyS1, etc. (and usually correspond respectively to COM1, COM2, etc. in DOS/Windows). The /dev directory has a special file for each port. Type "ls /dev/ttyS*" to see them. Just because there may be (for example) a ttyS3 file, doesn't necessarily mean that there exists a phyical serial port there.

Which one of these names (ttyS0, ttyS1, etc.) refers to which physical serial port is determined as follows. The serial driver (software) maintains a table showing which I/O address corresponds to which ttyS. This mapping of names (such as ttyS1) to I/O addresses (and IRQ's) may be both set and viewed by the "setserial" command. See What is Setserial. This does `not` set the I/O address and IRQ in the hardware itself (which is set by jumpers or by plug-and-play software). Thus what physical port

TOC

corresponds to say ttyS1 depends both on what the serial driver thinks (per setserial) and what is set in the hardware. If a mistake has been made, the phyical port may not correspond to any name (such as ttyS2) and thus it can't be used. See Serial Port Devices /dev/ttyS2, etc. for more details>

3.4 Interrupts

When the serial port receives several bytes (may be set to 1, 4, 8, or 14) into its buffer, it signals the CPU to fetch it (them) by sending an electrical signal known as an interrupt on a certain wire. On the ISA bus, each device usually has it own private interrupt line (wire). This interrupt will also be sent if there is an unexpected delay while waiting for the next byte to arrive (known as a timeout). Thus if the bytes are being received slowly (such as someone typing on a terminal keyboard) there may be an interrupt issued for every byte received. For some UART chips if 4 bytes could have received, but none of these 4 show up, then the port gives up waiting for the next byte and issues an interrupt to fetch the 1 or more bytes waiting in the buffer.

Each interrupt conductor (inside the computer) has a number (IRQ) and the serial port must know which conductor to use to signal on. For example, ttyS0 normally uses IRQ number 4 known as IRQ4 (or IRQ 4). A list of them and more will be found in "man setserial" (search for "Configuring Serial Ports"). Interrupts are issued whenever the serial port needs to get the CPU's attention. It's important to do this in a timely manner since the buffer inside the serial port can hold only 16 (1 in old modems) incoming bytes. If the CPU fails to remove such received bytes promptly, then there will not be any space left for any more incoming bytes and the small buffer may overflow (overrun). Bytes will be lost. There is no Flow Control to prevent this.

Interrupts are also issued when the serial port has just sent out all 16 of its bytes from its small transmit buffer out the external cable. It then has space for 16 more outgoing bytes. The interrupt is to notify the CPU of that fact so that it may put more bytes in the small transmit buffer to be transmitted. Also, when a modem control line changes state an interrupt is issued.

The buffers mentioned above are all hardware buffers. The serial port also has large buffers in main memory. This will be explained later

Interrupts convey a lot of information but only indirectly. The interrupt itself just tells a chip called the interrupt controller that a certain serial port needs attention. The interrupt controller then signals the CPU. The CPU runs a special program to service the serial port. That program is called an interrupt service routine (part of the serial driver software). It tries to find out what has happened at the serial port and then deals with the problem such a transferring bytes from (or to) the serial port's hardware buffer. This program can easily find out what has happened since the serial port has registers at IO addresses known to the the serial driver software. These registers contain status information about the serial port. The software reads these registers and by inspecting the contents, finds out what has happened and takes appropriate action.

TOC

3.5 Data Flow (Speeds)

Data (bytes representing letters, pictures, etc.) flows into and out of your serial port. Flow rates (such as 56k (56000) bits/sec) are (incorrectly) called "speed". But almost everyone says "speed" instead of "flow rate".

It's important to understand that the average speed is often less than the specified speed. Waits (or idle time) result in a lower average speed. These waits may include long waits of perhaps a second due to Flow Control. At the other extreme there may be very short waits (idle time) of several micro-seconds between bytes. If the device on the serial port (such as a modem) can't accept the full serial port speed, then the average speed must be reduced.

3.6 Flow Control

Flow control means the ability to stop the flow of bytes in a wire. It also includes provisions to restart the flow without any loss of bytes. Flow control is needed for modems to allow a jump in instantaneous flow rates.

Example of Flow Control

For example, consider the case where you connect a 36.6k modem to your serial port that simply sends and receives bytes over the phone line at exactly 36.6k bits per second (bps). It's not doing any data compression or error correction. You have set the serial port speed to 115,200 bits/sec (bps), and you are sending data from your computer to the phone line. Then the flow from the your computer to your modem is at 115.2k bps. However the flow from your modem out the phone line is at best only 33.6k bps. Since a faster flow (115.2k) is going into your modem than is coming out of it, the modem is storing the excess flow (115.2k -33.6k = 81.6k bps) in one of its buffers. This buffer would eventually overrun (run out of storage space) unless the 115.2k flow is stopped.

But now flow control comes to the rescue. When the modem's buffer is almost full, the modem sends a stop signal to the serial port. The serial port passes on the stop signal on to the device driver and the 115.2k bps flow is halted. Then the modem continues to send out data at 33.6k bps drawing on the data it previous accumulated in its (the modem's) buffer. Since nothing is coming into the buffer, the level of bytes in it starts to drop. When almost no bytes are left in the buffer, the modem sends a start signal to the serial port and the 115.2k flow from the computer to the modem resumes. In effect, flow control creates an average flow rate (in this case 33.6k) which is significantly less than the "on" flow rate of 115.2k bps. This is "start-stop" flow control.

The above is a simple example of flow control for flow from the computer to a modem , but there is also flow control which is used for the opposite direction of flow: from a modem (or other device) to a computer. Each direction of flow involve 3 buffers: 1. in the modem 2. in the uart (serial port) 3. in main memory managed by the serial driver. Flow control protects certain buffers from overflowing. The small serial hardware buffers are not protected in this way but rely instead on a fast response to the interrupts they issue. This is the essence of flow control but there are still some more details.

TOC

Symptoms of No Flow Control

Understanding flow-control theory can be of practical use. The symptom of no flow control is chunks of data missing from files sent without the benefit of flow control. This is because when overflow happens, it's usually more than just a few bytes that overflow and are lost. Often hundreds or even thousands of bytes get lost, and all in contiguous chunks.

Hardware vs. Software Flow Control

If feasible best to use "hardware" flow control that uses two dedicated "modem control" wires to send the "stop" and "start" signals. Software flow control uses the main receive and transmit wires to send the start and stop signals. It uses the ASCII control characters DC1 (start) and DC3 (stop) for this purpose. They are just inserted into the regular stream of data. Software flow control is not only slower in reacting but also does not allow the sending of binary data unless special precautions are taken. Since binary data will likely contain DC1 and DC3, special means must be taken to distinguish between a DC3 that means a flow control stop and a DC3 that is part of the binary code. Likewise for DC1. To get software flow control to work for binary data requires both modem (hardware) and software support.

3.7 Data Flow Path; Buffers

Although much has been explained about this including flow control, a pair of 16-byte serial port buffers (in the hardware), and a pair of larger buffers inside a device connected to the serial port there is still another pair of buffers. These are large buffers (perhaps 8k) in main memory also known as serial port buffers. When an application program sends bytes to the serial port they first get stashed in the the transmit serial port buffer in main memory. The pair consists of both this transmit buffer and a receive buffer for the opposite direction of byte-flow.

The serial device driver takes out say 16 bytes from this transmit buffer, one byte at a time and puts them into the 16-byte transmit buffer in the serial hardware for transmission. Once in that transmit buffer, there is no way to stop them from being transmitted. They are then transmitted to the device connected to the serial port which also has a fair sized (say 1k) buffer. When the device driver (on orders from flow control) stops the flow of outgoing bytes from the computer, what it actually stops is the flow of outgoing bytes from the large transmit buffer in main memory. Even after this has happened and the flow to the device connected to the serial port has stopped, an application program may keep sending bytes to the 8k transmit buffer until it becomes fill.

When it gets fill, the application program can't send any more bytes to it (a "write" statement in a C_program blocks) and the application program temporarily stops running and waits until some buffer space becomes available. Thus a flow control "stop" is ultimately able to stop the program that is sending the bytes. Even though this program stops, the computer does not necessarily stop computing. It may switch to running other processes while it's waiting at a flow control stop. The above was a little oversimplified since there is another alternative of having the application program itself do something else while it is waiting to "write".

3.8 Complex Flow Control Example

TOC

For many situations, there is a transmit path involving several links, each with its own flow control. For example, I type at a text-terminal connected to a PC with a modem to access a BBS. For this I use the application program "minicom" which deals with 2 serial ports: one connected to a modem and another connected to the text-terminal. What I type at the text terminal goes into the first serial port to minicom, then from minicom out the second serial port to the modem, and then onto the telephone line to the BBS. The text-terminal has a limit to the speed at which bytes can be displayed on its screen and issues a flow control "stop" from time to time to slow down the flow. What happens when such a "stop" is issued? Let's consider a case where the "stop" is long enough to get thru to the BBS and stop the program at the BBS which is sending out the bytes.

Let's trace out the flow of this "stop" (which may be "hardware" on some links and "software" on others). First, suppose I'm "capturing" a long file from the BBS which is being sent simultaneously to both my text-terminal and a to file on my hard-disk. The bytes are coming in faster than the terminal can handle them so it sends a "stop" out its serial port to the first serial port on my PC. The device driver detects it and stops sending bytes from the 8k serial buffer (in main memory) to the terminal. Now minicom still keeps sending out bytes for the terminal into this 8k buffer.

When this 8k transmit buffer (on the first serial port) is full, minicom must stop writing to it. Minicom stops and waits. But this also causes minicom to stop reading from the 8k receive buffer on the 2nd serial port connected to the modem. Flow from the modem continues until this 8k buffer too fills up and sends a different "stop" to the modem. Now the modem's buffer ceases to send to the serial port and also fills up. The modem (assuming error correction is enabled) sends a "stop signal" to the other modem at the BBS. This modem stops sending bytes out of its buffer and when its buffer gets fill, another stop signal is sent to the serial port of the BBS. At the BBS, the 8-k (or whatever) buffer fills up and the program at the BBS can't write to it anymore and thus temporarily halts.

Thus a stop signal from a text terminal has halted a programs on a BBS computer. What a complex sequence of events! Note that the stop signal passed thru 4 serial ports, 2 modems, and one application program (minicom). Each serial port has 2 buffers (in one direction of flow): the 8k one and the hardware 16-byte one. The application program may have a buffer in its C_code. This adds up to 11 different buffers the data is passing thru. Note that the small serial hardware buffers do not participate directly in flow control.

If the terminal speed limitation is the bottleneck in the flow from the BBS to the terminal, then its flow control "stop" is actually stopping the program that is sending from the BBS as explained above. But you may ask: How can a "stop" last so long that 11 buffers (some of them large) all get filled up? It can actually happen this way if all the buffers were near their upper limits when the terminal sent out the "stop".

But if you were to run a simulation on it you would discover that it's usually more complicated than this. At an instant of time some links are flowing and others are stopped (due to flow control). A "stop" from the terminal seldom propagates back to the BBS neatly as described above. It may take a few "stops" from the terminal to result in one "stop" at the BBS. To understand what is going on you really need to observe a simulation which can be done for a simple case with coins on a table. Use only a few buffers and set the upper level for each buffer at only a few coins.

Does one really need to understand all this? Well, understanding this explained to me why capturing from a BBS was loosing text. The situation was exactly the above example but modem-to-modem flow control was disabled. Chunks of captured text that were supposed to also get to my hard-disk never got there because of an overflow at my modem buffer due to flow control "stops" from the terminal. Even though the BBS had a flow path to the hard-disk without bottlenecks, the overflow due to the terminal happened on this path and chunks of text were lost and never even made it to the hard-disk. Note that the flow to the hard-disk passed thru my modem and since the overflow happened there, bytes intended for the hard-disk were lost.

3.9 Serial Software: Device Driver Module

The device driver for the serial port is the software that operates the serial port. It is now provided as a serial module. This module will normally get loaded automatically if it's needed. The kernel 2.2 + will do this. In earlier kernels, you had to have `kerneld` running in order to do auto-load modules on demand. Otherwise the serial module needed to be explicitly listed in /etc/modules. Before modules became popular with Linux, the serial driver was usually built into the kernel.

When the serial module is loaded it displays a message on the screen about the existing serial ports (often showing a wrong IRQ). But once the module is used by `setserial` to tell the device driver the (hopefully) correct IRQ then you should see a second display with the correct IRQ, etc. See What is Setserial for more info on `setserial`.)

One may modify the driver by editing the kernel source code. Much of the serial driver is found in the file serial.c. For details regarding writing of programs for the serial port see Serial-Programming-HOWTO (currently being revised by Vern Hoxie).

---

4. Is the Serial Port Obsolete?

4.1 Introduction

The answer is yes, the serial port is obsolete but it's still needed, especially for Linux. The serial port has many shortcomings but almost all new PC's seem to come with them them. Linux supports ordinary telephone modems only if they work thru a serial port.

The serial port must pass data between the computer and the external cable. Thus it has two interfaces and both of these interfaces are slow. First we'll consider the interface via external cable to the outside world.

4.2 EIA-232 Cable Is Low Speed & Short Distance

The conventional EIA-232 serial port is inherently low speed and is severely limited in distance. Ads often read "high speed" but it can only work at "high speed" over very short distances such as to a modem located right next to the computer. Compared to a network card, even this "high speed" is low speed. All of the serial cable wires use a common ground return wire so that twisted-pair technology (needed for high speeds) can't be used without additional hardware. More modern interfaces for serial ports exist but they are not standard on PC's like the EIA-232 is. See Successors to EIA-232. Some multiport serial cards support them.

It is somewhat tragic that the RS-232 standard from 1969 did not use twisted pair technology which could operate about a hundred times faster. Twisted pairs have been used in telephone cables since the late 1800's. In 1888 (over 110 years ago) the "Cable Conference" reported its support of twisted-pair (for telephone systems) and pointed out its advantages. But over 80 years after this approval by the "Cable Conference", RS-232 failed to utilize it. Since RS-232 was originally designed for connecting a terminal to a low speed modem located nearby, the need for high speed and longer distance transmission was apparently not recognized.

4.3 Inefficient Interface to the Computer

To communicate with the computer, any I/O device needs to have an address so that the computer can write to it and read from it. For this purpose many I/O devices (such as serial ports) use a special type of address known as an I/O addresses (sometimes called an I/O port). It's actually a range of addresses and the lower address in this range is the base address. If someone only says (or writes) "address" it likely really means "base address"

Instead of using I/O, addresses some I/O devices read and write directly from/to main memory. This provides more bandwidth since the conventional serial I/O system only moves a byte at a time. There are various ways to read/write directly to main memory. One way is called shared memory I/O (where the

shared memory is usually on the same card as the I/O device). Other methods are DMA (direct memory access) on the ISA bus and what is about the same as DMA (only much faster): "bus mastering" on the PCI bus. These methods are a lot faster than those used for the serial port. Thus the conventional serial port with it's interrupt driven (every 14 bytes) interface and single bytes transfers on a bus which could accommodate 4 (or 8) bytes at a time is not suited for very high speed I/O.

5. Multiport Serial Boards/Cards/Adapters

5.1 Intro to Multiport Serial

Multiport serial cards install in a slot in a PC on the ISA or PCI bus. Instead of being called "... cards" they are also called "... adapters" or "... boards". Each such card provides you with many serial ports. Each card has a number of external connecters (DB-25 or RJ45 (telephone-like)) so that one may connect up a number of modems, etc. Each such modem is thus connected to its own serial port. Since the space on the external-facing part of the card is limited there is often not enough room for all the serial port connectors. To solve this problem, the connectors may be on the ends of cables which come out (externally) from the card. They may also be on little boxes which are connected by cable to the multiport card.

Dumb ones are not too much different than ordinary serial ports. They are interrupt driven and the CPU of the computer does most all the work servicing them. They usually have a system of sharing a single interrupt for all the ports. They usually require special drivers that you must put into the kernel or activate by modifying source code.

5.2 Making "devices" in the /dev directory

The serial ports your multiport board uses depends on what kind of board you have. Some of these may be listed in detail in `rc.serial` or in `0setserial`. These files may be in the `setserial` package. I highly recommend getting the latest version of `setserial` if you are trying to use multiport boards. You will probably need to create these devices. Either use the `mknod` command, or the `MAKEDEV` script. Devices (in the /dev directory) for serial ports are made by adding ``64 + port number''. So, if you wanted to create devices for `ttyS17`, you would type:

```
linux# mknod -m 666 /dev/cua17 c 5 81
linux# mknod -m 666 /dev/ttyS17 c 4 81
```

Note that ``64 + 17 = 81''. Using the `MAKEDEV` script, you would type:

```
linux# cd /dev
linux# ./MAKEDEV ttyS17
```

Besides the listing of various brands of multiports found in this HOWTO there is Gary's Encyclopedia - Serial Cards. It's not as complete, but may have some different links.

5.3 Standard PC Serial Boards

TOC

Standard PC serial boards (COM1 - COM4) can be used to, to connect external serial devices (modems, serial mice, etc...). Since PC's no longer come with them (but have the chips for this purpose mounted on the motherboard), they are hard to find in retail stores. An internal modem for the ISA bus may include a built-in serial port.

Note: due to address conflicts, you may not be able to use COM4 and IBM8514 video board (or clones) simultaneously. See Avoiding IO Address Conflicts with Certain Video Boards

5.4 Dumb Multiport Serial Boards (with 8250/16450/16550A UART's)

They are also called "serial adapters".
* => The file that ran setserial in Debian shows some details of configuring # => See note below for this board

- AST FourPort and clones (4 ports) * #
- Accent Async-4 (4 ports) *
- Arnet Multiport-8 (8 ports)
- Bell Technologies HUB6 (6 ports)
- Boca BB-1004 (4 ports), BB-1008 (8 ports), BB-2016 (16 ports; See the mini-howto) * #
- Boca IOAT66 or? ATIO66 (6 ports, Linux doesn't support it's IRQ sharing ?? Uses odd-ball 10-cond RJ45-like connectors)
- Boca 2by4 (4 serial ports, 2 parallel ports)
- Byterunner (claims low prices)
- Computone ValuePort V4-ISA (AST FourPort compatible) *
- Digi PC/8 (8 ports) #
- GTEK BBS-550 (8 ports; See the mini-howto)
- Hayes ESP (after kernel 2.1.15)
- HUB-6 See Bell Technologies.
- Longshine LCS-8880, Longshine LCS-8880+ (AST FourPort compatible) *
- Moxa C104, Moxa C104+ (AST FourPort compatible) *
- PC-COMM (4 ports)
- Sealevel Systems COMM-2 (2 ports), COMM-4 (4 ports) and COMM-8 (8 ports)
- SIIG I/O Expander 2S IO1812 (4 ports) #
- STB-4COM (4 ports)
- Twincom ACI/550
- Usenet Serial Board II (4 ports) *

TOC

In general, Linux will support any serial board which uses a 8250, 16450, 16550, 16550A, 16650, et UART. See the latest man page for "setserial" for a more complete list.

Notes:

AST Fourport: You might need to specify `skip_test` in `rc.serial`.

BB-1004 and BB-1008 do not support DCD and RI lines, and thus are not usable for dialin modems. They will work fine for all other purposes.

Digi PC/8 Interrupt Status Register is at 0x140.

SIIG IO1812 manual for the listing for COM5-COM8 is wrong. They should be COM5=0x250, COM6=0x258, COM7=0x260, and COM8=0x268.

5.5 Intelligent Multiport Serial Boards

Make sure that a Linux-compatible driver is available and read the information that comes with it. These boards use special devices (in the /dev directory), and not the standard ones. This information varies depending on your hardware. If you have updated info please send it to me.

- Comtrol RocketPort (36MHz ASIC; 4, 8, 16, 32, up to 128 ports)
  contact: `info@comtrol.com` or `http://www.comtrol.com`
  driver status: supported by Comtrol
  driver location: `ftp://tsx-11.mit.edu/pub/linux/packages/comtrol`
- Computone IntelliPort II (ISA, PCI and EISA busses up to 64 ports)
  contact: Michael H. Warfield, mailto:mhw@wittsend.com, http://www.computone.com
  driver status: Beta
  driver location: ftp://ftp.computone.com/PUB/Products/IntelliPortII/Linux/BETA, patch at http://www.wittsend.com/computone/linux-2.2.10-ctone.patch.gz
  mailing list: mailto:majordomo@lazuli.wittsend.com with "subscribe linux-computone" in body
  note: Old ATvantage and Intelliport cards are not supported by Computone
- Cyclades Cyclom-Y (Cirrus Logic CD1400 UARTs; 8 - 32 ports),
  Cyclom-Z (25MHz MIPS R3000; 8 - 128 ports)
  contact: `sales@cyclades.com` or `http://www.cyclades.com`
  driver status: supported by Cyclades
  driver location: `ftp://ftp.cyclades.com/pub/cyclades` and included in Linux kernel since version 1.1.75
- Decision PCCOM8 (8 ports)
  contact: `pccom8@signum.se`
  driver location: `ftp://ftp.signum.se/pub/pccom8`
- Digi PC/Xi (12.5MHz 80186; 4, 8, or 16 ports),
  PC/Xe (12.5/16MHz 80186; 2, 4, or 8 ports),
  PC/Xr (16MHz IDT3041; 4 or 8 ports),
  PC/Xem (20MHz IDT3051; 8 - 64 ports)

TOC

contact: sales@dgii.com or http://www.dgii.com
driver status: supported by Digi
driver location: ftp://ftp.dgii.com/drivers/linux and included in Linux kernel since version 2.0

- Digi COM/Xi (10MHz 80188; 4 or 8 ports)
  contact: Simon Park, si@wimpol.demon.co.uk
  driver status: ALPHA
  note: Simon is often away from email for months at a time due to his job. Mark Hatle, mailto:fray@krypton.mankato.msus.edu has graciously volunteered to make the driver available if you need it. Mark is not maintaining or supporting the driver.
- Equinox SuperSerial Technology (30MHz ASIC; 2 - 128 ports)
  contact: sales@equinox.com or http://www.equinox.com
  driver status: supported by Equinox
  driver location: ftp://ftp.equinox.com/library/sst
- GTEK Cyclone (16C654 UARTs; 6, 16 and 32 ports),
  SmartCard (24MHz Dallas DS80C320; 8 ports),
  BlackBoard-8A (16C654 UARTs; 8 ports),
  PCSS (15/24MHz 8032; 8 ports)
  contact: spot@gtek.com or http://www.gtek.com
  driver status: supported by GTEK
  driver location: ftp://ftp.gtek.com/pub
- Hayes ESP (COM-bic; 1 - 8 ports)
  contact: Andrew J. Robinson, arobinso@nyx.net or http://www.nyx.net/~arobinso
  driver status: Will be supported by Linux kernel (1998). Also supported by author
  driver location: http://www.nyx.net/~arobinso and included in Linux kernel since version 2.1.15. Setserial 2.15+ supports.
- Maxpeed SS (Toshiba; 4, 8 and 16 ports)
  contact: info@maxpeed.com or http://www.maxpeed.com
  driver status: supported by Maxpeed
  driver location: ftp://maxpeed.com/pub/ss
- Moxa C218 (12MHz 80286; 8 ports),
  Moxa C320 (40MHz TMS320; 8 - 32 ports)
  contact: info@moxa.com.tw or http://www.moxa.com.tw
  driver status: supported by Moxa
  driver location: ftp://ftp.moxa.com.tw/drivers/c218-320/linux
- SDL RISCom/8 (Cirrus Logic CD180; 8 ports)
  contact: sales@sdlcomm.com or http://www.sdlcomm.com
  driver status: supported by SDL
  driver location: ftp://ftp.sdlcomm.com/pub/drivers
- Specialix SX (25MHz T225; 8? - 32 ports),

SIO/XIO (20 MHz Zilog Z280; 4 - 32 ports)
contact: mailto:R.E.Wolff@BitWizard.nl,
http://members.aa.net/~swear/pedia/kerneldocs/specialix.txt
driver status: BETA / Supported by Specialix
driver location: http://www.BitWizard.nl/specialix/
old driver location: ftp://metalab.unc.edu/pub/Linux/kernel/patches/serial

- Stallion EasyIO-4 (4 ports), EasyIO-8 (8 ports), and
  EasyConnection (8 - 32 ports) - each with Cirrus Logic CD1400 UARTs,
  Stallion (8MHz 80186 CPU; 8 or 16 ports),
  Brumby (10/12 MHz 80186 CPU; 4, 8 or 16 ports),
  ONboard (16MHz 80186 CPU; 4, 8, 12, 16 or 32 ports),
  EasyConnection 8/64 (25MHz 80186 CPU; 8 - 64 ports)
  contact: `sales@stallion.com` or `http://www.stallion.com`
  driver status: supported by Stallion
  driver location: `ftp://ftp.stallion.com/drivers/ata5/Linux` and included in linux kernel since 1.3.27

A review of Comtrol, Cyclades, Digi, and Stallion products was printed in the June 1995 issue of the *Linux Journal*. The article is available at `http://www.ssc.com/lj/issue14`.

---

6. Configuring the Serial Port

6.1 Configuring Overview

In many cases, configuring will happen automatically and you have nothing to do. But sometimes you need to configure (or just want to check out the configuration). If so, first you need to know about the two parts to configuring the serial port under Linux:

The first part (low-level configuring) is assigning it an IO address, IRQ, and name (such as ttyS2). This IO-IRQ pair must be set in both the hardware and told to the serial driver. We might just call this "io-irq" configuring for short. The `setserial` is used to tell the driver. PnP methods, jumpers, etc, are used to set the hardware. Details will be supplied later. If you need to configure but don't understand certain details it's easy to get into trouble.

The second part (high-level configuring) is assigning it a speed (such as 38.4K bits/sec), selecting flow control, etc. This is often done by communication programs such as PPP, minicom, or by getty (which you may run on the port so that others may log into it). This high-level configuring may also be done with the `stty` program. `stty` is also useful to view the current status if you're having problems and to reset the terminal interface to close to "normal" by typing something like ``stty sane". See also the section Stty When Linux starts, some effort is made to detect and configure (low-level) a few serial ports. Exactly what happens depends on your BIOS, hardware, Linux distribution, etc. If the serial ports work OK, there may be no need for you to do any configuring. Application programs often do the high-level configuring but you may need to supply them some information so that they can do it. With Plug-and-Play serial ports (often built into an internal modem), the situation has become more complex. Here are cases when you need to do low-level configuring (set IRQ and IO addresses):

- You want to use more than 2 serial ports
- Installing a new serial port (such as an internal modem)
- Having problems with serial port(s)

For kernel 2.2+ you may be able to use more that 2 serial ports without low-level configuring by sharing interrupts but this may be no easier than low-level configuring. See Interrupt sharing and Kernels 2.2+

The low-level configuring (setting the IRQ and IO address) seems to cause people more trouble (than high-level), although for many it's fully automatic and there is no configuring to be done. Thus most all of this section is on that topic. Until the serial driver knows the correct IRQ and IO address the port will probably not work at all. It likely cannot even be found by Linux. Even if it can be found, it may work extremely slow if the IRQ is wrong. See Slow: Text appears on the screen slowly after long delays.

In the Wintel world, the IO address and IRQ are called "resources" and we are thus configuring certain resources. In review, the low-level configuring consists of putting two values (an IRQ number and IO address) into two places:

1. the device driver (often by running "`setserial`" at boot-time)
2. memory registers of the serial port hardware itself

TOC

You may watch the start-up (= boot-time) messages. They are usually correct. But if you're having problems there's a good chance that some of these messages don't show the true configuration of the hardware (and are not supposed to). See I/O Address & IRQ: Boot-time messages.

6.2 Common mistakes made re low-level configuring

Here are some common mistakes people make:

- setserial command: They run it (without the "autoconfig" option) and think it has checked out the hardware (it hasn't).
- setserial messages: They see them displayed on the screen at boot-time, and erroneously think that the result shows how their hardware is actually configured.
- /proc/interrupts: When their serial device isn't in use they don't see its interrupt there, and erroneously conclude that their serial port can't be found (or doesn't have an interrupt set).
- /proc/ioports: People think this shows the hardware configuration when it only shows about the same data (possibly erroneous) as setserial.

6.3 I/O Address & IRQ: Boot-time messages

In many cases your ports will automatically get low-level configured at boot-time (but not always correctly). To see what is happening, look at the start-up messages on the screen. Don't neglect to check the messages from the BIOS before Linux is loaded (no examples shown here). Use shift-PageUp to see the messages after they have flash by. The `dmesg` command may be used at any time to view some of the messages but it often misses important ones. Here's an example of the start-up messages (as of mid 1999). Note that ttyS00 is the same as /dev/ttyS0.

```
At first you see what was detected (but the irq is only a wild guess):

Serial driver version 4.27 with no serial options enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
ttyS01 at 0x02f8 (irq = 3) is a 16550A
ttyS02 at 0x03e8 (irq = 4) is a 16550A

Later you see what was saved, but it's not necessarily correct either:

Loading the saved-state of the serial devices...
/dev/ttyS0 at 0x03f8 (irq = 4) is a 16550A
/dev/ttyS1 at 0x02f8 (irq = 3) is a 16550A
/dev/ttyS2 at 0x03e8 (irq = 5) is a 16550A
```

Note that there is a slight disagreement: The first message shows ttyS2 at irq=4 while the second shows it at irq=5. Your may only have the first message. In most cases the last message is the correct one. But if your having trouble it may be misleading. Before reading the explanation of all of this complexity in the rest of this section, you might just try using your serial port and see if it works OK. If so it may not be essential to read further.

The second message is from the `setserial` program being run at boot-time. It shows what the device driver

thinks is the correct configuration. But this too could be wrong. For example, the irq could actually be set irq=8 in the hardware (both messages wrong). The irq=5 is there because someone incorrectly put this into configuration file (or the like). The fact that Linux sometimes gets IRQs wrong is because it doesn't probe for IRQs. It just assumes the "standard" ones (first message) or accepts what you told it when you configured it (second message). Neither of these is necessarily correct. If the serial driver has the wrong IRQ the serial port is very slow or doesn't seem to work at all.

The first message is a result of Linux probing the serial ports. If a port shows up here it exists but the irq may be wrong. Linux doesn't check IRQs because doing so is not foolproof. It just assumes the IRQs are as shown because they are the "standard" values. Your may check them manually with `setserial` using the `autoconfig` option but this isn't guaranteed to be correct.

The data shown by the BIOS messages (which you see at first) is what is set in the hardware. If your serial port is Plug-and-Play PnP then it's possible that the `isapnp` will run and change these settings. Look for messages about this after Linux starts. The last serial port message shown in the example above should agree with the BIOS messages (as possibly modified by isapnp). If they don't agree then you either need to change the setting in the port hardware or use setserial to tell the driver what is actually set in the hardware.

Also, if you have Plug-and-Play (PnP) serial ports, Linux will not find them unless the IRQ and IO has been set inside the hardware by Plug-and-Play software. This is a common reason why the start-up messages do not show a serial port that physically exists. The PC hardware (a PnP BIOS) may automatically low-level configure this. PnP configuring will be explained later.

6.4 What is the current IO address and IRQ of my Serial Port ?

The previous section indicated how to attempt to do this by looking at the start-up messages. An alternate title for this section is "Other ways of finding out your IO Address and IRQ". If they give you sufficient info then you may not need to read this section. If they don't then there are some other ways to look into this.

There are really two answers to the question "What is my IO and IRQ?" 1. What the device driver thinks has been set (This is what setserial sets and shows). 2. What is actually set in the hardware. They both should be the same. If you're having trouble (including communication programs that can't communicate) it may mean that these two items are not set the same. In other words, this means that the driver has incorrect info on the physical serial port. If the driver has the wrong IO address it will try to send data to a non-existing serial port --or even worse, to an actual device that is not a serial port. If it has the wrong IRQ the driver will not get interrupt service requests from the serial port, resulting in the possible overflow of the serial port's buffer and in very slow response. See Slow: Text appears on the screen slowly after long delays. If it has the wrong model of UART there is also apt to be trouble. To determine if both I0-IRQ pairs are identical you must find out the answers to the questions posed in the next two subsections.

What does the device driver think?

This is easy to find out. Just look at the start-up messages or type "setserial -g /dev/ttyS*". If everything works OK then what it tells you is likely also set in the hardware. There are some other ways to find this info by looking at "files" in the /proc directory. An important reason for understanding these other ways is to warn you that they only show what the device driver thinks. Some people view certain "files" in the /proc directory and erroneously think that what they see is set in the hardware but "it ain't necessarily so".

/proc/ioports will show the IO addresses that the drivers are using. /proc/interrupts shows the IRQs that are used by drivers of currently running processes (that have devices open). Note that in both cases above you are only seeing what the driver thinks and not necessarily what is actually set in the hardware. /proc/interrupts also shows how many interrupts have been issued (often thousands) for each device. You can get a clue from this because if you see a large number of interrupts that have been issued it means that there is a piece of hardware somewhere that is using that interrupt. Sometimes a showing of just a few interrupts doesn't mean that that interrupt is actually being physically generated by any serial port. Thus if you see almost no interrupts for a port that you're trying to use, that interrupt might not be set in the hardware and it implies that the driver is using the wrong interrupt. To view /proc/interrupts to check on a program that you're currently running (such as "minicom") you need to keep the program running while you view it. To do this, try to jump to a shell without exiting the program.

What is set in my serial port hardware ?

How do you find out what IO address and IRQ are actually set in the device hardware? Perhaps the BIOS messages will tell you some info before Linux starts booting. Use the shift-PageUp key to step back thru the boot-time messages and look at the very first ones which are from the BIOS. This is how it was before Linux started. Setserial can't change it but isapnp or pciutils can.

One crude method is try probing with setserial using the "autoconfig" option. You'll need to guess the addresses to probe at. See What is Setserial. For a PCI serial port, look at /proc/pci or /proc/bus/pci/devices. If your serial port is is Plug-and-Play see the next two subsections.

For a port set with jumpers, its how the jumpers were set. If the port is not Plug-and-Play (PnP) but has been setup by using a DOS program then it's set at whatever the person who ran that program set it to.

What is set in my PnP serial port hardware ?

PnP ports don't store their configuration in the hardware when the power is turned off. This is in contrast to Jumpers (non-PnP) which remain the same with the power off. If you have an ISA PnP port, it can reach a state where it doesn't have any IO address or IRQ and is in effect disabled. It should still be possible to find the port using the `pnpdump` program.

For Plug-and-Play (PnP) on the ISA bus one may try the `pnpdump` program (part of `isapnptools`). If you use the --dumpregs option then it should tell you the actual IO address and IRQ set in the port. The address it "trys" is not the device's IO address, but a special

For PnP ports checking on how it's configured under DOS/Windows may not be of much help. Windows stores its configuration info in its Registry which is not used by Linux. It may supply the BIOS's non-volatile memory with some info but it may not be kept in sync with the current Window configuration in the Registry ?? If you let a PnP BIOS automatically do the configuring when you start Linux (and have told the BIOS that you don't have a PnP operating system when running Linux) then Linux should use whatever configuration is in the BIOS's non-volatile memory.

6.5 Choosing Serial IRQs

If you have a true Plug-and-Play set up where either the OS or a PnP BIOS configures all your devices, then you don't choose your IRQs. PnP determines what it thinks is best and assigns them. But if you use the tools in

TOC

Linux for Plug-and-Play (isapnp and pcitools) then you have to choose. If you already know what IRQ you want to use you could skip this section except that you may want to know that IRQ 0 has a special use (see following paragraph).

IRQ 0 is not an IRQ

While IRQ 0 is actually the timer (in hardware) it has a special meaning for setting a serial port with setserial. It tells the driver that there is no interrupt for the port and the driver then will use polling methods. This is quite inefficient but can be tried if there is an interrupt conflict or mis-set interrupt. The advantage of assigning this is that you don't need to know what interrupt is set in the hardware. It should be used only as a temporary expedient until you are able to find a real interrupt to use.

Interrupt sharing and Kernels 2.2+

The general rule is that every device should use a unique IRQ and not share them. But there are situations where sharing is permitted such as with most multi-port boards. Even when it is permitted, it may not be as efficient since every time a shared interrupt is given a check must be made to determine where it came from. Thus if it's feasible, it's nice to allocate every device it's own interrupt.

Prior to kernel 2.2, serial IRQs could be shared with each other only for most multiport boards. Starting with kernel 2.2 serial IRQs may be sometimes shared between all serial ports. In order for sharing to work in 2.2 the kernel must have been compiled with CONFIG_SERIAL_SHARE_IRQ, and the serial port hardware must support sharing (so that if two serial cards put different voltages on the same interrupt wire, only the voltage that means "this is an interrupt" will prevail). Thus even if you have 2.2, it may be best to avoid sharing.

What IRQs to choose?

The serial hardware often has only a limited number of IRQs it can be set at. Also you don't want IRQ conflicts. So there may not be much of a choice. Your PC may normally come with `ttyS0` and `ttyS2` at IRQ 4, and `ttyS1` and `ttyS3` at IRQ 3. Looking at `/proc/interrupts` will show which IRQs are being used by programs currently running. You likely don't want to use one of these. Before IRQ 5 was used for sound cards, it was often used for a serial port.

Here is how Greg (original author of Serial-HOWTO) set his up in /etc/rc.d/rc.serial. rc.serial is a file (shell script) which runs at start-up (it may have a different name of location). For versions of "setserial" after 2.15 it's not always done this way anymore but this example does show the choice of IRQs.

```
/sbin/setserial /dev/ttyS0 irq 3        # my serial mouse
/sbin/setserial /dev/ttyS1 irq 4        # my Wyse dumb terminal
/sbin/setserial /dev/ttyS2 irq 5        # my Zoom modem
/sbin/setserial /dev/ttyS3 irq 9        # my USR modem
```

Standard IRQ assignments:

```
IRQ  0    Timer channel 0 (May mean "no interrupt".  See below.)
IRQ  1    Keyboard
IRQ  2    Cascade for controller 2
IRQ  3    Serial port 2
IRQ  4    Serial port 1
```

```
IRQ  5    Parallel port 2, Sound card
IRQ  6    Floppy diskette
IRQ  7    Parallel port 1
IRQ  8    Real-time clock
IRQ  9    Redirected to IRQ2
IRQ 10    not assigned
IRQ 11    not assigned
IRQ 12    not assigned
IRQ 13    Math coprocessor
IRQ 14    Hard disk controller 1
IRQ 15    Hard disk controller 2
```

There is really no Right Thing to do when choosing interrupts. Just make sure it isn't being used by the motherboard, or any other boards. 2, 3, 4, 5, 7, 10, 11, 12 or 15 are possible choices. Note that IRQ 2 is the same as IRQ 9. You can call it either 2 or 9, the serial driver is very understanding. If you have a very old serial board it may not be able to use IRQs 8 and above.

Make sure you don't use IRQs 1, 6, 8, 13 or 14! These are used by your motherboard. You will make her very unhappy by taking her IRQs. When you are done, double-check `/proc/interrupts` when programs that use interrupts are being run and make sure there are no conflicts.

6.6 Choosing Addresses --Video card conflict with ttyS3

The IO address of the IBM 8514 video board (and others like it) is allegedly 0x?2e8 where ? is 2, 4, 8, or 9. This may conflict (but shouldn't it the serial port is well designed) with the IO address of `ttyS3` at 0x02e8 if the serial port ignores the leading 0 hex digit (many do). That is bad news if you try to use `ttyS3` at this IO address.

In most cases you should use the default addresses if feasible. Addresses shown represent the first address of an 8-byte range. For example 3f8 is really the range 3f8-3ff. Each serial device (as well as other types of devices that use IO addresses) needs its own unique address range. There should be no overlaps (conflicts). Here are the default addresses for the serial ports:

```
ttyS0 address 0x3f8
ttyS1 address 0x2f8
ttyS2 address 0x3e8
ttyS3 address 0x2e8
```

6.7 Set IO Address & IRQ in the hardware (mostly for PnP)

After it's set in the hardware don't forget to insure that it also gets set in the driver by using `setserial`. For non-PnP serial ports they are either set in hardware by jumpers or by running a DOS program ("jumperless") to set them (it may disable PnP). The rest of this subsection is only for PnP serial ports. Here's a list of the possible methods of configuring PnP serial ports:

TOC

- Using a PnP BIOS CMOS setup menu (usually only for external devices on ttyS0 (Com1) and ttyS1 (Com2))
- Letting a PnP BIOS automatically configure a PnP serial port See Using a PnP BIOS to I0-IRQ Configure
- Doing nothing if you have both a PnP serial port and a PnP Linux operating system (see Plug-and-Play-HOWTO).
- Using `isapnp` for a PnP serial port non-PCI)
- Using pciutils (pcitools) for the PCI bus

The IO address and IRQ must be set (by PnP) in their registers each time the system is powered on since PnP hardware doesn't remember how it was set when the power is shut off. A simple way to do this is to let a PnP BIOS know that you don't have a PnP OS and the BIOS will automatically do this each time you start. This might cause problems in Windows (which is a PnP OS) if you start Windows with the BIOS thinking that Windows is not a PnP OS. See Plug-and-Play-HOWTO.

Plug-and-Play was designed to automate this io-irq configuring, but for Linux at present, it has made life more complicated. The standard kernels for Linux don't support plug-and-play very well. If you use a patch to the Linux kernel to covert it to a plug-and-play operating system, then all of the above should be handled automatically by the OS. But when you want to use this to automate configuring devices other that the serial port, you may find that you'll still have to configure the drivers manually since many Linux drivers are not written to support a Linux PnP OS. If you use `isapnptools` or the BIOS for configuring plug-and-play this will only put the two values into the registers of the serial port section of the modem card and you will likely still need to set up setserial. None of this is easy or very well documented as of early 1999. See Plug-and-Play-HOWTO and the isapnptools FAQ.

Using a PnP BIOS to I0-IRQ Configure

While the explanation of how to use a PnP OS or isapnp for io-irq configuring should come with such software, this is not the case if you want to let a PnP BIOS do such configuring. Not all PnP BIOS can do this. The BIOS usually has a CMOS menu for setting up the first two serial ports. There is often little to choose from. Unless otherwise indicated in menus, these first two ports get set at the standard IO addresses and IRQs. See Serial Port Device Names & Numbers

Whether you like it or not, when you start up a PC a PnP BIOS starts to do PnP (io-irq) configuring of hardware devices. It may do the job partially and turn the rest over to a PnP OS (which you probably don't have) or if thinks you don't have a PnP OS it may fully configure all the PnP devices but not configure the device drivers. This is what you want but it's not always easy to figure out exactly what the PnP BIOS has done.

If you tell the BIOS that you don't have a PnP OS, then the PnP BIOS should do the configuring of all PnP serial ports --not just the first two. If you have MS Windows9x on the same PC, the BIOS might ?? have saved the io-irq configuration used for MS Windows in its non-volatile memory. If you're now reading the latest version of this HOWTO let me know if Windows puts it there. If so the BIOS will use this same configuration for Linux. In this case, if you can find out how MS Windows has set up io-irq then it should be the same under Linux.

If you add a new PnP device, the BIOS should change its PnP configuration to accommodate it. It could even change the io-irq of existing devices if required to avoid any conflicts. For this purpose, it keeps a list of

TOC

non-PnP devices provided that you have told the BIOS how these non-PnP devices are io-irq configured. (way to tell the BIOS this is by running a program called ICU under DOS/Windows.

But how do you find out what the BIOS has done so that you set up the device drivers with this info? The BIOS itself may provide some info, either in it's setup menus of via messages on the screen when you turn on your computer. See What is set in my serial port hardware?

6.8 Giving the IRQ and IO Address to Setserial

Once you've set the IRQ and IO address in the hardware (or arranged for it to be done by PnP) you also need to insure that the "setserial" command is run each time you start Linux. See the subsection Boot-time Configuration

6.9 High-level Configuring: stty, etc.

As a rule, your application program will do most (or all) of this. The command which does it is `stty`. See Stty

Configuring Flow Control: Hardware Flow Control is Best

See Flow Control for an explanation of it. It's usually better to use hardware flow control rather than software flow control using Xon/Xoff. To use full hardware flow control you must normally have two wires for it in the cable between the serial port and the device. If the device is on a card, then it should always be possible to use hardware flow control.

Many applications (and the getty program) give you an option regarding flow control and will set it for you. It might even set hardware flow control by default. Like the IRQ and IO address, it must be set both in the serial driver and the hardware connected to the serial port. How it's set into the hardware is hardware dependent. Often there is a certain "init string" you send to the hardware device via the serial port from your PC. For a modem, the communication program should set it in both places.

If a program you use doesn't set flow control in the serial driver, then you may do it yourself using the `stty` command. Since the driver doesn't remember the setting after you stop Linux, you could put the stty command in a file that runs at start-up or when you login (such as /etc/profile for the bash shell). Heres what you would add for hardware flow control for port ttyS2:

```
stty crtscts < /dev/ttyS2
or for stty version >= 1.17:
stty -F /dev/ttyS2 crtscts
```

`crtscts` stands for a Control setting to use the RTS and CTS pins of the serial port for hardware flow control. The upper case letters of the last sentence spell: `crtscts`.

---

7. Serial Port Devices /dev/ttyS2, etc.

For creating devices in the device directory see

Creating Devices In the /dev directory

7.1 Serial Port Device Names & Numbers

Devices in Linux have major and minor numbers. Each serial port may have 2 possible names in the /dev directory: ttyS and cua. Their drivers behave slightly differently. The cua device is deprecated and will not be used in the future. See the Modem-HOWTO section: "The cua Device".

Dos/Windows use the COM name while the `setserial` program uses tty00, tty01, etc. Don't confuse these with dev/tty0, dev/tty1, etc. which are used for the console (your PC monitor) but are not serial ports. The dos names (COM1, etc.) and IO address is shown below for the "standard" case (but yours could be different).

```
        set-                                                  IO
dos    serial          major minor         major minor   address
COM1   tty00  /dev/ttyS0  4,  64;  /dev/cua0  5,  64       3F8
COM2   tty01  /dev/ttyS1  4,  65;  /dev/cua1  5,  65       2F8
COM3   tty02  /dev/ttyS2  4,  66;  /dev/cua2  5,  66       3E8
COM4   tty03  /dev/ttyS3  4,  67;  /dev/cua3  5,  67       2E8
```

Note that all distributions should come with ttyS devices (and cua devices until cua is finally abolished). You can verify this by typing:

```
linux% ls -l /dev/cua*
linux% ls -l /dev/ttyS*
```

7.2 Link ttySN to /dev/modem ?

On some installations, two extra devices will be created, `/dev/modem` for your modem and `/dev/mouse` for your mouse. Both of these are symbolic links to the appropriate device in `/dev` which you specified during the installation (unless you have a bus mouse, then `/dev/mouse` will point to the bus mouse device).

There has been some discussion on the merits of `/dev/mouse` and `/dev/modem`. The use of these links is discouraged. In particular, if you are planning on using your modem for dialin you may run into problems because the lock files may not work correctly if you use `/dev/modem`. Use them if you like, but *be sure they point to the right device*. However, if you change or remove this link, some applications might need reconfiguration.

TOC

7.3 Notes For Multiport Boards

For board addresses, and IRQs, look at the `rc.serial` or `/etc/rc.boot/0setserial` that comes with the `setserial` program. It has a lot of detail on multiport boards, including I/O addresses and device names.

7.4 Creating Devices In the /dev directory

If you don't have a device, you will have to create it with the `mknod` command. Example, suppose you needed to create devices for `ttyS0`:

```
linux# mknod -m 666 /dev/cua0 c 5 64
linux# mknod -m 666 /dev/ttyS0 c 5 64
```

You can use the `MAKEDEV` script, which lives in `/dev`. See the man page for it. This simplifies the making of devices. For example, if you needed to make the devices for `ttyS0` you would type:

```
linux# cd /dev
linux# ./MAKEDEV ttyS0
```

This handles the devices creation and should set the correct permissions.

---

8. Interesting Programs You Should Know About

Most info on getty has been moved to Modem-HOWTO with a little info on the use of getty with directly connected terminals now found in Text-Terminal-HOWTO.

8.1 Serial Monitoring/Diagnostics Programs

A few Linux programs will monitor the modem control lines and indicate if they are positive (1) or negative (0).

- statserial
- modemstat (only works on Linux PC consoles. Will coexist with command line)
- serialmon (doesn't monitor RTS, CTS, DSR but logs other functions)

You may already have them. If not, download them from Serial Software. As of June 1998, I know of no diagnostic program in Linux for the serial port.

8.2 Changing Interrupt Priority

- `irqtune` will give serial port interrupts higher priority to improve performance.
- `hdparm` for hard-disk tuning may help some more.

8.3 What is Setserial ?

Introduction

Don't ever use `setserial` with Laptops (PCMCIA). `setserial` is a program which allows you to tell the device driver software the I/O address of the serial port, which interrupt (IRQ) is set in the port's hardware, what type of UART you have, etc. It can also show how the driver is currently set. In addition, it can probe the hardware (if certain options are given). Besides the manual for `setserial`, check out info in `/usr/doc/setserial.../` or the like. It should tell you how setserial is handled in your distribution of Linux.

`Setserial` is often run automatically at boot-time by a start-up shell-script. It will only work if the serial module is loaded. If you should for some reason unload the serial module later on, the changes previously made by `setserial` will be forgotten. So `setserial` must be run again to reestablish them. In addition to running via a start-up script, something akin to `setserial` runs when the serial module is loaded. Thus when you watch the start-up messages on the screen it may look like it ran twice, and in fact it has.

m4_define TorS

TOC

With appropriate options, `setserial` can probe (at a given I/O address) for a serial port but you may guess the I/O address. If you ask it to probe for /dev/ttyS2 for example, it will only probe at the address it thinks ttyS2 is at. If you tell setserial that ttyS2 is at a different address, then it will probe at that address, etc. See Probing

Setserial does not set either IRQ's nor I/O addresses in the serial port hardware itself. You must tell setserial the identical values that have been set in the hardware. It's set in the hardware either by jumpers or by plug-and-play. Do not just invent some values that you think would be nice to use and tell them to setserial. However, if you know the I/O address but don't know the IRQ you may command setserial to attempt to determine the IRQ.

You can see a list of possible commands to use (but not the one-letter options such as -v for verbose --which you should normally use when troubleshooting) by just typing `setserial` with no arguments. Note that setserial calls an I/O address a "port". If you type:

```
setserial-g /dev/ttyS*
```

you'll see some info about how that device driver is configured for your ports. Add a "v" to the option "-g" to see more. But this doesn't tell you if the hardware actually has these values set in it. If fact, you can run setserial and assign a purely fictitious I/O address, any IRQ, and whatever uart type you would like to have. Then the next time you type "setserial ..." it will display these bogus values without complaint. Note that assignments made by setserial are lost when the PC is powered down so it is usually run automatically somewhere each time that Linux is booted.

Probing

In order to try to find out if you have a certain piece of serial hardware you must first know (or guess) its I/O address (or the device driver must have an I/O address for it, likely previously set by setserial). To try to detect the physical hardware use the -v (verbose) and `autoconfig` command to `setserial`. If the resulting message shows a uart type such as 16550A, then you're OK. If instead it shows "`unknown`" for the uart type, then there is likely no serial port at all at that I/O address. Some cheap serial ports don't identify themselves correctly so if you see "`unknown`" you still might have something there.

Besides auto-probing for uart type, setserial can auto-probe for IRQ's but this doesn't always work right either. In versions of setserial >= 2.15, your last probe test may be saved and put into the configuration file `/etc/serial.conf` which will be used next time you start Linux. The script that runs `setserial` at boot-time does not usually probe, but you could change it so that it does. See the next section.

Can Linux Configure The Serial Devices Automagically?

Yes, but ... You may set Linux up to detect and set up the serial devices automatically on startup (probing). It's easy to do for setserial < 2.15. Just add some lines to the file that runs setserial on start-up. See Old configuration method: edit a script For example, for ttyS3 you would add:

```
/sbin/setserial /dev/ttyS3 auto_irq skip_test autoconfig
```

to the file that runs setserial on startup. Do this for every serial port you want to auto configure. Be s to give a device name that really does exist on your machine.

For versions >= 2.15 (provided your distribution implemented the change, Redhat didn't) it's much harder to do since the file that runs setserial on startup, /etc/init.d/setserial or the like was not intended to be edited by the user. There may be no helpful comments in it like there were in earlier versions.

Boot-time Configuration

When the kernel loads the serial module (or if the "module" is built into the kernel) then only `ttyS{0-3}` are auto-detected and the driver is set to IRQs 4 and 3 (regardless of what the hardware is actually set at). You see this as a boot-time message just like as if `setserial` had been run.. If you use 3 or more ports, this may result in IRQ conflicts.

To fix such conflicts by telling setserial the true IRQs (or for other reasons) there may be a file somewhere that runs `setserial` again. This happens early at boot-time before any process uses the serial port. In fact, your distribution may have set things up so that the setserial program runs automatically from a start-up script at boot-time. More info about how to handle this situation should be found in /usr/doc/setserial.../ or the like.

New configuration method using /etc/serial.conf

Starting with version 2.15 (1999) of `setserial` the shell-script that runs from a file at boot-time gets its data from a configuration file: /etc/serial.conf. But serial.conf is normally never edited. Instead just use `setserial` on the command line. Normally, what you changed with the setserial command is saved to the configuration file (serial.conf) when you shutdown (normally) or reboot. This only works if "###AUTOSAVE### or the like is on the first line of serial.conf. If you should use `setserial` experimentally and it doesn't work out right, then don't forget to redo it so that the experimental settings don't get saved by mistake. The file most commonly used to run setserial at boot-time (in conformance with the configuration file) is now /etc/init.d/setserial (Debian) or /etc/init.d/serial (Redhat), or etc., but it also should not normally be edited.

To disable a port, use `setserial` to set it to "uart none". The format of /etc/serial.conf appears to be just like that of the parameters placed after "setserial" on the command line with one line for each port. If you don't use autosave, you may edit /etc/serial.conf manually. For 2.15, the Debian distribution installs the system with autosave enabled, but Redhat 6.0 just had a file /usr/doc/setserial-2.15/rc.serial which you have to move to /etc/init.d/.

BUG: As of July 1999 there is a bug/problem since with ###AUTOSAVE### only the setserial parameters displayed by "setserial -G /dev/ttyS?" (where ? = 0, 1, 2, ...) get saved but the other parameters don't get saved. This will only affect a small minority of users since the parameters not saved are seldom used anyway. It's been reported as a bug and may be fixed by now.

In order to force the current settings set by setserial to be saved to the configuration file (serial.conf) without shutting down, do what normally happens when you shutdown: Run the shell-script `/etc/init.d/{set}serial stop`. The "stop" command will save the current configuration but the serial ports still keep working OK.

TOC

Old configuration method: edit a script

Prior to 2.15 (1999), or if your distribution chose not to use /etc/serial.conf then you need to look for a file that runs setserial at boot-time. If it doesn't exist, you need to create one (or place the commands in a file that runs early at boot-time). If such a file is currently being used it's likely somewhere in the /etc directory-tree. But Redhat has supplied it in /usr/doc/setserial/ although you need to move it to the /etc tree before using it. You might use "locate" to try to find such a file. For example, you could type: locate "*serial*".

What you are looking for could be named rc.serial, or 0setserial (Debian). If such a file is supplied, it should contain a number of commented-out examples. By uncommenting some of these and/or modifying them, you should be able to set things up correctly. Make sure that you are using a valid path for `setserial`, and a valid device name. You could do a test by executing this file manually (just type its name as the super-user) to see if it works right. Testing like this is a lot faster than doing repeated reboots to get it right. Of course you can also test a single `setserial` command by just typing it on the command line.

The script `/etc/rc.d/rc.serial` was commonly used in the past. The Debian distribution used `/etc/rc.boot/0setserial`. Another file once used is `/etc/rc.d/rc.local` but it's not a good idea to use this since it may not be run early enough. It's been reported that other processes may try to open the serial port before rc.local runs resulting in serial communication failure.

IRQs

By default, both ttyS0 and ttyS2 share IRQ 4, while ttyS0 and ttyS3 share IRQ 3. But sharing serial interrupts is not permitted unless you: 1. have kernel 2.2 or better, and 2. you've complied in support for this, and 3. your serial hardware supports it. See

Interrupt sharing and Kernels 2.2+

If you only have two serial ports, ttyS0 and ttyS1, you're still OK since IRQ sharing conflicts don't exist for non-existent devices.

If you add an internal modem and retain ttyS0 and ttyS1, then you should attempt to find an unused IRQ and set it both on your serial port (or modem card) and then use setserial to assign it to your device driver. If IRQ 5 is not being used for a sound card, this may be one you can use for a modem. To set the IRQ in hardware you may need to use isapnp, a PnP BIOS, or patch Linux to make it PnP. To help you determine which spare IRQ's you might have, type "man setserial" and search for say: "IRQ 11".

----------From TT-HOWTO----------

Setserial tells the serial port driver both the interrupt (IRQ) number and the port IO address. It can tell you what type of UART chips you have if you use the autoconfig parameter. For the serial port to work, the serial module (for the Linux kernel) must be loaded. If you use more than 2 serial ports and want to assign unique IRQ numbers (or use more than 4 serial ports), then you need to use setserial. If your serial port is plug-and-play you should look in the Serial-HOWTO. In addition to the man page for it, but sure to look at /usr/doc/setserial/ if it exists. It may tell you how setserial is run automatically at boot time and how to alter the configuration. Don't ever use setserial with Laptops (PCMCIA bus).

TOC

starting with version 2.15 (1999) of `setserial` changes made to setserial on the command line are sometimes automatically saved so that they will remain after you shut off the computer.

Where to Run Setserial ?

Starting with version 2.15 of `setserial` the method of configuration changed. Before 2.15 you configured it by editing a shell-script that ran `setserial`. After 2.15 there's a configuration file /etc/serial.conf which is automatically edited to conform with whatever you do with the `setserial` command.

You may find you have both the old and new way of configuring installed. If the shell-script is named "...pre-2.15" or the like, it's probably not being used anymore so everything should be OK if you keep it.

Prior to version 2.15

Prior to 2.15 there was no configuration file for `setserial`. One needs to edit the shell-script that runs setserial at boot-time. You need to find it or create one if it doesn't exist. Where it is depends on your distribution. It might be in a file named "rc.serial" . In the Debian distribution it was the 0setserial file in /etc/rc.boot. One way to try to find it is to search for it with the "locate" command.

Version 2.15 and later

You neither edit the file that runs setserial nor the configuration file. Changes made by using `setserial` on the command line are *sometimes* saved in /etc/serial.conf during normal shutdown. This feature is called autosave. It only works if the first line of serial.conf contains ###AUTOSAVE### or the like.

There's a bug I've reported in 2.15 and it's been fixed (but you could still have it): With the bug, seldom used changes made by setserial don't get saved. Only the most important parameters that you see if you give the command "setserial -g /dev/ttyS?" do get saved.

If autosave is enabled, then each time you boot, setserial is run by /etc/init.d/setserial (or "serial", etc.) using the saved configuration.

8.4 Stty

Introduction

`stty` does much of the configuration of the serial port but since application programs (and the getty program) often handle it, you may not need to use it much. It's handy if your having problems or want to see how the port is set up. Try typing ``stty -a" at your terminal/console to see how it's now set. Also try typing it without the -a (all) for a short listing which shows how it's set different than normal. Don't try to learn all the setting unless you want to become a serial guru. Most of the defaults should work OK and some of the settings are needed only for certain obsolete dumb terminals made in the 1970's. Dumb terminals made after about 1980 don't use them.

Whereas `setserial` only deals with actual serial ports, stty is used both for serial ports and for vir terminals such as the standard Linux text interface at a PC monitor. For the PC monitor, many of the stty settings are meaningless. Changing the baud rate, etc. doesn't appear to actually do anything.

Here are some of the items stty configures: speed (bits/sec), parity, bits/byte, # of stop bits, strip 8th bit?, modem control signals, flow control, break signal, end-of-line markers, change case, padding, beep if buffer overrun?, echo, allow background tasks to write to terminal?, define special (control) characters (such as what key to press for interrupt). See the `stty` man or info page for more details. Also see the man page: `termios` which covers the same options set by stty but (as of mid 1999) covers features which the stty man page fails to mention.

With some implementations of getty (getty_ps package), the commands that one would normally give to stty are typed into a getty configuration file: /etc/gettydefs. Even without this configuration file, the getty command line may be sufficient to set things up so that you don't need stty.'")

One may write C programs which change the stty configuration, etc. Looking at some of the documentation for this may help one better understand the use of the stty command (and its many possible arguments). Serial-Programming-HOWTO is useful. The manual page: termios contains a description of the C-language structure (of type termios) which stores the stty configuration in computer memory. Many of the flag names in this C-structure are almost the same (and do the same thing) as the arguments to the stty command.

Using stty for another port

Using `stty` to inspect or configure another terminal or serial port is tricky. For example, let's say you are at the PC monitor (tty1) and want to use `stty` for the serial port ttyS2. You need to use the redirection operator <. First, be warned that if there is a terminal on ttyS2 and a shell running on that terminal, then what you see will likely be deceptive and trying to set it will not work. See Two Interfaces at a Terminal to understand it.

Type ``stty -a < /dev/ttyS2'' to look at the settings of ttyS2. Use the same redirection operator < to set ttyS2. Some people try to use ``stty ... > /dev/ttyS2 to set the terminal. This will not do it. Instead, it takes the message normal displayed by the stty command for the terminal you are on (tty1) and sends this message to ttyS2 but it doesn't change any settings for ttyS2.

Versions starting with 1.17 (still not released as of mid 1999) will no longer need to use redirection (<) but instead will use ``stty ... -F /dev/ttyS2'' (or --file instead of F) etc.

Two interfaces at a terminal

When using a shell (such as bash) with command-line-editing enabled there are two different terminal interfaces (what you see when you type stty -a). When you type at the command line you have a temporary "raw" interface (or raw mode) where each character is read by the command-line-editor as you type it. Once you hit the <return> key, the command-line-editor is exited and the terminal interface is changed to the nominal "cooked" interface (cooked mode) for the terminal. This "cooked" mode lasts until the next prompt is sent to the terminal. Note that one never types anything to this "cooked" mode but it does have a purpose.

When a prompt is sent to the terminal the terminal goes from "cooked" to "raw" mode (just like it do when you start an editor since you are starting the command-line editor). The settings for the "raw" mode are based only on the basic settings taken from the "cooked" mode. Raw mode keeps these setting but changes several other settings in order to change the mode to "raw". It is not at all based on the settings used in the previous "raw" mode. Thus if one uses stty to change settings for the raw mode, such settings will be lost as soon as one hits the <return> key at the terminal that has supposedly been "set".

Now when one types stty to look at the terminal interface, one may either get a view of the cooked mode or the raw mode. You need to figure out which one you're looking at. It you use stty from another terminal to deal with a terminal that is displaying a command line, then the view is that of the raw mode. Any changes made will only be made to the raw mode and will be lost when someone presses <return> at the terminal you tried to "set". But if you type a stty command at your terminal (without using < for redirection) and then hit <return> it's a different story. The <return> puts the terminal in cooked mode. Your changes are saved and will still be there when the terminal goes back into raw mode (unless of course it's a setting not allowed in raw mode).

This situation can create problems. For example, suppose you corrupt your terminal interface and to restore it you go to another terminal and "stty sane <dev/ttyS1" to restore it. It will not work! Of course you can try to type "stty sane ..." at the terminal that is corrupted but you can't see what you typed. All the above not only applies to dumb terminals but to virtual terminals used on a PC Monitor as well as to the terminal windows in X. In other words, it applies to almost everyone who uses Linux. Luckily, a file that runs stty at boot-time will likely deal with a terminal (or serial port with no terminal) that has no shell running on it so there's no problem.

Where to put the stty command ?

Should you need to have `stty` set up the serial interface each time the computer starts up then you need to put the `stty` command in a file that will be executed each time the computer is started up (Linux boots). It should be run before the serial port is used (including running getty on the port). There are many possible places to put it. If it gets put in more than one place and you only know about (or remember) one of those places, then a conflict is likely. So make sure to document what you do.

One good place to put it would be in the same file that runs setserial when the system is booted. See Where to Run Setserial?. It would seem best to put it after the setserial command so that the low level stuff is done first.

8.5 What is isapnp ?

`isapnp` is a program to configure Plug-and-Play (PnP) devices on the ISA bus including internal modems. It comes in a package called "isapnptools" and includes another program, "pnpdump" which finds all your ISA PnP devices and shows you options for configuring them in a format which may be added to the PnP configuration file: /etc/isapnp.conf. The isapnp command may be put into a startup file so that it runs each time you start the computer and thus will configure ISA PnP devices. It is able to do this even if your BIOS doesn't support PnP. See Plug-and-Play-HOWTO.

---

9. Speed (Flow Rate)

By "speed" we really mean the "data flow rate" but almost everybody incorrectly calls it speed. The speed is measured in bits/sec (or baud). Speed is set using the "stty" command or by a program which uses the serial port. See Stty

9.1 Can't Set a High Enough Speed

You need to find out the highest speed supported by your hardware. As of late 1998 most hardware only supported speeds up to 115.2K bps. A few 56K internal modems support 230.4K bps. Recent Linux kernels support high speeds (over 115.2K) but you might have difficulty using it because of one or both of the following reasons:

1. The application program (or stty) will not accept the high speed.
2. Setserial has a default speed of 115,200 (but this default is easy to change)

How speed is set in hardware: the divisor and baud_base

Here's a list of commonly used divisors and their corresponding speeds (assuming a maximum speed of 115,200): 1 (115.2K), 2 (57.6K), 3 (38.4K), 6 (19.2K), 12 (9.6K), 24 (4.8K), 48 (2.4K), 96 (1.2K), etc. The serial driver sets the speed in the hardware by sending the hardware only a "divisor" (a positive integer). This "divisor" divides the maximum speed of the hardware resulting in a slower speed (except a divisor of 1 obviously tells the hardware to run at maximum speed).

Normally, if you specify a speed of 115.2K (in your communication program or by stty) then the serial driver sets the port hardware to divisor 1 which obviously sets the highest speed. If you happen to have hardware with a maximum speed of say 230.4K, then specifying 115.2K will result in divisor 1 and will actually give you 230.4K. This is double the speed that you set. In fact, for any speed you set, the actual speed will be double. If you had hardware that could run at 460.8K then the actual speed would be quadruple what you set.

Work-arounds for setting speed

To correct this accounting (but not always fix the problem) you may use "setserial" to change the baud_base to the actual maximal speed of your port such as 230.4K. Then if you set the speed (by your application or by stty) to 230.4K, a divisor of 1 will be used and you'll get the same speed as you set. PROBLEM: stty and many communication programs (as of mid 1999) still have 115.2K as their maximum speed setting and will not let you set 230.4K, etc. So in these cases one solution is not to change anything with `setserial` but mentally keep in mind that the actual speed is always double what you set.

There's another work-around which is not much better. To use it you set the baud_base (with setserial) to

the maximal speed of your hardware. This corrects the accounting so that if you set say 115.2K you actually get 115.2K. Now you still have to figure out how to set the highest speed if your communication program (or the like) will not let you do it. Fortunately, setserial has a way to do this: use the "spd_cust" parameter with "divisor 1". Then when you set the speed to 38400 in a communication program, the divisor will be set to 1 in the port and it will operate at maximum speed. For example:
setserial /dev/ttyS2 spd_cust baud_base 230400 divisor 1
Don't try using "divisor" for any other purpose other than the special use illustrated above (with spd_cust).

If there are two or more high speeds that you want to use that your communication program can't set, then it's not quite as easy as above. But the same principles apply. You could just keep the default baud_base and understand that when you set a speed you are really only setting a divisor. So your actual speed will always be your maximum speed divided by whatever divisor is set by the serial driver. See How speed is set in hardware: the divisor and baud_base

Crystal frequency is not baud_base

Note that the baud_base setting is usually much lower than the frequency of the crystal oscillator in the hardware since the crystal frequency is often divided by 16 in the hardware to get the actual top speed. The reason the crystal frequency needs to be higher is so that this high crystal speed can be used to take a number of samples of each bit to determine if it's a 1 or a 0.

9.2 Higher Serial Throughput

If you are seeing slow throughput and serial port overruns on a system with (E)IDE disk drives, you can get `hdparm`. This is a utility that can modify (E)IDE parameters, including unmasking other IRQs during a disk IRQ. This will improve responsiveness and will help eliminate overruns. Be sure to read the man page very carefully, since some drive/controller combinations don't like this and may corrupt the filesystem.

Also have a look at a utility called `irqtune` that will change the IRQ priority of a device, for example the serial port that your modem is on. This may improve the serial throughput on your system. The `irqtune` FAQ is at http://www.best.com/~cae/irqtune

---

10. Communications Programs And Utilities

10.1 List of Software

Here is a list of some communication software you can choose from, available via FTP, if they didn't come with your distribution.

- `ecu` - a communications program
- C-Kermit - portable, scriptable, serial and TCP/IP communications including file transfer, character-set translation, and zmodem support
- `minicom` - `telix`-like communications program
- `procomm` - `procomm`-like communications program with zmodem
- `seyon` - X based communication program
- `xc` - xcomm communication package
- `term` and `SLiRP` offer TCP/IP functionality using a shell account.
- `screen` is another multi-session program. This one behaves like the virtual consoles.
- `callback` is where you dial out to a remote modem and then that modem hangs up and calls you back (to save on phone bills).
- `mgetty+fax` handles FAX stuff, and provides an alternate `ps_getty`.
- `ZyXEL` is a control program for ZyXEL U-1496 modems. It handles dialin, dialout, dial back security, FAXing, and voice mailbox functions.
- SLIP and PPP software can be found at `ftp://metalab.unc.edu/pub/Linux/system/network/serial`.

10.2 kermit and zmodem

To use zmodem with `kermit`, add the following to your `.kermrc`:

```
define rz !rz < /dev/ttyS3 > /dev/ttyS3
define sz !sz \%0 > /dev/ttyS3 < /dev/ttyS3
```

Be sure to put in the correct port your modem is on. Then, to use it, just type `rz` or `sz <filename>` at the `kermit` prompt.

11. Serial Tips And Miscellany

Here are some serial tips you might find helpful...

11.1 Line Drivers

For a text terminal, the EIA-232 speeds are fast enough but the usable cable length is often too short. Balanced technology could fix this. The common method of obtaining balanced communication with a text terminal is to install 2@ line drivers in the serial line to convert unbalanced to balanced (and conversely). They are a specialty item and are expensive if purchased new.

11.2 Known Defective Hardware

Avoiding IO Address Conflicts with Certain Video Boards

The IO address of the IBM 8514 video board (and others) is allegedly 0x?2e8 where ? is 2, 4, 8, or 9. This may conflict (but shouldn't it the serial port is well designed) with the IO address of `ttyS3` at 0x02e8 if the serial port ignores the leading 0 hex digit when it decodes the address (many do). That is bad news if you try to use `ttyS3` at this IO address. Another story is that Linux will not detect your internal modem on `ttyS3` but that you can use `setserial` to put `ttyS3` at this address and the modem will work fine.

Problem with AMD Elan SC400 CPU (PC-on-a-chip)

This has a race condition between an interrupt and a status register of the UART. An interrupt is issued when the UART transmitter finishes the transmission of a byte and the UART transmit buffer becomes empty (waiting for the next byte). But a status register of the UART doesn't get updated fast enough to reflect this. As a result, the interrupt service routine rapidly checks and determines (erroneously) that nothing has happened. Thus no byte is sent to the port to be transmitted and the UART transmitter waits in vain for a byte that never arrives. If the interrupt service routine had waited just a bit longer before checking the status register, then it would have been updated to reflect the true state and all would be OK.

There is a proposal to fix this by patching the serial driver. But Should linux be patched to accommodate defective hardware, especially if this patch may impair performance of good hardware?

11.3 What Are Lock Files ?

TOC

A lock file is simply a file saying that a particular device is in use. They are kept in `/var/lock`. Formerly they were in `/usr/spool/uucp`. Linux lock files are named `LCK..`*name*, where *name* is either a device name, or a UUCP site name. Certain processes create these locks so that they can have exclusive access to devices. For instance if you dial out on your modem, a lock will appear telling other processes that someone is using the modem already. Locks mainly contain the PID of the process that has locked the device. Most programs look at the lock, and try to determine if that lock is still valid by checking the process table for the process that has locked the device. If the lock is found to be valid, the program (should) exit. If not, some programs remove the stale lock, and use the device, creating their own lock in the process. Other programs just exit and tell you that the device is in use.

Having the same physical serial port known by two different device names (such as ttyS0 and cua0) could cause problems. The lock checking software is aware of ttyS vs. cua but it will make things simpler in this regard by the planned elimination of cua. In other cases, assigning an alternate name to the same device is asking for trouble.

---

12. Troubleshooting

See Modem-HOWTO for troubleshooting related to modems or getty for modems.

12.1 Serial Electrical Test Equipment

Breakout Gadgets, etc.

While a multimeter (used as a voltmeter) may be all that you need for just a few terminals, simple special test equipment has been made for testing serial port lines. Some are called "breakout ... " where breakout means to break out conductors from a cable. These gadgets have a couple of connectors on them and insert into the serial cable. Some have test points for connecting a voltmeter. Others have LED lamps which light when certain modem control lines are asserted (turned on). Still others have jumpers so that you can connect any wire to any wire. Some have switches.

Radio Shack sells (in 1998) a "RS-232 Troubleshooter" or "RS-232 Line Tester" which checks TD, RD, CD, RTS, CTS, DTR, and DSR. A green light means on (+12 v) while red means off (-12 v). They also sell a "RS-232 Serial Jumper Box" which permits connecting the pins anyway you choose.

Measuring Voltages

Any voltmeter or multimeter, even the cheapest that sells for about $10, should work fine. Trying to use other methods for checking voltage is tricky. Don't use a LED unless it has a series resistor to reduce the voltage across the LED. A 470 ohm resistor is used for a 20 ma LED (but not all LED's are 20 ma). The LED will only light for a certain polarity so you may test for + or - voltages. Does anyone make such a gadget for automotive circuit testing?? Logic probes may be damaged if you try to use them since the TTL voltages for which they are designed are only 5 volts. Trying to use a 12 V incandescent light bulb is not a good idea. It won't show polarity and due to limited output current of the UART it probably will not even light up.

To measure voltage on a female connector you may plug in a bent paper clip into the desired opening. The paper clip's diameter should be no larger than the pins so that it doesn't damage the contact. Clip an alligator clip (or the like) to the paper clip to connect up.

Taste Voltage

As a last resort, if you have no test equipment and are willing to risk getting shocked (or even electrocuted) you can always taste the voltage. Before touching one of the test leads with your tongue, test them to make sure that there is no high voltage on them. Touch both leads (at the same time) to one hand to see if they shock you. Then if no shock, wet the skin contact points by licking and repeat. If this test gives you a shock, you certainly don't want to use your tongue.

TOC

For the test for 12 V, Lick a finger and hold one test lead in it. Put the other test lead on your tongue. the lead on your tongue is positive, there will be a noticeable taste. You might try this with flashlight batteries first so you will know what taste to expect.

12.2 Serial Monitoring/Diagnostics

A few Linux programs will monitor the modem control lines and indicate if they are positive (1) or negative (0). See section Serial Monitoring/Diagnostics

12.3 The following subsections are in both the Serial and Modem HOWTOs:

12.4 My Serial Port is Physically There but Can't be Found

Check the BIOS menus and BIOS messages. If it's an ISA bus PnP serial port, try "pnpdump --dumpregs" and/or see Plug-and-Play-HOWTO. For the PCI bus look at /proc/pci. You may try probing with setserial. See Probing If nothing seems to get thru the port it may be there but have a bad interrupt. See Slow: Text appears on the screen slowly after long delays

12.5 Slow: Text appears on the screen slowly after long delays

It's likely mis-set/conflicting interrupts. Here are some of the symptoms which will happen the first time you try to use a modem, terminal, or printer. In some cases you type something but nothing appears on the screen until many seconds later. Only the last character typed may show up (and it may be just an invisible <return> character but you notice the cursor jumps down one line). In other cases where a lot of data should appear on the screen, only a batch of about 16 characters appear. Then there is a long wait of many seconds for the next batch of characters. You might also get "input overrun" error messages (or find them in logs).

For more details on the symptoms and why this happens see

Interrupt Problem Details and/or Interrupt Conflicts and/or Mis-set Interrupts.

As a final check to see if it really is an interrupt problem, set the IRQ to zero with "setserial". This will tell the driver not to use interrupts but to use inefficient (but faster) polling methods. If this seemingly fixes the "slow" problem then you had an interrupt problem but should still try to fix it since polling uses excessive computer resources.

Checking to find the interrupt conflict may not be easy since looking in the /proc directory may be misleading. Make sure there are no IRQs being illegally shared. Check all your boards (serial, ethernet,

SCSI, etc...). Make sure the jumper (or PnP) settings, and the `setserial` parameters are correct for your serial devices. Also check `/proc/ioports` and `/proc/interrupts` and `/proc/pci` for conflicts. For more details see Serial-HOWTO: Interrupt Problem Details. If it involves Plug-and-Play devices, see also Plug-and-Play-HOWTO.

12.6 The Startup Screen Show Wrong IRQs for the Serial Ports.

Linux does not do any IRQ detection on startup. When the serial module loads it only does serial device detection. Thus, disregard what it says about the IRQ, because it's just assuming the standard IRQs. This is done, because IRQ detection is unreliable, and can be fooled. But if and when setserial runs from a start-up script, it changes the IRQ's and displays the new (and hopefully correct) state on on the startup screen. If the wrong IRQ is not corrected by a later display on the screen, then you've got a problem.

So, even though I have my `ttyS2` set at IRQ 5, I still see

```
ttyS02 at 0x03e8 (irq = 4) is a 16550A
```

at first when Linux boots. (Older kernels may show "ttyS02" as "tty02") You have to use `setserial` to tell Linux the IRQ you are using.

12.7 "Cannot open /dev/ttyS?: Permission denied"

Check the file permissions on this port with "ls -l /dev/ttyS?"_ If you own the ttyS? then you need read and write permissions: crw with the c (Character device) in col. 1. It you don't own it then it should show rw- in cols. 8 & 9 which means that everyone has read and write permission on it. Use "chmod" to change permissions. There are more complicated ways to get access like belonging to a "group" that has group permission.

12.8 "Operation not supported by device" (error message) for ttySx

This means that an operation requested by setserial, stty, etc. couldn't be done because the kernel doesn't support doing it. Formerly this was often due to the "serial" module not being loaded. But with the advent of PnP, it may likely mean that there is no modem (or other serial device) at the address where the driver (and setserial) thinks it is. If there is no modem there, commands (for operations) sent to that address obviously don't get done. See What is set in my serial port hardware?

If the "serial" module wasn't loaded but "lsmod" shows you it's now loaded it might be the case that it's loaded now but wasn't loaded when you got the error message. In many cases the module will automatically loaded when needed (if it can be found). To force loading of the "serial" module it may be listed in the file: /etc/modules.conf or /etc/modules. The actual module should reside in: /lib/modules/.../misc/serial.o.

12.9 "Cannot create lockfile. Sorry" (error messag

When a port is "opened" by a program a lockfile is created in /var/lock/. Wrong permissions for the lock directory will not allow a lockfile to be created there. Use "ls -ld /var/lock" to see if the permissions are OK: usually rwx for everyone (repeated 3 times). If it's wrong, use "chmod" to fix it. Of course, if there is no "lock" directory no lockfile can be created there. For more info on lockfiles see What Are Lock Files

12.10 Software which may help

- `modemstat` and `statserial` show the current state of various modem signal lines (such as DTR, CTS, etc.)
- `irqtune` will give serial port interrupts higher priority to improve performance.
- `hdparm` for hard-disk tuning may help some more.

---

13. Interrupt Problem Details

While the section Troubleshooting lists problems by symptom, this section explains what will happen if interrupts are set incorrectly. This section helps you understand what caused the symptom, what other symptoms might be due to the same problem, and what to do about it.

13.1 Symptoms of Mis-set or Conflicting Interrupts

The symptoms depend on whether or not you have a modern serial port with FIFO buffers or an obsolete serial port without FIFO buffers. It's important to understand the symptoms for the obsolete ones also since sometimes modern ports seem to behave that way.

For the obsolete serial ports, only one character gets thru every several seconds. This is so slow that it seems that nothing is working (especially if the character that gets thru is invisible (such a space or newline). For the modern ports with FIFO buffers you will likely see bursts of up to 16 characters every several seconds.

If you have a modem on the port and dial a number, it seemingly doesn't connect since the CONNECT message doesn't make it thru. But after a long wait it finally does connect and you finally do see part of a login message (or the like).

If you use minicom, a common test to see if things are working is to type the simplest "AT" command and see if the modem responds. Typing just at<enter> should normally (if interrupts are OK) result in an "OK" response from the modem. With bad interrupts you type at<enter> and may see nothing. But then after 10 seconds or so you see the cursor drop down one line. What is going on is that the FIFO is behaving like it can only hold one byte. The "at" you typed caused it to overrun and both letters were lost. But the final <enter> got thru since you waited for it and you "see" this invisible character by noting that the cursor jumped down one line. If you were to type a single letter and then wait about 10 seconds, you should see it echo back to the screen. This is fine if your typing speed is less that one word per minute :-)

13.2 Mis-set Interrupts

If you don't understand what an interrupt does see Interrupts. If a serial port has one IRQ set in the hardware but a different one set in the device driver, the device driver will not receive any interrupts sent by the serial port. Since the serial port uses interrupts to tell its driver when it needs service (fetching bytes from it's 16-byte receive buffer or putting another 16-bytes in its transmit buffer) one might expect that the serial port would not work at all.

But it still may work anyway --sort of. Why? Well, besides the interrupt method of servicing the port there's a slow polling method that doesn't need interrupts. The way it works is that every so often the device driver checks the serial port to see if it needs anything such as if it has some bytes that need

TOC

fetching from its receive buffer. If interrupts don't work, the serial driver falls back to this polling method. But this polling method was not intended to be used a substitute for interrupts. It's so slow that it's not practical to use and may cause buffer overruns. Its purpose may have been to get things going again if just one interrupt is lost or fails to do the right thing. It's also useful in showing you that interrupts have failed.

For the 16-byte transmit buffer, 16 bytes will be transmitted and then it will wait until the next polling takes place (several seconds later) before the next 16 bytes is sent out. Thus transmission is very slow and in small chunks. Receiving is slow too since bytes that are received by the receive buffer are likely to remain there for several seconds until it is polled.

This explains why it takes so long before you see what you typed. When you type say AT to a modem, the AT goes out the serial port to the modem. The modem then echos the AT back thru the serial port to the screen. Thus the AT characters have to pass twice thru the serial port. Normally this happens so fast that AT seems to appear on the screen at the same time you hit the keys on the keyboard. With polling delays thru the serial port, you don't see what you typed until many seconds later.

What about overruns of the 16-byte receive buffer? This will happen with an external modem since the modem just sends to the serial port at high speed which is likely to overrun the 16-byte buffer. But for an internal modem, the serial port is on the same card and it's likely to check that this receive buffer has room for more bytes before putting received bytes into it. In this case there will be no overrun of this receive buffer, but text will just appear on your screen in 16-byte chunks at intervals of several seconds.

Even with an external modem you might not get overruns. If just a few characters (under 16) are sent you don't get overruns since the buffer likely has room for them. But attempts to send a larger number of bytes from your modem to your screen may result in overruns. However, more than 16 (with no gaps) can get thru OK if the timing is right. For example, if 32 bytes were received (and no more bytes followed), the polling might just happen after the first 16 bytes had been received. Then there would be space for the next 16 bytes so that 32 bytes gets thru OK. Similar conditions might pass between 16 to 31 bytes thru OK. But it's also likely that only an occasional 16-byte chunk will get thru and huge gaps of missing data will be lost.

If you have an obsolete serial port with only a 1-byte buffer (or it's been incorrectly set to work like a 1-byte buffer) then the situation will be much worse than described above and only one character will occasionally make it thru the port. Every character received causes an overrun (and is lost) except for the last character received. This character is likely to be just a line-feed since this is often the last character to be transmitted in a burst of characters sent to your screen. Thus you may type AT<return> to the modem but never see AT on the screen. All you see several seconds later is that the cursor drops down one line (a line feed). This has happened to me with a 16-byte FIFO buffer that was behaving like a 1-byte buffer.

When a communication program starts up, it expects interrupts to be working. It's not geared to using this slow polling-like mode of operation. Thus all sorts of mistakes may be made such as setting up the serial port and/or modem incorrectly. It may fail to realize when a connection has been made. If a script is being used for login, it may fail (caused by timeout) due to the polling delays.

TOC

13.3 Interrupt Conflicts

When two devices have the same IRQ number it's called sharing interrupts. Under some conditions this sharing works out OK. Starting with kernel version 2.2, serial ports may, in some cases, share interrupts with other serial ports. Devices on the PCI bus may share the same IRQ interrupt with other devices on the PCI bus. In other cases where there is potential for conflict, there should be no problem if no two devices with the same IRQ are ever "in use" at the same time. More precisely, "in use" really means "open" (in programmer jargon). In cases other than the exceptions mentioned above (unless special software and hardware permit sharing), sharing is not allowed and conflicts arise if sharing is attempted.

Even if two processes with conflicting IRQs run at the same time, one of the devices will likely have its interrupts sent to its device driver and may work OK. The other device will not have its interrupts sent to the correct driver and will likely behave just like a process with mis-set interrupts. See Mis-set Interrupts for more details.

13.4 Resolving Interrupt Problems

If you are getting a very slow response as described above, then one test is to change the IRQ to 0 (uses polling instead of interrupts) and see if the problem goes away. If so, then there was likely something wrong with the interrupt and you need to fix it. Check /proc/interrupts to see if it is currently in use by another process. If it's in use by another serial port you could try "top" (type f and then enable the TTY display) or "ps -e" to find out which serial ports are in use. If you suspect that setserial has a wrong IRQ then see What is the current IO address and IRQ of my Serial Port ?

---

14. What Are UARTs? How Do They Affect Performance?

14.1 Introduction to UARTS

(This section is also in the Modem-HOWTO) UARTs (**U**niversal **A**synchronous **R**eceiver **T**ransmitter) are serial chips on your PC motherboard (or on an internal modem card). The UART function may also be done on a chip that does other things as well. On older computers like many 486's, the chips were on the disk IO controller card. Still older computer have dedicated serial boards.

The UART's purpose is to convert bytes from the PC's parallel bus to a serial bit-stream. The cable going out of the serial port is serial and has only one wire for each direction of flow. The serial port sends out a stream of bits, one bit at a time. Conversely, the bit stream that enters the serial port via the external cable is converted to parallel bytes that the computer can understand. UARTs deal with data in byte sized pieces, which is conveniently also the size of ASCII characters.

Say you have a terminal hooked up to your PC. When you type a character, the terminal gives that character to it's transmitter (also a UART). The transmitter sends that byte out onto the serial line, one bit at a time, at a specific rate. On the PC end, the receiving UART takes all the bits and rebuilds the (parallel) byte and puts it in a buffer.

Along with converting between serial and parallel, the UART does some other things as a byproduct (side effect) of it's primary task. The voltage used to represent bits is also converted (changed). Extra bits (called start and stop bits) are added to each byte before it is transmitted. See the Serial-HOWTO section, ``Voltage Waveshapes'' for details. Also, while the flow rate (in bytes/sec) on the parallel bus inside the computer is very high, the flow rate out the UART on the serial port side of it is much lower. The UART has a fixed set of rates (speeds) which it can use at its serial port interface.

14.2 Two Types of UARTs

There are two basic types of UARTs: dumb UARTS and FIFO UARTS. Dumb UARTs are the 8250, 16450, early 16550, and early 16650. They are obsolete but if you understand how they work it's easy to understand how the modern ones work with FIFO UARTS (late 16550, 16550A, 16c552, late 16650, 16750, and 16C950).

There is some confusion regarding 16550. Early models had a bug and worked properly only as 16450's (no FIFO). Later models with the bug fixed were named 16550A but many manufacturers did not accept the name change and continued calling it a 16550. Most all 16550's in use today are like 16550A's. Linux will report it as being a 16550A even though your hardware manual (or a label note) says it's a 16550. A similar situation exists for the 16650 (only it's worse since the manufacturer allegedly didn't admit anything was wrong). Linux will report a late 16650 as being a 16650V2. If it reports it as 16650 it is bad

news and only is used as if it had a one-byte buffer.

14.3 FIFOs

To understand the differences between dumb and FIFO (First In, First Out queue discipline) first let's examine what happens when a UART has sent or received a byte. The UART itself can't do anything with the data passing thru it, it just receives and sends it. For the original dumb UARTS, the CPU gets an interrupt from the serial device every time a byte has been sent or received. The CPU then moves the received byte out of the UART's buffer and into memory somewhere, or gives the UART another byte to send. The 8250 and 16450 UARTs only have a 1 byte buffer. That means, that every time 1 byte is sent or received, the CPU is interrupted. At low transfer rates, this is OK. But, at high transfer rates, the CPU gets so busy dealing with the UART, that is doesn't have time to adequately tend to other tasks. In some cases, the CPU does not get around to servicing the interrupt in time, and the byte is overwritten, because they are coming in so fast. This is called an "overrun" or "overflow".

That's where the FIFO UARTs are useful. The 16550A (or 16550) FIFO chip comes with 16 byte FIFO buffers. This means that it can receive up to 14 bytes (or send 16 bytes) before it has to interrupt the CPU. Not only can it wait for more bytes, but the CPU then can transfer all 14 (or more) bytes at a time. Although the interrupt threshold (trigger level) may be set at 8 instead of 14, this is still a significant advantage over the other UARTs, which only have 1 byte buffers. The CPU receives less interrupts, and is free to do other things. Data is not lost, and everyone is happy.

While most PC's only have a 16550 with 16-byte buffers, better UARTS have even larger buffers. Note that the interrupt is issued slightly before the buffer get full (at say a "trigger level" of 14 bytes for a 16-byte buffer). This allows room for a few more bytes to be received during the time that the interrupt is being serviced. The trigger level may be set to various permitted values by kernel software. A trigger level of 1 will be almost like a dumb UART (except that it still has room for 15 more bytes after it issues the interrupt).

If you type something while visiting a BBS, the characters you type go out thru the serial port. Your typed characters that you see on the screen are what was echoed back thru the telephone line thru your modem and then thru your serial port to the screen. If you had a 16-byte buffer on the serial port which held back characters until it had 14 of them, you would need to type many characters before you could see what you typed (before they appeared on the screen). This would be very confusing but there is a "timeout" to prevent this. Thus you normally see a character on the screen just as soon as you type it.

The "timeout" works like this for the receive UART buffer: If characters arrive one after another, then an interrupt is issued only when say the 14th character reaches the buffer. But if a character arrives and the next character doesn't arrive soon thereafter, then an interrupt is issued. This happens even though there are not 14 characters in the buffer (there may only be one character in it). Thus when what you type goes thru this buffer, it acts almost like a 1-byte buffer even though it is actually a 16-byte buffer (unless your typing speed is a hundred times faster than normal). There is also "timeout" for the transmit buffer as well.

14.4 UART Model Numbers

TOC

Here's a list of UARTs. *TL* is *T*rigger *L*evel

- 8250, 16450, early 16550: Obsolete with 1-byte buffers
- 16550, 16550A, 16c552: 16-byte buffers, TL=1,4,8,14
- 16650: 32-byte buffers. Speed up to 460.8 Kbps
- 16750: 64-byte buffer for send, 56-byte for receive. Speed up to 921.6 Kbps
- Hayes ESP: 1K-byte buffers.

The obsolete ones are only good for modems no higher than 14.4k (DTE speeds up to 38400 bps). For modern modems you need at least a 16550 (and not an early 16550). For V.90 56k modems, it may be a several percent faster with a 16650 (especially if you are downloading uncompressed files). The main advantage of the 16650 is its larger buffer size as the extra speed isn't needed unless the modem compression ratio is high. Some 56k internal modems may come with a 16650 ??

Non-UART, and intelligent multiport boards use DSP chips to do additional buffering and control, thus relieving the CPU even more. For example, the Cyclades Cyclom, and Stallion EasyIO boards use a Cirrus Logic CD1400 RISC UART, and many boards use 80186 CPUs or even special RISC CPUs, to handle the serial IO.

Most newer PC's (486's, Pentiums, or better) come with 16550A's (usually called just 16550's). If you have something really old the chip may unplug so that you may be able to upgrade by buying a 16550A chip and replacing your existing 16450 UART. If the functionality has been put on another type of chip, you are out of luck. If the UART is socketed, then upgrading is easy (if you can find a replacement). The new and old are pin-to-pin compatible. It may be more feasible to just buy a new serial board on the Internet (few retail stores stock them today).

---

15. Pinout and Signals

15.1 Pinout

```
        PINOUT of the SERIAL PORT    (--> direction is out of PC)
                (Note DCD is sometimes labeled CD)
Pin #   Pin #   Acronym  Full-Name   Direction  What-it-May-Do/Mean
9-pin   25-pin
 3       2      TxD     Transmit Data     -->   Transmits byte out of PC
 2       3      RxD     Receive Data      <--        Receives bytes into PC
 7       4      RTS     Request To Send   -->   RTS/CTS flow control
 8       5      CTS     Clear To Send     <--        RTS/CTS flow control
 6       6      DSR     Data Set Ready    <--        I'm ready to communicate
 4      20      DTR     Data Terminal Ready-->  I'm ready to communicate
 1       8      DCD     Data Carrier Detect<--       Modem connected to another
 9      22      RI      Ring Indicator    <--        Telephone line ringing
 5       7              Signal Ground
```

15.2 Signals May Have No Fixed Meaning

Only 3 of the 9 pins have a fixed assignment: transmit, receive and signal ground. This is fixed by the hardware and you can't change it. But the other signal lines are controlled by software and may do (and mean) almost anything at all. However they can only be in one of two states: asserted (+12 volts) or negated (-12 volts). Asserted is "on" and negated is "off". For example, Linux software may command that DTR be negated and the hardware only carries out this command and puts -12 volts on the DTR pin. A modem (or other device) that receives this DTR signal may do various things. If a modem has been configured a certain way it will hang-up the telephone line when DTR is negated. In other cases it may ignore this signal or do something else when DTR is negated (turned off).

It's like this for all the 6 signal lines. The hardware only sends and receives the signals, but what action (if any) they perform is up to the Linux software and the configuration/design of devices that you connect to the serial port. However, most pins have certain functions which they normally perform but this may vary with the operating system and the device driver configuration. Under Linux, one may modify the source code to make these signal lines behave differently (some people have).

15.3 Cabling Between Serial Ports

A cable from a serial port always connects to another serial port. A modem or other device that connects to the serial port has a serial port built into it. For modems, the cable is always straight thru: pin 2 goes to pin 2, etc. The modem is said to be DCE (Data Communications Equipment) and the computer is said to be DTE (Data Terminal Equipment). Thus for connecting DTE-to-DCE you use straight-thru cable. For connecting DTE-to-DTE you must use a null-modem cable and there are many ways to wire such cable (see examples in Text-Terminal-HOWTO).

There are good reasons why it works this way. One reason is that the signals are unidirectional. If pin 1 sends a signal out of it (but is unable to receive any signal) then obviously you can't connect it to pin 1 of the same type of device. If you did, they would both send out signals on the same wire to each other but neither would be able to receive any

signal. There are two ways to deal with this situation. One way is to have a two different types of equipment wh pin 1 of the first type sends the signal to pin 1 of the second type (which receives the signal). That's the way it's d when you connect a PC (DTE) to a modem (DCE). There's a second way to do this without having two different types of equipment: Connect pin 1 (a sending pin) to a receiving pin (not pin 1) on same type of equipment. That's the way it's done when you connect 2 PC's together or a PC to a terminal (DTE-to-DTE). The cable used for this is called a null-modem cable.

The serial pin designations were originally intended for connecting a dumb terminal to a modem. The terminal was DTE (Data Terminal Equipment) and the modem was DCE (Data Communication Equipment). Today the PC is usually used as DTE instead of a terminal (but real terminals may still be used this way). The names of the pins are the same on both DTE and DCE. The words: "receive" and "transmit" are in this case from the point of view of the PC (DTE). The transmit pin from the PC transmits to the "transmit" pin of the modem (but actually the modem is receiving the data from this pin so from the point of view of the modem it would be a receive pin).

The serial port was originally intended to be used for connecting DTE to DCE which makes cabling simple: just use a straight-thru cable. Thus when one connects a modem one seldom needs to worry about which pin is which. But people wanted to connect DTE to DTE (for example a computer to a terminal) and various ways were found to do this by fabricating various type of special cables. In this case what pin connects to what pin becomes more important.

15.4 RTS/CTS and DTR/DSR Flow Control

This is "hardware" flow control. Flow control was previously explained in the Flow Control subsection but the pins and voltage signals were not. Linux only supports RTS/CTS flow control at present (but a special driver may exist for a specific application which supports DTR/DSR flow control). Only RTS/CTS flow control will be discussed since DTR/DSR flow control works the same way. To get RTS/CTS flow control one needs to either select hardware flow control in an application program or use the command:
stty crtscts < /dev/ttyS2 (or the like). This enables RTS/CTS hardware flow control in the Linux device driver.

Then when a DTE (such as a PC) wants to stop the flow into it, it negates RTS. Negated "Request To Send" (-12 volts) means "Request NOT To Send to me" (stop sending). When the PC is ready for more bytes it asserts RTS (+12 volts) and the flow of bytes to it resumes. Flow control signals are always sent in a direction opposite to the flow of bytes that is being controlled. DCE equipment (modems) works the same way but sends the stop signal out the CTS pin. Thus it's RTS/CTS flow control using 2 lines.

On what pins is this stop signal received? That depends on whether we have a DCE-DTE connection or a DTE-DTE connection. For DCE-DTE it's a straight-thru connection so obviously the signal is received on a pin with the same name as the pin it's sent out from. It's RTS-->RTS (PC to modem) and CTS<--CTS (modem to PC). For DTE-to-DTE the connection is also easy to figure out. The RTS pin always sends and the CTS pin always receives. Assume that we connect two PCs (PC1 and PC2) together via their serial ports. Then it's RTS(PC1)-->CTS(PC2) and CTS(PC1)<--RTS(PC2). In other words RTS and CTS cross over. Such a cable (with other signals crossed over as well) is called a "null modem" cable. See Cabling Between Serial Ports

What is sometimes confusing is that there is the original use of RTS where it means about the opposite of the previous explanation above. This original meaning is: I Request To Send to you. This request was intended to be sent from a terminal (or computer) to a modem which, if it decided to grant the request, would send back an asserted CTS from its CTS pin to the CTS pin of the computer: You are Cleared To Send to me. Note that in contrast to the modern RTS/CTS bi-directional flow control, this only protects the flow in one direction: from the computer (or terminal) to the modem. This original use appears to be little used today on modern equipment (including modems).

The DTR and DSR Pins

Just like RTS and CTS, these pins are paired. For DTE-to-DTE connections they are likely to cross over. There are two ways to use these pins. One way is to use them as a substitute for RTS/CTS flow control. The DTR pin is just like the RTS pin while the DSR pin behaves like the CTS pin. Although Linux doesn't support DTR/DSR flow control, it can be obtained by connecting the RTS/CTS pins at the PC to the DSR/DTR pins at the device that uses DTR/DSR flow control. DTR flow control is the same as DTR/DSR flow control but it's only one-way and the DSR pin is not used. Many text terminals and some printers use this type of flow control.

The normal use of DTR and DSR is as follows: A device asserting DTR says that its powered on and ready to operate. For a modem, the meaning of a DTR signal from the PC depends on how the modem is configured. To send a DTR signal manually from a PC using the stty command set the baud rate to 0. Negating DTR is sometimes called "hanging up" but it doesn't always do this.

15.5 Preventing a Port From Opening

If "stty -clocal" (or getty is used with the "local" flag negated) then a serial port can't open until DCD gets an assert (+12 volts) signal.

---

16. Voltage Waveshapes

16.1 Voltage for a Bit

At the EIA-232 serial port, voltages are bipolar (positive or negative with respect to ground) and should be about 12 volts in magnitude (some are 5 or 10 volts). For the transmit and receive pins +12 volts is a 0-bit (sometimes called "space") and -12 volts is a 1-bit (sometimes called "mark"). This is known as inverted logic since normally a 0-bit is both false and negative while a one is normally both true and positive. Although the receive and transmit pins are inverted logic, other pins (modem control lines) are normal logic with a positive voltage being true (or "on" or "asserted") and a negative voltage being false (or "off" or "negated"). Zero voltage has no meaning (except it usually means that the unit is powered off).

A range of voltages is allowed. The specs say the magnitude of a transmitted signal should be between 5 and 15 volts but must never exceed 25 V. Any voltage received under 3 V is undefined (but some devices will accept a lower voltage as valid). One sometimes sees erroneous claims that the voltage is commonly 5 volts (or even 3 volts) but it's usually 11-12 volts. If you are using a EIA-422 port on a Mac computer as an EIA-232 (requires a special cable) or EIA-423 then the voltage will actually be only 5 V. The discussion here assumes 12 V.

Note that normal computer logic normally is just a few volts (5 volts was once the standard) so that if you try to use test equipment designed for testing 3-5 volt computer logic (TTL) on the 12 volts of a serial port, it may damage the test equipment.

16.2 Voltage Sequence for a Byte

The transmit pin (TxD) is held at -12 V (mark) at idle when nothing is being sent. To start a byte it jumps to +12 V (space) for the start bit and remains at +12 V for the duration (period) of the start bit. Next comes the low-order bit of the data byte. If it's a 0-bit nothing changes and the line remains at +12 V for another bit-period. If it's a 1-bit the voltage jumps from +12 to -12 V. After that comes the next bit (-12 V if a 1 or +12 V if a 0), etc., etc. After the last data bit a parity bit may be sent and then a -12 V (mark) stop bit. Then the line remains at -12 V (idle) until the next start bit. Note that there is no return to 0 volts and thus there is no simple way (except by a synchronizing signal) to tell where one bit ends and the next one begins for the case where 2 consecutive bits are the same polarity (both zero or both one).

A 2nd stop bit would also be -12 V, just the same as the first stop bit. Since there is no signal to mark the boundaries between these bits, the only effect of the 2nd stop bit is that the line must remain at -12 V idle twice as long. The receiver has no way of detecting the difference between a 2nd stop bit and a longer idle time between bytes. Thus communications works OK if one end uses one stop bit and the other end uses 2 stop bits, but using only one stop bit is obviously faster. In rare cases 1 1/2 stop bits are used. This means that the line is kept at -12 V for 1 1/2 time periods (like a stop bit 50% wider than normal).

TOC

16.3 Parity Explained

Characters are normally transmitted with either 7 or 8 bits of data. An additional parity bit may (or may not) be appended to this resulting in a byte length of 7, 8 or 9 bits. Some terminal emulators and older terminals do not allow 9 bits. Some prohibit 9 bits if 2 stop bits are used (since this would make the total number of bits too large: 12 bits total after adding the start bit).

The parity may be set to odd, even or none (mark and space parity may be options on some terminals). With odd parity, the parity bit is selected so that the number of 1-bits in a byte, including the parity bit, is odd. If a such a byte gets corrupted by a bit being flipped, the result is an illegal byte of even parity. This error will be detected and if it's an incoming byte to the terminal an error-character symbol will appear on the screen. Even parity works in a similar manner with all legal bytes (including the parity bit) having an even number of 1-bits. During set-up, the number of bits per character usually means only the number of data bits per byte (7 for true ASCII and 8 for various ISO character sets).

A "mark" is a 1-bit (or logic 1) and a "space" is a 0-bit (or logic 0). For mark parity, the parity bit is always a one-bit. For space parity it's always a zero-bit. Mark or space parity only wastes bandwidth and should be avoided if feasible. "No parity" means that no parity bit is added. For terminals that don't permit 9 bit bytes, "no parity" must be selected when using 8 bit character sets since there is no room for a parity bit.

16.4 Forming a Byte (Framing)

In serial transmission of bytes via EIA-232 ports, the low-order bit is always sent first. Serial ports on PC's use asynchronous communication where there is a start bit and a stop bit to mark the beginning and end of a byte. This is called framing and the framed byte is sometimes called a frame. As a result a total of 9, 10, or 11 bits are sent per byte with 10 being the most common. 8-N-1 means 8 data bits, No parity, 1 stop bit. This adds up to 10 bits total when one counts the start bit. One stop bit is almost universally used. At 110 bits/sec (and sometimes at 300 bits/sec) 2 stop bits were once used but today the 2nd stop bit is used only in very unusual situations (or by mistake since it still works OK that way but wastes bandwidth).

16.5 How "Asynchronous" is Synchronized

The EIA-232 serial port as implemented on PC is asynchronous which in effect means that there is no "clock" signal sent with "ticks" to mark when each bit is sent.. There are only two states of the transmit (or receive) wire: mark (-12 V) or space (+12 V). There is no state of 0 V. Thus a sequence of 1-bits is transmitted by just a steady -12 V with no markers of any kind between bits. For the receiver to detect individual bits it must always have a clock signal which is in synchronization with the transmitter clock. Such a clock would generate a "tick" in synchronization with each transmitted (or received) bit.

For asynchronous transmission, synchronization is achieved by framing each byte with a start bit and a stop bit (done by hardware). The receiver listens on the negative line for a positive start bit and when it detects one it starts its clock ticking. It uses this clock tick to time the reading of the next 7, 8 or 9 bits. (It actually is a little more complex than this since several samples of a bit are normally taken and this

requires additional timing ticks.) Then the stop bit is read, the clock stops and the receiver waits for t next start bit. Thus async is actually synchronized during the reception of a single byte but there is n synchronization between one byte and the next byte.

17. Other Serial Devices (not async EIA-232)

17.1 Successors to EIA-232

A number of EIA standards have been established for higher speeds and longer distances using twisted-pair (balanced) technology. Balanced transmission can sometimes be a hundred times faster than unbalanced EIA-232. For a given speed, the distance (maximum cable length) may be many times longer with twisted pair. But PC-s keep being made with the "obsolete" EIA-232 since it works OK with modems and mice since the cable length is short. If this appears in the latest version of this HOWTO, please let me know if any of the non-EIA-232 listed below are supported by Linux.

17.2 EIA-422-A (balanced) and EIA-423-A (unbalanced)

EIA-423 is just like the unbalanced EIA-232 except that the voltage is only 5 volts. Since this falls within EIA-232 specs it can be connected to a EIA-232 port. Its specs call for somewhat higher speeds than the EIA-232 (but this may be of little help on a long run where it's the unbalance that causes interference).

Apple's Mac computer prior to mid-1998 with its EIA-232/EIA-422 Port provided twisted-pairs (balanced) for transmit and receive (when used as a 422). It is (per specs) exactly 100 times as fast as EIA-423 (which in turn is somewhat faster than EIA-232) The Mac used a small round "mini-DIN-8" connector. It also provided conventional EIA-232 but at only at 5 volts (which is still legal EIA-232). To make it work like at EIA-232 one must use a special cable which (signal) grounds RxD+ (one side of a balanced pair) and use RxD- as the receive pin. While TxD- is used as the transmit pin, for some reason TxD+ should not be grounded. See Macintosh Communications FAQ. However, due to the fact that Macs (and upgrades for them) cost more than PC's, they are not widely as host computers for Linux.

17.3 EIA-485

This is like EIA-422 (balanced). It is half-duplex. It's not just point-to-point but may be used for a multidrop LAN (up to 32 nodes). There are no connector specs

17.4 EIA-530

EIA-530-A (balanced but can also be used unbalanced) at 2Mbits/s (balanced) was intended to be a replacement for EIA-232 but few have been installed. It uses the same 25-pin connector as EIA-232.

TOC

17.5 EIA-612/613

The High Speed Serial Interface (HSSI = EIA-612/613) uses a 50-pin connector and goes up to about 50 Mbits/s but the distance is limited to only several meters. For Linux there are PCI cards supporting HSSI. The companies that sell the cards often provide (or point you to) a Linux driver. A mini-howto or the like is needed for this topic.

17.6 The Universal Serial Bus (USB)

The Universal Serial Bus (USB) is being built into PCI chips. New PC's have them. It is 12 Mbits/s over a twisted pair with a 4-pin connector (2 wires are power supply) but it also is limited to short distances of at most 5 meters (depends on configuration).

Another HOWTO is needed for it. Work is underway for supporting it in Linux (but no HOWTO). It is synchronous and transmits in special packets like a network. Just like a network, it can have several devices attached to it. Each device on it gets a time-slice of exclusive use for a short time. A device can also be guaranteed the use of the bus at fixed intervals. One device can monopolize it if no other device wants to use it. It's not simple to describe in detail.

17.7 Synchronization & Synchronous

Beside the asynchronous EIA-232 (and others) there are a number of synchronous serial port standards. In fact EIA-232 includes synchronous specifications but they aren't normally implemented for serial ports on PC's. But first we'll explain what a synchronous means.

Defining Asynchronous vs Synchronous

Asynchronous (async) means "not synchronous". In practice, an async signal is what the async serial port sends and receives which is a stream of bytes each delimited by a start and stop bit. Synchronous (sync) is most everything else. But this doesn't explain the basic concepts.

In theory, synchronous means that bytes are sent out at a constant rate one after another in step with a clock signal tick. There is often a separate wire or channel for sending the clock signal. Asynchronous bytes may be sent out erratically with various time intervals between bytes (like someone typing characters at a keyboard).

There are certain situations that need to be classified as either sync or async. The async serial port often sends out bytes in a steady stream which would make this a synchronous case but since they still have the start/stop bits (which makes it possible to send them out erratically) it's called async. Another case is where data bytes (without any start-stop bits) are put into packets with possible erratic spacing between one packet and the next. This is called sync since the bytes within each packet must be transmitted synchronously.

Synchronous Communication

Did you ever wonder what all the unused pins are for on a 25-pin connector for the serial port? Most of them are for use in synchronous communication which is seldom implemented on PC's. There are pins for sync timing signals as well as for a sync reverse channel. The EIA-232 spec provides for both sync and async but PC's use a UART (Universal Asynchronous Receiver/Transmitter) chip such as a 16450, 16550A, or 16650 and can't deal with sync. For sync one needs a USART chip or the equivalent where the "S" stands for Synchronous. Since sync is a niche market, a sync serial port is likely to be quite expensive.

Besides the sync part of the EIA-232, there are various other EIA synchronous standards. For EIA-232, 3 pins of the connector are reserved for clock (or timing) signals. Sometimes it's a modem's task to generate some timing signals making it impossible to use synchronous communications without a synchronous modem (or without a device called a "synchronous modem eliminator" which provides the timing signals).

Although few serial ports are sync, synchronous communication does often take place over telephone lines using modems which use V.42 error correction. This strips off the start/stop bits and puts the date bytes in packets resulting in synchronous operation over the phone line.

---

18. Other Sources of Information

18.1 Books

1. Axleson, Jan: Serial Port Complete, Lakeview Research, Madison, WI, 1998.
2. Black, Uyless D.: Physical Layer Interfaces & Protocols, IEEE Computer Society Press, Los Alamitos, CA, 1996.
3. Campbell, Joe: The RS-232 Solution, 2nd ed., Sybex, 1982.
4. Levine, Donald: POSIX Programmer's Guide, (ISBN 0-937175-73-0; O'Reilly)
5. Putnam, Byron W.: RS-232 Simplified, Prentice Hall, 1987.
6. Seyer, Martin D.: RS-232 Made Easy, 2nd ed., Prentice Hall, 1991.
7. Stevens, Richard W.: Advanced Programming in the UNIX Environment, (ISBN 0-201-56317-7; Addison-Wesley)
8. Tischert, Michael & Bruno Jennrich: PC Intern, Abacus 1996. Chapter 7: Serial Ports

Notes re books:

1. "... Complete" has hardware details (including register) but the programming aspect is Window oriented.
2. "Physical Layer ..." covers much more than just EIA-232.

18.2 Serial Software

It's best to use the nearest mirror site, but here's the main sites:
Serial Software for Linux software for the serial ports including getty and port monitors.
Serial Communications for communication programs.

- `irqtune` will give serial port interrupts higher priority to improve performance. Using `hdparm` for hard-disk tuning may help some more.
- `modemstat` and `statserial` show the current state of various modem control lines. See Serial Monitoring/Diagnostics

18.3 Linux Documents

- man pages for: `setserial(8) stty`
- libc (or glibc) docs package: "Low Level Terminal Interface"
- Modem-HOWTO: modems on the serial port
- PPP-HOWTO: help with PPP (using a modem on the serial port)

TOC

- Printing-HOWTO: for setting up a serial printer
- Serial-Programming-HOWTO: for some aspects of serial-port programming
- Text-Terminal-HOWTO: how they work and how to install and configure
- UPS-HOWTO: setting up UPS sensors connected to your serial port
- UUCP-HOWTO: for information on setting up UUCP

18.4 Usenet newsgroups:

- comp.os.linux.answers
- comp.os.linux.hardware: Hardware compatibility with the Linux operating system.
- comp.os.linux.networking: Networking and communications under Linux.
- comp.os.linux.setup: Linux installation and system administration.

18.5 Serial Mailing List

The Linux serial mailing list. To join, send email to `majordomo@vger.rutgers.edu`, with ``subscribe linux-serial'' in the message body. If you send ``help'' in the message body, you get a help message. The server also serves many other Linux lists. Send the ``lists'' command for a list of mailing lists.

18.6 Internet

- Serial Suite by Vern Hoxie is a collection of blurbs about the care and feeding of the Linux serial port plus some simple programs. When logging into 'scicom' as "anonymous", you must use your full e-mail address as the password. For example: greg.hankins@cc.gatech.edu
- A white paper discussing serial communications and multiport serial boards is available from Cyclades at `http://www.cyclades.com`.

END OF Serial-HOWTO

The Linux Serial Programming HOWTO

by Peter H. Baumann, Peter.Baumann@dlr.de

v1.0, 22 January 1998

This document describes how to program communications with devices over a serial port on a Linux box.

1. Introduction

- 1.1 Copyright
- 1.2 New Versions Of This Document
- 1.3 Feedback

2. Getting started

- 2.1 Debugging
- 2.2 Port Settings
- 2.3 Input Concepts for Serial Devices

3. Program Examples

- 3.1 Canonical Input Processing
- 3.2 Non-Canonical Input Processing
- 3.3 Asynchronous Input
- 3.4 Waiting for Input from Multiple Sources

4. Other Sources of Information

5. Contributions

1. Introduction

This is the Linux Serial Programming HOWTO. All about how to program communications with other devices / computers over a serial line under Linux. Different techniques are explained: Canonical I/O (only complete lines are transmitted/received), asyncronous I/O, and waiting for input from multiple sources.

This document does not describe how to set up serial ports, because this has been described by Greg Hankins in the Serial-HOWTO.

I have to emphasize that I am not an expert in this field, but have had problems with a project that involved such communication. The code examples presented here were derived from the miniterm code available from the LDP programmers guide (`ftp://sunsite.unc.edu/pub/Linux/docs/LDP/programmers-guide/lpg-0.4.tar.gz` and mirrors) in the examples directory.

Since I wrote this document in June 1997, I have moved to WinNT to satisfy customers need, so I have not built up more in depth knowledge. If anybody has any comments, I will gladly incorporate them into this document (see sect. Feedback). If someone would like to take over and do a better job, please e-mail me.

All examples were tested using a i386 Linux Kernel 2.0.29.

1.1 Copyright

The Linux Serial-Programming-HOWTO is copyright (C) 1997 by Peter Baumann. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would *like* to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would *like* to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at `linux-howto@sunsite.unc.edu` via email.

1.2 New Versions Of This Document

New versions of the Serial-Programming-HOWTO will be available at `ftp://sunsite.unc.edu:/pub/Linux/docs/HOWTO/Serial-Programming-HOWTO` and mirror sites. There are other formats, such as PostScript and DVI versions in the `other-formats`

directory. The Serial-Programming-HOWTO is also available at http://sunsite.unc.edu/LDP/HOWTO/Serial-Programming-HOWTO.html and will be posted to comp.os.linux.answers monthly.

1.3 Feedback

Please send me any corrections, questions, comments, suggestions, or additional material. I would like to improve this HOWTO! Tell me exactly what you don't understand, or what could be clearer. You can reach me at Peter.Baumann@dlr.de via email. Please include the version number of the Serial-Programming-HOWTO when writing, this is version 0.3.

---

2. Getting started

2.1 Debugging

The best way to debug your code is to set up another Linux box, and connect the two computers via a null-modem cable. Use miniterm (available from the LDP programmers guide (`ftp://sunsite.unc.edu/pub/Linux/docs/LDP/programmers-guide/lpg-0.4.tar.gz` in the examples directory) to transmit characters to your Linux box. Miniterm can be compiled very easily and will transmit all keyboard input raw over the serial port. Only the define statement `#define MODEMDEVICE "/dev/ttyS0"` has to be checked. Set it to `ttyS0` for COM1, `ttyS1` for COM2, etc.. It is essential for testing, that *all* characters are transmitted raw (without output processing) over the line. To test your connection, start miniterm on both computers and just type away. The characters input on one computer should appear on the other computer and vice versa. The input will not be echoed to the attached screen.

To make a null-modem cable you have to cross the TxD (transmit) and RxD (receive) lines. For a description of a cable see sect. 7 of the Serial-HOWTO.

It is also possible to perform this testing with only one computer, if you have two unused serial ports. You can then run two miniterms off two virtual consoles. If you free a serial port by disconnecting the mouse, remember to redirect `/dev/mouse` if it exists. If you use a multiport serial card, be sure to configure it correctly. I had mine configured wrong and everything worked fine as long as I was testing only on my computer. When I connected to another computer, the port started loosing characters. Executing two programs on one computer just isn't fully asynchronous.

2.2 Port Settings

The devices `/dev/ttyS*` are intended to hook up terminals to your Linux box, and are configured for this use after startup. This has to be kept in mind when programming communication with a raw device. E.g. the ports are configured to echo characters sent from the device back to it, which normally has to be changed for data transmission.

All parameters can be easily configured from within a program. The configuration is stored in a structure `struct termios`, which is defined in `<asm/termbits.h>`:

```
#define NCCS 19
struct termios {
        tcflag_t c_iflag;               /* input mode flags */
        tcflag_t c_oflag;               /* output mode flags */
        tcflag_t c_cflag;               /* control mode flags */
        tcflag_t c_lflag;               /* local mode flags */
        cc_t c_line;                    /* line discipline */
        cc_t c_cc[NCCS];                /* control characters */
};
```

TOC

This file also includes all flag definitions. The input mode flags in `c_iflag` handle all input processing, which means that the characters sent from the device can be processed before they are read with `read`. Similarly `c_oflag` handles the output processing. `c_cflag` contains the settings for the port, as the baudrate, bits per character, stop bits, etc.. The local mode flags stored in `c_lflag` determine if characters are echoed, signals are sent to your program, etc.. Finally the array `c_cc` defines the control characters for end of file, stop, etc.. Default values for the control characters are defined in `<asm/termios.h>`. The flags are described in the manual page `termios(3)`. The structure `termios` contains the `c_line` (line discipline) element, which is not used in POSIX compliant systems.

2.3 Input Concepts for Serial Devices

Here three different input concepts will be presented. The appropriate concept has to be chosen for the intended application. Whenever possible, do not loop reading single characters to get a complete string. When I did this, I lost characters, whereas a `read` for the whole string did not show any errors.

Canonical Input Processing

This is the normal processing mode for terminals, but can also be useful for communicating with other dl input is processed in units of lines, which means that a `read` will only return a full line of input. A line is by default terminated by a `NL` (ASCII `LF`), an end of file, or an end of line character. A `CR` (the DOS/Windows default end-of-line) will not terminate a line with the default settings.

Canonical input processing can also handle the erase, delete word, and reprint characters, translate `CR` to `NL`, etc..

Non-Canonical Input Processing

Non-Canonical Input Processing will handle a fixed amount of characters per read, and allows for a character timer. This mode should be used if your application will always read a fixed number of characters, or if the connected device sends bursts of characters.

Asynchronous Input

The two modes described above can be used in synchronous and asynchronous mode. Synchronous is the default, where a `read` statement will block, until the read is satisfied. In asynchronous mode the `read` statement will return immediatly and send a signal to the calling program upon completion. This signal can be received by a signal handler.

Waiting for Input from Multiple Sources

This is not a different input mode, but might be useful, if you are handling multiple devices. In my application I was handling input over a TCP/IP socket and input over a serial connection from another computer quasi-simultaneously. The program example given below will wait for input from two different input sources. If input from one source becomes available, it will be processed, and the program will then wait for new input.

The approach presented below seems rather complex, but it is important to keep in mind that Linux is a multi-processing operating system. The `select` system call will not load the CPU while waiting for input,

whereas looping until input becomes available would slow down other processes executing at the same ti

3. Program Examples

All examples have been derived from `miniterm.c`. The type ahead buffer is limited to 255 characters, just like the maximum string length for canonical input processing (`<linux/limits.h>` or `<posix1_lim.h>`).

See the comments in the code for explanation of the use of the different input modes. I hope that the code is understandable. The example for canonical input is commented best, the other examples are commented only where they differ from the example for canonical input to emphasize the differences.

The descriptions are not complete, but you are encouraged to experiment with the examples to derive the best solution for your application.

Don't forget to give the appropriate serial ports the right permissions (e. g.: `chmod a+rw /dev/ttyS1`)!

3.1 Canonical Input Processing

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
#include <stdio.h>

/* baudrate settings are defined in <asm/termbits.h>, which is
included by <termios.h> */
#define BAUDRATE B38400
/* change this definition for the correct port */
#define MODEMDEVICE "/dev/ttyS1"
#define _POSIX_SOURCE 1 /* POSIX compliant source */

#define FALSE 0
#define TRUE 1

volatile int STOP=FALSE;

main()
{
  int fd,c, res;
  struct termios oldtio,newtio;
  char buf[255];
/*
  Open modem device for reading and writing and not as controlling tty
  because we don't want to get killed if linenoise sends CTRL-C.
*/
 fd = open(MODEMDEVICE, O_RDWR | O_NOCTTY );
 if (fd <0) {perror(MODEMDEVICE); exit(-1); }

 tcgetattr(fd,&oldtio); /* save current serial port settings */
 bzero(&newtio, sizeof(newtio)); /* clear struct for new port settings */
```

TOC

```
/*
  BAUDRATE: Set bps rate. You could also use cfsetispeed and cfsetospeed.
  CRTSCTS : output hardware flow control (only used if the cable has
            all necessary lines. See sect. 7 of Serial-HOWTO)
  CS8     : 8n1 (8bit,no parity,1 stopbit)
  CLOCAL  : local connection, no modem contol
  CREAD   : enable receiving characters
*/
 newtio.c_cflag = BAUDRATE | CRTSCTS | CS8 | CLOCAL | CREAD;

/*
  IGNPAR  : ignore bytes with parity errors
  ICRNL   : map CR to NL (otherwise a CR input on the other computer
            will not terminate input)
  otherwise make device raw (no other input processing)
*/
 newtio.c_iflag = IGNPAR | ICRNL;

/*
 Raw output.
*/
 newtio.c_oflag = 0;

/*
  ICANON  : enable canonical input
  disable all echo functionality, and don't send signals to calling program
*/
 newtio.c_lflag = ICANON;

/*
  initialize all control characters
  default values can be found in /usr/include/termios.h, and are given
  in the comments, but we don't need them here
*/
 newtio.c_cc[VINTR]    = 0;     /* Ctrl-c */
 newtio.c_cc[VQUIT]    = 0;     /* Ctrl-\ */
 newtio.c_cc[VERASE]   = 0;     /* del */
 newtio.c_cc[VKILL]    = 0;     /* @ */
 newtio.c_cc[VEOF]     = 4;     /* Ctrl-d */
 newtio.c_cc[VTIME]    = 0;     /* inter-character timer unused */
 newtio.c_cc[VMIN]     = 1;     /* blocking read until 1 character arrives */
 newtio.c_cc[VSWTC]    = 0;     /* '\0' */
 newtio.c_cc[VSTART]   = 0;     /* Ctrl-q */
 newtio.c_cc[VSTOP]    = 0;     /* Ctrl-s */
 newtio.c_cc[VSUSP]    = 0;     /* Ctrl-z */
 newtio.c_cc[VEOL]     = 0;     /* '\0' */
 newtio.c_cc[VREPRINT] = 0;     /* Ctrl-r */
 newtio.c_cc[VDISCARD] = 0;     /* Ctrl-u */
 newtio.c_cc[VWERASE]  = 0;     /* Ctrl-w */
 newtio.c_cc[VLNEXT]   = 0;     /* Ctrl-v */
 newtio.c_cc[VEOL2]    = 0;     /* '\0' */
```

TOC

```
/*
  now clean the modem line and activate the settings for the port
*/
 tcflush(fd, TCIFLUSH);
 tcsetattr(fd,TCSANOW,&newtio);

/*
  terminal settings done, now handle input
  In this example, inputting a 'z' at the beginning of a line will
  exit the program.
*/
 while (STOP==FALSE) {     /* loop until we have a terminating condition */
 /* read blocks program execution until a line terminating character is
    input, even if more than 255 chars are input. If the number
    of characters read is smaller than the number of chars available,
    subsequent reads will return the remaining chars. res will be set
    to the actual number of characters actually read */
    res = read(fd,buf,255);
    buf[res]=0;             /* set end of string, so we can printf */
    printf(":%s:%d\n", buf, res);
    if (buf[0]=='z') STOP=TRUE;
 }
 /* restore the old port settings */
 tcsetattr(fd,TCSANOW,&oldtio);
}
```

3.2 Non-Canonical Input Processing

In non-canonical input processing mode, input is not assembled into lines and input processing (erase, kill, delete, etc.) does not occur. Two parameters control the behavior of this mode: `c_cc[VTIME]` sets the character timer, and `c_cc[VMIN]` sets the minimum number of characters to receive before satisfying the read.

If MIN > 0 and TIME = 0, MIN sets the number of characters to receive before the read is satisfied. As TIME is zero, the timer is not used.

If MIN = 0 and TIME > 0, TIME serves as a timeout value. The read will be satisfied if a single character is read, or TIME is exceeded (t = TIME *0.1 s). If TIME is exceeded, no character will be returned.

If MIN > 0 and TIME > 0, TIME serves as an inter-character timer. The read will be satisfied if MIN characters are received, or the time between two characters exceeds TIME. The timer is restarted every time a character is received and only becomes active after the first character has been received.

If MIN = 0 and TIME = 0, read will be satisfied immediately. The number of characters currently available, or the number of characters requested will be returned. According to Antonino (see contributions), you could issue a `fcntl(fd, F_SETFL, FNDELAY);` before reading to get the same result.

By modifying `newtio.c_cc[VTIME]` and `newtio.c_cc[VMIN]` all modes described above can be tested.

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
```

TOC

```
#include <stdio.h>

#define BAUDRATE B38400
#define MODEMDEVICE "/dev/ttyS1"
#define _POSIX_SOURCE 1 /* POSIX compliant source */
#define FALSE 0
#define TRUE 1

volatile int STOP=FALSE;

main()
{
  int fd,c, res;
  struct termios oldtio,newtio;
  char buf[255];

 fd = open(MODEMDEVICE, O_RDWR | O_NOCTTY );
 if (fd <0) {perror(MODEMDEVICE); exit(-1); }

 tcgetattr(fd,&oldtio); /* save current port settings */

 bzero(&newtio, sizeof(newtio));
 newtio.c_cflag = BAUDRATE | CRTSCTS | CS8 | CLOCAL | CREAD;
 newtio.c_iflag = IGNPAR;
 newtio.c_oflag = 0;

 /* set input mode (non-canonical, no echo,...) */
 newtio.c_lflag = 0;

 newtio.c_cc[VTIME]    = 0;   /* inter-character timer unused */
 newtio.c_cc[VMIN]     = 5;   /* blocking read until 5 chars received */

 tcflush(fd, TCIFLUSH);
 tcsetattr(fd,TCSANOW,&newtio);


 while (STOP==FALSE) {       /* loop for input */
   res = read(fd,buf,255);   /* returns after 5 chars have been input */
   buf[res]=0;               /* so we can printf... */
   printf(":%s:%d\n", buf, res);
   if (buf[0]=='z') STOP=TRUE;
 }
 tcsetattr(fd,TCSANOW,&oldtio);
}
```

3.3 Asynchronous Input

```
#include <termios.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
```

TOC

```
#include <sys/signal.h>
#include <sys/types.h>

#define BAUDRATE B38400
#define MODEMDEVICE "/dev/ttyS1"
#define _POSIX_SOURCE 1 /* POSIX compliant source */
#define FALSE 0
#define TRUE 1

volatile int STOP=FALSE;

void signal_handler_IO (int status);   /* definition of signal handler */
int wait_flag=TRUE;                    /* TRUE while no signal received */

main()
{
  int fd,c, res;
  struct termios oldtio,newtio;
  struct sigaction saio;           /* definition of signal action */
  char buf[255];

  /* open the device to be non-blocking (read will return immediatly) */
  fd = open(MODEMDEVICE, O_RDWR | O_NOCTTY | O_NONBLOCK);
  if (fd <0) {perror(MODEMDEVICE); exit(-1); }

  /* install the signal handler before making the device asynchronous */
  saio.sa_handler = signal_handler_IO;
  saio.sa_mask = 0;
  saio.sa_flags = 0;
  saio.sa_restorer = NULL;
  sigaction(SIGIO,&saio,NULL);

  /* allow the process to receive SIGIO */
  fcntl(fd, F_SETOWN, getpid());
  /* Make the file descriptor asynchronous (the manual page says only
     O_APPEND and O_NONBLOCK, will work with F_SETFL...) */
  fcntl(fd, F_SETFL, FASYNC);

  tcgetattr(fd,&oldtio); /* save current port settings */
  /* set new port settings for canonical input processing */
  newtio.c_cflag = BAUDRATE | CRTSCTS | CS8 | CLOCAL | CREAD;
  newtio.c_iflag = IGNPAR | ICRNL;
  newtio.c_oflag = 0;
  newtio.c_lflag = ICANON;
  newtio.c_cc[VMIN]=1;
  newtio.c_cc[VTIME]=0;
  tcflush(fd, TCIFLUSH);
  tcsetattr(fd,TCSANOW,&newtio);

  /* loop while waiting for input. normally we would do something
     useful here */
  while (STOP==FALSE) {
    printf(".\n");usleep(100000);
```

```
      /* after receiving SIGIO, wait_flag = FALSE, input is available
         and can be read */
      if (wait_flag==FALSE) {
        res = read(fd,buf,255);
        buf[res]=0;
        printf(":%s:%d\n", buf, res);
        if (res==1) STOP=TRUE; /* stop loop if only a CR was input */
        wait_flag = TRUE;      /* wait for new input */
      }
    }
    /* restore old port settings */
    tcsetattr(fd,TCSANOW,&oldtio);
  }

  /***************************************************************************
  * signal handler. sets wait_flag to FALSE, to indicate above loop that     *
  * characters have been received.                                           *
  ***************************************************************************/

  void signal_handler_IO (int status)
  {
    printf("received SIGIO signal.\n");
    wait_flag = FALSE;
  }
```

3.4 Waiting for Input from Multiple Sources

This section is kept to a minimum. It is just intended to be a hint, and therefore the example code is kept short. This will not only work with serial ports, but with any set of file descriptors.

The select call and accompanying macros use a `fd_set`. This is a bit array, which has a bit entry for every valid file descriptor number. `select` will accept a `fd_set` with the bits set for the relevant file descriptors and returns a `fd_set`, in which the bits for the file descriptors are set where input, output, or an exception occurred. All handling of `fd_set` is done with the provided macros. See also the manual page `select(2)`.

```
  #include <sys/time.h>
  #include <sys/types.h>
  #include <unistd.h>

  main()
  {
     int    fd1, fd2;  /* input sources 1 and 2 */
     fd_set readfs;    /* file descriptor set */
     int    maxfd;     /* maximum file desciptor used */
     int    loop=1;    /* loop while TRUE */

     /* open_input_source opens a device, sets the port correctly, and
        returns a file descriptor */
     fd1 = open_input_source("/dev/ttyS1");   /* COM2 */
     if (fd1<0) exit(0);
     fd2 = open_input_source("/dev/ttyS2");   /* COM3 */
     if (fd2<0) exit(0);
```

```
    maxfd = MAX (fd1, fd2)+1;  /* maximum bit entry (fd) to test */

    /* loop for input */
    while (loop) {
      FD_SET(fd1, &readfs);  /* set testing for source 1 */
      FD_SET(fd2, &readfs);  /* set testing for source 2 */
      /* block until input becomes available */
      select(maxfd, &readfs, NULL, NULL, NULL);
      if (FD_ISSET(fd1))         /* input from source 1 available */
        handle_input_from_source1();
      if (FD_ISSET(fd2))         /* input from source 2 available */
        handle_input_from_source2();
    }

  }
```

The given example blocks indefinitely, until input from one of the sources becomes available. If you need to timeout on input, just replace the select call by:

```
  int res;
  struct timeval Timeout;

  /* set timeout value within input loop */
  Timeout.tv_usec = 0;  /* milliseconds */
  Timeout.tv_sec  = 1;  /* seconds */
  res = select(maxfd, &readfs, NULL, NULL, &Timeout);
  if (res==0)
  /* number of file descriptors with input = 0, timeout occurred. */
```

This example will timeout after 1 second. If a timeout occurs, select will return 0, but beware that `Timeout` is decremented by the time actually waited for input by `select`. If the timeout value is zero, select will return immediatly.

---

4. Other Sources of Information

- The Linux Serial-HOWTO describes how to set up serial ports and contains hardware information.
- Serial Programming Guide for POSIX Compliant Operating Systems, by Michael Sweet. This link is obsolete and I could not find a new location for it. Does somebody know where we can find it again? It was a well prepared document!
- The manual page `termios(3)` describes all flags for the `termios` structure.

5. Contributions

As mentioned in the introduction, I am no expert in this field, but had problems myself, and found a solution with the help of others. Thanks for the help from Mr. Strudthoff from the European Transonic Windtunnel, Cologne, Michael Carter (`mcarter@rocke.electro.swri.edu`, and Peter Waltenberg (`p.waltenberg@karaka.chch.cri.nz`)

Antonino Ianella (`antonino@usa.net` wrote the Serial-Port-Programming Mini HOWTO, at the same time I prepared this document. Greg Hankins asked me to incorporate Antonino's Mini-HOWTO into this document.

The structure of this document and SGML formatting was derived from the Serial-HOWTO by Greg Hankins. Thanks also for various corrections made by : Dave Pfaltzgraff (`Dave_Pfaltzgraff@patapsco.com`), Sean Lincolne (`slincol@tpgi.com.au`), Michael Wiedmann (`mw@miwie.in-berlin.de`), and Adrey Bonar (`andy@tipas.lt`).

Linux Shadow Password HOWTO

Michael H. Jackson, mhjack@tscnet.com

v1.3, 3 April 1996

This document aims to describe how to obtain, install, and configure the Linux password Shadow Suite. It also discusses obtaining, and reinstalling other software and network daemons that require access to user passwords. This other software is not actually part of the Shadow Suite, but these programs will need to be recompiled to support the Shadow Suite. This document also contains a programming example for adding shadow support to a program. Answers to some of the more frequently asked questions are included near the end of this document.

1. Introduction.

- 1.1 Changes from the previous release.
- 1.2 New versions of this document.
- 1.3 Feedback.

2. Why shadow your passwd file?

- 2.1 Why you might NOT want to shadow your passwd file.
- 2.2 Format of the /etc/passwd file
- 2.3 Format of the shadow file
- 2.4 Review of crypt(3).

3. Getting the Shadow Suite.

- 3.1 History of the Shadow Suite for Linux
- 3.2 Where to get the Shadow Suite.
- 3.3 What is included with the Shadow Suite.

TOC

4. Compiling the programs.

- 4.1 Unpacking the archive.
- 4.2 Configuring with the config.h file
- 4.3 Making backup copies of your original programs.
- 4.4 Running make

5. Installing

- 5.1 Have a boot disk handy in case you break anything.
- 5.2 Removing duplicate man pages
- 5.3 Running make install
- 5.4 Running pwconv
- 5.5 Renaming npasswd and nshadow

6. Other programs you may need to upgrade or patch

- 6.1 Slackware adduser program
- 6.2 The wu_ftpd Server
- 6.3 Standard ftpd
- 6.4 pop3d (Post Office Protocol 3)
- 6.5 xlock
- 6.6 xdm
- 6.7 sudo
- 6.8 imapd (E-Mail pine package)
- 6.9 pppd (Point-to-Point Protocol Server)

7. Putting the Shadow Suite to use.

- 7.1 Adding, Modifying, and deleting users
- 7.2 The passwd command and passwd aging.
- 7.3 The login.defs file.
- 7.4 Group passwords.

- 7.5 Consistency checking programs
- 7.6 Dial-up passwords.

TOC

8. Adding shadow support to a C program

- 8.1 Header files
- 8.2 libshadow.a library
- 8.3 Shadow Structure
- 8.4 Shadow Functions
- 8.5 Example

9. Frequently Asked Questions.

10. Copyright Message.

11. Miscellaneous and Acknowledgments.

---

1. Introduction.

This is the Linux Shadow-Password-HOWTO. This document describes why and how to add shadow password support on a Linux system. Some examples of how to use some of the *Shadow Suite's* features is also included.

When installing the *Shadow Suite* and when using many of the utility programs, you must be logged in as *root*. When installing the *Shadow Suite* you will be making changes to system software, and it is highly recommended that you make backup copies of programs as indicated. I also recommend that you read and understand all the instructions before you begin.

1.1 Changes from the previous release.

```
Additions:
        Added a sub-section on why you might not want to install shadow
        Added a sub-section on updating the xdm program
        Added a section on how to put Shadow Suite features to work
        Added a section containing frequently asked questions

Corrections/Updates:
        Corrected html references on Sunsite
        Corrected section on wu-ftp to reflect adding -lshadow to the Makefile
        Corrected minor spelling and verbiage errors
        Changed section on wu-ftpd to support ELF
        Updated to reflect security problems in various login programs
        Updated to recommend the Linux Shadow Suite by Marek Michalkiewicz
```

1.2 New versions of this document.

The latest released version of this document can always be retrieved by anonymous FTP from:

sunsite.unc.edu

`/pub/Linux/docs/HOWTO/Shadow-Password-HOWTO`

or:

`/pub/Linux/docs/HOWTO/other-formats/Shadow-Password-HOWTO{-html.tar,ps,dvi}.gz`

or via the World Wide Web from the Linux Documentation Project Web Server, at page: Shadow-Password-HOWTO or directly from me, `<mhjack@tscnet.com>`. It will also be posted to the newsgroup: `comp.os.linux.answers`

This document is now packaged with the Shadow-YYDDMM packages.

1.3 Feedback.

Please send any comments, updates, or suggestions to me: Michael H. Jackson <mhjack@tscnet.com> The sooner I get feedback, the sooner I can update and correct this document. If you find any problems with it, please mail me directly as I very rarely stay up-to-date on the newsgroups.

---

2. Why shadow your passwd file?

By default, most current Linux distributions do not contain the *Shadow Suite* installed. This includes Slackware 2.3, Slackware 3.0, and other popular distributions. One of the reasons for this is that the copyright notices in the original *Shadow Suite* were not clear on redistribution if a fee was charged. Linux uses a GNU Copyright (sometimes refereed to as a Copyleft) that allows people to package it into a convenient package (like a CD-ROM distribution) and charge a fee for it.

The current maintainer of the *Shadow Suite*, Marek Michalkiewicz <marekm@i17linuxb.ists.pwr.wroc.pl> received the source code from the original author under a BSD style copyright that allowed redistribution. Now that the copyright issues are resolved, it is expected that future distributions will contain password shadowing by default. Until then, you will need to install it yourself.

If you installed your distribution from a CD-ROM, you may find that, even though the distribution did not have the *Shadow Suite* installed, some of the files you need to install the *Shadow Suite* may be on the CD-ROM.

However, Shadow Suite versions 3.3.1, 3.3.1-2, and shadow-mk all have security problems with their login program and several other suid root programs that came with them, and should no longer be used.

All of the necessary files may be obtained via anonymous FTP or through the World Wide Web.

On a Linux system without the *Shadow Suite* installed, user information including passwords is stored in the `/etc/passwd` file. The password is stored in an *encrypted* format. If you ask a cryptography expert, however, he or she will tell you that the password is actually in an *encoded* rather than *encrypted* format because when using crypt(3), the text is set to null and the password is the key. Therefore, from here on, I will use the term *encoded* in this document.

The algorithm used to encode the password field is technically referred to as a *one way hash function*. This is an algorithm that is easy to compute in one direction, but very difficult to calculate in the reverse direction. More about the actual algorithm used can be found in section 2.4 or your crypt(3) manual page.

When a user picks or is assigned a password, it is encoded with a randomly generated value called the *salt*. This means that any particular password could be stored in 4096 different ways. The *salt* value is then stored with the encoded password.

When a user logs in and supplies a password, the *salt* is first retrieved from the stored encoded password. Then the supplied password is *encoded* with the *salt* value, and then compared with the *encoded* password. If there is a match, then the user is authenticated.

It is computationally difficult (but not impossible) to take a randomly *encoded* password and recover the original password. However, on any system with more than just a few users, at least some of the passwords will be common words (or simple variations of common words).

System crackers know all this, and will simply encrypt a dictionary of words and common passwords using all possible 4096 *salt* values. Then they will compare the encoded passwords in your `/etc/passwd` file with their database. Once they have found a match, they have the password for another account. This is referred to as a *dictionary attack*, and is one of the most common methods for gaining or expanding unauthorized access to a system.

TOC

If you think about it, an 8 character password encodes to 4096 * 13 character strings. So a dictionary of say 400,000 common words, names, passwords, and simple variations would easily fit on a 4GB hard drive. The attacker need only sort them, and then check for matches. Since a 4GB hard drive can be had for under $1000.00, this is well within the means of most system crackers.

Also, if a cracker obtains your `/etc/passwd` file first, they only need to encode the dictionary with the `salt` values actually contained in your `/etc/passwd` file. This method is usable by your average teenager with a couple of hundred spare Megabytes and a 486 class computer.

Even without lots of drive space, utilities like crack(1) can usually break at least a couple of passwords on a system with enough users (assuming the users of the system are allowed to pick their own passwords).

The `/etc/passwd` file also contains information like user ID's and group ID's that are used by many system programs. Therefore, the `/etc/passwd` file *must* remain world readable. If you were to change the `/etc/passwd` file so that nobody can read it, the first thing that you would notice is that the `ls -l` command now displays user ID's instead of names!

The *Shadow Suite* solves the problem by relocating the passwords to another file (usually `/etc/shadow`). The `/etc/shadow` file is set so that it cannot be read by just anyone. Only *root* will be able to read and write to the `/etc/shadow` file. Some programs (like xlock) don't need to be able to change passwords, they only need to be able to verify them. These programs can either be run *suid root* or you can set up a group *shadow* that is allowed read only access to the `/etc/shadow` file. Then the program can be run *sgid shadow*.

By moving the passwords to the `/etc/shadow` file, we are effectively keeping the attacker from having access to the encoded passwords with which to perform a *dictionary attack*.

Additionally, the *Shadow Suite* adds lots of other nice features:

- A configuration file to set login defaults (`/etc/login.defs`)
- Utilities for adding, modifying, and deleting user accounts and groups
- Password aging and expiration
- Account expiration and locking
- Shadowed group passwords (optional)
- Double length passwords (16 character passwords) NOT RECOMMENDED
- Better control over user's password selection
- Dial-up passwords
- Secondary authentication programs NOT RECOMMENDED

Installing the *Shadow Suite* contributes toward a more secure system, but there are many other things that can also be done to improve the security of a Linux system, and there will eventually be a series of Linux Security HOWTO's that will discuss other security measures and related issues.

For current information on other Linux security issues, including warnings on known vulnerabilities see the Linux Security home page.

2.1 Why you might NOT want to shadow your passwd file.

There are a few circumstances and configurations in which installing the *Shadow Suite* would *NOT* be a good

TOC

idea:

- The machine does not contain user accounts.
- Your machine is running on a LAN and is using NIS (Network Information Services) to get or supply user names and passwords to other machines on the network. (This can actually be done, but is beyond the scope of this document, and really won't increase security much anyway)
- Your machine is being used by terminal servers to verify users via NFS (Network File System), NIS, or some other method.
- Your machine runs other software that validates users, and there is no shadow version available, and you don't have the source code.

2.2 Format of the /etc/passwd file

A non-shadowed `/etc/passwd` file has the following format:

```
username:passwd:UID:GID:full_name:directory:shell
```

Where:

`username`

The user (login) name

`passwd`

The encoded password

`UID`

Numerical user ID

`GID`

Numerical default group ID

`full_name`

The user's full name - Actually this field is called the GECOS (General Electric Comprehensive Operating System) field and can store information other than just the full name. The Shadow commands and manual pages refer to this field as the comment field.

`directory`

User's home directory (Full pathname)

`shell`

User's login shell (Full Pathname)

For example:

```
username:Npge08pfz4wuk:503:100:Full Name:/home/username:/bin/sh
```

Where `Np` is the salt and `ge08pfz4wuk` is the *encoded* password. The encoded salt/password could just as easily have been `kbeMVnZM0oL7I` and the two are exactly the same password. There are 4096 possible encodings for the same password. (The example password in this case is 'password', a really *bad* password).

Once the shadow suite is installed, the `/etc/passwd` file would instead contain:

TOC

```
username:x:503:100:Full Name:/home/username:/bin/sh
```

The `x` in the second field in this case is now just a place holder. The format of the `/etc/passwd` file really didn't change, it just no longer contains the *encoded* password. This means that any program that reads the `/etc/passwd` file but does not actually need to verify passwords will still operate correctly.

The passwords are now relocated to the shadow file (usually `/etc/shadow` file).

2.3 Format of the shadow file

The `/etc/shadow` file contains the following information:

```
username:passwd:last:may:must:warn:expire:disable:reserved
```

Where:

`username`

The User Name

`passwd`

The Encoded password

`last`

Days since Jan 1, 1970 that password was last changed

`may`

Days before password may be changed

`must`

Days after which password must be changed

`warn`

Days before password is to expire that user is warned

`expire`

Days after password expires that account is disabled

`disable`

Days since Jan 1, 1970 that account is disabled

`reserved`

A reserved field

The previous example might then be:

```
username:Npge08pfz4wuk:9479:0:10000::::
```

2.4 Review of crypt(3).

TOC

From the crypt(3) manual page:

"*crypt* is the password encryption function. It is based on the *Data Encryption Standard* algorithm with variations intended (among other things) to discourage use of hardware implementations of a key search.

The key is a user's typed password. The encoded string is all NULLs

The *salt* is a two-character string chosen from the set a-zA-Z0-9./. This string is used to perturb the algorithm in one of 4096 different ways.

By taking the lowest 7 bits of each character of the key, a 56-bit key is obtained. This 56-bit key is used to encrypt repeatedly a constant string (usually a string consisting of all zeros). The returned value points to the encrypted password, a series of 13 printable ASCII characters (the first two characters represent the salt itself). The return value points to static data whose content is overwritten by each call.

Warning: The key space consists of 2**56 equal 7.2e16 possible values. Exhaustive searches of this key space **are possible** using massively parallel computers. Software, such as `crack(1)`, is available which will search the portion of this key space that is generally used by humans for passwords. Hence, password selection should, at minimum, avoid common words and names. The use of a `passwd(1)` program that checks for crackable passwords during the selection process is recommended.

The DES algorithm itself has a few quirks which make the use of the `crypt(3)` interface a very poor choice for anything other than password authentication. If you are planning on using the `crypt(3)` interface for a cryptography project, don't do it: get a good book on encryption and one of the widely available DES libraries."

Most *Shadow Suites* contain code for doubling the length of the password to 16 characters. Experts in `des` recommend against this, as the encoding is simply applied first to the left half and then to the right half of the longer password. Because of the way `crypt` works, this may make for a *less* secure encoded password then if double length passwords were not used in the first place. Additionally, it is less likely that a user will be able to remember a 16 character password.

There is development work under way that would allow the authentication algorithm to be replaced with something more secure and with support for longer passwords (specifically the MD5 algorithm) and retain compatibility with the `crypt` method.

If you are looking for a good book on encryption, I recommend:

```
"Applied Cryptography: Protocols, Algorithms, and Source Code in C"
by Bruce Schneier <schneier@chinet.com>
ISBN: 0-471-59756-2
```

---

3. Getting the Shadow Suite.

3.1 History of the Shadow Suite for Linux

DO NOT USE THE PACKAGES IN THIS SECTION, THEY HAVE SECURITY PROBLEMS

The original *Shadow Suite* was written by `John F. Haugh II`.

There are several versions that have been used on Linux systems:

- `shadow-3.3.1` is the original.
- `shadow-3.3.1-2` is Linux specific patch made by Florian La Roche <flla@stud.uni-sb.de> and contains some further enhancements.
- `shadow-mk` was specifically packaged for Linux.

The `shadow-mk` package contains the `shadow-3.3.1` package distributed by `John F. Haugh II` with the `shadow-3.3.1-2 patch` installed, a few fixes made by Mohan Kokal <magnus@texas.net> that make installation a lot easier, a patch by `Joseph R.M. Zbiciak` for `login1.c` (login.secure) that eliminates the -f, -h security holes in /bin/login, and some other miscellaneous patches.

The `shadow.mk` package was the *previously* recommended package, but should be replaced due to a *security problem* with the `login` program.

There are *security problems* with Shadow versions 3.3.1, 3.3.1-2, and shadow-mk involving the `login` program. This `login` bug involves not checking the length of a login name. This causes the buffer to overflow causing crashes or worse. It has been rumored that this buffer overflow can allow someone with an account on the system to use this bug and the shared libraries to gain *root* access. I won't discuss exactly how this is possible because there are a lot of Linux systems that are affected, but systems with these *Shadow Suites* installed, and most pre-ELF distributions *without* the *Shadow Suite* are vulnerable!

For more information on this and other Linux security issues, see the Linux Security home page (Shared Libraries and login Program Vulnerability)

3.2 Where to get the Shadow Suite.

The only recommended *Shadow Suite* is still in BETA testing, however the latest versions are safe in a production environment and don't contain a vulnerable `login` program.

The package uses the following naming convention:

```
shadow-YYMMDD.tar.gz
```

where `YYMMDD` is the issue date of the Suite.

This version will eventually be *Version 3.3.3* when it is released from Beta testing, and is maintained by Marek Michalkiewicz <marekm@i17linuxb.ists.pwr.wroc.pl>. It's available as: shadow-current.tar.gz.

The following mirror sites have also been established:

- ftp://ftp.icm.edu.pl/pub/Linux/shadow/shadow-current.tar.gz
- ftp://iguana.hut.fi/pub/linux/shadow/shadow-current.tar.gz
- ftp://ftp.cin.net/usr/ggallag/shadow/shadow-current.tar.gz
- ftp://ftp.netural.com/pub/linux/shadow/shadow-current.tar.gz

You should use the currently available version.

You should NOT use a version *older* than `shadow-960129` as they also have the `login` security problem discussed above.

When this document refers to the *Shadow Suite* I am referring to the this package. It is assumed that this is the package that you are using.

For reference, I used `shadow-960129` to make these installation instructions.

If you were previously using `shadow-mk`, you should upgrade to this version and rebuild everything that you originally compiled.

3.3 What is included with the Shadow Suite.

The *Shadow Suite* contains replacement programs for:

`su, login, passwd, newgrp, chfn, chsh, and id`

The package also contains the new programs:

`chage, newusers, dpasswd, gpasswd, useradd, userdel, usermod, groupadd, groupdel, groupmod, groups, pwck, grpck, lastlog, pwconv, and pwunconv`

Additionally, the library: `libshadow.a` is included for writing and/or compiling programs that need to access user passwords.

Also, manual pages for the programs are also included.

There is also a configuration file for the login program which will be installed as `/etc/login.defs`.

4. Compiling the programs.

4.1 Unpacking the archive.

The first step after retrieving the package is unpacking it. The package is in the tar (tape archive) format and compressed using gzip, so first move it to `/usr/src`, then type:

```
tar -xzvf shadow-current.tar.gz
```

This will unpack it into the directory: `/usr/src/shadow-YYMMDD`

4.2 Configuring with the config.h file

The first thing that you need to do is to copy over the `Makefile` and the `config.h` file:

```
cd /usr/src/shadow-YYMMDD
cp Makefile.linux Makefile
cp config.h.linux config.h
```

You should then take a look at the `config.h` file. This file contains definitions for some of the configuration options. If you are using the *recommended* package, I recommend that you disable group shadow support for your first time around.

By default shadowed group passwords are enabled. To disable these edit the `config.h` file, and change the `#define SHADOWGRP` to `#undef SHADOWGRP`. I recommend that you disable them to start with, and then if you really want group passwords and group administrators that you enable it later and recompile. If you leave it enabled, you *must* create the file `/etc/gshadow`.

Enabling the long passwords option is NOT recommended as discussed above.

Do NOT change the setting: `#undef AUTOSHADOW`

The `AUTOSHADOW` option was originally designed so that programs that were shadow ignorant would still function. This sounds good in theory, but does not work correctly. If you enable this option, and the program runs as root, it may call `getpwnam()` as root, and later write the modified entry back to the `/etc/passwd` file (with the *no-longer-shadowed password*). Such programs include chfn and chsh. (You can't get around this by swapping real and effective uid before calling `getpwnam()` because root may use chfn and chsh too.)

The same warning is also valid if you are building libc, it has a `SHADOW_COMPAT` option which does the same thing. It *should NOT* be used! If you start getting encoded passwords back in your `/etc/passwd` file, this is the problem.

If you are using a `libc` version prior to 4.6.27, you will need to make a couple more changes to `config.h` and the `Makefile`. To `config.h` edit and change:

```
#define HAVE_BASENAME
```

to:

```
#undef HAVE_BASENAME
```

And then in the `Makefile`, change:

```
SOBJS = smain.o env.o entry.o susetup.o shell.o \
        sub.o mail.o motd.o sulog.o age.o tz.o hushed.o

SSRCS = smain.c env.c entry.c setup.c shell.c \
        pwent.c sub.c mail.c motd.c sulog.c shadow.c age.c pwpack.c rad64.c \
        tz.c hushed.c
```

```
SOBJS = smain.o env.o entry.o susetup.o shell.o \
        sub.o mail.o motd.o sulog.o age.o tz.o hushed.o basename.o

SSRCS = smain.c env.c entry.c setup.c shell.c \
        pwent.c sub.c mail.c motd.c sulog.c shadow.c age.c pwpack.c rad64.c \
        tz.c hushed.c basename.c
```

These changes add the code contained in `basename.c` which is contained in `libc 4.6.27` and later.

4.3 Making backup copies of your original programs.

It would also be a good idea to track down and make backup copies of the programs that the shadow suite will replace. On a Slackware 3.0 system these are:

- /bin/su
- /bin/login
- /usr/bin/passwd
- /usr/bin/newgrp
- /usr/bin/chfn
- /usr/bin/chsh
- /usr/bin/id

The BETA package has a *save* target in the Makefile, but it's commented out because different distributions place the programs in different places.

You should also make a backup copy of your `/etc/passwd` file, but be careful to name it something else if you place it in the same directory so you don't overwrite the `passwd` command.

4.4 Running make

You need to be logged as root to do most of the installation.

Run make to compile the executables in the package:

```
make all
```

You may see the warning: `rcsid defined but not used`. This is fine, it just happens because the author is using a version control package.

---

5. Installing

5.1 Have a boot disk handy in case you break anything.

If something goes terribly wrong, it would be handy to have a boot disk. If you have a boot/root combination from your installation, that will work, otherwise see the Bootdisk-HOWTO, which describes how to make a bootable disk.

5.2 Removing duplicate man pages

You should also move the manual pages that are about to be replaced. Even if you are brave enough install the Shadow Suite without making backups, you will still want to remove the old manual pages. The new manual pages won't normally overwrite the old ones because the old ones are probably compressed.

You can use a combination of: `man -aW command` and `locate command` to locate the manual pages that need to be (re)moved. It's generally easier to figure out which are the older pages before you run `make install`.

If you are using the Slackware 3.0 distribution, then the manual pages you want to remove are:

- /usr/man/man1/chfn.1.gz
- /usr/man/man1/chsh.1.gz
- /usr/man/man1/id.1.gz
- /usr/man/man1/login.1.gz
- /usr/man/man1/passwd.1.gz
- /usr/man/man1/su.1.gz
- /usr/man/man5/passwd.5.gz

There may also be man pages of the same name in the `/var/man/cat[1-9]` subdirectories that should also be deleted.

5.3 Running make install

You are now ready to type: (do this as root)

```
make install
```

TOC

This will install the new and replacement programs and fix-up the file permissions. It will also install man pages.

This also takes care of installing the Shadow Suite include files in the correct places in `/usr/include/shadow`.

Using the BETA package you must manually copy the file `login.defs` to the `/etc` subdirectory and make sure that only *root* can make changes to it.

```
cp login.defs /etc
chmod 700 /etc/login.defs
```

This file is the configuration file for the *login* program. You should review and make changes to this file for your particular system. This is where you decide which tty's root can login from, and set other security policy settings (like password expiration defaults).

5.4 Running pwconv

The next step is to run `pwconv`. This must also be done as *root*, and is best done from the `/etc` subdirectory:

```
cd /etc
/usr/sbin/pwconv
```

`pwconv` takes your `/etc/passwd` file and strips out the fields to create two files: `/etc/npasswd` and `/etc/nshadow`.

A `pwunconv` program is also provided if you need to make a normal `/etc/passwd` file out of an `/etc/passwd` and `/etc/shadow` combination.

5.5 Renaming npasswd and nshadow

Now that you have run `pwconv` you have created the files `/etc/npasswd` and `/etc/nshadow`. These need to be copied over to `/etc/passwd` and `/etc/shadow`. We also want to make a backup copy of the original `/etc/passwd` file, and make sure only root can read it. We'll put the backup in root's home directory:

```
cd /etc
cp passwd ~passwd
chmod 600 ~passwd
mv npasswd passwd
mv nshadow shadow
```

You should also ensure that the file ownerships and permissions are correct. If you are going to be using *X-Windows*, the `xlock` and `xdm` programs need to be able to read the `shadow` file (but not write it).

TOC

There are two ways that this can be done. You can set `xlock` to suid root (`xdm` is usually run as root anyway). Or you can make the `shadow` file owned by `root` with a group of `shadow`, but before you do this, make sure that you have a shadow group (look in `/etc/group`). None of the users on the system should actually be in the shadow group.

```
chown root.root passwd
chown root.shadow shadow
chmod 0644 passwd
chmod 0640 shadow
```

Your system now has the password file shadowed. You *should* now pop over to another virtual terminal and verify that you can login.

Really, do this now!

If you can't, then something is wrong! To get back to a non-shadowed state, do the following the following:

```
cd /etc
cp ~passwd passwd
chmod 644 passwd
```

You would then restore the files that you saved earlier to their proper locations.

---

6. Other programs you may need to upgrade or patch

Even though the shadow suite contains replacement programs for most programs that need to access passwords, there are a few additional programs on most systems that require access to passwords.

If you are running a *Debian Distribution* (or even if you are not), you can obtain Debian sources for the programs that need to be rebuild from: ftp://ftp.debian.org/debian/stable/source/

The remainder of this section discusses how to upgrade `adduser`, `wu_ftpd`, `ftpd`, `pop3d`, `xlock`, `xdm` and `sudo` so that they support the shadow suite.

See the section Adding Shadow Support to a C program for a discussion on how to put shadow support into any other program that needs it (although the program must then be run SUID root or SGID shadow to be able to actually access the shadow file).

6.1 Slackware adduser program

Slackware distributions (and possibly some others) contain a interactive program for adding users called `/sbin/adduser`. A shadow version of this program can be obtained from ftp://sunsite.unc.edu/pub/Linux/ system/Admin/accounts/adduser.shadow-1.4.tar.gz.

I would encourage you to use the programs that are supplied with the *Shadow Suite* (`useradd`, `usermod`, and `userdel`) instead of the slackware `adduser` program. They take a little time to learn how to use, but it's well worth the effort because you have much more control and they perform proper file locking on the `/etc/passwd` and `/etc/shadow` file (`adduser` doesn't).

See the section on Putting the Shadow Suite to use for more information.

But if you gotta have it, here is what you do:

```
tar -xzvf adduser.shadow-1.4.tar.gz
cd adduser
make clean
make adduser
chmod 700 adduser
cp adduser /sbin
```

6.2 The wu_ftpd Server

TOC

Most Linux systems some with the `wu_ftpd` server. If your distribution does not come with shadow installed, then your `wu_ftpd` will not be compiled for shadow. `wu_ftpd` is launched from `inetd/tcpd` as a *root* process. If you are running an old `wu_ftpd` daemon, you will want to upgrade it anyway because older ones had a bug that would allow the *root* account to be compromised (For more info see the Linux security home page).

Fortunately, you only need to get the source code and recompile it with shadow enabled.

If you are not running an ELF system, The `wu_ftp` server can be found on Sunsite as wu-ftp-2.4-fixed.tar.gz

Once you retrieve the server, put it in `/usr/src`, then type:

```
cd /usr/src
tar -xzvf wu-ftpd-2.4-fixed.tar.gz
cd wu-ftpd-2.4-fixed
cp ./src/config/config.lnx.shadow ./src/config/config.lnx
```

Then edit `./src/makefiles/Makefile.lnx`, and change the line:

```
LIBES    = -lbsd -support
```

to:

```
LIBES    = -lbsd -support -lshadow
```

Now you are ready to run the build script and install:

```
cd /usr/src/wu-ftpd-2.4-fixed
/usr/src/wu-ftp-2.4.fixed/build lnx
cp /usr/sbin/wu.ftpd /usr/sbin/wu.ftpd.old
cp ./bin/ftpd /usr/sbin/wu.ftpd
```

This uses the Linux shadow configuration file, compiles and installs the server.

On my Slackware 2.3 system I also had to do the following before running `build`:

```
cd /usr/include/netinet
ln -s in_systm.h in_system.h
cd -
```

Problems have been reported compiling this package under ELF systems, but the Beta version of the next release works fine. It can be found as wu-ftp-2.4.2-beta-10.tar.gz

Once you retrieve the server, put it in `/usr/src`, then type:

```
cd /usr/src
tar -xzvf wu-ftpd-2.4.2-beta-9.tar.gz
cd wu-ftpd-beta-9
cd ./src/config
```

Then edit `config.lnx`, and change:

```
#undef SHADOW.PASSWORD
```

to:

```
#define SHADOW.PASSWORD
```

Then,

```
cd ../Makefiles
```

and edit the file `Makefile.lnx` and change:

```
LIBES = -lsupport -lbsd # -lshadow
```

to:

```
LIBES = -lsupport -lbsd -lshadow
```

Then build and install:

```
cd ..
build lnx
cp /usr/sbin/wu.ftpd /usr/sbin/wu.ftpd.old
cp ./bin/ftpd /usr/sbin/wu.ftpd
```

Note that you should check your `/etc/inetd.conf` file to make sure that this is where your wu.ftpd server really lives. It has been reported that some distributions place the server daemons in different places, and then wu.ftpd in particular may be named something else.

6.3 Standard ftpd

If you are running the standard `ftpd` server, I would recommend that you upgrade to the `wu_ftpd` server. Aside from the known bug discussed above, it's generally thought to be more secure.

If you insist on the standard one, or you need *NIS* support, Sunsite has ftpd-shadow-nis.tgz

6.4 pop3d (Post Office Protocol 3)

TOC

If you need to support the third *Post Office Protocol (POP3)*, you will need to recompile a `pop3d` program. `pop3d` is normally run by `inetd/tcpd` as `root`.

There are two versions available from Sunsite: pop3d-1.00.4.linux.shadow.tar.gz and pop3d+shadow+elf.tar.gz

Both of these are fairly straight forward to install.

6.5 xlock

If you install the shadow suite, and then run *X Windows System* and lock the screen without upgrading your `xlock`, you will have to use `CNTL-ALT-Fx` to switch to another *tty*, login, and kill the `xlock` process (or use `CNTL-ALT-BS` to kill the X server). Fortunately it's fairly easy to upgrade your `xlock` program.

If you are running XFree86 Versions 3.x.x, you are probably using `xlockmore` (which is a great screen-saver in addition to a lock). This package supports *shadow* with a recompile. If you have an older `xlock`, I recommend that you upgrade to this one.

`xlockmore-3.5.tgz` is available at:
ftp://sunsite.unc.edu/pub/Linux/X11/xutils/screensavers/xlockmore-3.7.tgz

Basically, this is what you need to do:

Get the `xlockmore-3.7.tgz` file and put it in `/usr/src` unpack it:

```
tar -xzvf xlockmore-3.7.tgz
```

Edit the file: `/usr/X11R6/lib/X11/config/linux.cf`, and change the line:

```
#define HasShadowPasswd    NO

to

#define HasShadowPasswd    YES
```

Then build the executables:

```
cd /usr/src/xlockmore
xmkmf
make depend
make
```

Then move everything into place and update file ownerships and permissions:

```
cp xlock /usr/X11R6/bin/
cp XLock /var/X11R6/lib/app-defaults/
chown root.shadow /usr/X11R6/bin/xlock
chmod 2755 /usr/X11R6/bin/xlock
chown root.shadow /etc/shadow
chmod 640 /etc/shadow
```

Your xlock will now work correctly.

6.6 xdm

`xdm` is a program that presents a login screen for X-Windows. Some systems start `xdm` when the system is told to goto a specified run level (see `/etc/inittab`.

With the *Shadow Suite* install, `xdm` will need to be updated. Fortunately it's fairly easy to upgrade your `xdm` program.

`xdm.tar.gz` is available at: ftp://sunsite.unc.edu/pub/Linux/X11/xutils/xdm.tar.gz

Get the `xdm.tar.gz` file and put it in `/usr/src`, then to unpack it:

```
tar -xzvf xdm.tar.gz
```

Edit the file: `/usr/X11R6/lib/X11/config/linux.cf`, and change the line:

```
#define HasShadowPasswd    NO

to

#define HasShadowPasswd    YES
```

Then build the executables:

```
cd /usr/src/xdm
xmkmf
make depend
make
```

Then move everything into place:

```
cp xdm /usr/X11R6/bin/
```

`xdm` is run as *root* so you don't need to change it file permissions.

TOC

6.7 sudo

The program `sudo` allows a system administrator to let users run programs that would normally require root access. This is handy because it lets the administrator limit access to the root account itself while still allowing users to do things like mounting drives.

`sudo` needs to read passwords because it verifies the users password when it's invoked. `sudo` already runs SUID root, so accessing the `/etc/shadow` file is not a problem.

`sudo` for the shadow suite, is available as at:
ftp://sunsite.unc.edu/pub/Linux/system/Admin/sudo-1.2-shadow.tgz

Warning: When you install `sudo` your `/etc/sudoers` file will be replaced with a default one, so you need to make a backup of it if you have added anything to the default one. (you could also edit the Makefile and remove the line that copies the default file to `/etc`).

The package is already setup for shadow, so all that's required is to recompile the package (put it in `/usr/src`):

```
cd /usr/src
tar -xzvf sudo-1.2-shadow.tgz
cd sudo-1.2-shadow
make all
make install
```

6.8 imapd (E-Mail pine package)

`imapd` is an e-mail server similar to `pop3d`. `imapd` comes with the *Pine E-mail* package. The documentation that comes with the package states that the default for Linux systems is to include support for shadow. However, I have found that this is not true. Furthermore, the build script / Makefile combination on this package is makes it very difficult to add the `libshadow.a` library at compile time, so I was unable to add shadow support for `imapd`.

If anyone has this figured out, please E-mail me, and I'll include the solution here.

6.9 pppd (Point-to-Point Protocol Server)

The pppd server can be setup to use several types of authentication: *Password Authentication Protocol* (PAP) and *Cryptographic Handshake Authentication Protocol* (CHAP). The pppd server usually reads the password strings that it uses from `/etc/ppp/chap-secrets` and/or `/etc/ppp/pap-secrets`. If you are using this default behavior of pppd, it is not necessary to reinstall pppd.

pppd also allows you to use the *login* parameter (either on the command line, or in the configuration or `options` file). If the *login* option is given, then pppd will use the `/etc/passwd` file for the username

and passwords for the *PAP*. This, of course, will no longer work now that our password file is shado
For pppd-1.2.1d this requires adding code for shadow support.

The example given in the next section is adding shadow support to `pppd-1.2.1d` (an older version of pppd).

`pppd-2.2.0` already contains shadow support.

---

7. Putting the Shadow Suite to use.

This section discusses some of the things that you will want to know now that you have the *Shadow Suite* installed on your system. More information is contained in the manual pages for each command.

7.1 Adding, Modifying, and deleting users

The *Shadow Suite* added the following command line oriented commands for adding, modifying, and deleting users. You may also have installed the `adduser` program.

useradd

The `useradd` command can be used to add users to the system. You also invoke this command to change the default settings.

The first thing that you should do is to examine the default settings and make changes specific to your system:

```
    useradd -D
```

```
GROUP=1
HOME=/home
INACTIVE=0
EXPIRE=0
SHELL=
SKEL=/etc/skel
```

The defaults are probably not what you want, so if you started adding users now you would have to specify all the information for each user. However, we can and should change the default values.

On my system:

- I want the default group to be 100
- I want passwords to expire every 60 days
- I don't want to lock an account because the password is expired
- I want to default shell to be `/bin/bash`

To make these changes I would use:

```
    useradd -D -g100 -e60 -f0 -s/bin/bash
```

Now running `useradd -D` will give:

```
GROUP=100
```

```
HOME=/home
INACTIVE=0
EXPIRE=60
SHELL=/bin/bash
SKEL=/etc/skel
```

Just in case you wanted to know, these defaults are stored in the file `/etc/default/useradd`.

Now you can use `useradd` to add users to the system. For example, to add the user `fred`, using the defaults, you would use the following:

```
useradd -m -c "Fred Flintstone" fred
```

This will create the following entry in the `/etc/passwd` file:

```
fred:*:505:100:Fred Flintstone:/home/fred:/bin/bash
```

And the following entry in the `/etc/shadow` file:

```
fred:!:0:0:60:0:0:0:0
```

`fred`'s home directory will be created and the contents of `/etc/skel` will be copied there because of the `-m` switch.

Also, since we did not specify a UID, the next available one was used.

`fred`'s account is created, but `fred` still won't be able to login until we unlock the account. We do this by changing the password.

```
passwd fred
```

```
Changing password for fred
Enter the new password (minimum of 5 characters)
Please use a combination of upper and lower case letters and numbers.
New Password: *******
Re-enter new password: *******
```

Now the `/etc/shadow` will contain:

```
fred:J0C.WDR1amIt6:9559:0:60:0:0:0:0
```

And `fred` will now be able to login and use the system. The nice thing about `useradd` and the other programs that come with the *Shadow Suite* is that they make changes to the `/etc/passwd` and `/etc/shadow` files atomically. So if you are adding a user, and another user is changing their password at the same time, both operations will be performed correctly.

You should use the supplied commands rather than directly editing `/etc/passwd` and `/etc/shadow`. If you were editing the `/etc/shadow` file, and a user were to change his password while you are editing, and then you were to save the file you were editing, the user's password change would be lost.

Here is a small interactive script that adds users using `useradd` and `passwd`:

```
#!/bin/bash
#
# /sbin/newuser - A script to add users to the system using the Shadow
#                 Suite's useradd and passwd commands.
#
# Written my Mike Jackson <mhjack@tscnet.com> as an example for the Linux
# Shadow Password Howto.  Permission to use and modify is expressly granted.
#
# This could be modified to show the defaults and allow modification similar
# to the Slackware Adduser program.  It could also be modified to disallow
# stupid entries.  (i.e. better error checking).
#
##
#  Defaults for the useradd command
##
GROUP=100        # Default Group
HOME=/home       # Home directory location (/home/username)
SKEL=/etc/skel   # Skeleton Directory
INACTIVE=0       # Days after password expires to disable account (0=never)
EXPIRE=60        # Days that a passwords lasts
SHELL=/bin/bash  # Default Shell (full path)
##
#  Defaults for the passwd command
##
PASSMIN=0        # Days between password changes
PASSWARN=14      # Days before password expires that a warning is given
##
#  Ensure that root is running the script.
##
WHOAMI=`/usr/bin/whoami`
if [ $WHOAMI != "root" ]; then
        echo "You must be root to add news users!"
        exit 1
fi
##
#  Ask for username and fullname.
##
echo ""
echo -n "Username: "
read USERNAME
echo -n "Full name: "
read FULLNAME
#
echo "Adding user: $USERNAME."
#
# Note that the "" around $FULLNAME is required because this field is
# almost always going to contain at least on space, and without the "'s
# the useradd command would think that you we moving on to the next
```

TOC

```
# parameter when it reached the SPACE character.
#
/usr/sbin/useradd -c"$FULLNAME" -d$HOME/$USERNAME -e$EXPIRE \
        -f$INACTIVE -g$GROUP -m -k$SKEL -s$SHELL $USERNAME
##
#  Set password defaults
##
/bin/passwd -n $PASSMIN -w $PASSWARN $USERNAME >/dev/null 2>&1
##
#  Let the passwd command actually ask for password (twice)
##
/bin/passwd $USERNAME
##
#  Show what was done.
##
echo ""
echo "Entry from /etc/passwd:"
echo -n "   "
grep "$USERNAME:" /etc/passwd
echo "Entry from /etc/shadow:"
echo -n "   "
grep "$USERNAME:" /etc/shadow
echo "Summary output of the passwd command:"
echo -n "   "
passwd -S $USERNAME
echo ""
```

Using a script to add new users is really much more preferable than editing the `/etc/passwd` or `/etc/shadow` files directly or using a program like the Slackware `adduser` program. Feel free to use and modify this script for your particular system.

For more information on the `useradd` see the online manual page.

usermod

The `usermod` program is used to modify the information on a user. The switches are similar to the `useradd` program.

Let's say that you want to change `fred`'s shell, you would do the following:

```
usermod -s /bin/tcsh fred
```

Now `fred`'s `/etc/passwd` file entry would be change to this:

```
fred:*:505:100:Fred Flintstone:/home/fred:/bin/tcsh
```

Let's make `fred`'s account expire on 09/15/97:

```
usermod -e 09/15/97 fred
```

Now `fred`'s entry in `/etc/shadow` becomes:

TOC

```
fred:J0C.WDR1amIt6:9559:0:60:0:0:10119:0
```

For more information on the `usermod` command see the online manual page.

userdel

`userdel` does just what you would expect, it deletes the user's account. You simply use:

```
userdel -r username
```

The `-r` causes all files in the user's home directory to be removed along with the home directory itself. Files located in other file system will have to be searched for and deleted manually.

If you want to simply lock the account rather than delete it, use the `passwd` command instead.

7.2 The passwd command and passwd aging.

The `passwd` command has the obvious use of changing passwords. Additionally, it is used by the *root* user to:

- Lock and unlock accounts (`-l` and `-u`)
- Set the maximum number of days that a password remains valid (`-x`)
- Set the minimum days between password changes (`-n`)
- Sets the number of days of warning that a password is about to expire (`-w`)
- Sets the number of days after the password expires before the account is locked (`-i`)
- Allow viewing of account information in a clearer format (`-S`)

For example, let look again at `fred`

```
passwd -S fred
fred P 03/04/96 0 60 0 0
```

This means that `fred`'s password is valid, it was last changed on 03/04/96, it can be changed at any time, it expires after 60 days, fred will not be warned, and and the account won't be disabled when the password expires.

This simply means that if `fred` logs in after the password expires, he will be prompted for a new password at login.

If we decide that we want to warn `fred` 14 days before his password expires and make his account inactive 14 days after he lets it expire, we would need to do the following:

```
passwd -w14 -i14 fred
```

Now `fred` is changed to:

```
fred P 03/04/96 0 60 14 14
```

For more information on the `passwd` command see the online manual page.

TOC

7.3 The login.defs file.

The file `/etc/login` is the configuration file for the `login` program and also for the *Shadow Suite* as a whole.

`/etc/login` contains settings from what the prompts will look like to what the default expiration will be when a user changes his password.

The `/etc/login.defs` file is quite well documented just by the comments that are contained within it. However, there are a few things to note:

- It contains flags that can be turned on or off that determine the amount of logging that takes place.
- It contains pointers to other configuration files.
- It contains defaults assignments for things like password aging.

From the above list you can see that this is a rather important file, and you should make sure that it is present, and that the settings are what you desire for your system.

7.4 Group passwords.

The `/etc/groups` file may contain passwords that permit a user to become a member of a particular group. This function is enabled if you define the constant `SHADOWGRP` in the `/usr/src/shadow-YYMMDD/config.h` file.

If you define this constant and then compile, you must create an `/etc/gshadow` file to hold the group passwords and the group administrator information.

When you created the `/etc/shadow`, you used a program called `pwconv`, there no equivalent program to create the `/etc/gshadow` file, but it really doesn't matter, it takes care of itself.

To create the initial `/etc/gshadow` file do the following:

```
touch /etc/gshadow
chown root.root /etc/gshadow
chmod 700 /etc/gshadow
```

Once you create new groups, they will be added to the `/etc/group` and the `/etc/gshadow` files. If you modify a group by adding or removing users or changing the group password, the `/etc/gshadow` file will be changed.

The programs `groups`, `groupadd`, `groupmod`, and `groupdel` are provided as part of the *Shadow Suite* to modify groups.

The format of the `/etc/group` file is as follows:

```
groupname:!:GID:member,member,...
```

Where:

`groupname`

The name of the group

`!`

TOC

The field that normally holds the password, but that is now relocated to the `/etc/gshadow` file.

GID

The numerical group ID number

member

List of group members

The format of the `/etc/gshadow` file is as follows:

```
groupname:password:admin,admin,...:member,member,...
```

Where:

groupname

The name of the group

password

The encoded group password.

admin

List of group administrators

member

List of group members

The command `gpasswd` is used only for adding or removing administrators and members to or from a group. `root` or someone in the list of administrators may add or remove group members.

The groups password can be changed using the `passwd` command by *root* or anyone listed as an administrator for the group.

Despite the fact that there is not currently a manual page for `gpasswd`, typing `gpasswd` without any parameters gives a listing of options. It's fairly easy to grasp how it all works once you understand the file formats and the concepts.

7.5 Consistency checking programs

pwck

The program `pwck` is provided to provide a consistency check on the `/etc/passwd` and `/etc/shadow` files. It will check each username and verify that it has the following:

- the correct number of fields
- unique user name
- valid user and group identifier
- valid primary group
- valid home directory
- valid login shell

It will also warn of any account that has no password.

It's a good idea to run `pwck` after installing the *Shadow Suite*. It's also a good idea to run it periodically, perhaps weekly or monthly. If you use the `-r` option, you can use `cron` to run it on a regular basis and have the report mailed to you.

grpck

`grpck` is the consistency checking program for the `/etc/group` and `/etc/gshadow` files. It performs the following checks:

- the correct number of fields
- unique group name
- valid list of members and administrators

It also has the `-r` option for automated reports.

7.6 Dial-up passwords.

Dial-up passwords are another optional line of defense for systems that allow dial-in access. If you have a system that allows many people to connect locally or via a network, but you want to limit who can dial in and connect, then dial-up passwords are for you. To enable dial-up passwords, you must edit the file `/etc/login.defs` and ensure that `DIALUPS_CHECK_ENAB` is set to `yes`.

Two files contain the dial-up information, `/etc/dialups` which contains the ttys (one per line, with the leading "/dev/" removed). If a tty is listed then dial-up checks are performed.

The second file is the `/etc/d_passwd` file. This file contains the fully qualified path name of a shell, followed by an optional password.

If a user logs into a line that is listed in `/etc/dialups`, and his shell is listed in the file `/etc/d_passwd` he will be allowed access only by suppling the correct password.

Another useful purpose for using dial-up passwords might be to setup a line that only allows a certain type of connect (perhaps a PPP or UUCP connection). If a user tries to get another type of connection (i.e. a list of shells), he must know a password to use the line.

Before you can use the dial-up feature, you must create the files.

The command `dpasswd` is provided to assign passwords to the shells in the `/etc/d_passwd` file. See the manual page for more information.

---

8. Adding shadow support to a C program

Adding shadow support to a program is actually fairly straightforward. The only problem is that the program must be run by root (or SUID root) in order for the the program to be able to access the `/etc/shadow` file.

This presents one big problem: very careful programming practices must be followed when creating SUID programs. For instance, if a program has a shell escape, this must not occur as root if the program is SUID root.

For adding shadow support to a program so that it can check passwords, but otherwise does need to run as root, it's a lot safer to run the program SUID shadow instead. The `xlock` program is an example of this.

In the example given below, `pppd-1.2.1d` already runs SUID as root, so adding shadow support should not make the program any more vulnerable.

8.1 Header files

The header files should reside in `/usr/include/shadow`. There should also be a `/usr/include/shadow.h`, but it will be a symbolic link to `/usr/include/shadow/shadow.h`.

To add shadow support to a program, you need to include the header files:

```
#include <shadow/shadow.h>
#include <shadow/pwauth.h>
```

It might be a good idea to use compiler directives to conditionally compile the shadow code (I do in the example below).

8.2 libshadow.a library

When you installed the *Shadow Suite* the `libshadow.a` file was created and installed in `/usr/lib`.

When compiling shadow support into a program, the linker needs to be told to include the `libshadow.a` library into the link.

This is done by:

```
    gcc program.c -o program -lshadow
```

However, as we will see in the example below, most large programs use a `Makefile`, and usually have a variable called `LIBS=...` that we will modify.

8.3 Shadow Structure

The `libshadow.a` library uses a structure called `spwd` for the information it retrieves from the `/etc/shadow` file. This is the definition of the `spwd` structure from the `/usr/include/shadow/shadow.h` header file:

TOC

```
struct spwd
{
  char *sp_namp;                /* login name */
  char *sp_pwdp;                /* encrypted password */
  sptime sp_lstchg;             /* date of last change */
  sptime sp_min;                /* minimum number of days between changes */
  sptime sp_max;                /* maximum number of days between changes */
  sptime sp_warn;               /* number of days of warning before password
                                   expires */
  sptime sp_inact;              /* number of days after password expires
                                   until the account becomes unusable. */
  sptime sp_expire;             /* days since 1/1/70 until account expires
*/
  unsigned long sp_flag;        /* reserved for future use */
};
```

The *Shadow Suite* can put things into the `sp_pwdp` field besides just the encoded passwd. The password field could contain:

```
username:Npge08pfz4wuk;@/sbin/extra:9479:0:10000::::
```

This means that in addition to the password, the program `/sbin/extra` should be called for further authentication. The program called will get passed the username and a switch that indicates why it's being called. See the file `/usr/include/shadow/pwauth.h` and the source code for `pwauth.c` for more information.

What this means is that we should use the function `pwauth` to perform the actual authentication, as it will take care of the secondary authentication as well. The example below does this.

The author of the *Shadow Suite* indicates that since most programs in existence don't do this, and that it may be removed or changed in future versions of the *Shadow Suite*.

8.4 Shadow Functions

The `shadow.h` file also contains the function prototypes for the functions contained in the `libshadow.a` library:

```
extern void setspent __P ((void));
extern void endspent __P ((void));
extern struct spwd *sgetspent __P ((__const char *__string));
extern struct spwd *fgetspent __P ((FILE *__fp));
extern struct spwd *getspent __P ((void));
extern struct spwd *getspnam __P ((__const char *__name));
extern int putspent __P ((__const struct spwd *__sp, FILE *__fp));
```

The function that we are going to use in the example is: `getspnam` which will retrieve for us a `spwd` structure for the supplied name.

8.5 Example

This is an example of adding shadow support to a program that needs it, but does not have it by default.

This example uses the *Point-to-Point Protocol Server* (pppd-1.2.1d), which has a mode in which it performs *PAP* authentication using user names and passwords from the `/etc/passwd` file instead of the *PAP* or *CHAP* files. You would not need to add this code to `pppd-2.2.0` because it's already there.

This feature of pppd probably isn't used very much, but if you installed the *Shadow Suite*, it won't work anymore because the passwords are no longer stored in `/etc/passwd`.

The code for authenticating users under `pppd-1.2.1d` is located in the `/usr/src/pppd-1.2.1d/pppd/auth.c` file.

The following code needs to be added to the top of the file where all the other `#include` directives are. We have surrounded the `#includes` with conditional directives (i.e. only include if we are compiling for shadow support).

```
#ifdef HAS_SHADOW
#include <shadow.h>
#include <shadow/pwauth.h>
#endif
```

The next thing to do is to modify the actual code. We are still making changes to the `auth.c` file.

Function `auth.c` before modifications:

```
/*
 * login - Check the user name and password against the system
 * password database, and login the user if OK.
 *
 * returns:
 *      UPAP_AUTHNAK: Login failed.
 *      UPAP_AUTHACK: Login succeeded.
 * In either case, msg points to an appropriate message.
 */
static int
login(user, passwd, msg, msglen)
    char *user;
    char *passwd;
    char **msg;
    int *msglen;
{
    struct passwd *pw;
    char *epasswd;
    char *tty;
```

```
    if ((pw = getpwnam(user)) == NULL) {
        return (UPAP_AUTHNAK);
    }
     /*
     * XXX If no passwd, let them login without one.
     */
    if (pw->pw_passwd == '\0') {
        return (UPAP_AUTHACK);
    }

    epasswd = crypt(passwd, pw->pw_passwd);
    if (strcmp(epasswd, pw->pw_passwd)) {
        return (UPAP_AUTHNAK);
    }

    syslog(LOG_INFO, "user %s logged in", user);

    /*
     * Write a wtmp entry for this user.
     */
    tty = strrchr(devname, '/');
    if (tty == NULL)
        tty = devname;
    else
        tty++;
    logwtmp(tty, user, "");             /* Add wtmp login entry */
    logged_in = TRUE;

    return (UPAP_AUTHACK);
}
```

The user's password is placed into `pw->pw_passwd`, so all we really need to do is add the function `getspnam`. This will put the password into `spwd->sp_pwdp`.

We will add the function `pwauth` to perform the actual authentication. This will automatically perform secondary authentication if the shadow file is setup for it.

Function `auth.c` after modifications to support shadow:

```
/*
 * login - Check the user name and password against the system
 * password database, and login the user if OK.
 *
 * This function has been modified to support the Linux Shadow Password
 * Suite if USE_SHADOW is defined.
 *
 * returns:
 *      UPAP_AUTHNAK: Login failed.
```

```
 *      UPAP_AUTHACK: Login succeeded.
 * In either case, msg points to an appropriate message.
 */
static int
login(user, passwd, msg, msglen)
    char *user;
    char *passwd;
    char **msg;
    int *msglen;
{
    struct passwd *pw;
    char *epasswd;
    char *tty;

#ifdef USE_SHADOW
    struct spwd *spwd;
    struct spwd *getspnam();
#endif

    if ((pw = getpwnam(user)) == NULL) {
        return (UPAP_AUTHNAK);
    }

#ifdef USE_SHADOW
        spwd = getspnam(user);
        if (spwd)
                pw->pw_passwd = spwd->sp-pwdp;
#endif

     /*
     * XXX If no passwd, let NOT them login without one.
     */
    if (pw->pw_passwd == '\0') {
        return (UPAP_AUTHNAK);
    }
#ifdef HAS_SHADOW
    if ((pw->pw_passwd && pw->pw_passwd[0] == '@'
         && pw_auth (pw->pw_passwd+1, pw->pw_name, PW_LOGIN, NULL))
        || !valid (passwd, pw)) {
        return (UPAP_AUTHNAK);
    }
#else
    epasswd = crypt(passwd, pw->pw_passwd);
    if (strcmp(epasswd, pw->pw_passwd)) {
        return (UPAP_AUTHNAK);
    }
#endif

    syslog(LOG_INFO, "user %s logged in", user);
```

```
    /*
     * Write a wtmp entry for this user.
     */
    tty = strrchr(devname, '/');
    if (tty == NULL)
        tty = devname;
    else
        tty++;
    logwtmp(tty, user, "");             /* Add wtmp login entry */
    logged_in = TRUE;

    return (UPAP_AUTHACK);
}
```

Careful examination will reveal that we made another change as well. The original version allowed access (returned `UPAP_AUTHACK` if there was NO password in the `/etc/passwd` file. This is *not* good, because a common use of this login feature is to use one account to allow access to the PPP process and then check the username and password supplied by PAP with the username in the `/etc/passwd` file and the password in the `/etc/shadow` file.

So if we had set the original version up to run as the shell for a user i.e. `ppp`, then anyone could get a ppp connection by setting their PAP to user `ppp` and a password of null.

We fixed this also by returning `UPAP_AUTHNAK` instead of `UPAP_AUTHACK` if the password field was empty.

Interestingly enough, `pppd-2.2.0` has the same problem.

Next we need to modify the Makefile so that two things occur: `USE_SHADOW` must be defined, and `libshadow.a` needs to be added to the linking process.

Edit the Makefile, and add:

```
LIBS = -lshadow
```

Then we find the line:

```
COMPILE_FLAGS = -I.. -D_linux_=1 -DGIDSET_TYPE=gid_t
```

And change it to:

```
COMPILE_FLAGS = -I.. -D_linux_=1 -DGIDSET_TYPE=gid_t -DUSE_SHADOW
```

Now make and install.

---

9. Frequently Asked Questions.

Q: I used to control which tty's *root* could log into using the file `/etc/securettys`, but it doesn't seem to work anymore, what's going on?

A: The file `/etc/securettys` does absolutely nothing now that the *Shadow Suite* is installed. The tty's that *root* can use are now located in the login configuration file `/etc/login.defs`. The entry in this file may point to another file.

Q: I installed the *Shadow Suite*, but now I can't login, what did I miss?

A: You probably installed the Shadow programs, but didn't run `pwconv` or you forgot to copy `/etc/npasswd` to `/etc/passwd` and `/etc/nshadow` to `/etc/shadow`. Also, you may need to copy `login.defs` to `/etc`.

Q: In the section on xlock, it said to change the group ownership of the `/etc/shadow` file to `shadow`. I don't have a `shadow` group, what do I do?

A: You can add one. Simply edit the `/etc/group` file, and insert a line for the shadow group. You need to ensure that the group number is not used by another group, and you need to insert it before the `nogroup` entry. Or you can simply suid `xlock` to root.

Q: Is there a mailing list for the Linux Shadow Password Suite?

A: Yes, but it's for the development and beta testing of the next Shadow Suite for Linux. You can get added to the list by mailing to: `shadow-list-request@neptune.cin.net` with a subject of: `subscribe`. The list is actually for discussions of the Linux `shadow-YYMMSS` series of releases. You should join if you want to get involved in further development or if you install the Suite on your system and want to get information on newer releases.

Q: I installed the *Shadow Suite*, but when I use the `userdel` command, I get "userdel: cannot open shadow group file", what did I do wrong?

A: You compiled the *Shadow Suite* with the `SHADOWGRP` option enabled, but you don't have an `/etc/gshadow` file. You need to either edit the `config.h` file and recompile, or create an `/etc/group` file. See the section on shadow groups.

Q: I installed the *Shadow Suite* but now I'm getting encoded passwords back in my `/etc/passwd` file, what's wrong?

A: You either enabled the `AUTOSHADOW` option in the Shadow `config.h` file, or your `libc` was compiled with the `SAHDOW_COMPAT` option. You need to determine which is the problem, and recompile.

10. Copyright Message.

The Linux Shadow Password HOWTO is Copyright (c) 1996 Michael H. Jackson.

Permission is granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copies above, provided a notice clearly stating that the document is a modified version is also included in the modified document.

Permission is granted to copy and distribute translations of this document into another language, under the conditions specified above for modified versions.

Permission is granted to convert this document into another media under the conditions specified above for modified versions provided the requirement to acknowledge the source document is fulfilled by inclusion of an obvious reference to the source document in the new media. Where there is any doubt as to what defines 'obvious' the copyright owner reserves the right to decide.

11. Miscellaneous and Acknowledgments.

The code examples for `auth.c` are taken from pppd-1.2.1d and ppp-2.1.0e, Copyright (c) 1993 and The Australian National University and Copyright (c) 1989 Carnegie Mellon University.

Thanks to Marek Michalkiewicz <marekm@i17linuxb.ists.pwr.wroc.pl> for writing and maintaining the *Shadow Suite* for Linux, and for his review and comments on this document.

Thanks to Ron Tidd <rtidd@tscnet.com> for his helpful review and testing.

Thanks to everyone who has sent me feedback to help improve this document.

Please, if you have any comments or suggestions then mail them to me.

regards

Michael H. Jackson <mhjack@tscnet.com>

Slovenian HOWTO

Primoz Peterlin <primoz.peterlin@biofiz.mf.uni-lj.si>

v2.9, 15 February 1999

Zbirka receptov za poslovenjenje sistema Linux. (Note to English readers: this document addresses Linux localization issues specific to Slovenian users.)

1. Uvod

- 1.1 Kaj je Linux?
- 1.2 Dogovor o zapisu

2. Za nestrpne

- 2.1 Nasi znaki na konzoli
- 2.2 Nasi znaki v okolju X Window System

3. Prikaz nasih znakov na zaslonu

- 3.1 Nabor znakov ``ISO Latin 2''
- 3.2 Konzola
- 3.3 X Window System
- 3.4 PostScript pod X11
- 3.5 TrueType pod X11

4. Tipkovnica

- 4.1 Konzola
- 4.2 X Windows System

TOC

5. Slovenski locale

- 5.1 Programiranje z GNU gettext
- 5.2 Poslovenjenje katalogov sporocil

6. Poslovenjenje razlicnih programov

- 6.1 Ukazne lupine
- 6.2 Terminalski emulatorji
- 6.3 Pisarniski paketi
- 6.4 Urejevalniki
- 6.5 Elektronska posta
- 6.6 Svetovni splet
- 6.7 Delo na oddaljenem racunalniku
- 6.8 TeX in LaTeX
- 6.9 groff
- 6.10 Perl
- 6.11 Crkovanje: ispell
- 6.12 Pretvarjanje med razlicnimi kodiranji
- 6.13 Drugi programi

7. Tiskanje

- 7.1 Tiskalniki, ki podpirajo ISO Latin 2
- 7.2 Tiskalniki, ki podpirajo kak drug nabor z nasimi znaki
- 7.3 Tiskalniki, ki podpirajo kak drug nabor z nasimi znaki II
- 7.4 Tiskalniki PostScript

8. Linux v Sloveniji

- 8.1 Skupina uporabnikov Linuxa
- 8.2 Strezniki FTP
- 8.3 Usenet
- 8.4 Zastopniki za Linux

9. O tem spisu

- 9.1 Opombe k drugi izdaji
- 9.2 Zahvala
- 9.3 Vprasanja
- 9.4 Uporaba in razsirjanje
- 9.5 Note to English readers

---

1. Uvod

1.1 Kaj je Linux?

Linux je izvedba vecopravilnega, vecuporabniskega operacijskega sistema Unix za osebne racunalnike s procesorji Intel (80386, 80486, Pentium, Pentium MMX, Pentium Pro in Pentium II), Digital AXP 21x64, Sun SPARC in Motorola 680x0, v delu pa so priredbe za Power PC, MIPS in ARM. Jedro operacijskega sistema je ob pomoci in sodelovanju stevilnih zanesenjakov z Interneta napisal Linus Torvalds z univerze v Helsinkih na Finskem. Skupaj s programi iz projektov GNU, X Window System in BSD predstavlja sodobno programsko okolje, enakovredno komercialnim razlicicam sistema Unix.

Linux je prost program, dostopen skupaj z izvorno kodo. Najnovejse razlicice programja za Linux je moc najti na:

`ftp://metalab.unc.edu/pub/Linux/`
`ftp://tsx-11.mit.edu/pub/linux/`

Obe mesti sta navadno tezko dostopni in zelo zasedeni, zato uporabimo eno od stevilnih mest, kjer se zrcali vsebina zgornjih dveh; pametno je, da zacnemo na katerem od domacih zrcal; nasteta so v poglavju Linux v Sloveniji.

1.2 Dogovor o zapisu

V tem spisu so zgledi zamaknjeni od roba in izpisani v pisavi fiksne sirine, npr.

```
echo "Pozdravljeni"
```

Psevdoimena v zgledih so izpisana s lezeco razlicico iste pisave, npr.

`ls datoteka`

Tu se od bralca ali bralke pricakuje, da bo psevdoime *datoteka* nadomestil(-a) z dejanskim imenom datoteke.

Z znakom ~ se po ustaljenem obicaju na sistemih Unix oznacuje osnovno podrocje uporabnika (za uporabnika z imenom *uporabnik* je to navadno `/home/uporabnik`). Nanj kaze tudi sistemska spremenljivka `$HOME`.

Nadalje je za nastavitve spremenljivk okolja v tem spisu vseskozi uporabljan zapis, ki se uporablja v Bournovi ukazni lupini in njenih izpeljankah (Kornova lupina, lupina `bash`). Bralec ali bralka, ki uporablja ukazno lupino C (`csh` ali `tcsh`), bo znal(a) sam(a) prevesti primere

`SPREMENLJIVKA=vrednost; export SPREMENLJIVKA`

(ali celo `export SPREMENLJIVKA=vrednost`, kar dopusca `bash`) v analogne

`setenv SPREMENLJIVKA vrednost`

2. Za nestrpne

To poglavje ponuja nekaksna ``Izbrana poglavja iz slovenskega HOWTO'' za tiste, ki se jim zdajle mudi, da bi lahko bodisi v znakovnem, bodisi v graficnem nacinu, takoj zaceli tipkati z nasimi znaki, k branju celotnega spisa pa bi se vrnili kdaj kasneje in v miru. Omenim naj edino se to, da sta prilagoditev konzole in prilagoditev okolja X Window System povsem neodvisna.

2.1 Nasi znaki na konzoli

Trije koraki so potrebni, da znakovni terminal (ki se mu iz razlogov, po starosti dalec presegajocih povprecnega uporabnika Linuxa, vcasih pravi tudi *konzola*) prilagodimo za delo z nasimi znaki:

1. Omogociti moramo prikaz nasih znakov v znakovnem nacinu. Opisano v razdelku Konzola poglavja ``Prikaz nasih znakov na zaslonu''.
2. Nastaviti moramo ustrezno preslikavo tipkovnice. Opisano v razdelku Konzola poglavja ``Tipkovnica''.
3. Ukazno lupino moramo nastaviti tako, da bo dovoljevala vnos osembitnih znakov. Opisano v razdelku bash v podpoglavju o ukaznih lupinah (ali v razdelku tcsh, ce uporabljate to ukazno lupino).

2.2 Nasi znaki v okolju X Window System

Z okoljem X Window System je podobno kot s konzolo, le pot je malenkost daljsa.

1. Namestiti moramo nase znake in pripraviti graficno okolje X Window System do tega, da jih bodo uporabljali. Opisano v razdelku X Window System poglavja ``Prikaz nasih znakov na zaslonu''.
2. Nastaviti moramo preslikavo tipkovnice. Opisano v razdelku X Window System poglavja ``Tipkovnica''.
3. Nastaviti moramo terminalski emulator, da bo prikazoval nase znake. Opisano v razdelku xterm poglavja o terminalskih emulatorjih.
4. Ukazno lupino moramo nastaviti tako, da bo dovoljevala vnos osembitnih znakov. Opisano v razdelku bash v podpoglavju o ukaznih lupinah (ali v razdelku tcsh, ce uporabljate to ukazno lupino).

3. Prikaz nasih znakov na zaslonu

3.1 Nabor znakov ``ISO Latin 2''

ISO 8859-2 oz. ISO Latin 2 je clan druzine osembitnih naborov znakov ISO 8859, ki pokriva potrebe alfabetskih pisav (latinica in cirilica ter arabska, hebrejska in grska pisava; v pripravi sta tudi armenska in gruzinska). Druzino je zasnovalo evropsko zdruzenje proizvajalcev racunalnikov (European Computer Manufacturer's Association, ECMA), kot mednarodni standard pa potrdila organizacija za standardizacijo ISO s sedezem v Zenevi.

Nabor ISO Latin 2 vsebuje vse potrebne znake za pisanje v albanscini, anglescini, bosnjascini, cescini, finscini, hrvascini, irscini, luziski srbscini, madzarscini, nemscini, poljscini, romunscini, slovascini, slovenscini, ter srbscini v latinicnem precrkovanju.

Nekaj proizvajalcev strojne in programske opreme, med njimi IBM, Apple in Microsoft, uporablja svoje kodne strani za kodiranje znakov srednje- in vzhodnoevropskih narodov. Te kodne strani so praviloma nezdruzljive tako med seboj, kot tudi s standardom ISO.

3.2 Konzola

Nabor znakov za sistemsko konzolo najdemo v paketu kbd Andriesa Brouwerja z univerze v Eindhovnu, Nizozemska. Paket je del vseh distribucij Linuxa in se avtomaticno nalozi ob namestitvi sistema. Pisave in pretvorne tabele so zlozene v imeniku /usr/lib/kbd. (Avanturisti se lahko namesto s paketom `kbd` igrajo s prototipnimi Linux Console Tools avtorja Yanna Dirsona.)

V starejsih distribucijah Slackware se je prevedeni paket imenoval keytbls. Namestimo ga enako kot druge pakete, z ukazom `pkgtool` ali `installpkg`.

Ko imamo paket enkrat namescen, izberemo drugo pisavo na zaslonu z ukazom `setfont`. Spodnji zgled izbere pisavo z naborom znakov po standardu ISO Latin 2 v velikosti sestnajst pik:

```
setfont lat2-16
```

Za nas zanimive pisave so iso02.f08, iso02.f14 in iso02.f16, ter lat2-08.psf, lat2-10.psf, lat2-12.psf, lat2-14.psf in lat2-16.psf (pripono `.psf` lahko pri ukazu `setfont` izpuscamo).

Izbrana pisava velja v vseh virtualnih terminalih (glej `console(4)`) na konzoli. Ce nam to ni vsec, ampak bi radi imeli v vsakem virtualnem terminalu svojo pisavo, si lahko pomagamo s programom `vtfontd` Iana Zimmermana. Ta vsakic, ko zapustimo virtualni terminal, shrani izbrano konzolno pisavo, in jo ponovno restavrira, ko se vrnemo vanj.

`ftp://sizif.mf.uni-lj.si/pub/linux/vtfontd-1.0.1.tar.gz`

Z izbiro pisave smo opravili sele nekako tretjino vsega dela, potrebnega, da bi lahko v tekstovnem okolju pisali po slovensko. Izvesti moramo vsaj se preslikavo tipkovnice ter omogociti vnos osembitnih znakov v ukazni lupini. Nestrpna bralka ali bralec si lahko potrebno prebere v razdelku Konzola poglavja ``Tipkovnica'' ter v o razdelku ukazni lupini bash v poglavju o ukaznih lupinah.

3.3 X Window System

Kratka navodila za namestitev pisav z nasimi znaki za uporabo z X Window System in streznikom XFree86.

1. Pisave snamemo z najblizjega streznika FTP:

ftp://ftp.arnes.si/software/unix/Latin-2-fonts/

Datoteka ISO8859-2-bdf.tar.gz vsebuje najpopolnejsi nabor pisav, kodiranih po ISO 8859-2, vecino uporabnih pisav pa vsebuje tudi xfonts-iso2-0.84.tar.gz. Po obicajnem postopku (`tar -xzf ...`) jih dekomprimiramo in razpakiramo.

2. Pisavam z nasimi znaki dolocimo mesto v drevesni strukturi imenikov. Obicajno je to podimenik v imeniku /usr/X11R6/lib/X11/fonts/ (vcasih tudi /usr/lib/X11/fonts/, /usr/lib/X11R6/lib/X11/fonts ali celo /usr/X386/lib/X11/fonts/ kar pa naj nas ne moti). Mozen primer:

```
/usr/X11R6/lib/X11/fonts/iso_8859.2/75dpi/
/usr/X11R6/lib/X11/fonts/iso_8859.2/100dpi/
/usr/X11R6/lib/X11/fonts/iso_8859.2/misc/
```

   Ce imenikov s tem imenom se ni, jih ustvarimo.

3. Streznik X ne zna uporabiti pisav v enakem formatu, kot se distribuirajo, zato jih moramo s programom `bdftopcf` prevesti iz formata BDF (Binary Distribution Format) v format PCF (Portable Compiled Format). Starejse izdaje X Window System so namesto slednjega uporabljale format SNF (Server Native Format), ki pa se danes ne uporablja vec.

```
for FILE in *.bdf
do
  bdftopcf $FILE -o `basename $FILE .bdf`.pcf
done
```

   (da, ukazi v `bash` lahko segajo tudi prek vecih vrstic). Pisav v obliki BDF zdaj ne potrebujemo vec in jih lahko pobrisemo.

4. Streznik X zna uporabiti stisnjene datoteke s pisavami, kar lahko izkoristimo in prihranimo se nekaj prostora na disku.

```
gzip *.pcf
```

   Starejse izdaje streznika X podpirajo le stiskanje s `compress`.

5. V vsakem imeniku s pisavami pricakuje streznik datoteko fonts.dir, ki podaja preslikavo med opisi pisav (X Logical Font Description, XLFD) in imeni datotek. Ustvarimo jo z ukazom

```
mkfontdir
```

   Ukaz moramo pognati v vsakem podimeniku s pisavami.

6. Koncno lahko streznik obvestimo o tem, da smo namestili nove pisave, kar storimo z ukazom `xset`:

```
xset +fp /usr/X11R6/lib/X11/fonts/iso_8859.2/75dpi/
xset +fp /usr/X11R6/lib/X11/fonts/iso_8859.2/100dpi/
xset +fp /usr/X11R6/lib/X11/fonts/iso_8859.2/misc/
xset fp rehash
```

   Ce je slo vse gladko, mora ukaz `xlsfonts -fn "*-iso8859-2"` pokazati seznam vseh novo namescenih pisav.

7. Ukaz `xset` velja samo, dokler ne zapustimo X Window System. Trajno spremembo dosezemo s popravkom konfiguracijske datoteke. Streznik XFree86 hrani svoje nastavitve v datoteki /etc/XF86Config (pisec teh vrstic je v nekaj letih ukvarjanja z XFree86 na razlicnih distribucijah nastavitve videl tudi ze v datotekah /etc/X11/XF86Config, /usr/X11R6/lib/X11/XF86config in /usr/X11/lib/X11/Xconfig). Poiscemo niz `FontPath` in dodamo vrstice:

```
FontPath "/usr/X11R6/lib/X11/fonts/iso_8859.2/75dpi"
FontPath "/usr/X11R6/lib/X11/fonts/iso_8859.2/100dpi"
FontPath "/usr/X11R6/lib/X11/fonts/iso_8859.2/misc"
```

TOC

To je vse. Od zdaj naprej zna X Window System na zahtevo programa prikazati tudi nase znake. Vec o tem, kako program, npr. `xterm`, pripravimo do tega, da zahteva nase znake, si preberemo v razdelku xterm poglavja o terminalskih emulatorjih.

3.4 PostScript pod X11

Streznik X zna uporabljati rastrske pisave v oblikah PCF, SNF in BDF ter vektorske pisave v obliki Speedo in Type 1. Pri majhnih velikostih (do 10 tipografskih pik) so rastrske pisave na zaslonu navadno lepse, pri vecjih velikostih, npr. v graficnih programih, pa so vektorske pisave znatno lepse.

Nekaj vektorskih pisav Type 1 v kodnem razporedu ISO Latin 2 je pripravil Peter Soos. Pri namescanju v celoti sledimo postopku, opisanem v razdelku X Window System poglavja Prikaz nasih znakov na zaslonu. S streznika snamemo datoteko v obliki `zip`:

`ftp://ftp.arnes.si/software/unix/Latin-2-fonts/l2pfb005.zip`

Se vec pisav je zbrala ceska skupina, paket RPM (vsebino namestimo z ukazom `rpm -Uvh ...`) najdemo na

`ftp://crash.fce.vutbr.cz/pub/linux_fonts/X11fonts-ulT1mo-beta-1.0-4.noarch.rpm`

Ustvarimo nov imenik, npr. /usr/X11R6/lib/X11/fonts/iso_8859.2/Type1, in vanj stresemo vsebino arhiva l2pfb004.zip. Avtor je ze pripravil datoteki fonts.scale in fonts.dir. Ponovimo vajo z `xset` in v nastavitveno datoteko dodamo vrstico:

```
FontPath "/usr/X11R6/lib/X11/fonts/iso_8859.2/Type1"
```

Namesto `xlsfonts` tokrat iz okenskega okolja X Window System pozenimo `xfontsel`. Novo namescene pisave najdemo najhitreje, ce jih iscemo po ``crkolivnici'' (angl. type foundry) `sp` (po avtorjevih zacetnicah) v prvem stolpcu (`fndry`). Izberite kaksno veliko povecavo (`pxlsz`) in primerjajte z rastrskimi pisavami!

3.5 TrueType pod X11

Rasterizator za pisave TrueType se ni standardni del paketa X Window System. Fantje iz The XFree86 Project, Inc. nacrtujejo vkljucitev podpore zanje v izdaji XFree86 4.0.

Freetype in xfsft

Mark Leisher in Juliusz Chroboczek sta standardni streznik pisav iz okolja X Window System dopolnila s FreeType, prostim rasterizatorjem za pisave TrueType, ki so ga napisali David Turner, Robert Wilhelm, Werner Lemberg in sodelavci. Streznik je na voljo kot izvorna koda, s katero lahko pokrpate izvorno kodo streznika X, ali pa kot ze preveden program za Linux.

`http://www.freetype.org/`
`http://www.dcs.ed.ac.uk/home/jec/programs/xfsft/`

Predpostavili bomo, da ste na kakrsenkoli nacin ze uspeli priti do izvedljive datoteke xfsft.

1. Ustvarimo imenik za pisave TrueType, npr. /usr/X11R6/lib/X11/fonts/TrueType/ in vanj prepisemo datoteke s pisavami TrueType.
2. V omenjenem imeniku ustvarimo datoteko fonts.scale. Preprost primer taksne datoteke je naslednji:

5

```
times.ttf -monotype-times-medium-r-normal--0-0-0-0-p-0-iso8859-1
times.ttf -monotype-times-medium-r-normal--0-0-0-0-p-0-iso8859-2
times.ttf -monotype-times-medium-r-normal--0-0-0-0-p-0-iso8859-5
times.ttf -monotype-times-medium-r-normal--0-0-0-0-p-0-koi8-r
times.ttf -monotype-times-medium-r-normal--0-0-0-0-p-0-iso10646-1
```

Opazimo lahko, da se vseh pet vrstic nanasa na eno in isto datoteko. Razlikujejo se le po kodnem naboru. Ker datoteka vsebuje vse potrebne znake, jo streznik pisav aplikacijam lahko predstavi kot katerokoli od nastetih petih kodnih naborov. Joerg Pomnitz je napisal programcek ttmkfdir, ki avtomaticno pregleda vse datoteke TrueType in ustvari datoteko fonts.scale.

http://www.darmstadt.gmd.de/~pommnitz/xfsft.html

3. V istem imeniku zatem pozenemo se mkfontdir. Ta ne bo napravil nic drugega, kot obstojeco datoteko fonts.scale nespremenjeno prepisal v datoteko z imenom fonts.dir.
4. S streznikom pisav xfsft bodisi prepisemo obstojeci streznik (navadno /usr/X11R6/bin/xfs, ali pa ga premaknemo na kaksno drugo primerno mesto.
5. Streznik pisav potrebuje konfiguracijsko datoteko. Spodnja naj sluzi za zgled, ce nimate boljse:

```
clone-self = off
use-syslog = off
client-limit = 20

catalogue = /usr/X11R6/lib/X11/fonts/TrueType,/usr/X11R6/lib/X11/fonts/Type1

error-file = /usr/X11R6/lib/X11/fs/fs-errors

# in decipoints
default-point-size = 120

# x,y
default-resolutions = 100,100,75,75
```

Zdaj lahko pozenemo streznik pisav X. Pri zgledu smo uporabili vrata 7100. Ce so na vasem sistemu ze zasedena, izberite druga. Za podrobnosti si oglejte prirocnik `xfs(1)`.

```
xfsft -port 7100 -config /usr/X11R6/lib/X11/fs/config &
```

6. Ne da bi spreminjali pot, v kateri iscemo pisave, lahko preverimo, ce streznik deluje:

```
fslsfonts -server localhost:7100
```

7. Ce smo zadovoljni, dodamo streznik pisav med vire, kjer iscemo pisave, npr.

```
xset fp+ tcp/localhost:7100
```

Streznik pisav lahko seveda tece tudi na drugem racunalniku, tako da ni potrebe, da imamo na vseh racunalnikih v lokalni mrezi namescene vse pisave. Dokoncno dodamo streznik pisav v konfiguracijsko datoteko /etc/XF86Config:

```
FontPath "tcp/localhost:7100"
```

Xfstt

Drugi streznik pisav za X11, ki podpira pisave TrueType, je Xfstt. Pisec teh vrstic z njim nima izkusenj, je pa z veseljem pripravljen na to mesto vkljuciti izkusnje drugih. Xfstt najdete na navedenem mestu in vseh zrcalih le-tega.

ftp://metalab.unc.edu/pub/Linux/X11/fonts/Xfstt-0.9.10.tgz

---

4. Tipkovnica

4.1 Konzola

Nabor znakov za sistemsko konzolo in tabele za premapiranje tipkovnice najdemo v paketu kbd Andriesa Brouwerja z univerze v Eindhovnu, Nizozemska. Avtor paketa je napisal tudi Keyboard HOWTO, kjer si lahko o tej temi preberemo vec. Paket je del vseh obicajnih distribucij in se nalozi avtomaticno ob namestitvi sistema.

V starejsih distribucijah Slackware se je prevedeni paket imenoval keytbls. Namestimo ga enako kot druge pakete, z ukazom `pkgtool` ali `installpkg`.

Paket `kbd` vsebuje tabelo za slovensko tipkovnico s 102 tipkama in zaslonske pisave za kodni nabor ISO Latin 2. Vsi, ki iz kakrsnihkoli razlogov uporabljajo drugacno tipkovnico, si lahko na osnovi zgledov v /usr/lib/kbd/ pripravijo svojo. Marko Macek je ze pripravil tabelo preslikav za amerisko tipkovnico s 101 tipko:

http://www.kiss.uni-lj.si/~k4fr0235/kbd/

Celoten postopek za izbiro zaslonske pisave, tabele za tipkovnico in uporabo le-te je taksen:

```
# Izberemo pisavo z naborom ISO Latin 2 v velikosti 16 pik
setfont lat2-16
# Nalozimo prireditveno tabelo za slovensko tipkovnico...
loadkeys slovene
# ...in jo izberemo
echo -e -n "\\033(K" > /dev/console
```

Distribucija Debian pozna tudi ukaz `kbdconfig`, ki opravi isto.

Ce zelimo, da je taksna nastavitev trajna, dodamo zgornjih sest vrstic v nastavitveno datoteko `/etc/rc.d/rc.local`, ki se izvede ob zagonu operacijskega sistema. Ce ob tem omogocimo se vnos osembitnih znakov v ukazni lupini bash, je to najnujnejse, kar je potrebno za pisanje po slovensko v tekstovnem nacinu.

4.2 X Windows System

V X Windows System lahko spreminjamo razpored tipk na tipkovnici z ukazom `xmodmap`. Tabelo preslikav za obicajno tipkovnico s 102 ali 105 tipkami, kot je v rabi pri nas, lahko snamemo z

ftp://sizif.mf.uni-lj.si/pub/i18n/Xmodmap.sl.pc102

Uporaba je enostavna:

```
xmodmap Xmodmap.sl.pc102
```

Preslikava tipkovnice velja, dokler ne zapustimo okolja X Window System. Zamenjamo jo lahko s ponovnim ukazom `xmodmap`, ki mu kot argument podamo datoteko s tabelo preslikav. Tabela za standardno amerisko tipkovnico navadno najdemo kot /usr/X11R6/lib/X11/etc/xmodmap.std. Ce je ne, si jo lahko pripravimo sami: *preden* nalozimo slovensko preslikavo tipkovnice, pozenemo:

```
xmodmap -pke > Xmodmap.us
```

To potem nalozimo enako kot slovensko tabelo.

Ce zelimo ta razpored trajno namestiti, preimenujemo datoteko bodisi v ~/.Xmodmap (osebna nastavitev) ali pa v sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap (skupna nastavitev za vse uporabnike; ponekod se ta datoteka imenuje tudi /etc/X11/xinit/Xmodmap), ki se avtomaticno nalozi ob vsakem zagonu X Window System. Ce se *ne nalozi*, preverite svojo datoteko ~/.xinitrc. V njej mora biti nekaj takega kot:

```
usermodmap=$HOME/.Xmodmap
sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap

if [ -f $sysmodmap ]; then
    xmodmap $sysmodmap
fi

if [ -f $usermodmap ]; then
    xmodmap $usermodmap
fi
```

Ob preslikavi tipkovnice ne pozabimo na to, da moramo tudi v ukazni lupini omogociti vnos osembitnih znakov (poglavje Ukazne lupine).

Xks

Cesko podjetje Informatica s.r.o. se je problema preslikave tipkovnice lotilo drugace. Njihov produkt Xks je streznik, ki se vgnezdi med streznik X in namenske programe ter prestreze vse dogodke `KeyPress` ter jih po potrebi modificira. Ta resitev deluje tudi pri slabo napisanih programih, pri katerih preslikava z `xmodmap` ne deluje, obljublja pa tudi resitev za aktivacijo nemih tipk (tipke v zgornji vrstici slovenske tipkovnice lahko v kombinaciji s tipko `AltGr` delujejo kot neme tipke).

http://www.informatica.cz/

Xks je program na pokusino (shareware); registracija izvedbe za Linux stane 900 ceskih kron (okrog 4500 tolarjev). Pisec teh vrstic se zaenkrat se ni utegnil ukvarjati s njim, bi pa z veseljem vkljucil na to mesto izkusnje drugih.

---

5. Slovenski locale

Z izrazom *locale* oznacujemo skupek lastnosti programskega okolja, odvisnih od jezikovnega in kulturnega okolja uporabnika. Mednje sodi informacija o tem, kateri znaki so crke in kaksen je njihov vrstni red, nacin izpisa datuma in ure, izbiro locila v zapisu decimalnih stevil (vejica oz. pika), zapis valute in sporocila programov uporabniku.

Programski vmesnik za internacionalizacijo in lokalizacijo na sistemih, zdruzljivih s standardi POSIX, je definiran v standardu ISO/IEC 9899 (Programski jezik ISO C) ter IEEE 1003.2 (POSIX.2). Informacije, pomembne za lokalizacijo, razvrsca v sest kategorij:

- `LC_CTYPE` (znaki, ki predstavljajo abecedo)
- `LC_COLLATE` (vrstni red znakov v abecedi)
- `LC_TIME` (izpis datuma in ure)
- `LC_NUMERIC` (izpis stevilcnih vrednosti)
- `LC_MONETARY` (izpis valute)
- `LC_MESSAGES` (komunikacije z uporabnikom)

`LC_ALL` zajema vseh nastetih sest kategorij. Medtem ko projekt GNU pri izvedbi prvih petih kategorijah zvesto sledi standardu, se je v sesti odlocil za njegovo razsiritev. Standardizacijski komite se namrec ni mogel zediniti okrog izvedbe ravnanja s prevodi sporocil, zato je standard v tej tocki prakticno neuporaben: doloca le prevode za ``da'' in ``ne''. Projekt GNU je zato razvil svojo metodo, to je paket GNU `gettext`. Prevedeni katalogi sporocil, ki jih uporablja mehanizem `gettext` se navadno nahajajo v podimenikih imenika /usr/share/locale/.

Sledi nekaj primerov odziva poslovenjenega sistema. Primeri so pobrani s HP-UX, ker je lokalizacija Linuxa zaenkrat se v teku -- knjiznica GNU libc 2.0 in kasnejse izpeljanke sicer ze ponujajo okvir za lokalizacijo, izvedba pa zaenkrat se zaostaja, zato koncepta zaenkrat se ne moremo ilustrirati s programi iz paketov `textutils`, `fileutils` ipd.

Z nastavitvijo spremenljivke `LANG` na slovenski locale se spremenijo vsi prilastki:

```
~> date
Sat Oct 19 22:32:04 METDST 1996
~> LANG=sl_SI.iso88592 date
Sob, 19 okt 1996 22:32:04
```

Lahko pa jih spreminjamo tudi vsakega posebej. Ce zelimo obdrzati angleske izpise, obliko datuma, decimalno piko ipd., zeleli pa bi, da gre urejanje crk po slovenski abecedi, spremenimo samo `LC_COLLATE`:

```
~> sort abc.tex
abc
```

```
abca
abd
abc
~> LANG=C LC_COLLATE=sl_SI.iso88592 sort abc.tex
abc
abca
abc
abd
```

Seveda `LC_COLLATE` ne resi vseh problemov v zvezi z urejanjem. Pri razvrscanju kastiljskih kraljev bi, recimo, zeleli, da pride Alfonz IX. za Alfonzem VIII. in pred Alfonzem X. Kar pa je verjetno ze problem za umetno inteligenco...

Spremenljivka `LANG` ima v splosnem obliko:

`LANG=`*`jezik`*`[_`*`drzava`*`[.`*`nabor`*`]]`

Pri tem je *jezik* dvocrkovna koda jezika, kot jo doloca standard ISO 639, *drzava* pa dvocrkovna koda drzave, dolocena s standardom ISO 3166. Zgled za slovenscino, kot se govori v Sloveniji in pise s kodnim naborom ISO 8859-2, je torej

`LANG=sl_SI.iso88592`

Dolocitev drzave in nabora znakov je neobvezna (pomembna je seveda pri jezikih, ki se govorijo v vec drzavah; kaksen locale uporabljajo Slovenci v zamejstvu, je piscu teh vrstic neznano), tako lahko pisemo na kratko

`LANG=sl`

5.1 Programiranje z GNU gettext

Paket GNU `gettext` ponuja nekaj orodij za lokalizacijo:

- `xgettext`, ki iz izvorne kode izlusci sporocila
- `msgcmp`, ki primerja dva kataloga sporocil
- `msgmerge`, zdruzi star preveden katalog sporocil z novim, neprevedenim
- `msgfmt`, ki prenosen katalog sporocil prevede v binarno obliko
- `msgunfmt`, obratno od `msgfmt`

S kratkim zgledom ilustrirajmo, kako poteka pisanje lokaliziranega programa s paketom GNU `gettext`.

1. Najprej napisemo program zgled.c. Pri pisanju programa ze vkljucimo mehanizme internacionalizacije:

```
#include <locale.h>
#include <stdio.h>
#include <libintl.h>

#define PACKAGE   "zgled"
#define LOCALEDIR "/usr/local/share/locale"
```

TOC

```
char main() {
    setlocale (LC_MESSAGES, "");
    bindtextdomain (PACKAGE, LOCALEDIR);
    textdomain (PACKAGE);

    printf("%s\n", gettext("This is a short example."));
    printf("%s\n", gettext("Author: Primoz Peterlin"));
    return(0);
}
```

2. Z ukazom `xgettext` iz izvorne kode izluscimo izpise na zaslon:

   ```
   xgettext zgled.c
   ```

   S tem smo ustvarili datoteko messages.po. To je obicajna besedilna datoteka, ki vsebuje skelet prevodov sporocil.
3. Datoteko messages.po z urejevalnikom dopolnimo s prevodi, recimo takole:

```
# Slovenski prevod zgleda
# Primoz Peterlin, <primoz.peterlin@biofiz.mf.uni-lj.si>
#
msgid ""
msgstr ""
"Project-Id-Version: zgled 1.0\n"
"POT-Creation-Date: 1998-12-06 14:05:53+0100\n"
"PO-Revision-Date: 1998-12-06 15:00:00+0100\n"
"Last-Translator: Primoz Peterlin <peterlin@biofiz.mf.uni-lj.si>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=ISO-8859-2\n"
"Content-Transfer-Encoding: 8bit\n"
"Xgettext-Options: \n"
"Files: zgled.c\n"

#: zgled.c:13
msgid "This is a short example."
msgstr "To je kratek zgled."

#: zgled.c:14
msgid "Author: Primoz Peterlin"
msgstr "Avtor: Primoz Peterlin"
```

4. Popravljeno prenosljivo datoteko s sporocili (angl. portable message file) messages.po prevedemo v binarno obliko:

   ```
   msgfmt -v -o zgled.mo messages.po
   ```
5. Binarno datoteko prestavimo v imenik, kjer jo bo `gettext` nasel:

```
mv zgled.mo /usr/local/share/locale/sl/LC_MESSAGES
```

6. Program prevedemo in povezemo s knjiznico libintl.a:

```
gcc -o zgled zgled.c -lintl
```

7. Pa poskusimo, kaksna je razlika!

```
LANG=C ./zgled
LANG=sl ./zgled
LC_ALL=sl ./zgled
LC_MESSAGES=sl ./zgled
LC_CTYPE=sl ./zgled
LC_ALL=sl_SI.iso88592 ./zgled
```

V prvem in zadnjem primeru je izpisano sporocilo anglesko, v predzadnjem pa verjetno tudi. V prvem zato, ker smo eksplicitno zahtevali locale ``C'', v zadnjem pa zato, ker smo zahtevali locale ``sl_SI.iso88592'', v katerem ni prevoda sporocil -- to namrec obstaja za locale ``sl''. V predzadnjem zgledu smo uporabili pravi locale, a irelevantno kategorijo `LC_CTYPE` namesto uporabljene `LC_MESSAGES`.

Se dve opombi: imenik, kamor prestavimo katalog sporocil, se mora ujemati z imenikom `LOCALEDIR`, navedenim v izvorni kodi. Tezavo, da GNU `gettext` razume ``sl'' in ``sl_SI.iso88592'' kot razlicna locala, je pisec teh vrstic resil z simbolno povezavo v imeniku /usr/local/share/locale:

```
ln -s sl sl_SI.iso88592
```

5.2 Poslovenjenje katalogov sporocil

V teku je internacionalizacija projekta GNU (programi iz projekta GNU predstavljajo znaten del celotnega sistema Linux in vecino tistega, s cimer se sooca koncni uporabnik). Trenutno so v slovenscino ze prevedena sporocila za naslednje pakete GNU: `enscript`, `fileutils`, `gettext`, `grep`, `hello`, `recode` in `tar`. Z nastavitvijo spremenljivke okolja

```
LANG=sl
```

bodo programi iz teh paketov izpisovali slovenska sporocila namesto angleskih.

Razprava o poslovenjenju projekta GNU poteka med prejemniki elektronskega spiska `<sl@li.org>`, nanj se lahko narocite tako, da na naslov sl-request@li.org posljete vrstico

```
subscribe
```

v telesu sporocila.

Pomoc prostovoljcev, ki bi prevedli preostale pakete, ali pa se samo zavezali za vzdrzevanje ze prevedenih, je nadvse zazelena. Vabljeni!

---

6. Poslovenjenje razlicnih programov

Na tem mestu bi se v idealnem svetu opis poslovenjenja Linuxa lahko koncal. Povedali smo vse potrebno o vnosu in izpisu nasih znakov, ter kako z definiranjem spremenljivk okolja programom dopovemo, da delamo s slovenskimi besedili. To poglavje je takorekoc v celoti posveceno *izjemam* -- programom, ki iz taksnih ali drugacnih razlogov ne uporabljajo lokalizacijskih spremenljivk, alternativnim metodam za vnos ali izpis znakov, ki v izjemnih primerih (npr. ce nimamo tipkovnice s slovenskimi znaki) lahko vseeno pridejo prav, in podobnemu.

6.1 Ukazne lupine

bash

Nastavitve ukazne lupine `bash` so v datoteki ~/.inputrc (ki jo po potrebi ustvarimo):

```
set meta-flag on
set convert-meta off
set output-meta on
```

Te tri vrstice je morda pametno dodati tudi v `/etc/skel`, tako da jih podedujejo tudi vsi na novo ustvarjeni uporabniki.

Tole se sicer ne navezuje na slovenscino, ko ste ze pri poslu, si lahko zivljenje olajsate se s tem, da tipkam `Home`, `Delete` in `End` vrnete njihov obicajni pomen: v isto datoteko (~/.inputrc) dodate se vrstice

```
"\e[1~": beginning-of-line
"\e[3~": delete-char
"\e[4~": end-of-line
```

tcsh

Vnos osembitnih znakov omogocimo, ce v nastavitveno datoteko (/etc/csh.login za skupne nastavitve oz. ~/.tcshrc ali ~/.cshrc za osebne nastavitve) dopisemo vrstico:

```
setenv LC_CTYPE sl_SI.iso88592
```

Med nami: kaze, da je ukazni lupini `tcsh` povsem vseeno, na kaj nastavite spremenljivko `LC_CTYPE`, samo da ni nedefinirana.

Pazite: ukazna lupina mora biti prevedena z vklopljeno izbiro ``8-bit''! To lahko preverite z ukazom ``set'':

```
rufko:~% set
...
version tcsh 6.06.00 (Cornell) 1995-05-13 (i386-intel-linux) options 8b,dl,al
rufko:~%
```

Pomemben je tisti ``8b''. Ce se pri vas ne izpise, vam verjetno ne preostane drugega, kot da snamete izvorno kodo in jo prevedete.

6.2 Terminalski emulatorji

xterm

V datoteki ~/.Xresources (osebna nastavitev) ali pa v datoteki /usr/lib/X11/app-defaults/XTerm (skupna nastavitev) dolocimo pisave, ki jih uporablja XTerm. Izberemo pisavo, ki ima vse znake enako siroke, npr. ,,fixed`` ali ,,Courier``, in je kodirana po razporedu ISO Latin 2. Zgled:

```
XTerm*font:     -*-fixed-medium-r-normal--16-*-iso8859-2
XTerm*boldFont: -*-fixed-bold-r-normal-sans-16-*-iso8859-2
```

Primerne pisave lahko izpisemo z ukazom `xlsfonts` (glej se razdelek X Window System poglavja ``Prikaz nasih znakov na zaslonu"):

```
xlsfonts -fn '*-fixed-*-iso8859-2'
```

rxvt

Po podatkih v ceskem HOWTO `rxvt` zaenkrat (razvojna izdaja 2.4.6) ne deluje z nasimi znaki brez posegov v izvorno kodo. Potrebne posege najdete v ``Czech HOWTO" (ki *ni* del arhiva LDP; glejte razdelek Zahvala, kje ga najdete).

6.3 Pisarniski paketi

WordPerfect

(Jaka Mele, Vse naredim pod Linuxom!, Win.Ini, l.6, st.9, str.60)

WordPerfect ima vse potrebne znake ze v svoji tabeli. Vse, kar moramo napraviti, je da za nase crke definiramo makroukaze. Recimo, da bi radi definirali crko S (velika crka S s stresico). Pricnemo s snemanjem makroukaza (recimo mu Scaron), skocimo v tabelo znakov, v njej izberemo crko S, zakljucimo makro in nazadnje pri nastavitvah povezemo makro z izbrano tipko.

(Jaka Mele, `Message-id: <Pine.LNX.4.05.9812241251460.15368-101000@ro.zrsss.si>`

```
---------- Forwarded message ----------
Date: Thu, 24 Dec 1998 11:04:33 +0000
From: "[ALAMUN ALEN" <alen.salamun@uni-mb.si>
To: jaka.mele@lugos.si
Subject: WordPerfect SLO tipkovnica

Hi!

Daj prosim spravi to v mojem imenu na lugos listo, ker je verjetno
trazena roba. Uspelo mi je definirati SLO tipkovnico brez makrojev. Sem na
fax-u pa ne morem poslati jaz na Lugos listo.

LP, Alen
```

Datoteko SLO.WGK s prireditveno tabelo za slovensko tipkovnico najdete na

`ftp://sizif.mf.uni-lj.si/pub/i18n/SLO.WGK`

(Andrej Komelj, `Message-id: <19981224182341.A3794@perun.mist.si>`)


```
Odgovor na vprasanje: "Kako je s podporo sumnikom v WP-ju 8.0 pod
Linuxom?" se glasi: SLABO.

1. Resitev v obliki "insert symbol -> multinational" je v teoriji
uporabna, vendar... Hja, multinational znaki so v WP Multinational
fontu, le-ta pa je najbolj podoben Roman-WP fontu. Ima "serife", zato
Helvetica in Zurich odpadeta. Vendar se tudi pri Roman-WP pojavijo
tezave, ce se le-tega uporablja v bold, italic ali bold-italic obliki.

2. Resitev v obliki uporabe Type 1 fontov z definiranimi slovenskimi
znaki (kje se jih dobi? odg: poglejte si v Slovenian-HOWTO, ali jih
skopirajte iz OS/2-ja, ce ga imate) pa zal tudi odpade, ker "brezplacna"
verzija WP-ja nima vsebovanega font installerja (xwpfi) in zato kakrsno
koli dodajanje fontov v wp.drs odpade... Skratka, tako kot je nekdo ze
napisal; brezplacen-in-neoskubljen WP je ena velika potegavscina... Za
Americane je mogoce se nekako uporaben, za povprecnega Slovenca, ki pise
tudi slovenska besedila, pa je popolnoma brez vrednosti. :-(

Nazaja k LyX-u in ze sestavljenim vzorcem. :-)

Mimogrede, kaj se je zgodilo z WP-jem?!? Dokler se ni bil v lasti
Corel-a, je znal postavljati stresice na katere koli znake v katerem
koli fontu. Z WP-jem 6.0 za DOS sem tako pisal tekste v TrueType, Type 1
in CG fontih... POVSOD sem lahko uporabljal sumnike. Huh, "napredek" pa
tak.

--
Kome.
```

Star Office

(Jaka Mele, Vse naredim pod Linuxom!, Win.Ini, l.6, st.9, str.60)

StarOffice je malce tezavnejsi, zato, ker rabi dodatne Type1 fonte, pa vendarle povejmo recept se zanj. Kot prvo si torej moramo dobiti slovenske oziroma latin2 pisave (pfb, afm, fonts.dir in fonts.scale) iz recimo ftp streznika drustva uporabnikov linuxa slovenije ftp://ftp.lugos.si/pub/lugos/arhiv/Type1-Latin2-Fonts/, datoteki l2afm004.zip in l2pfb004.zip). Nato arhiva odpremo v /usr/lib/X11/fonts/Type1.Latin2/, ter ta imenik dodamo v spisek imenikov s pisavami, ki jih uporabi XWindow streznik. To naredimo tako, da v datoteko /etc/X11/XF86Config dopisemo:

```
FontPath "/usr/X11R6/lib/X11/fonts/Type1.Latin2/"
```

Za tem pozenemo

```
xset fp+ /usr/X11R6/lib/X11/fonts/Type1.Latin2/
```

ter se `xset fp rehash`. Nato je potrebno v direktoriju /usr/local/StarOffice3-1/x3p/fontmetrics/ za `.afm` datoteke narediti povezave iz /usr/lib/X11/fonts/Type1.Latin2/. Podobno naredimo se za direktorij /usr/local/StarOffice3-1/x3p/pssoftfonts/, le da sem povezemo .pfb datoteke. Skoraj smo ze na koncu, v datoteko /usr/local/StarOffice-3.1/xp3/psstd.fonts dodamo po vzoru ze dodanih pisav v datoteki fonts.dir iz Latin2 podrocja, vse nove pisave, tako, da namesto npr. arial.pfb pride `arial`, namesto `-0-0-0-0` pa `-%d-%d-%d-%d...` Saj se takoj vidi. Na koncu si premapirajmo se tipkovnico in sicer s programom xmodmap, tako da izvrsimo `xmodmap xmodmap.si`. Ce imate starejsi Xwindow sistem, si boste morda morali xmodmap.si se popraviti, da vam bosta tipki Backspace in Delete delali normalno, sicer pa samo pozenete swriter3 in uzivate v dodatnih slovenskih pisavah.

Applixware

(Boris Pozar, `Message-Id: <346044BF.FE97E3AF@guest.arnes.si>`)

Z Lugosovega streznika snamemo pisave TrueType v kodnem naboru ISO 8859-2. Potem jih kot `root` s FontManagerjem dodamo v katalog pisav in zadeva je resena.

ftp://ftp.lugos.si/pub/lugos/arhiv/Type1-Latin2-Fonts/l2ttf004.zip

6.4 Urejevalniki

GNU emacs

Emacs je, tako kot njegov avtor, kontroverzni Richard M. Stallman, urejevalnik, glede katerega ljudje niso indiferentni: ali ga obozujejo, ali pa sovrazijo. Emacs je vec kot urejevalnik; je takorekoc nacin zivljenja. V njem lahko pisete programe, iz njega poganjate prevajalnik, pisete besedila, poganjate crkovalnik, poganjate TeX, berete posto in novice... Ko enkrat v resnici dobro obvladate Emacs, imate realne moznosti, da se vam se dobrsen del naslednjega stoletja ne bo treba nauciti nobenega novega urejevalnika. Dokazano pa ljudje shajajo tudi brez njega.

Starejse izdaje programa GNU emacs

Privzeta nastavitev urejevalnika je omejitev na sedembitne znake (ASCII). Nastavitev spremenimo tako, da v datoteko z nastavitvami dodamo naslednjih nekaj vrstic:

```
;;  Omogocimo izpis znakov s kodo vecjo od 127
(standard-display-european t)

;;  Omogocimo vnos znakov s kodo vecjo od 127
(set-input-mode (car (current-input-mode))
        (nth 1 (current-input-mode))
        0)
```

Prilagoditev Emacsa je mozno izvesti v eni od naslednjih datotek z nastavitvami:

- /usr/lib/emacs/site-lisp/site-start.el (skupne nastavitve)
- ~/.emacs (osebne nastavitve)
- /usr/lib/emacs/site-lisp/default.el (skupne nastavitve)

Z vnosom in izpisom nasih znakov seveda se ni konec zgodbe o prilagoditvi Emacsa za slovenscino. Zelimo na primer tudi, da se ukaza `forward-word` in `backward-word` ne bosta zatikala na nasih znakih, ter da tudi za nase znake pravilno tece pretvarjanje med velikimi in malimi crkami. S starejsimi izdajami Emacsa to ni vedno povsem trivialno. Bralcu bo v veliko pomoc zbirka nasvetov Slovenscina in racunalniki avtorja Alesa Kosirja, v kateri je programu GNU emacs so posveceno obsirno poglavje. Najdete jo na

http://nl.ijs.si/gnusl/tex/tslovene/slolang/slolang.html

GNU emacs 20

Z zadnjo izdajo urejevalnika Emacs in standardno slovensko tipkovnico je prilagoditev posebno enostavna. Vse, kar potrebujemo, je

```
;;; Turn on Latin-2 both as input and output method (Emacs 20.*
;;; and onwards)
(set-language-environment "Latin-2")
```


Kasneje lahko nacin Latin-2 izberemo za vnos in izpis v menuju Mule / Set Language Environment / European / Latin-2.

crisp

Crisp je urejevalnik v duhu urejevalnika Brief z burno zgodovino. Napisal ga je Paul Fox iz Anglije in ga distribuiral kot ``program na pokusino'' (shareware). Zadnja taksna razlicica je 2.2e iz leta 1991 in je po nekaterih vesteh avtor ne podpira vec. Novejse (trenutno 5.1) pa trzi podjetje Vital, Inc. iz Teksasa, ZDA. Crisp je nekako blizje pecejevskemu duhu, kot pa sta Emacs in Vi; ce se nobenega od teh dveh ne morete navaditi, je morda Crisp prava izbira.

Privzeta nastavitev urejevalnika Crisp 2.2e je, da znakov s kodami od 127 dalje ne prikaze, ampak izpise njihovo sestnajstisko kodo, ki jo uvede z znakoma `\x`. Pomaga, ce nalozimo paket z izvorno kodo Crisp in v datoteki $(SRC)/crisp2.2e/src/crunch/tty/xterm.cr popravimo drugi argument funkcije `set_term_features()` iz `NULL` na `"%c"`. Popravljeni makroukaz prevedemo:

```
crunch xterm.cr
```

Rezultat, popravljeni xterm.cm, postavimo med ostale, navadno v imenik /usr/local/lib/crisp/macros/tty.

Makro `xterm` velja, kot bi iz imena sami uganili, za Crisp, pognan v oknu XTerm znotraj X Window System. Enaka intervencija piscu teh vrstic ni uspela na linux.cr (konzola) in xcrisp.cr (razlicica Crisp za X11). Poleg tega se je na starejsi verziji Linuxa (1.0.8) `crunch` pritozeval nad predprocesorjem za C, /lib/cpp. Pomagala je logicna povezava:

```
ln -s  /usr/lib/gcc-lib/i486-linux/2.5.8/cpp /lib/cpp
```

Ales Kosir je napisal za Crisp makro `slovene`:

http://nl.ijs.si/gnusl/tex/tslovene/src/slovene.cr

Z njim lahko vnasamo nase znake tudi na tipkovnicah, ki nimajo primernih tipk na nacin: namesto c vtipkamo `"c`, makro pa pretvori zaporedje v ustrezen znak. Poleg ISO Latin 1 podpira se nekaj drugih kodiranj, ki so v uporabi v nasih krajih. Navodila za uporabo so na voljo tudi prek WWW:

http://nl.ijs.si/gnusl/tex/tslovene/slolang/

Del distribucije Caldera OpenLinux je tudi ``lahka'' izvedba urejevalnika Crisp, CRiSP LiTE. Caldera ponuja tudi moznost kasnejse ugodnejse nadgraditve na polno izvedbo. CRiSP LiTE nima zgoraj opisanih tezav z vnosom osembitnih znakov.

vi

Urejevalnik `vi` izvira iz sive davnine Unixa. Zaradi tezav z licencnino najdete na sistemih Linux namesto originalnega Vi navadno urejevalnik `vim` (Vi IMproved) avtorja Brama Moolenaarja, vcasih pa tudi ``elvis'', delo Steva Kirkendalla, ali `nvi` iz distribucije BSD 4.4. Ideje, na katerih temelji ``vi'' so v temeljih ortogonalne na vse, kar ste se o urejevalnikih naucili pod MS-DOS. Kljub temu pa je glede na to, da je ``vi'' edini urejevalnik, ki zanesljivo obstaja na cisto vsakem Unixu, pametno poznati nekaj osnovnih ukazov za delo z njim.

Po izkusnjah pisca tako originalni `vi` kot tudi `vim` sama po sebi brez tezav prepuscata osembitne znake. Tezave lahko vcasih povzrocajo zunanji programi, ki jih klicemo iz urejevalnika (npr. `fmt` za poravnavanje besedila). Navadno pomaga, ce v `~/.profile` dolocimo spremenljivko `LC_CTYPE` (glej poglavje Slovenski locale).

6.5 Elektronska posta

Pine

Pine najprej prebere skupne nastavitve iz /usr/local/lib/pine.conf (ali /etc/pine.conf, zatem pa se osebne nastavitve iz ~/.pinerc. V enem ali drugem poiscemo niz `character-set` in postavimo to spremenljivko na `ISO-8859-2`:

```
...
# character-set should reflect the capabilities of the display
# you have. Normal default is US-ASCII.  Typical alternatives
# include ISO-8859-x, where x is a number between 1 and 9.
character-set=ISO-8859-2
...
```

elm

Opozorilo: pisec ze dolgo ne uporablja vec tega programa in zapisanega ni preveril. Danski in nemski HOWTO trdita, da je bodisi dovolj v ukazni lupini postaviti spremenljivko `LANG` na vrednost `iso_8859_1`, bodisi v konfiguracijski datoteki `~/.elm/elmrc` nastaviti vrednosti naslednjih spremenljivk:

```
...
charset=iso-8859-1
displaycharset=iso-8859-1
textencoding=8bit
...
```

Vse ISO 8859-1 bi bilo pri nas verjetno potrebno nadomestiti z ISO 8859-2.

Se enkrat prosim koga, ki ta program v resnici uporablja, da opise svoje izkusnje.

Netscape Mail

V Netscape Mail 4.06 je mozno pisati nase znake v telesu sporocila, ne pa tudi v glavi. Ko imamo enkrat namescene nase pisave in urejeno preslikavo tipkovnice (razdelka Pisave pod X11 in Tipkovnica pod X11), le se v menuju View postnega programa izberemo ``Encoding'', tam pa ``Central European (ISO-8859-2)''.

Vesel bom opisa izkusenj z drugimi izdajami programa Netscape Mail.

6.6 Svetovni splet

Lynx

Lynx pozna od izdaje 2.5 dalje tudi kodno stran ISO 8859-2. Nastavimo jo lahko na enega od treh nacinov. Prvi je, da izberemo `O)ptions`, zatem pa `C)harset`, ter z pritiskanjem na kurzorske puscice izberemo zeljeni nabor znakov. Drugi je, da v datoteki ~/.lynxrc (osebna nastavitev) nastavimo

```
character_set=ISO Latin 2
```

In zadnji je, da v datoteki /usr/lib/lynx.cfg (skupna nastavitev)

```
CHARACTER_SET:ISO Latin 2
```

Netscape Navigator

(Metod Kozelj, `Message-id: <Pine.HPP.3.96.981216083349.17728E-100000@hmljhp.rzs-hm.si>`)

Netscape Navigator, pa tudi brkljalnik iz Netscape Communicator, uporabljata nabor znakov, ki ga nudi X Windows System. Za prikaz nasih znakov je torej najprej namestiti pisave za X11 (poglavje Pisave pod X11). Ko Navigator zazenemo, sam ugotovi, da so na voljo tudi pisave za 'Central European (ISO-8859-2)'. Ce pregledujemo stran, ki brkljalniku pravilno sporoci uporabljeni nabor znakov, potem Navigator avtomaticno uporabi pisavo s pravim kodnim naborom. Seveda lahko spremenimo osnovno velikost crk ali pa obliko za proporcionalno pisavo.

To storimo tako, da v 'Edit / Preferences / Appearance / Fonts' dolocimo, da spreminjamo nastavitve za srednjeevropske pisave ('For the Encoding / Central European (iso-8859-2)'), potem pa izberemo, s katero pisavo naj Navigator prikazuje tekste. Navigator nam sam ponudi le pisave s pravim naborom znakov.

Ce pregledujemo strani, ki imajo sicer pravilen razpored crk (razpored ISO 8859-2 in ne Windows-1250), pa Navigator ne uporabi pisave s pravim razporedom crk, lahko razpored preklopimo rocno z izbiro ,,View / Encoding / Central European (ISO-8859-2)``.

Ob izbiri pisave s pravilnim razporedom crk in ob pravilnem razporedu tipk na tipkovnici lahko uporabljamo nase crke tudi pri pisanju v modulu 'Page Composer'.

Netscape Communicator 4.06 in novejsi znajo na Unixih, ki imajo namescene pisave ISO 8859-2, prikazati tudi strani, napisane v kodni strani Microsoft CP1250, oznacene kot `windows-1250` v glavi.

Spletni strezniki

Opomba: spodnje besedilo je prevod navodil ,,Configuring WWW Server for ISO 8859-2`` http://sizif.mf.uni-lj.si/linux/cee/app/httpd.html, ta pa se v veliki meri naslanjajo na besedilo Marka Martinca ,,Resitev problema slovenskih sumnikov v HTML dokumentih`` http://www.ijs.si/doc/www-csz.html.

Nacela

1. Dolocilo HTTP zahteva, da prenosni protokol HTTP prenasa tok podatkov brez omejitve na nabor ASCII ali izpisljive znake. To vkljucuje osembitne znake, sestnajstbitne znake kot npr. ISO10646 ali daljnovzhodne jezike, slike, animacije, zvok ipd.
2. Vsebino spisa doloca glava MIME (vrstica `Content-Type`. To je edina informacija o zvrsti informacije, ki jo ima na voljo brskalnik, zato mora odrazati pravo vsebino spisa.
3. Privzeto kodni razpored (terminologija MIME/HTTP uporablja izraz ,,nabor znakov``) za spise HTML je ISO 10646 oz. njegova osembitna podmnozica ISO 8859-1. *Vsak* spletni brskalnik mora biti zmozen prikazati spis HTML, ki uporablja kodni razpored ISO 8859-1. Glava MIME za spise HTML, kodirane po ISO 8859-1, je:

   ```
   Content-Type: text/html
   ```
4. Dolocilo HTTP dovoljuje, da z dodatnim dolocilom `charset` v vrstici `Content-Type` glave MIME izberemo drug kodni razpored. Naceloma je dovoljen katerikoli kodni razpored, registriran pri IANA Character Set Registry. Iz prakticnih razlogov pa je priporocljivo, da se omejimo na naslednje entitete HTTP:

   ```
   charset = "US-ASCII"
                 | "ISO-8859-1" | "ISO-8859-2" | "ISO-8859-3"
                 | "ISO-8859-4" | "ISO-8859-5" | "ISO-8859-6"
                 | "ISO-8859-7" | "ISO-8859-8" | "ISO-8859-9"
                 | "ISO-2022-JP" | "ISO-2022-JP-2" | "ISO-2022-KR"
                 | "UNICODE-1-1" | "UNICODE-1-1-UTF-7" | "UNICODE-1-1-UTF-8"
   ```

```
                    | token
```

Entiteta, ki je definirana pri IANA Character Set Registry, mora predstavljati nabor znakov, kot je bil prijavlj IANA. Prazna entiteta implicira nabor ISO 8859-1 (kar vkljucuje US-ASCII, ki je njegova podmnozica). Sp se HTML, kodirane v skladu s kodnim razporedom ISO 8859-2, morajo potemtakem vsebovati v glavi MIME vrstico

```
Content-Type: text/html; charset=ISO-8859-2
```

Pri tem presledek med podpicjem in `charset=ISO-8859-2` ni obvezen.

5. Numericne entitete oblike `&#`*nnn*`;` dolocajo pismenko s kodo *nnn* v referencnem razporedu znakov (ISO 8859-1 oz. ISO 10646), ne pa v trenutno izbranem. Na ta nacin torej sploh ne moremo uporabljati nasih znakov, kodiranih skladno s kodnim razporedom ISO 8859-2, lahko pa uporabimo 16-bitne kode ISO 10646. Crko c (mali C s kljukico) lahko torej zapisemo kot `č`. In samo tako.

Prakticni oziri

Kdaj je uporaba kodnega razporeda ISO 8859-2 priporocljiva, kdaj pa ne? Znotraj *intraneta* (notranjega informacijskega sistema) v okolju, kjer se uporablja ISO 8859-2, je morda smiselno ta razpored uporabiti za *vse* spise. Ce pa je informacija namenjena sirsemu obcinstvu, pa je potrebno vedeti, da je uporabniki zunaj srednje ali vzhodne Evrope, kjer se ta kodni razpored uporablja, informacije morda ne bodo mogli prikazati. Veljavno dolocilo prenosnega protokola HTTP namrec od spletnih brskalnikov ne zahteva, da prikazejo spise HTML, ki uporabljajo kodni razpored ISO 8859-2. V primeru, da brskalnik ne zmore prikazati strani v zahtevanem kodnem razporedu, lahko ta problem ignorira in stran (nepravilno) prikaze v nekem drugem kodnem razporedu (navadno ISO 8859-1), ali pa ponudi, da spis shranimo na disk. Zato je priporocljivo, da vstopne tocke v vas informacijski sistem ne uporabljajo kodnih razporedov, razlicnih od ISO 8859-1.

Podrobnosti izvedbe

Obdelali bomo tri metode, s katerimi lahko spletni streznik prilagodimo, da pravilno postreze uporabniku s spisom, kodiranim po ISO 8859-2:

- dolocitev nove pripone za staticne spise, kodirane po ISO-8859-2, in vezava le-te na na primeren tip spisa v glavi MIME
- dinamicno prekodiranje spisov
- uporaba posebnosti streznika ali brskalnika

Dolocitev nove pripone za staticne spise

Ob zahtevku za spis vecina spletnih streznikov izve zvrst staticnih spisov iz pripone datoteke ter na osnovi tega izdela glavo MIME, s katero opremi spis, preden ga izroci prejemniku. Vec razlicnih pripon lahko ustreza isti zvrsti spisa (priponi `.jpeg` in `.jpg` na primer obe dolocata zvrst `image/jpeg`), obratno pa ne - ena pripona lahko doloca kvecjemu eno zvrst spisa.

Nobene splosne metode ni, ki bi dovoljevala uporabo pripone `.html` za spletne spise, kodirane bodisi v skladu s kodnim razporedom ISO 8859-1, bodisi ISO 8859-2, in pricakovala od spletnega streznika, da se bo nekako odlocil, s kaksno glavi MIME bo opremil ene in druge. Ena moznost je vsekakor, da pripono `.html` trajno vezemo na zvrst spisov HTML, kodirani skladno s kodnim razporedom ISO 8859-2. Ta resitev ima seveda ocitno pomanjkljivost za obcinstvo, ki ne more prikazati tega nabora znakov.

Tukaj predlagana resitev izkorisca zmoznost spletnih streznikov, da dolocijo novo pripono datoteke in jo vezejo na zvrst spisa `text/html; charset=ISO-8859-2` v glavi MIME. To moznost podpira vecina spletnih streznikov. Streznika W3C ter NCSA `httpd` (ter vse njune izpeljanke, npr. Apache) uporabljata ukaz `AddType` v nastavitveni datoteki za vezavo dodatnih zvrsti spisov na nove pripone datotek. Z enostavno in domisleno shemo (glej Resitev problema slovenskih sumnikov v HTML dokumentih lahko zagotovimo pravilne glave MIME za zvrsti spisov, ki

TOC

uporabljajo kodne razporede, razlicne od ISO 8859-1. Ce na primer dolocimo pripono `.html-l2` za spise, kodiran skladno s kodnim razporedom ISO 8859-2, moramo v nastavitveni datoteki dodati vrstico:

- Streznik W3C httpd (CERN httpd): v datoteki httpd.conf dodamo:

```
AddType  .html-l2     text/html;charset=ISO-8859-2  8bit 1.0
```

- Streznik NCSA httpd: v datoteki srm.conf dodamo:

```
AddType  text/html;charset=ISO-8859-2   .html-l2
```

- Streznik Apache se od NCSA httpd razlikuje samo po manjkajoci piki: v datoteki srm.conf dodamo:

```
AddType  text/html;charset=ISO-8859-2   html-l2
```

Dinamicno prekodiranje

Na vsakem spletnem strezniku, ki podpira vmesnik Common Gateway Interface (CGI) lahko izvedemo dinamicno prekodiranje spletnih spisov. Poleg dobrih strani (iz enega samega izvora lahko ponudimo spletni spis v mnogo razlicnih kodnih razporedih) ima ta metoda tudi slabe: pretvarjanje spisa po eni strani porablja procesorski cas streznika, po drugi pa dinamicnih spisov posredniki (angl. proxy) navadno ne shranjujejo v medpomnilniku, zato z njimi dodatno obremenjujemo komunikacijske vode.

Primera skriptov, ki izvajata dinamicno prekodiranje:

- slovene - paket za dinamicno pretvarjanje spletnih spisov iz kodnega razporeda JUS I.B1.002 v CP852, CP1250, ISO 8859-2 ter US-ASCII (s precrkovanjem iz CSZ v CSZ). Avtor Tadej Vodopivec, Ljubljana, Slovenija.
- pl-conv - skript v Perl za dinamicno pretvarjanje kodiranja spletnih spisov med ISO 8859-2, CP852, CP1250, Mazowia, Apple RomanCE in US-ASCII (s precrkovanjem). Avtor Jaroslaw Strzalkowski, Krakow, Poljska.

Sledijo tri resitve, ki uporabljajo posebnosti in nestandardne razsiritve programske opreme. Prvi dve se nanasata na spletni streznik, zadnja na priljubljeni spletni brskalnik Netscape Navigator. Istega problema se je lotil G. Nicol s predlogom ``MIME Header Supplemented File Type''.

Metainformacija za spletni streznik W3C

Spletni streznik W3C (nekoc poznan kot CERN WWW) omogoca dodajanje metainformacije v glave MIME odhodnih spisov.

Po privzeti nastavitvi isce streznik metainformacije v datotekah s pripono `.meta` podimenik `.web` imenika s spisi. Ce na primer spletne spise hranimo v imeniku `/WWW/Hypertext` in zelimo opremiti spis `isolatin2.html` v njem s pravilno glavo MIME, napravimo to takole:

- Ce se ne obstaja, najprej ustvarimo podimenik `.web`:

```
$ mkdir /WWW/Hypertext/.web
```

- Zatem v tem podimeniku ustvarimo datoteko z metainformacijami. Za datoteko isolatin2.html se bo ta imenovala isolatin2.html.meta. Datoteka mora vsebovati glavo MIME v obliki, kot jo predpisuje format RFC822. V nasem primeru je to ena sama vrstica:

```
Content-Type: text/html; charset=ISO-8859-2
```

TOC

Datoteke ASIS za streznik Apache

Streznik Apache doloca poseben tip datoteke, imenovan ASIS. Streznik poslje datoteko ASIS odjemniku natancno taksno, kot je (angl. ``as is''), ne da bi jo opremil z glavo, ki jo sicer zahteva protokol HTTP. Datoteka sama mora zato vsebovati vsa potrebna polja, ki jih zahteva protokol HTTP, tem sledi prazna vrstica in vsebina spisa HTML.

V nastavitveni datoteki za streznik dolocimo, da pripona `.asis` pripada zvrsti spisa `httpd/send-as-is`:

```
AddType httpd/send-as-is asis
```

Primer datoteke ASIS:

```
Status: 200 OK
Content-Type: text/html; charset=ISO-8859-2

<HTML>
<HEAD>
<TITLE>Hello world</TITLE>
</HEAD>
<BODY>
<H1>Zivjo svet!</H1>
</BODY>
</HTML>
```

Kot vidite, morate v datoteko vkljuciti tudi tromestno odzivno kodo streznika. Streznik *vedno* doda polji `Date:` in `Server:`, zato teh ne smemo vkljuciti v datoteko.

Znacka <META> z oznacbo HTTP-EQUIV

V dolocilu HTTP 2.0 predlagana znacka `<META>` z oznacbo `HTTP-EQUIV` je bila predlagana kot neobvezna metoda, s katero lahko streznik potegne dodatno metainformacijo iz zaglavja (`<HEAD>`) spletnega spisa in jo uporabi pri tvorjenju glave MIME. Piscu teh vrstic ni znano, da bi kateri streznik to v resnici tudi pocel. Tovrstno razclenjevanje spletnih spisov verjetno predstavlja preveliko obremenitev za procesor in bi se mocno poznalo pri odzivnosti spletnega streznika. Pac pa so namesto tega zaceli to informacijo uporabljati spletni brskalniki (Netscape Navigator, na primer, to pocenja od izdaje 2.0 dalje).

Primer spisa z metainformacijo o uporabljenem kodnem razporedu v zaglavju spisa:

```
<HTML>
<HEAD>
  <TITLE>Naslov<TITLE>
  <META HTTP-EQUIV="Content-Type"
      CONTENT="text/html; charset=ISO-8859-2">
</HEAD>

<BODY>
Besedilo...
</BODY>
</HTML>
```

6.7 Delo na oddaljenem racunalniku

TOC

Z ukazoma `telnet` ali `rlogin` lahko prek omrezja TCP/IP delamo na drugem racunalniku.

Tezave z osembitnimi znaki pri tem so redko posledica napacne nastavitve v maloprej omenjenih programov, saj privzeti nastavitvi pri obeh podpirata osembitne znake. V primeru tezav se poleg teh programov splaca pogledati se

- nastavitev ukazne lupine (poglavje Ukazne lupine)
- nastavitev terminala (fizicnega ali virtualnega). Eksplicitno ga lahko nastavimo z ukazom

```
stty -parenb -istrip cs8
```

- in na koncu res nastavitev programov `telnet` ali `rlogin`

Telnet

Telnet eksplicitno pripravimo do vzpostavitve osembitne zveze z ukazom `toggle binary`:

```
~> telnet
telnet> toggle binary
Negotiating binary mode with remote host.
telnet> open racunalnik
```

Telnet omogoca tudi, da si nastavitve za izbrane racunalnike vnesemo v nastavitveno datoteko, ~/.telnetrc. Vnosi imajo obliko:

```
...
biofiz.mf.uni-lj.si
        set binary true
...
```

rlogin

Privzet nacin zveze pri programu ``rlogin'' je osembiten, eksplicitno pa lahko to dolocimo z dolocilom ``-8'':

```
rlogin racunalnik -8
```

6.8 TeX in LaTeX

Poslovenjenje paketa LaTeX obsega

- 7- in 8-bitni nacin vnosa nasih znakov
- nacin izpisa nasih znakov
- poslovenjenje izpisov ``Contents'', ``Figure'' ipd.
- slovenska pravila za deljenje besed
- prilagoditev programa LyX: vizualnega okolja za vnos besedil v LaTeX in Linuxdoc DTD

Urejanje stvarnega kazala po slovenski abecedi zaenkrat se ne gre avtomaticno. Pokazemo, kako lahko tezavo z nekaj truda obvozimo.

Podporo za neangleske jezike v LaTeXu je poenotil paket `babel` Johannesa Braamsa z Nizozemske, ki je standardni del vseh sodobnih distribucij paketa TeX na Linuxu, kot sta teTeX in NTeX:

ftp://sunsite.informatik.rwth-aachen.de/pub/comp/tex/teTeX/
http://web.mathematik.uni-stuttgart.de/ntex/WWW/

Z uporabo izbire ``slovene'' v paketu `babel` se avtomaticno izberejo slovenska imena pod- in nadnapisov, naslovov ipd.,

slovenska pravila za zlogovanje besed (ce imamo namescene slovenske zlogovalne vzorce; poglavje Slovenska pravila za deljenje besed), in 7-bitni nacin vnosa znakov `"c"s"z`:

```
\usepackage[slovene]{babel}
```

V novi razlicici LaTeXa (LaTeX2e) lahko s paketoma `fontenc` in `inputenc` izberemo tudi 8-bitno kodno tabelo `T1` (znano tudi kot DC ali Cork) in 8-bitni vnos znakov. Primer celotnega spisa v LaTeXu:

```
\documentclass[a4paper]{article}
\usepackage[slovene]{babel}
\usepackage[T1]{fontenc}
\usepackage[latin2]{inputenc}
\begin{document}
Ceprav se pouk na zeleznicarski soli v Sentvidu obicajno konca
sele ob stirih, odpesacijo ucenci iz obcin Zelimlje in Zetale domov
ze cetrt cez tri.
\end{document}
```

LaTeX 2.09

V starejsi razlicici paketa LaTeX, LaTeX 2.09, ne moremo uporabiti mehanizma `inputenc`. Pomagamo si s stilom `latin2.sty` Andreja Brodnika. Primer uporabe:

```
\documentstyle[latin2]{article}
\begin{document}
...
\end{document}
```

Stil `latin2` je na voljo na vseh streznikih CTAN, npr.

`ftp://ftp.dante.de/tex-archive/macros/latex209/contrib/latin2/`

Slovenska pravila za deljenje besed

Datoteko s slovenskimi pravili za zlogovanje besed je v okviru svoje diplomske naloge leta 1990 izdelal Matjaz Vrecko, in jo predal v last slovenskemu drustvu uporabnikov sistema TeX (TeXCeH). Na voljo je za individualno uporabo brez namena pridobivanja premozenjske koristi. Od leta 1995 dalje zanjo skrbi dr. Leon Zlajpah z IJS, ki jo je posredoval tudi skrbniku sistema sistema `babel` za tujejezicno podporo v sistemu TeX, Johannesu Braamsu. Sistem `babel` praviloma dobimo skupaj z distribucijo sistema TeX. Trenutno zadnja izdaja je 2.3, datirana 1997-04-15.

V distribuciji *teTeX*, ki je najbolj priljubljena distribucija na Unixu, vklopimo podporo za slovenske delilne vzorce tako, da kot administrator pozenete program `texconfig`. V menuju izberemo moznost *HYPHEN - hyphenation table (tex/latex)*. Program `texconfig` pri tej izbiri pozene urejevalnik `vi` in vanj nalozi datoteko language.dat. Za tiste, ki ga ne poznate, najnujnejse o uporabi le-tega: levo, desno, gor in dol po besedilu se premikate s tipkami `h`, `l` (mali L), `j` in `k`. Znak pobrisete tako, da kurzor pripeljete nanj in pritisnete tipko `x`. Datoteko shranite in zakljucite z delom s pritiskom na tipko `Esc` in vpisom `:wq` (dvopicje, mali W, mali Q), ki mu sledi `Enter`. Oborozeni s tem znanjem v datoteki language.dat pobrisemo komentar (znak za procent) pred zapisom `slovene sihyph22.tex`, shranimo in zapustimo urejevalnik. Program `texconfig` bo avtomaticno ustvaril nove formatne datoteke z vgrajenimi delilnimi vzorci.

Pisec teh vrstic je imel nepojasnjene tezave z datoteko sihyph22.tex, ki so enako cudezno izginile, ko mu je dr. Zlajpah poslal novo izdajo slovenskih delilnih vzorcev. Dobimo jo lahko na ftp://sizif.mf.uni-lj.si/pub/i18n/tex/sihyph23.tex in jo

postavimo ob bok stari datoteki, navadno v imenik /usr/TeX/texmf/tex/generic/hyphen/. Pozenemo `texconfig`, ta najprej osvezimo seznam datotek (izbira *REHASH - rebuild ls-R database*), zatem pa ponovimo postopek iz prejsnjega odstavka, s tem, da ime `sihyph22.tex` popravimo v `sihyph23.tex`.

Dodatne informacije o slovenski skupini uporabnikov TeX najdete na

http://vlado.fmf.uni-lj.si/texceh/texceh.htm

LyX

(Roman Maurer, <roman.maurer@fmf.uni-lj.si>)

Ce uporabljamo LyX s standardno slovensko tipkovnico (102/105 tipk), ga lahko enostavno prilagodimo za delo s slovenscino:

1. Tipkovnica v strezniku Xwindow mora biti nastavljena tako, da z njo v splosnem lahko vnasamo nase znake (to je opisano v razdelku X Window System poglavja ``Tipkovnica'').
2. Pozenemo LyX in nastavimo:

   ```
   Options / Screen Fonts / Font Norm = iso8859-2
   ```

   Zal izgleda, da LyX tega podatka nikamor ne shrani. Zato moramo rocno dodati v datoteko lyxrc (sistemska je navadno v imeniku /usr/share/lyx/, uporabniska pa v ~/.lyx/) naslednjo vrstico:

   ```
   \font_norm iso8859-2
   ```
3. Odpremo (nek) dokument in izberemo:

   ```
   Layout / Document / Language = slovene
   Layout / Document / Encoding = latin2
   Layout / Paper / Papersize = A4
   Layout / Quotes = ,,text`` & Double
   ```
4. Ce zelimo te nastavitve uporabljati v vseh nasih dokumentih (in to seveda zelimo), uporabimo se izbiro

   ```
   Layout / Save layout as default
   ```
5. Kadar uporabimo razred "letter" namesto "article", odgovorimo pritrdilno na vprasanje:

   ```
   "Should I set some parameters to letter the defaults
   of this document class?"
   ```

Zaradi napake v LyXu do vkljucno 1.0.0pre2 ni mogoc vnos 8-bitnih znakov v matematicnem tekstovnem nacinu (math text mode). Pomagamo si tako, da postavimo kar celo formulo kot TeXovo kodo (`"tex-mode"` - s tem sicer izgubimo WYSIWYG). Razvijalci LyX-a so o napaki obvesceni in jo bodo verjetno popravili v naslednjih izdajah.

7-bitni vnos sumnikov kot `"c"s"z` je otezen, ker LyX avtomaticno pretvori dvojni narekovaj v dva enojna. Prva misel je, da si lahko spet pomagamo z izbiro `"tex-mode"`, udobnejsa moznost pa je, da predefiniramo dvojni narekovaj. V datoteko emacs.bind (ali pa cua.bind, kar ze uporabljamo) dodamo vrstico:

```
\bind "S-quotedbl" "accent-caron"
```

Nase znake bomo zdaj dobili s pritiskom na `"c"s"z`. S tem zal izgubimo avtomaticno pretvarjanje dvojnega narekovaja v dva enojna, a vsega pac ne moremo imeti...

Datoteka emacs.bind je v imeniku ~/.lyx/bind/. Ce je se ni, lahko napravimo kopijo sistemske z imenika /usr/share/lyx/bind/ in jo popravimo.

TOC

Ce imamo staro slovensko tipkovnico (ameriska s csz namesto ~{`), lahko uporabimo premapiranje tipkovnice v La
Alena Salamuna <alen.salamun@medinet.si>:

```
Options / Keyboard / Primary = slovene
```

Ustrezna datoteka slovene.kmap je tipicno v imeniku /usr/share/lyx/kbd/.

Takole pa lahko uporabimo crkovalnik ispell za crkovanje slovenskih spisov:

```
Edit / Spellchecker / Spellchecker Options
  * Use alternate language: slovensko
  * Extra special chars allowed in words: cszCSZ
```

V slovenski izdaji urejevalnika LyX pa:

```
Uredi / Crkovalnik / Izbire crkovalnika
  * Uporabi drug jezik: slovensko
  * Dodatni posebni znaki, dovoljeni v besedah: cszCSZ
```

Pregled besedila na zaslonu

V Evropi je standardni format papirja A4, kar moramo dopovedati tudi programom za pregled besedila na zaslonu:

XDvi

V datoteko ~/.Xresources (osebna nastavitev) dodamo vrstico

```
XDvi.paper:     a4
```

Ghostview

V datoteko ~/.Xresources (osebna nastavitev) ali pa v datoteko /usr/lib/X11/app-defaults/Ghostview (skupna nastavitev) dodamo vrstice:

```
!  Ghostview naj uporablja format A4
*pageMedia:             A4
!  Tudi Ghostscript naj uporablja format A4
*Ghostview.interpreter: gs
*Ghostview.arguments:   -sPAPERSIZE=a4
```

Ker tiskamo prek Ghostscripta tudi iz drugih programov, ne le iz Ghostview, je pametno, ce mu v nastavitveni datoteki (navadno `/usr/share/ghostscript/`*`izdaja`*`/gs_init.ps` ali `/usr/local/share/ghostscript/`*`izdaja`*`/gs_init.ps`) globalno izberemo velikost strani A4. Poiscemo spodnjo vrstico in jo odkomentiramo (pobrisemo zacetni znak za procent):

```
% (a4) /PAPERSIZE where { pop pop } { /PAPERSIZE exch def } ifelse
```

Stvarna kazala

Najpogostejse orodje za izdelavo stvarnih kazal v sistemu TeX je Makeindex. Ta je le omejeno lokaliziran (poleg angleskih pozna se nemska pravila za razvrscanje) in nase abecede ne pozna.

Stvarno kazalo v LaTeX izdelamo z ukazom `\makeindex` v preambuli; vsako besedo, ki jo zelimo vnesti v kazalo, pa

moramo posebej oznaciti z ukazom `\index{vnos}`. Ker navadno zelimo stvarno kazalo tudi izpisati, dodamo se `\printindex`, ta pa je definiran v makropaketu `makeidx`. Zgled, datoteki recimo zelez-sola.tex, naj ilustrira povedano:

TOC

```
\documentclass[a4paper]{article}
\usepackage[slovene]{babel}
\usepackage[T1]{fontenc}
\usepackage[latin2]{inputenc}
\usepackage{makeidx}
\makeindex
\begin{document}
Ceprav se pouk na zeleznicarski
soli\index{s~ola@sola!z~eleznic~arska@zeleznicarska} v
Sentvidu\index{S~entvid@Sentvid} obicajno konca sele ob stirih,
odpesacijo ucenci iz obcin Zelimlje\index{Z~elimlje@Zelimlje} in
Zetale\index{Z~etale@Zetale} domov ze cetrt cez tri.
\printindex
\end{document}
```

Trik je v tem, da `makeindex` dopusca skladnjo `\index{vnos1@vnos2}`, kjer je `vnos1` tisti, po katerem se kazalo ureja, `vnos2` pa besedilo, ki se v resnici izpise. Naso crko `C` zato pisemo kot `C~` (znak ~ je v kodnem razporedu ASCII za vsemi crkami), in tako bo zanesljivo uvrscena za vsemi C. Besedilo prevajamo v treh korakih:

```
latex zelez-sola
makeindex zelez-sola
latex zelez-sola
```

Prvi `latex` iz datoteke zelez-sola.tex izdela datoteko zelez-sola.idx, ukaz `makeindex` to uredi in iz nje izdela datoteko zelez-sola.ind, to pa vsrka drugi ukaz `latex`.

6.9 groff

Groff nudi precej omejeno podporo za nase znake. Pri izpisu na zaslon (nacin `nroff`) lahko goljufamo tako, da zapisemo datoteko z osembitnimi znaki ISO Latin 2, programu pa lazemo (izbira `-Tlatin1`), da so kodirani po razporedu ISO Latin 1.

Henryk Paluch je napisal paket `groff-latin2`, s katerim lahko nase znake vnasamo kot makroukaze, na primer:

```
\('a    a z ostrivcem
\(vc    c s stresico
\vou    u s krozcem
```

Izpis v obliki PostScript v casu pisanja tega spisa se vedno ne deluje, ceprav bi se mehanizme iz paketa ``ogonkify'' Juliusza Chroboczka podobno kot v programu ``a2ps'' verjetno dalo uporabiti tudi tu. Se najvecji problem je verjetno ta, da je paket ``groff'' osirotel, saj ga prvotni avtor, James Clark, ne vzdrzuje vec. Za paket `groff-latin2` zdaj skrbi Jan Kasprzak <kas@fi.muni.cz>, najdete ga na

ftp://ftp.fi.muni.cz/pub/localization/groff/

6.10 Perl

Od izdaje 5.004 dalje tudi Perl podpira lokalizacijo. Nastaviti moramo primerne spremenljivke okolja in bodisi v programu uporabiti klic

```
use locale;
```

bodisi poklicati Perl z izbiro `-Mlocale`. Nekaj primerov; preizkus smo si sposodili pri Janu Pazdziori:

```
~> LC_CTYPE=sl_SI.iso88592 perl -Mlocale -e '$_="tezisce\n"; s/\w/$&:/g; print;'
t:e:z:i:s:c:e:
~> LC_CTYPE=C perl -Mlocale -e '$_="tezisce\n"; s/\w/$&:/g; print;'
t:e:zi:sce:
~> LC_CTYPE=sl_SI.iso88592 perl -e '$_="tezisce\n"; s/\w/$&:/g; print;'
t:e:zi:sce:
```

Kot vidimo, je samo v prvem primeru, ko smo izvedli oboje: nastavili spremenljivko `LC_CTYPE` in Perl poklicali z izbiro `-Mlocale`, nase znake res uposteval kot crke (\).

6.11 Crkovanje: ispell

Vecina sodobnih distribucij Linuxa ze pride s crkovalnikom Ispell; manjka pa mu slovenski besedni zaklad. Tega najdete v datoteki

ftp://sizif.mf.uni-lj.si/pub/i18n/ispell/slovensko.tar.gz

Arhivska datoteka vsebuje dve datoteki, slovensko.hash in slovensko.aff. Postavimo ju tja, kjer Ispell ze hrani leksikone za ostale jezike. Pri piscu teh vrstic je to /usr/lib/ispell/; ce pri vas ni, si lahko morda pomagate s tem, da s find poskusite poiskati datoteko english.aff. Osnovni slovenski leksikon izvira iz baze, ki jo je podjetje Amebis d.o.o. prispevalo za mednarodni projekt MULTEXT-East; s posredovanjem dr. Tomaza Erjavca z odseka za inteligentne sisteme na Institutu Jozef Stefan pa je na voljo za nepridobitno uporabo.

Ispell uporabljamo tako, da mu v ukazni vrstici podamo jezik, v katerem je napisano besedilo, na primer english, deutsch, francais ali slovensko:

```
ispell -d slovensko besedilo.txt
```

Natancnejsa navodila za ravnanje s programom Ispell boste nasli v prirocniku, vec o prilagoditvi paketa `ispell` za slovenscino si lahko preberete na strani Slovenscina in racunalniki Alesa Kosirja. Ispell s slovenskim besednim zakladom poganja tudi spletni crkovalnik Trubar istega avtorja:

http://nl.ijs.si/cgi-bin/truform.pl

6.12 Pretvarjanje med razlicnimi kodiranji

Razen ISO Latin 2 vsebuje vse nase znake se nekaj drugih naborov: Microsoft Codepage 1250, IBM Codepage 852, JUS I.B1.002, Apple MacOS-CE, Apple MacOS-Croatian, Kamenicky, ISO Latin 4, ISO Latin 6, ISO Latin 7 itd.

Ceprav ni tezko napisati programcka za pretvorbo iz enega kodnega nabora v drugega, je pripravno uporabiti ze izdelan program. Eden takih je GNU recode, dostopen npr. na strezniku ARNES:

ftp://ftp.arnes.si/software/gnu/recode/recode-3.4.tar.gz

Starejse izdaje GNU recode (pred 3.5) ne poznajo Microsoftovih kodnih strani 1250-1259.

Uporaba je enostavna:

```
recode yu:latin2 datoteka
```

pretvori datoteko iz kodnega nabora JUS I.B1.002 v kodni nabor ISO Latin 2. Kot obicajno izda tudi tu izbira `--help` se dodatne moznosti.

Novejse izdaje programa recode (ki poznajo tudi nekaj vec kodnih naborov), uporabljajo malo drugacno skladnjo:

```
recode yu..latin2 datoteka
```

Po izkusnjah pisca pa je precrkovanje (npr. „nabori`` `latex` ali `flat`) se vedno omejeno na pretvorbo iz in v nabor ISO 8859-1. Novejse (preizkusne) izdaje programa recode najdete na:

ftp://ftp.iro.umontreal.ca/pub/recode/

6.13 Drugi programi

Pregledovalnik less

Less (avtor Mark Nudelman) je program za pregledovanje besedil na zaslonu, in je priljubljena zamenjava standardnega `more`.

Less prilagodimo za izpis osembitnih znakov z nastavitvijo v ~/.profile:

```
LESSCHARSET=latin1; export LESSCHARSET
```

`latin1` namesto `latin2` ni pomota. Less zaenkrat slednjega se ne pozna, ker pa ga pravzaprav zanima samo, kateri znaki so obicajni, kateri pa kontrolni (ki zahtevajo posebno obravnavo), ker pa je razpored znakov enak v celotni druzini ISO 8859, lahko izberemo kar latin1. Namesto tega lahko tudi eksplicitno navedemo zaporedje tipa znakov v naboru:

```
LESSCHARDEF="8bcccbcc18b95.33b."; export LESSCHARDEF
```

Vsi nabori iz druzine ISO 8859 imajo osem binarnih znakov (posebnih znakov, ki jih praviloma ni v besedilih), tri kontrolne znake, binarni znak, dva kontrolna znaka, osemnajst binarnih znakov, 95 navadnih znakov, 33 binarnih znakov, in normalne znake do konca tabele.

Koledar gcal

Vsi sistemi Unix imajo vgrajen koledarcek `cal`, ki pa je precej anglocentricen. Thomas Esken iz Nemcije je napisal alternativni koledarcek `gcal`. Od izdaje 2.40 dalje pozna tudi slovenske praznike. Najdemo ga na obicajnih mestih distribucije GNU. Najblizje je verjetno:

ftp://ftp.arnes.si/software/gnu/

Spodnja vrednost spremenljivke `GCAL` ustreza nasim razmeram:

```
GCAL="--cc-holidays=SI --gregorian-reform=1582"; export GCAL
```

---

7. Tiskanje

Pri tiskanju lahko naletimo na razlicne situacije:

- Tiskalnik neposredno podpira ISO Latin 2
- Tiskalnik podpira enega od ostalih osembitnih naborov z nasimi znaki (npr. kodno stran 852), ali pa sedembitni nabor JUS I.B1.002
- Tiskalnik razume PostScript
- Tiskalnik ne podpira nicesar drugega razen ASCII

Zadnja moznost ni predmet obravnave tega spisa.

7.1 Tiskalniki, ki podpirajo ISO Latin 2

Nekaj novejsih tiskalnikov ze neposredno podpira kodni razpored ISO Latin 2. Med njimi so

- Hewlett-Packard LaserJet 4 Plus in 4M Plus
- Hewlett-Packard LaserJet 5L
- Epson FX-2170 (NLSP)
- Epson LQ-870 (NLSP)
- Epson LQ-2070 (NLSP)
- Epson LQ-2170 (NLSP)
- Epson DLQ-3000 (NLSP; letnik 1996 in novejsi)
- Fujitsu DL700
- Fujitsu DL1150/1250
- Fujitsu DL3700/3800
- Fujitsu DL6400/6600

Vse, kar morate storiti, je izbrati pravilno kodno stran v tiskalniku. Ubezna sekvenca za tiskalnike Hewlett-Packard (PCL) in Epson (ESC/P2) so naslednje

PCL

Tole menda deluje:

```
\033(2N\033(s0p12h0s0b4099T
```

ESC/P2

Z ukazom `ESC ( t` prepisemo kodno tabelo ISO 8859-2 v drugo tabelo tiskalnika; zatem z ukazom `ESC t` izberemo drugo tabelo tiskalnika (strani C-73 in C-77 v ,,Epson ESC/P Reference

Manual``, december 1997):

```
\033(t\003\000\002\177\002\033t\002
```

Oznaka `\033` je osmiska koda znaka Escape.

Eden od nacinov, kako z /etc/printcap prilagodimo okolje za tiskanje na tak tiskalnik, je podrobneje opisan v razdelku Tiskalniki, ki podpirajo kak drug nabor z nasimi znaki.

7.2 Tiskalniki, ki podpirajo kak drug nabor z nasimi znaki

Ce tiskalnik podpira kak drug nabor z nasimi znaki (ne dolgo nazaj je bila ustaljena praksa, da je bil ROM v tiskalnikih zamenjan z EPROM, v njem pa svedski znaki zamenjani z znaki po JUS I.B1.002), lahko s primerno nastavitvijo v /etc/printcap se vedno tiskamo spise, kodirane po ISO Latin 2.

Primerno moramo dopolniti datoteko /etc/printcap. Za zgled si oglejmo tiskalnik Epson, ki ima znake po JUS I.B1.002 na mestu svedskega nabora.

```
lp|ascii|epson|Epson LQ-850:\
        :lp=/dev/lp1:\
        :sd=/usr/spool/lpd/epson:\
        :lf=/usr/spool/lpd/ERRORLOG:\
        :mx#0:\
        :sh:
jus|Epson LQ-850 z znaki JUS I.B1.002:\
        :lp=/dev/null:\
        :sd=/usr/spool/lpd/jus:\
        :if=/usr/spool/lpd/jus_filter:\
        :lf=/usr/spool/lpd/ERRORLOG:\
        :mx#0:\
        :sh:
latin2|text|Epson LQ-850 z znaki ISO Latin 2:\
        :lp=/dev/null:\
        :sd=/usr/spool/lpd/latin2:\
        :if=/usr/spool/lpd/latin2_filter:\
        :lf=/usr/spool/lpd/ERRORLOG:\
        :mx#0:\
        :sh:
```

Zgled ima tri tiskalniske vrste. Prva je dejanski tiskalnik z ameriskim (ASCII) naborom znakov. Druga uporablja filter `/usr/spool/lpd/jus_filter`:

TOC

```
#!/bin/sh
/usr/spool/lpd/jus_epson | lpr -Pascii
```

Datoteko torej precedimo skozi filter /usr/spool/lpd/jus_epson in jo postavimo v prvo tiskalnisko vrsto. Filter `jus_epson` ne napravi drugega, kot da na zacetku doda kodo za preklop v svedski nabor, prepise datoteko z vhoda na izhod, in na koncu doda kodo za preklop nazaj v ameriski nabor. To je tudi edini del programja, ki je odvisen od vrste tiskalnika:

```
#!/bin/sh
# Tiskanje besedil kodiranih po JUS I.B1.002 na tiskalnik Epson LQ
#
# Izberemo slovenski (svedski) nabor
/bin/echo -ne "\033R\005\c"
# Izpisemo dokument
cat
# Izberemo ameriski nabor
/bin/echo -ne "\033R\000\c"
```

Tretja vrsta uporablja podoben filter kot druga, le da ta najprej pretvori besedilo iz Latin 2 v JUS, zatem uporabi ze opisani filter za to, da doda na zacetku in na koncu potrebno kodo za tiskalnik, in koncno táko datoteko postavi v prvo vrsto.

```
#!/bin/sh
/usr/local/bin/recode --force latin2:yu | \
    /usr/spool/lpd/jus_epson | lpr -Pascii
```

Zgled predpostavlja, da je pot do programa recode /usr/local/bin/recode. Izbiro `--force` moramo uporabiti, ker je preslikava neobrnljiva -- v Latin 2 imamo tako sumnike kot zavite oklepaje, v JUS pa samo se prve.

Za ta zgled smo morali v imeniku /usr/spool/lpd ustvariti tri podimenike: /usr/spool/lpd/epson, /usr/spool/lpd/jus in /usr/spool/lpd/latin2.

Tiskanje gre zdaj tako:

1. Brez eksplicitne izbire vrste, ali pa z izbiro `-Pepson` ali `-Pascii` se datoteka odtisne kot ASCII:

   `lpr` *`datoteka`*

2. Z izbiro `-Pjus` se datoteka odtisne kot besedilo, kodirano po JUS I.B1.002:

   `lpr -Pjus` *`datoteka`*

3. Z izbiro `-Platin2` ali `-Ptext` se datoteka odtisne kot besedilo, kodirano po ISO Latin 2:

   `lpr -Platin2` *`datoteka`*

7.3 Tiskalniki, ki podpirajo kak drug nabor z nasimi znaki II

(Matjaz Terpin < matjaz.terpin@comcom.si>, `Message-id: <Pine.LNX.3.96.980709095750.24607B-400000@guru.comcom.si>`)

Matjaz Terpin < matjaz.terpin@comcom.si> je predlagal imenitno izboljsavo pravkar opisanega postopka, pri katerem lahko tiskamo nase znake *in* oglate in zavite oklepaje. Srz njegovega predloga je filter, ki pred vsakim nasim znakom preklopi v svedski nabor, takoj za njim pa spet nazaj v ASCII.

Datoteka /etc/printcap je v tem primeru taksna:

```
# /etc/printcap
lp|raw|yu:\
       :lp=/dev/lp1:\
       :sd=/var/spool/lpd/lp:\
       :lf=/var/spool/lpd/ERRORLOG:\
       :mx#0:\
       :sh:
lat2:\
       :lp=/dev/null:\
       :if=/etc/lp/lat2.filter:\
       :sd=/var/spool/lpd/lat2:\
       :lf=/var/spool/lpd/ERRORLOG:\
       :mx#0:\
       :sh:
```

Filter /etc/lp/lat2.filter precedi besedilo skozi naslednje sito, /etc/lp/lat2.switch, in ga postavi v vrsto `lp`:

```
#!/bin/sh
# /etc/lp/lat2.filter

/etc/lp/lat2.switch | lpr -Plp
```

„Meso`` postopka je v datoteki /etc/lp/lat2.switch:

```
#!/bin/sh
#/etc/lp/lat2.switch

echo -ne "\033R0"
```

```
sed -e "s/c/`echo -ne "\033R5\~\033R0"`/g" \
 -e "s/C/`echo -ne "\033R5\^\033R0"`/g" \
 -e "s/s/`echo -ne "\033R5\{\033R0"`/g" \
 -e "s/S/`echo -ne "\033R5\[\033R0"`/g" \
 -e "s/z/`echo -ne "\033R5\140\033R0"`/g" \
 -e "s/Z/`echo -ne "\033R5\@\033R0"`/g" \
 -e "s/c/`echo -ne "\033R5\}\033R0"`/g" \
 -e "s/C/`echo -ne "\033R5\]\033R0"`/g" \
 -e "s/d/`echo -ne "\033R5\|\033R0"`/g" \
 -e "s/D/`echo -ne "\033R5\\\\033R0"`/g"

echo -ne "\033R5"
```

Matjaz pravi se: sed slabo prebavi ``echo -ne "\000"'', zato uporabljam kar znak 0 (niclo), saj je tiskalnik (NEC pinwriter P7) zadovoljen tudi s tem.

7.4 Tiskalniki PostScript

``Naravni'' kodni nabor v PostScriptu je Adobe Standard Encoding. Nasih crk ne pozna. Pa ne le nasih, tudi mnogih drugih crk ne pozna, in pravzaprav sploh nima tovrstnih ambicij. Namesto tega je ASCII razsirjen z gradniki sestavljenih znakov: stresica, ostrivec, diereza, sedij itd. -- vsi so tam. Jezik PostScript pa omogoca definiranje sestavljenih znakov. Vse skupaj je cakalo samo se nekoga, ki bo definiral sestavljene znake za kodni nabor ISO Latin 2. Ta nekdo je bil Juliusz Chroboczek, ki je napisal paket ogonkify.

Filtra `a2ps` in GNU `enscript` sta bila napisana z namenom, da resita za nase kraje neobicajno zagato: kako odtisniti cisto obicajno besedilo na tiskalnik, ki razume edino PostScript. Eden in drugi postavita stran v PostScriptu z danim besedilom in jo posljeta na tiskalnik. Fleksibilnost PostScripta omogoca tudi transformacije (npr. pomanjsave) ali poudarjanje sintakse pri tiskanju programskih izpisov. Oba podpirata nabor ISO Latin 2.

- http://www.dcs.ed.ac.uk/home/jec/programs/ogonkify/
- http://www-inf.enst.fr/~demaille/a2ps/
- http://www.iki.fi/~mtr/genscript/
- ftp://ftp.arnes.si/gnu/enscript/enscript-1.6.1.tar.gz

ogonkify

Ogonkify je filter za izpise v PostScriptu, in omogoca pravilen odtis crk tudi v jezikih, ki ne uporabljajo nabora ISO Latin 1. Ker je filter, je ravno korak prepozen, da bi lahko pocel to elegantno, ampak mora prezveciti kakrsnokoli grdobijo pac izpljune program korak pred njim in iz dobljenega izdelati lep odtis. Programi pa izdelujejo datoteke PostScript vsak malo drugace, kar filtru ``ogonkify'' otezi delo. Zaenkrat podprti formati izpisa PostScript so: NCSA Mosaic, Netscape Navigator, StarOffice, Applixware in XFig.

TOC

Ogonkify za delovanje potrebuje Perl. Ko je namescen, `ogonkify` uporabimo takole za odtis spletn sestavka, shranjenega iz Navigatorja:

```
ogonkify -AT -N netscape.ps | lpr
```

GNU enscript

Ce GNU enscript se ni namescen, ga namestimo po navodilih, ki so prilozena programu. Poleg njega potrebujemo se vsaj eno pisavo z vsemi nasimi znaki, zapisan v obliki Type 1. Primeren je npr. IBM Courier, ki je prosto dostopen in se distribuira z X Window System, programom `ghostscript` ipd. Ce nismo prepricani, ali ga imamo ze namescenega, pozenemo:

```
find / -name cour.pf[ab]
```

V osebno (~/.enscriptrc) ali skupno konfiguracijsko datoteko (navadno /usr/local/etc/enscript.cfg ali /usr/local/etc/enscriptsite.cfg) dodamo pravilno pot do datotek s pisavami. Poleg tega potrebuje GNU enscript v vsakem imeniku s pisavami se datoteko z imenom font.map, ki vsebuje preslikavo med imeni pisav in datotekami, kjer so shranjene. Ustvarimo jo z ukazom `mkafmmap`.

Primer konfiguracijske datoteke:

```
...
#  Kje hranimo pisave PostScript?
AFMPath: /usr/lib/X11/fonts/Type1:/usr/local/lib/ghostscript/fonts
...
#  Privzeti nacin kodiranja:
DefaultEncoding: latin2
...
#  Privzeti format papirja:
DefaultMedia: A4
...
#  lpd izbira tiskalniske vrste z -P...
QueueParam: -P
...
#  Za tiskanje uporabljamo ukaz lpr, ne lp
Spooler: lpr
```

Ko je vse potrebno nastavljeno, odtisnemo besedilo enostavno z ukazom

```
enscript datoteka
```

a2ps

Z izdajo 4.7.21 je Akim Demaille v `a2ps` vgradil mehanizem za sestavljene znake, ki ga je Juliusz Chroboczek uporabil v filtru ogonkify.

Nasim krajem prilagodimo `a2ps` z majckeno spremembo nastavitvene datoteke. Sistemsko nastavitv datoteko najdemo navadno kot /usr/etc/a2ps.cfg, /usr/local/etc/a2ps.cfg ali kaj podobnega; osebna se imenuje ~/.a2psrc.

TOC

```
...
# Default encoding
Options: --encoding=latin2

# Default media
Options: --media=A4
...
```

---

8. Linux v Sloveniji

Harald T. Alvestrand z Norveske vodi evidenco o uporabnikih Linuxa po drzavah. Seznam slovenskih uporabnikov Linuxa je na naslovu

http://domen.uninett.no:29659/bycountry/SI.html

Ce vas ni med njimi, se prijavite!

http://domen.uninett.no:29659/

8.1 Skupina uporabnikov Linuxa

Od marca 1996 dalje deluje v Ljubljani organizirana skupina uporabnikov Linuxa v Sloveniji, drustvo LUGOS (Linux Users Group of Slovenia). Drustvo ima svojo stran na WWW:

http://www.lugos.si/

Drustvo ima tudi svoj elektronski spisek, na katerega se prijavite tako, da na naslov majordomo@lugos.si posljete vrstico

```
subscribe lugos-list
```

v telesu sporocila.

8.2 Strezniki FTP

Vzdrzevan seznam aktivnih arhivov v Sloveniji najdete na straneh streznika LUGOS,

http://www.lugos.si/arhiv/

Od marca 1997 dalje imamo v Sloveniji svoj Sun SITE, ki je usklajen z drugimi strezniki Sun SITE po svetu (Sun SITE na Univerzi Severne Karoline, ZDA, je eden glavnih centrov za Linux).

ftp://sunsite.fri.uni-lj.si/pub/linux/

Na strezniku ARNES se avtomaticno zrcalita distribucija Debian in programje iz projekta GNU,

ftp://ftp.arnes.si/software/unix/linux/debian/
ftp://ftp.arnes.si/software/gnu/

Streznik z zelo veliko izbiro programja za Linux je tudi na Gimnaziji v Murski Soboti,

ftp://mercur.s-gms.ms.edus.si/pub/UNIX/linux/

Studentski informacijski center je s pomocjo Zavoda za odprto druzbo uredil streznik FTP v Narodnem domu v Mariboru:

ftp://ftp.kibla.org/pub/Linux/

Distribucija Slackware se zrcali tudi na strezniku Fakultete za racunalnistvo in informatiko Univerze Ljubljani,

ftp://ftp.fri.uni-lj.si/pub/linux/

Wowen Goods for Linux se zrcalijo na strezniku Laboratorija za robotiko na Fakulteti za elektrotehniko, Univerza v Ljubljani:

http://robo.fe.uni-lj.si/woven/

Linux Documentation Project ter distribucija Debian se zrcalita na strezniku Centra za uporabno matematiko in teorijsko fiziko, Univerza v Mariboru:

http://www.camtp.uni-mb.si/linux/LDP/
ftp://ftp.camtp.uni-mb.si/pub/linux/debian/

8.3 Usenet

Linux nima svoje diskusijske skupine, ampak je vsa razprava o sistemih Unix zbrana v skupini si.comp.os.unix (kjer pa razprava o Linuxu res predstavlja znaten del prometa). Svojo skupino pa ima drustvo LUGOS: si.org.lugos.

8.4 Zastopniki za Linux

Agenda d.o.o., Lavriceva 3, Maribor (tel. 062/2297740), je uradni zastopnik podjetij Caldera, RedHat in SuSE v Sloveniji.

CD-ROM z distribucijami Linuxa se da navadno dobiti pri vseh bolje zalozenih trgovcih s tovrstno robo, npr. Infobia (tel. 061/614548) ali Mantis (tel. 061/1685473).

---

9. O tem spisu

Najnovejsa izdaja tega spisa je razen na uradnem mestu Linux Documentation Project dostopen tudi pri avtorju ter na domaci strani Slovenskega drustva uporabnikov Linuxa:

http://sizif.mf.uni-lj.si/linux/cee/Slovenian-HOWTO.html
http://www.lugos.si/delo/slo/HOWTO-sl/Slovenian-HOWTO.html

Vse pripombe na ta spis so dobrodosle na naslov `<primoz.peterlin@biofiz.mf.uni-lj.si>`, splosna vprasanja o Linuxu in poslovenjenju programov pa sodijo na dopisno listo lugos-slo@lugos.si ali v si.comp.os.unix.

9.1 Opombe k drugi izdaji

Druga izdaja je priblizno dvainpolkrat daljsa od prve, ki je izsla konec leta 1996. Izkazalo se je, da bi bil koristen kratek seznam stvari, ki jim moramo opraviti, ce zelimo usposobiti nase znake na konzoli ali v okolju X Window System, zato sem ga dodal na zacetek. Dodan je opis uporabe vektorskih pisav Type 1 in TrueType. Nastavitev *locale* predstavlja standarden nacin prilagoditve programov, zato je temu razdelku namenjena vecja pozornost. V celoti so novi opisi prilagoditev urejevalnika LyX ter pisarniskih paketov WordPerfect, Star Office in Applixware. Dodan je opis prilagoditve nove izdaje urejevalnika Emacs. Razdelek o tiskalnikih je razsirjen z domiselno alternativno razlicico tiskalniskega filtra; dodatno sta omenjena `ogonkify` in `a2ps`. Posodobljen je seveda tudi razdelek o Linuxu v Sloveniji. In morda se kaj, kar sem pozabil.

Ce je prva izdaja predstavljala nekaksne zbrane in urejene zapiske o problemih, s katerimi sem se sam sreceval, se zdaj nekoliko bolje zavedam tudi potreb drugih. Zato sem v to dodal tudi opis prilagoditev programov in programskih paketov, ki jih sam ne uporabljam, npr. Star Office, Applixware ali Word Perfect. Opisanih resitev nisem preizkusal; zaupam pac izkusnjam izvornih piscev. Te sem povsod tudi navedel in lahko stopite v stik z njimi.

9.2 Zahvala

Ce bralka ali bralec nista tega ze sama uganila, je cas, da priznam sam: noben strokovnjak za Linux nisem. Nekaj o lokalizaciji Linuxa sem se naucil iz spisov ``German HOWTO'' Winnfrieda Truemperja ter ``Danish/International HOWTO'' Thomasa Petersena, nekaj o internacionalizaciji kar tako iz spisov Michaela Gschwinda. Nadvse koristni so bila tudi ``Czech HOWTO'' skupine avtorjev, Polska strona ogonkowa, ter Slovenscina in racunalniki Alesa Kosirja. Prvi spis je spisala skupina avtorjev, obcasno ga objavijo v `cz.comp.linux`, zrcaljen je pa tudi na nekaj mestih na Ceskem in Slovaskem; drugo najdete na http://www.agh.edu.pl/ogonki/; stran vzdrzuje Jaroslaw Strzalkowski. ``Slovenscino in racunalnike'' najdete na http://nl.ijs.si/gnusl/tex/tslovene/slolang/.

Z namigi in nasveti so mi pomagali se (v abecednem vrstnem redu): Gorazd Bozic <gorazd.bozic@arnes.si>, Ales Casar <casar@uni-mb.si>, Tomaz Erjavec <tomaz.erjavec@ijs.si>, Matjaz Godec <gody@elgo.si>, Ales Kosir <ales.kosir@hermes.si>, Metod Kozelj <metod.kozelj@rzs-hm.si>, Renato Lukac <renato@s-gms.ms.edus.si>, Roman Maurer <roman.maurer@fmf.uni-lj.si>, Jaka Mele <jack@ro.zrsss.si>, Borut Mrak <borutm@bigfoot.com>, Boris Pozar <boris.pozar@guest.arnes.si>, Marko Samastur <marko.samastur@fmf.uni-lj.si>, Robert Susic <robert.susic@ijs.si>, Alen Salamun <alen.salamun@medinet.si>, Matjaz Terpin <matjaz.terpin@comcom.si>, in Leon Zlajpah <leon.zlajpah@ijs.si> Vsem hvala. Vsem po nemarnosti izpuscenim se opravicujem; poskusil bom popraviti do naslednje izdaje.

9.3 Vprasanja

Se pravi Slovenian ali Slovene?

Mnenja so deljena. Pisec teh vrstic se je pozanimal pri strokovnjakih s tega podrocja, ki so ga le potrdili v tem, da so mnenja **res** deljena. Slovnicno je pravilno eno ali drugo, odlocitev za eno ali drugo varianto je zaenkrat bolj domena jezikovne estetike. Iz zgodovinskih razlogov ta spis uporablja Slovenian, in tako bo do nadaljnjega tudi ostalo.

Nasi znaki v tem spisu

Cel spis govori o tem, kako pridemo do nasih znakov, sam jih pa nima. Kako to?

Kovaceva kobila... Sistem SGML-Tools (nekdaj Linuxdoc-SGML), ki je uradni standard za dokumentacijo HOWTO, zaenkrat podpira samo Latin 1. Cees de Groot, vzdrzevalec paketa SGML-Tools, kot tudi Tim Bynum, urednik zbirke HOWTO, se tezave zavedata. Pisec teh vrstic v upanju na boljse case pise v Latin-2; preden poslje material uredniku, pa poreze kljukice nad sumevci. Za domaco rabo pa lahko malo pogoljufamo in paketu `sgmltools` lazemo, da so osembitni znaki kodirani po Latin 1: pri pretvorbi v navadno besedilo je tako ali tako vseeno, pri pretvorbi v HTML moramo dodati vrstico <META> z definicijo uporabljenega nabora znakov, pri pretvorbi v LaTeX pa morebiti se malo telovaditi s `sed`.

9.4 Uporaba in razsirjanje

Copyright (c) 1996,1999 Primoz Peterlin <primoz.peterlin@biofiz.mf.uni-lj.si>. Ta spis lahko razsirjate pod pogoji, navedenimi v licenci LDP, dosegljivi na naslovu http://metalab.unc.edu/LDP/COPYRIGHT.html. Spisa ni dovoljeno razsirjati v spremenjeni obliki brez privolitve avtorja.

Informacije v tem spisu so podane v veri, da so tocne in koristne, vendar pa brez kakrsnegakoli jamstva, izrecnega ali impliciranega, vkljucno, a brez omejitve na implicirano jamstvo za primernost za trzenje ali uporabo. Uporabljate jih lahko izkljucno na lastno odgovornost. Avtor ne odgovarja za kakrsnekoli skodo, posredno ali neposredno, nastalo kot posledico uporabe informacij v tem spisu.

9.5 Note to English readers

Copyright (c) 1996,1999 by Primoz Peterlin <primoz.peterlin@biofiz.mf.uni-lj.si>. This document may be distributed under the terms set forth in the LDP license at http://metalab.unc.edu/LDP/COPYRIGHT.html. This document may not be distributed in modified form without consent of the author.

The information in this document is provided in belief of being accurate and useful, however without any warranty, expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. The entire risk of using the information goes with the user. The author can not be liable for any possible damage, direct or indirect, arising from the use of information provided in this document.

Building and Installing Software Packages for Linux

Mendel Cooper --- http://personal.riverusers.com/~thegrendel/

v1.91, 27 July 1999

This is a comprehensive guide to building and installing "generic" UNIX software distributions under Linux. Additionally, there is some coverage of "rpm" and "deb" pre-packaged binaries.

1. Introduction

2. Unpacking the Files

3. Using Make

4. Prepackaged Binaries

- 4.1 Whats wrong with rpms?
- 4.2 Problems with rpms: an example

5. Termcap and Terminfo Issues

6. Backward Compatibility With a.out Binaries

- 6.1 An Example

TOC

7. Troubleshooting

- 7.1 Link Errors
- 7.2 Other Problems
- 7.3 Tweaking and fine tuning
- 7.4 Where to go for more help

8. Final Steps

9. First Example: Xscrabble

10. Second Example: Xloadimage

11. Third Example: Fortune

12. Fourth Example: Hearts

13. Fifth Example: XmDipmon

14. Where to Find Source Archives

15. Final Words

16. References and Further Reading

17. Credits

---

1. Introduction

Many software packages for the various flavors of UNIX and Linux come as compressed archives of source files. The same package may be "built" to run on different target machines, and this saves the author of the software from having to produce multiple versions. A single distribution of a software package may thus end up running, in various incarnations, on an Intel box, a DEC Alpha, a RISC workstation, or even a mainframe. Unfortunately, this puts the responsibility of actually "building" and installing the software on the end user, the de facto "system administrator", the fellow sitting at the keyboard -- you. Take heart, though, the process is not nearly as terrifying or mysterious as it seems, as this guide will demonstrate.

2. Unpacking the Files

You have downloaded or otherwise acquired a software package. Most likely it is archived (*tarred*) and compressed (*gzipped*), in `.tar.gz` or `.tgz` form (familiarly known as a "tarball"). First copy it to a working directory. Then *untar* and *gunzip* it. The appropriate command for this is **tar xzvf *filename***, where *filename* is the name of the software file, of course. The de-archiving process will usually install the appropriate files in subdirectories it will create. Note that if the package name has a *.Z* suffix, then the above procedure will serve just as well, though running **uncompress**, followed by a **tar xvf** also works. You may preview this process by a **tar tzvf filename**, which lists the files in the archive without actually unpacking them.

The above method of unpacking "tarballs" is equivalent to either of the following:

- gzip -cd filename | tar xvf -
- gunzip -c filename | tar xvf -

(The '-' causes the *tar* command to take its input from `stdin`.)

Source files in the new *bzip2* (`.bz2`) format can be unarchived by a **bzip2 -cd filename | tar xvf -**, or, more simply by a **tar xyvf filename**, assuming that `tar` has been appropriately patched (refer to the Bzip2 HOWTO for details). Debian Linux uses a different patch for `tar`, one written by Hiroshi Takekawa, so that the *-I*, *--bzip2*, *--bunzip2* options work with that particular `tar` version.

[Many thanks to R. Brock Lynn and Fabrizio Stefani for corrections and updates on the above information.]

Sometimes the archived file must be untarred and installed from the user's home directory, or perhaps in a certain other directory, such as `/`, `/usr/src`, or `/opt`, as specified in the package's config info. Should you get an error message attempting to untar it, this may be the reason. Read the package docs, especially the `README` and/or `Install` files, if present, and edit the config files and/or `Makefiles` as necessary, consistent with the installation instructions. Note that you would **not** ordinarily alter the `Imake` file, since this could have unforseen consequences. Most software packages permit automating this process by running **make install** to emplace the binaries in the appropriate system areas.

- You might encounter `shar` files, or *shell archives*, especially in the source code newsgroups on the Internet. These remain in use because they are readable to humans, and this permits newsgroup moderators to sort through them and reject unsuitable ones. They may be unpacked by the **unshar filename.shar** command. Otherwise the procedure for dealing with them is the same as for "tarballs".
- Some source archives have been processed using nonstandard DOS, Mac, or even Amiga compression utilities such *zip*, *arc*, *lha*, *arj*, *zoo*, *rar*, and *shk*. Fortunately, Sunsite and other places have Linux uncompression utilities that can deal with most or all of these.

Occasionally, you may need to update or incorporate bug fixes into the unarchived source files using a `patch` or `diff` file that lists the changes. The doc files and/or `README` file will inform you should this

be the case. The normal syntax for invoking Larry Wall's powerful *patch* utility is **patch < patchfile**

You may now proceed to the build stage of the process.

3. Using Make

The `Makefile` is the key to the build process. In its simplest form, a Makefile is a script for compiling or building the "binaries", the executable portions of a package. The Makefile can also provide a means of updating a software package without having to recompile every single source file in it, but that is a different story (or a different article).

At some point, the Makefile launches `cc` or `gcc`. This is actually a preprocessor, a C (or C++) compiler, and a linker, invoked in that order. This process converts the source into the binaries, the actual executables.

Invoking *make* usually involves just typing **make**. This generally builds all the necessary executable files for the package in question. However, make can also do other tasks, such as installing the files in their proper directories (**make install**) and removing stale object files (**make clean**). Running **make -n** permits previewing the build process, as it prints out all the commands that would be triggered by a make, without actually executing them.

Only the simplest software uses a generic Makefile. More complex installations require tailoring the Makefile according to the location of libraries, include files, and resources on your particular machine. This is especially the case when the build needs the `X11` libraries to install. *Imake* and *xmkmf* accomplish this task.

An `Imakefile` is, to quote the man page, a "template" Makefile. The imake utility constructs a Makefile appropriate for your system from the Imakefile. In almost all cases, however, you would run **xmkmf**, a shell script that invokes imake, a front end for it. Check the README or INSTALL file included in the software archive for specific instructions. (If, after dearchiving the source files, there is an `Imake` file present in the base directory, this is a dead giveaway that **xmkmf** should be run.) Read the `Imake` and `xmkmf` man pages for a more detailed analysis of the procedure.

Be aware that `xmkmf` and `make` may need to be invoked as root, especially when doing a **make install** to move the binaries over to the `/usr/bin` or `/usr/local/bin` directories. Using make as an ordinary user without root privileges will likely result in *write access denied* error messages because you lack write permission to system directories. Check also that the binaries created have the proper execute permissions for you and any other appropriate users.

Invoking **xmkmf** uses the `Imake` file to build a new Makefile appropriate for your system. You would normally invoke **xmkmf** with the **-a** argument, to automatically do a *make Makefiles*, *make includes*, and *make depend*. This sets the variables and defines the library locations for the compiler and linker. Sometimes, there will be no `Imake` file, instead there will be an `INSTALL` or `configure` script that will accomplish this purpose. Note that if you run `configure`, it should be invoked as **./configure** to ensure that the correct `configure` script in the current directory is called. In most cases, the `README` file included with the distribution will explain the install procedure.

It is usually a good idea to visually inspect the `Makefile` that `xmkmf` or one of the install scripts

builds. The Makefile will normally be correct for your system, but you may occasionally be required "tweak" it or correct errors manually.

Installing the freshly built binaries into the appropriate system directories is usually a matter of running **make install** as root. The usual directories for system-wide binaries on modern Linux distributions are `/usr/bin`, `/usr/X11R6/bin`, and `/usr/local/bin`. The preferred directory for new packages is `/usr/local/bin`, as this will keep separate binaries not part of the original Linux installation.

Packages originally targeted for commercial versions of UNIX may attempt to install in the `/opt` or other unfamiliar directory. This will, of course, result in an installation error if the intended installation directory does not exist. The simplest way to deal with this is to create, as root, an `/opt` directory, let the package install there, then add that directory to the `PATH` environmental variable. Alternatively, you may create symbolic links to the `/usr/local/bin` directory.

Your general installation procedure will therefore be:

- Read the `README` file and other applicable docs.
- Run **xmkmf -a**, or the `INSTALL` or `configure` script.
- Check the `Makefile`.
- If necessary, run **make clean**, **make Makefiles**, **make includes**, and **make depend**.
- Run **make**.
- Check file permissions.
- If necessary, run **make install**.

Notes:

- You would not normally build a package as root. Doing an **su** to root is only necessary for installing the compiled binaries into system directories.
- After becoming familiar with *make* and its uses, you may wish to add additional optimization options passed to `gcc` in the standard `Makefile` included or created in the package you are installing. Some of these common options are *-O2*, *-fomit-frame-pointer*, *-funroll-loops*, and *-mpentium* (if you are running a Pentium cpu). Use caution and good sense when modifying a `Makefile`!
- After the *make* creates the binaries, you may wish to **strip** them. The **strip** command removes the symbolic debugging information from the binaries, and reduces their size, often drastically. This also disables debugging, of course.
- The Pack Distribution Project offers a different approach to creating archived software packages, based on a set of Python scripting tools for managing symbolic links to files installed in separate *collection directories*. These archives are ordinary *tarballs*, but they install in `/coll` and `/pack` directories. You may find it necessary to download the *Pack-Collection* from the above site should you ever run across one of these distributions.

---

4. Prepackaged Binaries

4.1 Whats wrong with rpms?

Manually building and installing packages from source is apparently so daunting a task for some Linux users that they have embraced the popular *rpm* and *deb* or the newer Stampede *slp* package formats. While it may be the case that an *rpm* install normally runs as smoothly and as fast as a software install in a certain other notorious operating system, some thought should certainly be given to the disadvantages of self-installing, prepackaged binaries.

First, be aware that software packages are normally released first as "tarballs", and that prepackaged binaries follow days, weeks, even months later. A current *rpm* package is typically at least a couple of minor version behind the latest "tarball". So, if you wish to keep up with all the 'bleeding edge' software, you might not wish to wait for an *rpm* or *deb* to appear. Some less popular packages may never be *rpm*'ed.

Second, the "tarball" package may well be more complete, have more options, and lend itself better to customization and tweaking. The binary rpm version may be missing some of the functionality of the full release. Source *rpm*'s contain the full source code and are equivalent to the corresponding "tarballs", and they likewise need to be built and installed using either of the **rpm --recompile packagename.rpm** or **rpm --rebuild packagename.rpm** options.

Third, some prepackaged binaries will not properly install, and even if they do install, they could crash and core-dump. They may depend on different library versions than are present in your system, or they may be improperly prepared or just plain broken. In any case, when installing an *rpm* or *deb* you necessarily trust the expertise of the persons who have packaged it.

Finally, it helps to have the source code on hand, to be able to tinker with and learn from it. It is much more straightforward to have the source in the archive you are building the binaries from, and not in a separate source *rpm*.

Installing an *rpm* package is not necessarily a no-brainer. If there is a dependency conflict, an *rpm* install will fail. Likewise, should the *rpm* require a different version of libraries than the ones present on your system, the install may not work, even if you create symbolic links to the missing libraries from the ones in place. Despite their convenience, *rpm* installs often fail for the same reasons "tarball" ones do.

You must install *rpm*'s and *deb*'s as root, in order to have the necessary write permissions, and this opens a potentially serious security hole, as you may inadvertently clobber system binaries and libraries, or even install a *Trojan horse* that might wreak havoc upon your system. It is therefore important to obtain *rpm* and *deb* packages from a "trusted source". In any case, you should run a 'signature check' (against the MD5 checksum) on the package, **rpm --checksig packagename.rpm**, before installing. Likewise highly recommended is running **rpm -K --nopgp packagename.rpm**. The corresponding commands for *deb* packages are **dpkg -I | --info packagename.deb** and **dpkg -e | --control packagename.deb**.

- `rpm --checksig gnucash-1.1.23-4.i386.rpm`

  `gnucash-1.1.23-4.i386.rpm: size md5 OK`
- `rpm -K --nopgp gnucash-1.1.23-4.i386.rpm`

```
gnucash-1.1.23-4.i386.rpm: size md5 OK
```

For the truly paranoid (and, in this case there is much to be said for paranoia), there are the *unrpm* and *rpmunpack* utilities available from the Sunsite utils/package directory for unpacking and checking the individual components of the packages.

Klee Diene has written an experimental *dpkgcert* package for verifying the integrity of installed *.deb* files against MD5 checksums. It is available from the Debian ftp archive. The current package name / version is *dpkgcert_0.2-4.1_all.deb*. The Jim Pick Software site maintains an experimental server database to provide *dpkgcert* certificates for the packages in a typical Debian installation.

In their most simple form, the commands **rpm -i packagename.rpm** and **dpkg --install packagename.deb** automatically unpack and install the software. Exercise caution, though, since using these commands blindly may be dangerous to your system's health!

Note that the above warnings also apply, though to a lesser extent, to Slackware's *pkgtool* installation utility. All "automatic" software installations require caution.

The martian and alien programs allow conversion between the *rpm*, *deb*, Stampede *slp*, and *tar.gz* package formats. This makes these packages accessible to all Linux distributions.

Carefully read the man pages for the *rpm* and *dpkg* commands, and refer to the RPM HOWTO, TFUG's Quick Guide to Red Hat's Package Manager, and The Debian Package Management Tools for more detailed information.

4.2 Problems with rpms: an example

Jan Hubicka wrote a very nice fractal package called *xaos*. At his home page, both `.tar.gz` and `rpm` packages are available. For the sake of convenience, let us try the rpm version, rather than the "tarball".

Unfortunately, the rpm of *xaos* fails to install. Two separate rpm versions misbehave.

rpm -i --test XaoS-3.0-1.i386.rpm

```
error: failed dependencies:
        libslang.so.0 is needed by XaoS-3.0-1
        libpng.so.0 is needed by XaoS-3.0-1
        libaa.so.1 is needed by XaoS-3.0-1
```

rpm -i --test xaos-3.0-8.i386.rpm

```
error: failed dependencies:
        libaa.so.1 is needed by xaos-3.0-8
```

The strange thing is that `libslang.so.0`, `libpng.so.0`, and `libaa.so.1` are all present in `/usr/lib` on the system tested. The rpms of *xaos* must have been built with slightly different versions of those libraries, even if the release numbers are identical.

As a test, let us try installing `xaos-3.0-8.i386.rpm` with the *--nodeps* option to force the install. A trial run of *xaos* crashes.

TOC

```
xaos: error in loading shared libraries: xaos: undefined symbol: __f
```

Let us stubbornly try to get to the bottom of this. Running *ldd* on the *xaos* binary to find its library dependencies shows all the necessary shared libraries present. Running *nm* on the `/usr/lib/libaa.so.1` library to list its symbolic references shows that it is indeed missing *__fabsl*. Of course, the absent reference *could* be missing from one of the other libraries... There is nothing to be done about that, short of replacing one or more libraries.

Enough! Download the "tarball", `XaoS-3.0.tar.gz`, available from the ftp site, as well as from the home page. Try building it. Running **./configure**, **make**, and finally (as root) **make install**, works flawlessly.

This is one of an number of examples of prepackaged binaries being more trouble than they are worth.

---

5. Termcap and Terminfo Issues

According to its man page, *"terminfo is a data base describing terminals, used by screen-oriented programs..."*. It defines a generic set of control sequences (escape codes) used to display text on terminals, and makes possible support for different terminal hardware without the need for special drivers. The *terminfo* libraries are located in `/usr/share/terminfo` on modern Linux distributions.

The *terminfo* database has largely supplanted the older *termcap* and the totally obsolete *termlib* ones. This is usually of no concern for program installation except when dealing with a package that requires *termcap*.

Most Linux distributions now use *terminfo*, but still retain the older termcap libraries for compatibility with legacy applications (see `/etc/termcap`). Sometimes there is a special compatibility package that needs to be installed to facilitate use of termcap linked binaries. Very occasionally, an *#define termcap* statement might need to be commented out of a source file. Check the appropriate doc files for your particular distribution for definitive information on this.

6. Backward Compatibility With a.out Binaries

In a very few cases, it is necessary to use a.out binaries, either because the source code is not available or because it is not possible to build new ELF binaries from the source for some reason.

As it happens, ELF installations almost always have a complete set of a.out libraries in the `/usr/i486-linuxaout/lib` directory. The numbering scheme for a.out libraries differs from that of ELF ones, cleverly avoiding conflicts that could cause confusion. The a.out binaries should therefore be able to find the correct libraries at runtime, but this might not always be the case.

Note that the kernel needs to have a.out support built into it, either directly or as a loadable module. It may be necessary to rebuild the kernel to enable this. Moreover, some Linux distributions require installation of a special compatibility package, such as Debian's `xcompat` for executing a.out X applications.

6.1 An Example

Jerry Smith wrote a very handy *rolodex* program some years back. It uses the Motif libraries, but fortunately is available as a statically linked binary in a.out format. Unfortunately, the source requires numerous tweaks to rebuild using the *lesstif* libraries. Even more unfortunately, the a.out binary bombs on an ELF system with the following error message.

```
xrolodex: can't load library '//lib/libX11.so.3'
No such library
```

As it happens, there is such a library, in `/usr/i486-linuxaout/lib`, but xrolodex is unable to locate it at run time. The simple solution is to provide a symbolic link in the `/lib` directory:

```
ln -s /usr/i486-linuxaout/lib/X11.so.3.1.0 libX11.so.3
```

It turns out to be necessary to provide similar links for the libXt.so.3 and libc.so.4 libraries. This needs to be done as root, of course. Note that you should make absolutely certain you will not overwrite or cause version number conflicts with pre-existing libraries. Fortunately, the new ELF libraries have higher version numbers than the older a.out ones, to anticipate and forestall just such problems.

After creating the three links, *xrolodex* runs fine.

The *xrolodex* package was originally posted on Spectro, but seems to vanished from there. It may currently be downloaded from Sunsite as a *tar.Z* format source file [512k].

7. Troubleshooting

If *xmkmf* and/or *make* succeeded without errors, you may proceed to the next section. However, in "real life", few things work right the first time. This is when your resourcefulness is put to the test.

7.1 Link Errors

- Suppose *make* fails with a `Link error: -lX11: No such file or directory`, even after xmkmf has been invoked. This may mean that the *Imake* file was not set up properly. Check the first part of the *Makefile* for lines such as:

  ```
  LIB=            -L/usr/X11/lib
  INCLUDE=        -I/usr/X11/include/X11
  LIBS=           -lX11 -lc -lm
  ```

  The `-L` and `-I` switches tell the compiler and linker where to look for the *library* and *include* files, respectively. In this example, the *X11 libraries* should be in the `/usr/X11/lib` directory, and the *X11 include files* should be in the `/usr/X11/include/X11` directory. If this is incorrect for your machine, make the necessary changes to the *Makefile* and try the *make* again.
- Undefined references to math library functions, such as the following:

  ```
  /tmp/cca011551.o(.text+0x11): undefined reference to `cos'
  ```

  The fix for this is to explicitly link in the `math library`, by adding an **-lm** to the *LIB* or *LIBS* flags in the `Makefile` (see previous example).
- Yet another thing to try if *xmkmf* fails is the following script:

  ```
  make -DUseInstalled -I/usr/X386/lib/X11/config
  ```

  This is a sort of bare bones equivalent of *xmkmf*.
- In a very few cases, running *ldconfig* as *root* may be the solution:

  # ldconfig updates the shared library symbolic links. *This may not be necessary* .
- Some `Makefiles` use unrecognized aliases for libraries present in your system. For example, the build may require `libX11.so.6`, but there exists no such file or link in `/usr/X11R6/lib`. Yet, there is a `libX11.so.6.1`. The solution is to do a **ln -s /usr/X11R6/lib/libX11.so.6.1 /usr/X11R6/lib/libX11.so.6**, as root. This may need to be followed by a **ldconfig**.
- Sometimes the source needs the older release X11R5 libraries to build. If you have the R5 libs in /usr/X11R6/lib (you were given the option of having them when first installing Linux), then you need only ensure that you have the links that the software needs to build. The `R5 libs` are named `libX11.so.3.1.0`, `libXaw.so.3.1.0`, and `libXt.so.3.1.0`. You generally need links, such as *libX11.so.3 -> libX11.so.3.1.0*. Possibly the software will also need a link of the form *libX11.so -> libX11.so.3.1.0*. Of course, to create a "missing" link, use the command **ln -s libX11.so.3.1.0 libX11.so**, *as root*.

- Some packages will require you to install updated versions of one or more libraries. For example, the versions of the *StarOffice* suite from StarDivision GmbH were notorious for needing a `libc` version or greater. Even the more recent *StarOffice* 5.0 will not run after installation with the new `glibc 2.1` libs. Fortunately, the newer *StarOffice* 5.1 solves these problems. If running an older version of *StarOffice* you would, as *root*, need to copy one or more libraries to the appropriate directories, remove the old libraries, then reset the symbolic links (check the latest version of the `StarOffice miniHOWTO` for more information on this). **Caution: Exercise extreme care in this, as you can render your system nonfunctional if you screw up.** You can usually find the latest updated libraries at Sunsite.

7.2 Other Problems

- An installed *Perl* or shell script gives you a `No such file or directory` error message. In this case, check the file permissions to make sure the file is executable and check the file header to ascertain whether the shell or program invoked by the script is in the place specified. For example, the scrip may begin with:

  ```
  #!/usr/local/bin/perl
  ```

  If *Perl* is in fact installed in your `/usr/bin` directory instead of the `/usr/local/bin` one, then the script will not run. There are two methods of correcting this. The script file header may be changed to `#!/usr/bin/perl`, or a symbolic link to the correct directory may be added, **ln -s /usr/bin/perl /usr/local/bin/perl**.
- Some X11 software requires the Motif libraries to build. The standard Linux distributions do not have the Motif libraries installed, and at present Motif costs an extra $100-$200 (though the freeware Lesstif also works in many cases). If you need Motif to build a certain package, but lack the Motif libraries, it may be possible to obtain *statically linked binaries*. Static linking incorporates the library routines in the binaries themselves. This results in much larger binary files, but the code will run on systems lacking the libraries.

  When a package requires libraries not present on your system for the build, it will result in link errors (`undefined reference` errors). The libraries may be expensive proprietary ones or difficult to find for sone other reason. In that case, obtaining a *statically linked* binary either from the author of the package or from a Linux user group may be the easiest to implement fix.
- Running a *configure* script creates a strange Makefile, one seemingly unrelated to the package you are attempting to build. This means the wrong *configure* ran, one found somewhere else in your path. Always invoke *configure* as **./configure** to prevent this.
- Most Linux distributions have changed over to the `libc 6 / glibc 2` libraries from the older `libc 5`. Precompiled binaries that worked with the older library may bomb if you have upgraded your library. The solution is to either recompile the applications from the source or to obtain newer precompiled binaries. If you are in the process of upgrading your system to `libc 6` and are experiencing problems, refer to Eric Green's *Glibc 2 HOWTO*.

  Note that there are some minor incompatibilities between `glibc` versions, so a binary built with `glibc 2.1` may not work with `glibc 2.0`, and vice versa.
- Sometimes it is necessary to remove the *-ansi* option from the compile flags in the `Makefile`. This enables gcc's extra, non-ANSI features, and allows building packages that require these extensions. (Thanks to Sebastien Blondeel for pointing this out.)

- Some programs require having *setuid root*, in order to run with *root privileges*. The command to implement this is **chmod u+s filename**, *as root* (note that the program must already be owned by root). This has the effect of setting the *setuid* bit in the file permissions. This issue comes up when the program accesses the system hardware, such as a modem or CD ROM drive, or when the SVGA libs are invoked from console mode, as in one particularly notorious emulation package. If a program works when run by root, but gives *access denied* error messages to an ordinary user, suspect this as the cause.

  Warning: A program with *setuid* as root may pose a security risk to your system. The program runs with root privileges and thus has the potential for doing significant damage. Make certain that you know what the program does, by looking at the source if possible, before setting the *setuid* bit.

7.3 Tweaking and fine tuning

You may wish to examine the `Makefile` to make certain that the best compilation options for your system are invoked. For example, setting the *-O2* flag chooses the highest level of optimization and the *-fomit-frame-pointer* flag results in a smaller binary (though debugging will then be disabled). **Do not play around with this unless you know what you are doing, and in any case, not until after a trial *build* works.**

7.4 Where to go for more help

In my experience, perhaps 25% of applications build "right out of the box". Another 50% or so can be "persuaded" to build with an effort ranging from trivial to herculean. That still means a significant number of packages will not build no matter what. Even then, the Intel `ELF` and/or `a.out` binaries for these might possibly be found at Sunsite or the TSX-11 archive. Red Hat and Debian have extensive archives of prepackaged binaries of most of the popular Linux software. Perhaps the author of the software can supply the binaries compiled for your particular flavor of machine.

`Note that if you obtain precompiled binaries, you will need to check for compatibility with your system:`

- `The binaries must run on your hardware (i.e., Intel x86).`
- `The binaries must be compatible with your kernel (i.e., a.out or ELF).`
- `Your libraries must be up to date.`
- `Your system must have the appropriate installation utility (rpm or deb).`

If all else fails, you may find help in the appropriate newsgroups, such as comp.os.linux.x or comp.os.linux.development.

If nothing at all works, at least you gave it your best effort, and you learned a lot.

---

8. Final Steps

Read the software package documentation to determine whether certain environmental variables need setting (in `.bashrc` or `.cshrc`) and if the `.Xdefaults` and `.Xresources` files need customizing.

There may be an applications default file, usually named `Xfoo.ad` in the original Xfoo distribution. If so, edit the Xfoo.ad file to customize it for your machine, then rename (**mv**) it Xfoo and install it in the `/usr/lib/X11/app-defaults` directory, *as root*. Failure to do this may cause the software to behave strangely or even refuse to run.

Most software packages come with one or more preformatted man pages. *As root*, copy the Xfoo.man file to the appropriate `/usr/man`, `/usr/local/man`, or `/usr/X11R6/man` directory (`man1` - `man9`), and rename it accordingly. For example, if Xfoo.man ends up in /usr/man/man4, it should be renamed Xfoo.4 (mv Xfoo.man Xfoo.4). By convention, user commands go in `man1`, games in `man6`, and administration packages in `man8` (see the *man docs* for more details). Of course, you may deviate from this on your own system, if you like.

A few packages will not install the binaries in the appropriate system directories, that is, they are missing the *install* option in the `Makefile`. Should this be the case, you can install the binaries manually by copying the binaries to the appropriate system directory, `/usr/bin`, `/usr/local/bin` or `/usr/X11R6/bin`, *as root*, of course. Note that `/usr/local/bin` is the preferred directory for binaries that are not part of the Linux distribution's base install.

Some or all of the above procedures should, in most cases, be handled automatically by a **make install**, and possibly a **make install.man** or **make install_man**. If so, the `README` or `INSTALL` doc file will specify this.

9. First Example: Xscrabble

Matt Chapman's `Xscrabble` seemed like a program that would be interesting to have, since I happen to be an avid Scrabble`TM` player. I downloaded it, uncompressed it, and built it following the procedure in the README file:

```
xmkmf
make Makefiles
make includes
make
```

Of course it did not work...

```
gcc -o xscrab -O2 -O -L/usr/X11R6/lib
init.o xinit.o misc.o moves.o cmove.o main.o xutils.o mess.o popup.o
widgets.o display.o user.o CircPerc.o
-lXaw -lXmu -lXExExt -lXext -lX11 -lXt -lSM -lICE -lXExExt -lXext -lX11
-lXpm -L../Xc -lXc

BarGraf.o(.text+0xe7): undefined reference to `XtAddConverter'
BarGraf.o(.text+0x29a): undefined reference to `XSetClipMask'
BarGraf.o(.text+0x2ff): undefined reference to `XSetClipRectangles'
BarGraf.o(.text+0x375): undefined reference to `XDrawString'
BarGraf.o(.text+0x3e7): undefined reference to `XDrawLine'
etc.
etc.
etc...
```

I enquired about this in the comp.os.linux.x newsgroup, and someone kindly pointed out that apparently the Xt, Xaw, Xmu, and X11 libs were not being found at the link stage. Hmmm...

There were two main Makefiles, and the one in the `src` directory caught my interest. One line in the Makefile defined LOCAL_LIBS as: LOCAL_LIBS = $(XAWLIB) $(XMULIB) $(XTOOLLIB) $(XLIB) Here were references to the libs not being found by the linker.

Looking for the next reference to LOCAL_LIBS, I saw on line 495 of that Makefile:

```
      $(CCLINK) -o $@ $(LDOPTIONS) $(OBJS) $(LOCAL_LIBS) $(LDLIBS)
$(EXTRA_LOAD_FLAGS)
```

Now what were these LDLIBS?

```
      LDLIBS = $(LDPOSTLIB) $(THREADS_LIBS) $(SYS_LIBRARIES)
$(EXTRA_LIBRARIES)
```

The SYS_LIBRARIES were:

```
SYS_LIBRARIES = -lXpm -L../Xc -lXc
```

Yes! Here were the missing libraries.

Possibly the linker needed to see the LDLIBS before the LOCAL_LIBS... So, the first thing to try was to modify the Makefile by transposing the $(LOCAL_LIBS) and $(LDLIBS) on line 495, so it would now read:

```
        $(CCLINK) -o $@ $(LDOPTIONS) $(OBJS) $(LDLIBS) $(LOCAL_LIBS)
$(EXTRA_LOAD_FLAGS)                                   ^^^^^^^^^^^^^^^^^^^^^^^^
```

I tried running *make* again with the above change, and lo and behold, it worked this time. Of course, Xscrabble still needed some fine tuning and twiddling, such as renaming the dictionary and commenting out some assert statements in one of the source files, but since then it has provided me with many hours of pleasure.

[Note that a newer version of Xscrabble is now available in rpm format, and this installs without problems.]

You may e-mail Matt Chapman, and download *Xscrabble* from his home page.

```
Scrabble is a registered trademark of the Milton Bradley Co., Inc.
```

---

10. Second Example: Xloadimage

This example poses an easier problem. The *xloadimage* program seemed a useful addition to my set of graphic tools. I copied the `xloadi41.gz` file directly from the source directory on the CD included with the excellent X User Tools book, by Mui and Quercia. As expected, *tar xzvf* unarchives the files. The *make*, however, produces a nasty-looking error and terminates.

```
gcc -c -O -fstrength-reduce -finline-functions -fforce-mem
-fforce-addr -DSYSV  -I/usr/X11R6/include
-DSYSPATHFILE=\"/usr/lib/X11/Xloadimage\" mcidas.c

In file included from /usr/include/stdlib.h:32,
                 from image.h:23,
                 from xloadimage.h:15,
                 from mcidas.c:7:
/usr/lib/gcc-lib/i486-linux/2.6.3/include/stddef.h:215:
conflicting types for `wchar_t'
/usr/X11R6/include/X11/Xlib.h:74: previous declaration of
`wchar_t'
make[1]: *** [mcidas.o] Error 1
make[1]: Leaving directory
`/home/thegrendel/tst/xloadimage.4.1'
make: *** [default] Error 2
```

The error message contains the essential clue.

Looking at the file `image.h`, line 23...

```
        #include <stdlib.h>
```

Aha, somewhere in the source for *xloadimage*, *wchar_t* has been redefined from what was specified in the standard include file, `stdlib.h`. Let us first try commenting out line 23 in `image.h`, as perhaps the *stdlib.h include* is not, after all, necessary.

At this point, the build proceeds without any fatal errors. The *xloadimage* package functions correctly now.

11. Third Example: Fortune

This example requires some knowledge of C programming. The majority of UNIX/Linux software is written in C, and learning at least a little bit of C would certainly be an asset for anyone serious about software installation.

The notorious *fortune* program displays up a humorous saying, a "fortune cookie", every time Linux boots up. Unfortunately (pun intended), attempting to build fortune on a Red Hat distribution with a 2.0.30 kernel generates fatal errors.

```
~/fortune# make all


gcc -O2 -Wall -fomit-frame-pointer -pipe   -c fortune.c -o
fortune.o
fortune.c: In function `add_dir':
fortune.c:551: structure has no member named `d_namlen'
fortune.c:553: structure has no member named `d_namlen'
make[1]: *** [fortune.o] Error 1
make[1]: Leaving directory `/home/thegrendel/for/fortune/fortune'
make: *** [fortune-bin] Error 2
```

Looking at `fortune.c`, the pertinent lines are these.

```
   if (dirent->d_namlen == 0)
            continue;
        name = copy(dirent->d_name, dirent->d_namlen);
```

We need to find the structure `dirent`, but it is not declared in the *fortune.c* file, nor does a **grep dirent** show it in any of the other source files. However, at the top of *fortune.c*, there is the following line.

```
#include <dirent.h>
```

This appears to be a system library include file, therefore, the logical place to look for *dirent.h* is in */usr/include*. Indeed, there does exist a *dirent.h* file in */usr/include*, but that file does not contain the declaration of the `dirent` structure. There is, however, a reference to another *dirent.h* file.

```
#include <linux/dirent.h>
```

At last, going to */usr/include/linux/dirent.h*, we find the structure declaration we need.

```
struct dirent {
```

```
        long            d_ino;
        __kernel_off_t  d_off;
        unsigned short  d_reclen;
        char            d_name[256]; /* We must not include
limits.h! */
};
```

Sure enough, the structure declaration contains no *d_namelen*, but there are a couple of "candidates" for its equivalent. The most likely of these is *d_reclen*, since this structure member probably represents the length of something and it is a short integer. The other possibility, *d_ino*, could be an inode number, judging by its name and type. As a matter of fact, we are probably dealing with a "directory entry" structure, and these elements represent attributes of a file, its name, inode, and length (in blocks). This would seem to validate our guess.

Let us edit the file `fortune.c`, and change the two *d_namelen* references in lines 551 and 553 to *d_reclen*. Try a *make all* again. **Success.** It builds without errors. We can now get our "cheap thrills" from fortune.

---

12. Fourth Example: Hearts

Here is the hoary old game of Hearts, written for UNIX systems by Bob Ankeney sometime in the '80's, revised in 1992 by Mike Yang, and currently maintained by Jonathan Badger. Its predecessor was an even older Pascal program by Don Backus of Oregon Software, later updated by Jeff Hemmerling. Originally intended as a multiplayer client, it also works well in single-player mode against computer opponents. The graphics are nice, though the game lacks sophisticated features and the computer players are not particularly strong. All the same, it seems to be the only decent Hearts game available for UNIX and Linux machines even at this late date.

Due to its age and lineage, this package is particularly difficult to build on a Linux system. It requires solving a long and perplexing series of puzzles. It is an exercise in patience and determination.

Before beginning, make certain that you have either the `motif` *or* `lesstif` *libraries installed.*

-

xmkmf

make

```
client.c: In function `read_card':
client.c:430: `_tty' undeclared (first use in this function)
client.c:430: (Each undeclared identifier is reported only once
client.c:430: for each function it appears in.)
client.c: In function `scan':
client.c:685: `_tty' undeclared (first use in this function)
make: *** [client.o] Error 1
```

These are the culprits in the file `client.c`:

```
#ifndef SYSV
        (buf[2] != _tty.sg_erase) && (buf[2] != _tty.sg_kill)) {
 #else
        (buf[2] != CERASE) && (buf[2] != CKILL)) {
#endif
```

-

In `client.c`, add

```
#define SYSV
```

at line 39. This will bypass the reference to *_tty*.

make

```
client.c:41: sys/termio.h: No such file or directory
make: *** [client.o] Error 1
```

-

The include file `termio.h` is in the `/usr/include` directory on a Linux system, rather than the `/usr/include/sys` one, as was the case on older UNIX machines. Therefore, change line 41 of `client.c` from

```
#include <sys/termio.h>
```

to

```
#include <termio.h>
```

make

```
gcc -o hearts -g      -L/usr/X11R6/lib client.o hearts.o select.o connect.o
sockio.o start_dist.o  -lcurses -ltermlib
/usr/bin/ld: cannot open -ltermlib: No such file or directory
collect2: ld returned 1 exit status
make: *** [hearts] Error 1
```

-

Modern Linux distributions use the *terminfo* and/or *termcap* database, rather than the obsolete *termlib* one.

Edit the `Makefile`.

Line 655:

```
CURSES_LIBRARIES = -lcurses -ltermlib
```

changes to:

```
CURSES_LIBRARIES = -lcurses -ltermcap
```

make

```
gcc -o xmhearts -g      -L/usr/X11R6/lib xmclient.o hearts.o select.o
connect.o sockio.o start_dist.o gfx.o  -lXm_s -lXt -lSM -lICE -lXext -lX11
-lPW
/usr/bin/ld: cannot open -lXm_s: No such file or directory
collect2: ld returned 1 exit status
```

-

The main *lesstif* library is `libXm`, rather than `libXm_s`. Therefore, edit the `Makefile`.

In line 653:

```
XMLIB = -lXm_s $(XTOOLLIB) $(XLIB) -lPW
```

changes to:

```
XMLIB = -lXm $(XTOOLLIB) $(XLIB) -lPW
```

make

```
gcc -o xmhearts -g      -L/usr/X11R6/lib xmclient.o hearts.o select.o
connect.o sockio.o start_dist.o gfx.o  -lXm -lXt -lSM -lICE -lXext -lX11 -lPW
/usr/bin/ld: cannot open -lPW: No such file or directory
collect2: ld returned 1 exit status
make: *** [xmhearts] Error 1
```

-

Round up the usual suspects.

There is no `PW` library. Edit the `Makefile`.

Line 653:

```
XMLIB = -lXm $(XTOOLLIB) $(XLIB) -lPW
```

changes to:

```
XMLIB = -lXm $(XTOOLLIB) $(XLIB) -lPEX5
```

(The PEX5 lib comes closest to PW.)

make

```
rm -f xmhearts
gcc -o xmhearts -g      -L/usr/X11R6/lib xmclient.o hearts.o select.o
connect.o sockio.o start_dist.o gfx.o  -lXm -lXt -lSM -lICE -lXext -lX11 -lPEX5
```

The make finally works (hurray!).

-

Installation:

As root,

```
[root@localhost hearts]# make install
install -c -s  hearts /usr/X11R6/bin/hearts
install -c -s  xmhearts /usr/X11R6/bin/xmhearts
install -c -s  xawhearts /usr/X11R6/bin/xawhearts
install in . done
```

-

Test run:

rehash

(We're running the tcsh shell.)

xmhearts

```
localhost:~/% xmhearts
Can't invoke distributor!
```

-

From README file in the hearts package:

```
     Put heartsd, hearts_dist, and hearts.instr in the HEARTSLIB
     directory defined in local.h and make them world-accessible.
```

From the file local.h:

```
/* where the distributor, dealer and instructions live */

#define HEARTSLIB "/usr/local/lib/hearts"
```

This is a classic case of RTFM.

As *root*,

cd /usr/local/lib

mkdir hearts

cd !$

Copy the distributor files to this directory.

cp /home/username/hearts/heartsd .

cp /home/username/hearts/hearts_dist .

cp /home/username/hearts/hearts.instr .

-

Try another test run.

xmhearts

It works some of the time, but more often than not crashes with a `dealer died!` message.

-

The "distributor" and "dealer" scan the hardware ports. We should thus suspect that those programs need root user privileges.

Try, as *root*,

chmod u+s /usr/local/lib/heartsd

chmod u+s /usr/local/lib/hearts_dist

(Note that, as previously discussed, *suid* binaries may create security holes.)

xmhearts

It finally works!

Hearts is available from Sunsite.

---

13. Fifth Example: XmDipmon

```
Bullwinkle: Hey Rocky, watch me pull a rabbit out of my hat.
Rocky:      But that trick never works.
Bullwinkle: This time for sure.
            Presto!
            Well, I'm gettin' close.
Rocky:      And now it's time for another special feature.
            --- "Rocky and His Friends"
```

XmDipmon is a nifty little application that displays a button showing the status of an Internet connection. It flashes and beeps when the connection is broken, as is all too often the case in on rural telephone systems. Unfortunately, XmDipmon works only with *dip*, making it useless for those people, the majority, who use *chat* to connect.

Building XmDipmon is not a problem. XmDipmon links to the *Motif* libraries, but it builds and works fine with *Lesstif*. The challenge is to alter the package to work when using *chat*. This involves actually tinkering with the source code, and necessarily requires some programming knowledge.

```
"When xmdipmon starts up, it checks for a file called /etc/dip.pid
 (you can let it look at another file by using the -pidfile
 command line option).  This file contains the PID of the dip
 deamon (dip switches itself into deamon mode once it has
 established a connection)."
                --- from the XmDipmon README file
```

Using the *-pidfile* option, the program can be directed to check for a different file upon startup, one that exists only during a successful *chat* login. The obvious candidate is the modem lock file. We could therefore try invoking the program with **xmdipmon -pidfile /var/lock/LCK..ttyS3** (this assumes that the modem is on com port #4, ttyS3). This only solves part of the problem, however. The program continually monitors the *dip daemon*, and we need to change this so it instead polls a process associated with *chat* or *ppp*.

There is only a single source file, and fortunately it is well-commented. Scanning the `xmdipmon.c` file, we find the *getProcFile* function, whose header description reads as follows.

```
/*****
* Name:                 getProcFile
* Return Type:  Boolean
* Description:  tries to open the /proc entry as read from the dip pid file.
<snip>
*****/
```

We are hot on the trail now. Tracing into the body of the function...

```
/* we watch the status of the real dip deamon */
sprintf(buf, "/proc/%i/status", pid);
procfile = (String)XtMalloc(strlen(buf)*sizeof(char)+1);
strcpy(procfile, buf);
procfile[strlen(buf)] = '\0';
```

The culprit is line 2383:

```
sprintf(buf, "/proc/%i/status", pid);
                ^^^^^^^^^^^^^^^^^^^^
```

This checks whether the dip daemon process is running . So, how can we change this to monitor the pppd daemon instead?

Looking at the *pppd* manpage:

```
FILES
       /var/run/pppn.pid (BSD or Linux), /etc/ppp/pppn.pid (others)
                     Process-ID for pppd process on ppp interface unit n.
```

Change line 2383 in `xmdipmon.c` to:

```
sprintf(buf, "/var/run/ppp0.pid" );
```

Rebuild the revised package. No problems with the build. Now test it with the new command line argument. It works like a charm. The little blue button indicates when a `ppp` connection to the ISP has been established, and flashes and beeps when the connection is broken. Now we have a fully functional *chat* monitor.

XmDipmon can be downloaded from Ripley Linux Tools.

---

14. Where to Find Source Archives

Now that you are eager to use your newly acquired knowledge to add utilities and other goodies to your system, you may find them online at the Linux Applications and Utilities Page, or on one of the very reasonably priced CD ROM archives by Red Hat, InfoMagic, Linux Systems Labs, Cheap Bytes, and others.

A comprehensive repository of source code is the comp sources UNIX archive.

Much UNIX source code is posted on the alt.sources newsgroup. If you are looking for particular source code packages, you may post on the related alt.sources.wanted newsgroup. Another good place to check is the comp.os.linux.announce newsgroup. To get on the Unix sources mailing list, send a *subscribe* message there.

Archives for the alt.sources newsgroup are at the following ftp sites:

- ftp.sterling.com/usenet/alt.sources/
- wuarchive.wustl.edu/usenet/alt.sources/articles
- src.doc.ic.ac.uk/usenet/alt.sources/articles

15. Final Words

To sum up, persistence makes all the difference (and a high frustration threshold certainly helps). As in all endeavors, learning from mistakes is critically important. Each misstep, every failure contributes to the body of knowledge that will lead to mastery of **the art of building software**.

16. References and Further Reading

BORLAND C++ TOOLS AND UTILITIES GUIDE, Borland International, 1992, pp. 9-42.
[One of the manuals distributed with Borland C++, ver. 3.1. Gives a fairly good intro to make syntax and concepts, using Borland's crippled implementation for DOS.]

DuBois, Paul: SOFTWARE PORTABILITY WITH IMAKE, O'Reilly and Associates, 1996, ISBN 1-56592-226-3.
[This is reputed to be the definitive imake reference, though I did not have it available when writing this article.]

Frisch, Aeleen: ESSENTIAL SYSTEM ADMINISTRATION (2nd ed.), O'Reilly and Associates, 1995, ISBN 1-56592-127-5.
[This otherwise excellent sys admin handbook has only sketchy coverage of software building.]

Hekman, Jessica: LINUX IN A NUTSHELL, O'Reilly and Associates, 1997, ISBN 1-56592-167-4.
[Good all-around reference to Linux commands.]

Lehey, Greg: PORTING UNIX SOFTWARE, O'Reilly and Associates, 1995, ISBN 1-56592-126-7.

Mayer, Herbert G.: ADVANCED C PROGRAMMING ON THE IBM PC, Windcrest Books, 1989, ISBN 0-8306-9363-7.
[An idea-filled book for the intermediate to advanced C programmer. Superb coverage of algorithms, quirks of the language, and even amusements. Unfortunately, out of print.]

Mui, Linda and Valerie Quercia: X USER TOOLS, O'Reilly and Associates, 1994, ISBN 1-56592-019-8, pp. 734-760.

Oram, Andrew and Steve Talbott: MANAGING PROJECTS WITH MAKE, O'Reilly and Associates, 1991, ISBN 0-937175-90-0.

Peek, Jerry and Tim O'Reilly and Mike Loukides: UNIX POWER TOOLS, O'Reilly and Associates / Random House, 1997, ISBN 1-56592-260-3.
[A wonderful source of ideas, and tons of utilities you may end up building from the source code, using the methods discussed in this article.]

Stallman, Richard M. and Roland McGrath: GNU MAKE, Free Software Foundation, 1995, ISBN 1-882114-78-7.
[Required reading.]

Waite, Mitchell, Stephen Prata, and Donald Martin: C PRIMER PLUS, Waite Group

Press, ISBN 0-672-22090-3,.
[Probably the best of the introductions to C programming. Extensive coverage for a primer. Newer editions now available.]

Welsh, Matt and Lar Kaufman: RUNNING LINUX, O'Reilly and Associates, 1996, ISBN 1-56592-151-8.
[Still the best overall Linux reference, though lacking in depth in some areas.]

The man pages for dpkg, gcc, gzip, imake, ldconfig, ldd, make, nm, patch, rpm, shar, strip, tar, termcap, terminfo, and xmkmf.

The BZIP2 HOWTO, by David Fetter.

The Glibc2 HOWTO, by Eric Green

The LINUX ELF HOWTO, by Daniel Barlow.

The RPM HOWTO, by Donnie Barnes.

The StarOffice miniHOWTO, by Matthew Borowski.

[These HOWTOs should be in the `/usr/doc/HOWTO` or `/usr/doc/HOWTO/mini` directory on your system. Updated versions are available in text, HTML, and SGML format from the LDP site, and usually from the respective authors' home sites.]

---

17. Credits

The author of this HOWTO would like to thank the following persons for their helpful suggestions, corrections, and encouragement.

- R. Brock Lynn
- Michael Jenner
- Fabrizio Stefani

Kudos also go to the fine people who have translated this HOWTO into Italian and Japanese.

And, of course, thanks, praise, benedictions and hosannahs to Greg Hankins and Tim Bynum of the Linux Documentation Project, which has made all this possible.

Software Release Practice HOWTO

Eric S. Raymond <esr@thyrsus.com>

1.2, 18 June 1999

This HOWTO describes good release practices for Linux open-source projects. By following these practices, you will make it as easy as possible for users to build your code and use it, and for other developers to understand your code and cooperate with you to improve it. This document is a must-read for novice developers. Experienced developers should review it when they are about to release a new project. It will be revised periodically to reflect the evolution of good-practice standards.

1. Introduction

- 1.1 Why this document?
- 1.2 New versions of this document

2. Good project- and archive- naming practice

- 2.1 Use GNU-style names with a stem and major.minor.patch numbering.
- 2.2 But respect local conventions where appropriate
- 2.3 Try hard to choose a name prefix that is unique and easy to type

3. Good development practice

- 3.1 Write either pure ANSI C or a portable scripting language
- 3.2 Follow good C portability practices
- 3.3 Use autoconf/automake/autoheader
- 3.4 Sanity-check your code before release

TOC

4. Good distribution-making practice

- 4.1 Make sure tarballs always unpack into a single new directory
- 4.2 Have a README
- 4.3 Respect and follow standard file naming practices

5. Good communication practice

- 5.1 Announce to c.o.l.a
- 5.2 Announce to a relevant topic newsgroup
- 5.3 Have a website
- 5.4 Host project mailing lists
- 5.5 Release to major archives
- 5.6 Provide RPMs

---

1. Introduction

1.1 Why this document?

There is a large body of good-practice traditions for open-source code that helps other people port, use, and cooperate with developing it. Some of these conventions are traditional in the Unix world and predate Linux; others have developed recently in response to particular new tools and technologies such as the World Wide Web.

This document will help you learn good practice. It is organized into topic sections, each containing a series of checklist items. Think of these as a pre-flight checklist for your distribution.

1.2 New versions of this document

This document will be posted monthly to the newsgroups `comp.os.linux.answers`. The document is archived on a number of Linux FTP sites, including `metalab.unc.edu` in `pub/Linux/docs/HOWTO`.

You can also view the latest version of this HOWTO on the World Wide Web via the URL http://metalab.unc.edu/LDP/HOWTO/Software-Release-Practice.html.

Feel free to mail any questions or comments about this HOWTO to Eric S. Raymond, esr@snark.thyrsus.com.

2. Good project- and archive- naming practice

As the load on maintainers of archives like Metalab, the PSA site and CPAN increases, there is an increasing trend for submissions to be processed partly or wholly by programs (rather than entirely by a human).

This makes it more important for project and archive-file names to fit regular patterns that computer programs can parse and understand.

2.1 Use GNU-style names with a stem and major.minor.patch numbering.

It's helpful to everybody if your archive files all have GNU-like names -- all-lower-case alphanumeric stem prefix, followed by a dash, followed by a version number, extension, and other suffixes.

Let's suppose you have a project you call `foobar' at version 1, release 2, level 3. If it's got just one archive part (presumably the sources), here's what its names should look

foobar-1.2.3.tar.gz

The source archive

foobar.lsm

The LSM file (assuming you're submitting to Metalab).

Please *don't* use these:

foobar123.tar.gz

This looks to many programs like an archive for a project called`foobar123' with no version number.

foobar1.2.3.tar.gz

This looks to many programs like an archive for a project called `foobar1' at version 2.3.

foobar-v1.2.3.tar.gz

Many programs think this goes with a project called `foobar-v1'.

foo_bar-1.2.3.tar.gz

The underscore is hard for people to speak, type, and remember

FooBar-1.2.3.tar.gz

Unless you *like* looking like a marketing weenie. This is also hard for people to speak, type, and remember.

TOC

If you have to differentiate between source and binary archives, or between different kinds of binary, express some kind of build option in the file name, please treat that as a file extension to go *after* the version number. That is, please do this:

foobar-1.2.3.src.tar.gz

sources

foobar-1.2.3.bin.tar.gz

binaries, type not specified

foobar-1.2.3.bin.ELF.tar.gz

ELF binaries

foobar-1.2.3.bin.ELF.static.tar.gz

ELF binaries statically linked

foobar-1.2.3.bin.SPARC.tar.gz

SPARC binaries

Please *don't* use names like `foobar-ELF-1.2.3.tar.gz', because programs have a hard time telling type infixes (like `-ELF') from the stem.

A good general form of name has these parts in order:

1. project prefix
2. dash
3. version number
4. dot
5. "src" or "bin" (optional)
6. dot or dash (dot preferred)
7. binary type and options (optional)
8. archiving and compression extensions

2.2 But respect local conventions where appropriate

Some projects and communities have well-defined conventions for names and version numbers that aren't necessarily compatible with the above advice. For instance, Apache modules are generally named like mod_foo, and have both their own version number and the version of Apache with which they work. Likewise, Perl modules have version numbers that can be treated as floating point numbers (e.g., you might see 1.303 rather than 1.3.3), and the distributions are generally named Foo-Bar-1.303.tar.gz for version 1.303 of module Foo::Bar.

Look for and respect the conventions of specialized communities and developers; for general use, follow the above guidelines.

2.3 Try hard to choose a name prefix that is unique and easy to type

The stem prefix should be common to all a project's files, and it should be easy to read, type, and remember. So please don't use underscores. And don't capitalize or BiCapitalize without extremely good reason -- it messes up the natural human-eyeball search order and looks like some marketing weenie trying to be clever.

It confuses people when two different projects have the same stem name. So try to check for collisions before your first release. A good place to check is the index file of Metalab.

---

3. Good development practice

Most of these are concerned with ensuring portability, not only across Linuxes but to other Unixes as well. Being portable to other Unixes is not just a worthy form of professionalism and hackerly politeness, it's valuable insurance against future changes in Linux itself.

Finally, other people *will* try to build your code on non-Linux systems; portability minimizes the number of annoying perplexed email messages you will get.

3.1 Write either pure ANSI C or a portable scripting language

For portability and stability, you should write either in ANSI C or a scripting language that is guaranteed portable because it has just one cross-platform implementation.

Scripting languages that qualify include Python, Perl, Tcl, and Emacs Lisp. Plain old shell does *not* qualify; there are too many different implementations with subtle idiosyncracies, and the shell environment is subject to disruption by user customizations such as shell aliases.

Java holds promise as a portable language, but the Linux-available implementations are still scratchy and poorly integrated with Linux. Java is still a bleeding-edge choice, though one likely to become more popular as it matures.

3.2 Follow good C portability practices

If you are writing C, do feel free to use the full ANSI features -- including function prototypes, which will help you spot cross-module inconsistancies. The old-style K&R compilers are history.

On the other hand, do *not* assume that GCC-specific features such as the `-pipe' option or nested functions are available. These will come around and bite you the second somebody ports to a non-Linux, non-GCC system.

3.3 Use autoconf/automake/autoheader

If you're writing C, use autoconf/automake/autoheader to handle portability issues, do system-configuration probes, and tailor your makefiles. People building from sources today expect to be able to type "configure; make" and get a clean build -- and rightly so.

3.4 Sanity-check your code before release

TOC

If you're writing C, test-compile with -Wall and clean up the errors at least once before each release. This catches a surprising number of errors. For real thoroughness, compile with -pedantic as well.

If you're writing Perl, check your code with perl -c (and maybe -T, if applicable). Use perl -w and 'use strict' religiously. (See the Perl documentation for discussion.)

---

4. Good distribution-making practice

These guidelines describe how your distribution should look when someone downloads, retrieves and unpacks it.

4.1 Make sure tarballs always unpack into a single new directory

The single most annoying mistake newbie developers make is to build tarballs that unpack the files and directories in the distribution into the current directory, potentially stepping on files already located there. *Never do this!*

Instead, make sure your archive files all have a common directory part named after the project, so they will unpack into a single top-level directory directly *beneath* the current one.

Here's a makefile trick that, assuming your distribution directory is named `foobar' and SRC contains a list of your distribution files, accomplishes this. It requires GNU tar 1.13

```
VERS=1.0
foobar-$(VERS).tar.gz:
        tar --name-prefix='foobar-$(VERS)/' -czf foobar-$(VERS).tar.gz $(SRC)
```

If you have an older tar program, do something like this:

```
foobar-$(VERS).tar.gz:
        @ls $(SRC) | sed s:^:foobar-$(VERS)/: >MANIFEST
        @(cd ..; ln -s foobar foobar-$(VERS))
        (cd ..; tar -czvf foobar/foobar-$(VERS).tar.gz `cat foobar/MANIFEST`)
        @(cd ..; rm foobar-$(VERS))
```

4.2 Have a README

Have a file called README or READ.ME that is a roadmap of your source distribution. By ancient convention, this is the first file intrepid explorers will read after unpacking the source.

Good things to have in the README include:

- A brief description of the project.
- A pointer to the project website (if it has one)
- Notes on the developer's build environment and potential portability problems.
- A roadmap describing important files and subdirectories.
- Either build/installation instructions or a pointer to a file containing same (usually INSTALL).
- Either a maintainers/credits list or a pointer to a file containing same (usually CREDITS).
- Either recent project news or a pointer to a file containing same (usually NEWS).

4.3 Respect and follow standard file naming practices

TOC

Before even looking at the README, your intrepid explorer will have scanned the filenames in the top-level directory of your unpacked distribution. Those names can themselves convey information. By adhering to certain standard naming practices, you can give the explorer valuable clues about what to look in next.

Here are some standard top-level file names and what they mean. Not every distribution needs all of these.

README or READ.ME

the roadmap file, to be read first

INSTALL

configuration, build, and installation instructions

CREDITS

list of project contributers

NEWS

recent project news

HISTORY

project history

COPYING

project license terms (GNU convention)

LICENSE

project license terms

MANIFEST

list of files in the distribution

FAQ

plain-text Frequently-Asked-Questions document for the project

TAGS

generated tag file for use by Emacs or vi

Note the overall convention that filenames with all-caps names are human-readable metainformation about the package, rather than build components.

---

5. Good communication practice

Your software won't do the world much good if nobody but you knows it exists. Also, developing a visible presence for the project on the Internet will assist you in recruiting users and co-developers. Here are the standard ways to do that.

5.1 Announce to c.o.l.a

Announce new releases to comp.os.linux.announce. Besides being widely read itself, this group is a major feeder for web-based what's-new sites like Freshmeat.

5.2 Announce to a relevant topic newsgroup

Find USENET topics group directly relevant to your application, and announce there as well. Post only where the *function* of the code is relevant, and exercise restraint.

If (for example) you are releasing a program written in Perl that queries IMAP servers, you should certainly post to comp.mail.imap. But you should probably not post to comp.lang.perl unless the program is also an instructive example of cutting-edge Perl techniques.

Your announcement should include the URL of a project website.

5.3 Have a website

If you intend try to build any substantial user or developer community around your project, it should have a website. Standard things to have on the website include:

- The project charter (why it exists, who the audience is, etc).
- Download links for the project sources.
- Instructions on how to join the project mailing list(s).
- A FAQ (Frequently Asked Questions) list.
- HTMLized versions of the project documentation
- Links to related and/or competing projects.

Some project sites even have URLs for anonymous access to the master source tree.

TOC

5.4 Host project mailing lists

It's standard practice to have a private development list through which project collaborators can communicate and exchange patches. You may also want to have an announcements list for people who want to be kept informed of the project's process

5.5 Release to major archives

For the last several years, the Metalab archive has been the most important interchange location for Linux software.

Other important locations include:

- the Python Software Activity site (for software written in Python).
- the CPAN, the Comprehensive Perl Archive Network, (for software written in Perl).

5.6 Provide RPMs

The de-facto standard format for installable binary packages is that used by the Red Hat Package manager, RPM. It's featured in the most popular Linux distribution, and supported by effectively all other Linux distributions (except Debian a Slackware).

Accordingly, it's a good idea for your project site to provide installable RPMs as well as source tarballs.

The Linux Sound HOWTO

Jeff Tranter, tranter@pobox.com

v1.20, 24 March 1999

This document describes sound support for Linux. It lists the supported sound hardware, describes how to configure the kernel drivers, and answers frequently asked questions. The intent is to bring new users up to speed more quickly and reduce the amount of traffic in the Usenet news groups and mailing lists.

1. Introduction

- 1.1 Acknowledgments
- 1.2 New versions of this document
- 1.3 Feedback
- 1.4 Distribution Policy

2. Sound Card Technology

3. Supported Hardware

- 3.1 Sound Cards
- 3.2 Alternate Sound Drivers
- 3.3 PC Speaker
- 3.4 Parallel Port

4. Installation

- 4.1 Installing the Sound Card
- 4.2 Configuring Plug and Play
- 4.3 Configuring the Kernel
- 4.4 Creating the Device Files

- 4.5 Booting Linux and Testing the Installation
- 4.6 Troubleshooting

5. Applications Supporting Sound

6. Answers To Frequently Asked Questions

- 6.1 What are the various sound device files?
- 6.2 How can I play a sound sample?
- 6.3 How can I record a sample?
- 6.4 Can I have more than one sound card?
- 6.5 Error: No such file or directory for sound devices
- 6.6 Error: No such device for sound devices
- 6.7 Error: No space left on device for sound devices
- 6.8 Error: Device busy for sound devices
- 6.9 I still get device busy errors!
- 6.10 Partial playback of digitized sound file
- 6.11 There are pauses when playing MOD files
- 6.12 Compile errors when compiling sound applications
- 6.13 SEGV when running sound binaries that worked previously
- 6.14 What known bugs or limitations are there in the sound driver?
- 6.15 Where are the sound driver ioctls() etc. documented?
- 6.16 What CPU resources are needed to play or record without pauses?
- 6.17 Problems with a PAS16 and an Adaptec 1542 SCSI host adaptor
- 6.18 Is it possible to read and write samples simultaneously?
- 6.19 My SB16 is set to IRQ 2, but configure does not allow this value.
- 6.20 If I run Linux, then boot DOS, I get errors and/or sound applications do not work properly.
- 6.21 Problems running DOOM under Linux
- 6.22 How can I reduce noise picked up by my sound card?
- 6.23 I can play sounds, but not record.
- 6.24 My "compatible" sound card only works if I first initialize under MS-DOS.
- 6.25 My 16-bit SoundBlaster "compatible" sound card only works in 8-bit mode under Linux.
- 6.26 Where can I find sound applications for Linux?
- 6.27 Can the sound driver be compiled as a loadable module?

- 6.28 Can I use a sound card to replace the system console beep?
- 6.29 What is VoxWare?
- 6.30 Sox/Play/Vplay reports "invalid block size 1024"
- 6.31 The mixer settings are reset whenever I load the sound driver module
- 6.32 Only user root can record sound
- 6.33 Is the sound hardware on the IBM ThinkPad supported?
- 6.34 Applications fail because my sound card has no mixer
- 6.35 Problems with a SB16 CT4170
- 6.36 How to connect a MIDI keyboard to a soundcard
- 6.37 Problems with IRQ 15 and Ensoniq PCI 128
- 6.38 Where can I get freely available MIDI patches to run SoftOSS?

7. References

1. Introduction

This is the Linux Sound HOWTO. It is intended as a quick reference covering everything you need to know to install and configure sound support under Linux. Frequently asked questions about sound under Linux are answered, and references are given to some other sources of information on a variety of topics related to computer generated sound and music.

The scope is limited to the aspects of sound cards pertaining to Linux. See the other documents listed in the *References* section for more general information on sound cards and computer sound and music generation.

1.1 Acknowledgments

Much of this information came from the documentation provided with the sound driver source code, by Hannu Savolainen (hannu@opensound.com). Thanks go to Hannu, Alan Cox, and the many other people who developed the Linux kernel sound drivers and utilities.

Thanks to the SGML Tools package, this HOWTO is available in several formats, all generated from a common source file.

1.2 New versions of this document

New versions of this document will be periodically posted to the comp.os.linux.answers newsgroup. They will also be uploaded to various anonymous ftp sites that archive such information including ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/.

Hypertext versions of this and other Linux HOWTOs are available on many world-wide web sites, including http://metalab.unc.edu/LDP/. Most Linux CD-ROM distributions include the HOWTOs, often under the /usr/doc directory, and you can also buy printed copies from several vendors. Sometimes the HOWTOs available from CD-ROM vendors, ftp sites, and printed format are out of date. If the date on this HOWTO is more than six months in the past, then a newer copy is probably available on the Internet.

Please note that, due to the dynamic nature of the Internet, all web and ftp links listed in this document are subject to change.

Translations of this document are available in several languages:

Chinese: http://www.linux.org.tw/CLDP/Sound-HOWTO.html

French: http://www.freenix.org/unix/linux/HOWTO/

Japanese: http://yebisu.ics.es.osaka-u.ac.jp/linux/

Korean: http://kldp.linux-kr.org/HOWTO/html/Sound/Sound-HOWTO.html

Russian: http://www.phtd.tpu.edu.ru/~ott/russian/linux/howto-rus/Sound-HOWTO.html

Spanish: ftp://ftp.insflug.org/es

Most translations of this and other Linux HOWTOs can also be found at http://metalab.unc.edu/pub/Linux/docs/HOWTO/translations/ and ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/translations/. If you make a translation of this document into another language, let me know and I'll include a reference to it here.

1.3 Feedback

I rely on you, the reader, to make this HOWTO useful. If you have any suggestions, corrections, or comments, please send them to me, tranter@pobox.com, and I will try to incorporate them in the next revision.

I am also willing to answer general questions on sound cards under Linux, as best I can. Before doing so, please read all of the information in this HOWTO, and send me detailed information about the problem. Please do not ask me about using sound cards under operating systems other than Linux.

If you publish this document on a CD-ROM or in hardcopy form, a complimentary copy would be appreciated. Mail me for my postal address. Also consider making a donation to the Linux Documentation Project to help support free documentation for Linux. Contact the Linux HOWTO coordinator, Tim Bynum mailto:linux-howto@metalab.unc.edu, for more information.

1.4 Distribution Policy

Copyright (c) 1995-1999 by Jeff Tranter. This document may be distributed under the terms set forth in the LDP license at http://metalab.unc.edu/LDP/COPYRIGHT.html.

---

2. Sound Card Technology

This section gives a *very* cursory overview of computer audio technology, in order to help you understand the concepts used later in the document. You should consult a book on digital audio or digital signal processing in order to learn more.

Sound is an *analog* property; it can take on any value over a continuous range. Computers are *digital*; they like to work with discrete values. Sound cards use a device known as an *Analog to Digital Converter* (A/D or ADC) to convert voltages corresponding to analog sound waves into digital or numeric values which can be stored in memory. Similarly, a *Digital to Analog Converter* (D/A or DAC) converts numeric values back to an analog voltage which can in turn drive a loudspeaker, producing sound.

The process of analog to digital conversion, known as sampling, introduces some error. Two factors are key in determining how well the sampled signal represents the original. *Sampling rate* is the number of samples made per unit of time (usually expresses as samples per second or Hertz). A low sampling rate will provide a less accurate representation of the analog signal. Sample size is the range of values used to represent each sample, usually expressed in bits. The larger the sample size, the more accurate the digitized signal will be.

Sound cards commonly use 8 or 16 bit samples at sampling rates from about 4000 to 44,000 samples per second. The samples may also be contain one channel (mono) or two (stereo).

FM Synthesis is an older technique for producing sound. It is based on combining different waveforms (e.g. sine, triangle, square). FM synthesis is simpler to implement in hardware that D/A conversion, but is more difficult to program and less flexible. Many sound cards provide FM synthesis for backward compatibility with older cards and software. Several independent sound generators or *voices* are usually provided.

Wavetable Synthesis combines the flexibility of D/A conversion with the multiple channel capability of FM synthesis. With this scheme digitized voices can be downloaded into dedicated memory, and then played, combined, and modified with little CPU overhead. State of the art sound cards all support wavetable synthesis.

Most sound cards provide the capability of *mixing*, combining signals from different input sources and controlling gain levels.

MIDI stands for Musical Instrument Digital Interface, and is a standard hardware and software protocol for allowing musical instruments to communicate with each other. The events sent over a MIDI bus can also be stored as MIDI files for later editing and playback. Many sound cards provide a MIDI interface. Those that do not can still play MIDI files using the on-board capabilities of the sound card.

MOD files are a common format for computer generated songs. As well as information about the musical notes to be played, the files contain digitized samples for the instruments (or voices). MOD files originated on the Amiga computer, but can be played on other systems, including Linux, with suitable

software.

3. Supported Hardware

This section lists the sound cards and interfaces that are currently supported under Linux. The information here is based on the latest Linux kernel, which at time of writing was version 2.2.4. This document only applies to the sound drivers included with the standard Linux kernel source distribution. There are other sound drivers available for Linux (see the later section entitled Alternate Sound Drivers).

For the latest information on supported sound cards and features see the files included with the Linux kernel source code, usually installed in the directory /usr/src/linux/Documentation/sound.

The information in this HOWTO is valid for Linux on the Intel platform.

The sound driver should also work with most sound cards on the Alpha platform. However, some cards may conflict with I/O ports of other devices on Alpha systems even though they work perfectly on i386 machines, so in general it's not possible to tell if a given card will work or not without actually trying it.

Users have reported that the sound driver was not yet working on the PowerPC version of Linux, but it should be supported in future.

Sound can be configured into the kernel under the MIPs port of Linux, and some MIPs machines have EISA slots and/or built in sound hardware. I'm told the Linux-MIPs group is interested in adding sound support in the future.

The Linux kernel includes a separate driver for the Atari and Amiga versions of Linux that implements a compatible subset of the sound driver on the Intel platform using the built-in sound hardware on these machines.

The SPARC port of Linux currently has sound support for some models of Sun workstations. I've been told that the on-board sound hardware works but the external DSP audio box is not supported because Sun has not released the specifications for it.

3.1 Sound Cards

The following sound cards are supported by the Linux kernel sound driver. Some of the items listed are audio chip sets rather than models of sound cards. The list is incomplete because there are many sound cards compatible with these that will work under Linux. To add further to the confusion, some manufacturers periodically change the design of their cards causing incompatibilities and continue to sell them as the same model.

- 6850 UART MIDI Interface
- AD1816/AD1816A based cards
- ADSP-2115
- ALS-007 based cards (Avance Logic)

TOC

- ATI Stereo F/X (no longer manufactured)
- Acer FX-3D
- AdLib (no longer manufactured)
- Audio Excel DSP 16
- AudioDrive
- CMI8330 sound chip
- Compaq Deskpro XL onboard sound
- Corel Netwinder WaveArtist
- Crystal CS423x
- ESC614
- ESS1688 sound chip
- ESS1788 sound chip
- ESS1868 sound chip
- ESS1869 sound chip
- ESS1887 sound chip
- ESS1888 sound chip
- ESS688 sound chip
- ES1370 sound chip
- ES1371 sound chip
- Ensoniq AudioPCI (ES1370)
- Ensoniq AudioPCI 97 (ES1371)
- Ensoniq SoundScape (and compatibles made by Reveal and Spea)
- Gallant SC-6000
- Gallant SC-6600
- Gravis Ultrasound
- Gravis Ultrasound ACE
- Gravis Ultrasound Max
- Gravis Ultrasound with 16 bit sampling option
- HP Kayak
- Highscreen Sound-Booster 32 Wave 3D
- IBM MWAVE
- Logitech Sound Man 16
- Logitech SoundMan Games
- Logitech SoundMan Wave
- MAD16 Pro (OPTi 82C928, 82C929, 82C930, 82C924 chipsets)
- Media Vision Jazz16
- MediaTriX AudioTriX Pro

TOC

- Microsoft Windows Sound System (MSS/WSS)
- MiroSOUND PCM12
- Mozart (OAK OTI-601)
- OPTi 82C931
- Orchid SW32
- Personal Sound System (PSS)
- Pinnacle MultiSound
- Pro Audio Spectrum 16
- Pro Audio Studio 16
- Pro Sonic 16
- Roland MPU-401 MIDI interface
- S3 SonicVibes
- SY-1816
- Sound Blaster 1.0
- Sound Blaster 2.0
- Sound Blaster 16
- Sound Blaster 16ASP
- Sound Blaster 32
- Sound Blaster 64
- Sound Blaster AWE32
- Sound Blaster AWE64
- Sound Blaster PCI 128
- Sound Blaster Pro
- Sound Blaster Vibra16
- Sound Blaster Vibra16X
- TI TM4000M notebook
- Terratec Base 1
- Terratec Base 64
- ThunderBoard
- Turtle Beach Maui
- Turtle Beach MultiSound Classic
- Turtle Beach MultiSound Fiji
- Turtle Beach MultiSound Hurricane
- Turtle Beach MultiSound Monterey
- Turtle Beach MultiSound Pinnacle
- Turtle Beach MultiSound Tahiti
- Turtle Beach WaveFront Maui

TOC

- Turtle Beach WaveFront Tropez
- Turtle Beach WaveFront Tropez+
- VIA chip set
- VIDC 16-bit sound
- Yamaha OPL2 sound chip
- Yamaha OPL3 sound chip
- Yamaha OPL3-SA1 sound chip
- Yamaha OPL3-SA2 sound chip
- Yamaha OPL3-SA3 sound chip
- Yamaha OPL3-SAx sound chip
- Yamaha OPL4 sound chip

A word about compatibility: even though most sound cards are claimed to be *SoundBlaster compatible*, very few currently sold cards are compatible enough to work with the Linux SoundBlaster driver. These cards usually work better using the MSS/WSS or MAD16 driver. Only real SoundBlaster cards made by Creative Labs, which use Creative's custom chips (e.g. SoundBlaster16 Vibra), MV Jazz16 and ESS688/1688 based cards generally work with the SoundBlaster driver. Trying to use a SoundBlaster Pro compatible 16 bit sound card with the SoundBlaster driver is usually just a waste of time.

The Linux kernel supports the SCSI port provided on some sound cards (e.g. ProAudioSpectrum 16) and the proprietary interface for some CD-ROM drives (e.g. Soundblaster Pro). See the Linux SCSI HOWTO and CDROM HOWTO documents for more information.

A kernel driver to support joystick ports, including those provided on some sound cards, is included as part of the 2.2 kernels.

Note that the kernel SCSI, CD-ROM, joystick, and sound drivers are completely independent of each other.

3.2 Alternate Sound Drivers

Sound support in the Linux kernel was originally written by Hannu Savolainen. Hannu then went on to develop the Open Sound system, a commercial set of sound drivers sold by 4Front Technologies that is supported on a number of Unix systems. Red Hat Software sponsored Alan Cox to enhance the kernel sound drivers to make them fully modular. Various other people also contributed bug fixes and developed additional drivers for new sound cards. These modified drivers were shipped by Red Hat in their 5.0 through 5.2 releases. These changes have now been integrated into the standard kernel as of version 2.0. Alan Cox is now the maintainer of the standard kernel sound drivers, although Hannu still periodically contributes code taken from the commercial driver.

The commercial Open Sound System driver from 4Front Technologies tends to be easier to configure and support more sound cards, particularly the newer models. It is also compatible with applications written for the standard kernel sound drivers. The disadvantage is that you need to pay for it, and you do not get source code. You can download a free evaluation copy of the product before deciding whether to

purchase it. For more information see the 4Front Technologies web page at http://www.opensound.c

Jaroslav Kysela and others started writing an alternate sound driver for the Gravis UltraSound Card. The project was renamed *Advanced Linux Sound Architecture* (ALSA) and has resulted in what they believe is a more generally usable sound driver that can be used as a replacement for the built-in kernel drivers. The ALSA drivers support a number of popular sound cards, are full duplex, fully modularized, and compatible with the sound architecture in the kernel. The main web site of the ALSA project is http://www.alsa-project.org. A separate "Alsa-sound-mini-HOWTO" is available which deals with compiling and installing these drivers.

Markus Mummert (mum@mmk.e-technik.tu-muenchen.de) has written a driver package for the Turtle Beach MultiSound (classic), Tahiti, and Monterey sound cards. The documentation states:

> It is designed for high quality hard disk recording/playback without losing sync even on a busy system. Other features such as wave synthesis, MIDI and digital signal processor (DSP) cannot be used. Also, recording and playback at the same time is not possible. It currently replaces VoxWare and was tested on several kernel versions ranging from 1.0.9 to 1.2.1. Also, it is installable on UN*X SysV386R3.2 systems.

It can be found at http://www.cs.colorado.edu/~mccreary/tbeach.

Kim Burgaard (burgaard@daimi.aau.dk) has written a device driver and utilities for the Roland MPU-401 MIDI interface. The Linux software map entry gives this description:

> A device driver for true Roland MPU-401 compatible MIDI interfaces (including Roland SCC-1 and RAP-10/ATW-10). Comes with a useful collection of utilities including a Standard MIDI File player and recorder.
>
> Numerous improvements have been made since version 0.11a. Among other things, the driver now features IRQ sharing policy and complies with the new kernel module interface. Metronome functionality, possibility for synchronizing e.g. graphics on a per beat basis without losing precision, advanced replay/record/overdub interface and much, much more.

It can be found at ftp://metalab.unc.edu/pub/Linux/kernel/sound/mpu401-0.2.tar.gz.

Another novel use for a sound card under Linux is as a modem for amateur packet radio. The 2.1 and later kernels include a driver that works with SoundBlaster and Windows Sound System compatible sound cards to implement 1200 bps AFSK and 9600 bps FSK packet protocols. See the Linux AX25 HOWTO for details (I'm a ham myself, by the way -- callsign VE3ICH).

3.3 PC Speaker

An alternate sound driver is available that requires no additional sound hardware; it uses the internal PC speaker. It is mostly software compatible with the sound card driver, but, as might be expected, provides much lower quality output and has much more CPU overhead. The results seem to vary, being dependent on the characteristics of the individual loudspeaker. For more information, see the documentation provided with the release.

The software, which has not been updated for some time, can be found at

ftp://ftp.informatik.hu-berlin.de/pub/Linux/hu-sound/.

3.4 Parallel Port

Another option is to build a digital to analog converter using a parallel printer port and some additional components. This provides better sound quality than the PC speaker but still has a lot of CPU overhead. The PC sound driver package mentioned above supports this, and includes instructions for building the necessary hardware.

---

4. Installation

Configuring Linux to support sound involves the following steps:

1. Installing the sound card.
2. Configuring Plug and Play (if applicable).
3. Configuring and building the kernel for sound support.
4. Creating the device files.
5. Booting the Linux kernel and testing the installation.

If you are running Red Hat Linux there is a utility called `sndconfig` that in most cases will detect your sound card and set up all of the necessary configuration files to load the appropriate sound drivers for your card. If you are running Red Hat I suggest you try using it. If it works for you then you can skip the rest of the instructions in this section.

If `sndconfig` fails, you are using another Linux distribution, or you want to follow the manual method in order to better understand what you are doing, then the next sections will cover each of these steps in detail.

4.1 Installing the Sound Card

Follow the manufacturer's instructions for installing the hardware or have your dealer perform the installation.

Older sound cards usually have switch or jumper settings for IRQ, DMA channel, etc; note down the values used. If you are unsure, use the factory defaults. Try to avoid conflicts with other devices (e.g. ethernet cards, SCSI host adaptors, serial and parallel ports) if possible.

Usually you should use the same I/O port, IRQ, and DMA settings that work under DOS. In some cases though (particularly with PnP cards) you may need to use different settings to get things to work under Linux. Some experimentation may be needed.

4.2 Configuring Plug and Play

Most sound cards now use the Plug and Play protocol to configure settings for i/o addresses, interrupts, and DMA channels. If you have one of the older sound cards that uses fixed settings or jumpers, then you can skip this section.

As of version 2.2 Linux does not yet have full Plug and Play support in the kernel. The preferred solution is to use the isapnp tools which ship with most Linux distributions (or you can download them from Red Hat's web site http://www.redhat.com/).

First check the documentation for your Linux distribution. It may already have Plug and Play support set up for you or it may work slightly differently than described here. If you need to configure it yourself,the details can be found in the man pages for the isapnp tools. Briefly the process you would normally follow is:

- Use `pnpdump` to capture the possible settings for all your Plug and Play devices, saving the result to the file /etc/isapnp.conf.
- Choose settings for the sound card that do not conflict with any other devices in your system and uncomment the appropriate lines in /etc/isapnp.conf. Don't forget to uncomment the `(ACT Y)` command near the end.
- Make sure that `isapnp` is run when your system boots up, normally done by one of the startup scripts. Reboot your system or run `isapnp` manually.

If for some reason you cannot or do not wish to use the isapnp tools, there are a couple of other options. If you use the card under Microsoft Windows 95 or 98, you can use the device manager to set up the card, then soft boot into Linux using the LOADLIN program. Make sure Windows and Linux use the same card setup parameters.

If you use the card under DOS, you can use the `icu` utility that comes with SoundBlaster16 PnP cards to configure it under DOS, then soft boot into Linux using the LOADLIN program. Again, make sure DOS and Linux use the same card setup parameters.

A few of the sound card drivers include the necessary software to initialize Plug and Play for the card. Check the documentation

for that card's driver for details.

4.3 Configuring the Kernel

When initially installing Linux you likely used a precompiled kernel. These kernels often do not provide sound support. It is best to recompile the kernel yourself with the drivers you need. You may also want to recompile the kernel in order to upgrade to a newer version or to free up memory resources by minimizing the size of the kernel. Later, when your sound card is working, you may wish to rebuild the kernel sound drivers as modules.

The Linux Kernel HOWTO should be consulted for the details of building a kernel. I will just mention here some issues that are specific to sound cards.

If you have never configured the kernel for sound support before it is a good idea to read the relevant documentation included with the kernel sound drivers, particularly information specific to your card type. The files can be found in the kernel documentation directory, usually installed in /usr/src/linux/Documentation/sound. If this directory is missing you likely either have a very old kernel version or you have not installed the kernel source code.

Follow the usual procedure for building the kernel. There are currently three interfaces to the configuration process. A graphical user interface that runs under X11 can be invoked using `make xconfig`. A menu-based system that only requires text displays is available as `make menuconfig`. The original method, using `make config`, offers a simple text-based interface.

When configuring the kernel there will be many choices for selecting the type of sound card you have and the driver options to use. The on-line help within the configuration tool should provide an explanation of what each option is for. Select the appropriate options to the best of your knowledge.

After configuring the options you should compile and install the new kernel as per the Kernel HOWTO.

4.4 Creating the Device Files

For proper operation, device file entries must be created for the sound devices. These are normally created for you during installation of your Linux system. A quick check can be made using the command listed below. If the output is as shown (the date stamp will vary) then the device files are almost certainly okay.

```
% ls -l /dev/sndstat
crw-rw-rw-   1 root     root      14,   6 Apr 25  1995 /dev/sndstat
```

Note that having the right device files there doesn't guarantee anything on its own. The kernel driver must also be loaded or compiled in before the devices will work (more on that later).

In rare cases, if you believe the device files are wrong, you can recreate them. Most Linux distributions have a `/dev/MAKEDEV` script which can be used for this purpose.

4.5 Booting Linux and Testing the Installation

You should now be ready to boot the new kernel and test the sound drivers. Follow your usual procedure for installing and rebooting the new kernel (keep the old kernel around in case of problems, of course).

During booting, check for a message such as the following on powerup (if they scroll by too quickly to read, you may be able to retrieve them with the `dmesg` command):

```
Sound initialization started
<Sound Blaster 16 (4.13)> at 0x220 irq 5 dma 1,5
<Sound Blaster 16> at 0x330 irq 5 dma 0
<Yamaha OPL3 FM> at 0x388
Sound initialization complete
```

This should match your sound card type and jumper settings (if any).

TOC

Note that the above messages are not displayed when using loadable sound driver module (unless you enable it, e.g. using `insmod sound trace_init=1`).

When the sound driver is linked into the kernel, the `Sound initialization started` and `Sound initialization complete` messages should be displayed. If they are not printed, it means that there is no sound driver present in the kernel. In this case you should check that you actually installed the kernel you compiled when enabling the sound driver.

If nothing is printed between the `Sound initialization started` and the `Sound initialization complete` lines, it means that no sound devices were detected. Most probably it means that you don't have the correct driver enabled, the card is not supported, the I/O port is bad or that you have a PnP card that has not been configured.

The driver may also display some error messages and warnings during boot. Watch for these when booting the first time after configuring the sound driver.

Next you should check the device file /dev/sndstat. Reading the sound driver status device file should provide additional information on whether the sound card driver initialized properly. Sample output should look something like this:

```
% cat /dev/sndstat
Sound Driver:3.5.4-960630 (Sat Jan 4 23:56:57 EST 1997 root,
Linux fizzbin 2.0.27 #48 Thu Dec 5 18:24:45 EST 1996 i586)
Kernel: Linux fizzbin 2.0.27 #48 Thu Dec 5 18:24:45 EST 1996 i586
Config options: 0

Installed drivers:
Type 1: OPL-2/OPL-3 FM
Type 2: Sound Blaster
Type 7: SB MPU-401

Card config:
Sound Blaster at 0x220 irq 5 drq 1,5
SB MPU-401 at 0x330 irq 5 drq 0
OPL-2/OPL-3 FM at 0x388 drq 0

Audio devices:
0: Sound Blaster 16 (4.13)

Synth devices:
0: Yamaha OPL-3

Midi devices:
0: Sound Blaster 16

Timers:
0: System clock

Mixers:
0: Sound Blaster
```

The command above can report some error messages. "No such file or directory" indicates that you need to create the device files (see section 4.3). "No such device" means that sound driver is not loaded or linked into kernel. Go back to section 4.2 to correct this.

If lines in the "Card config:" section of `/dev/sndstat` are listed inside parentheses (such as "(SoundBlaster at 0x220 irq 5 drq 1,5)"), it means that this device was configured but not detected.

Now you should be ready to play a simple sound file. Get hold of a sound sample file, and send it to the sound device as a basic check of sound output, e.g.

```
% cat endoftheworld >/dev/dsp
% cat crash.au >/dev/audio
```

(Make sure you don't omit the ">" in the commands above).

Note that, in general, using `cat` is not the proper way to play audio files, it's just a quick check. You'll want to get a proper sound player program (described later) that will do a better job.

This command will work only if there is at least one device listed in the audio devices section of `/dev/sndstat`. If the audio devices section is empty you should check why the device was not detected.

If the above commands return "I/O error", you should look at the end of the kernel messages listed using the "dmesg" command. It's likely that an error message is printed there. Very often the message is "Sound: DMA (output) timed out - IRQ/DRQ config error?". The above message means that the driver didn't get the expected interrupt from the sound card. In most cases it means that the IRQ or the DMA channel configured to the driver doesn't work. The best way to get it working is to try with all possible DMAs and IRQs supported by the device.

Another possible reason is that the device is not compatible with the device the driver is configured for. This is almost certainly the case when a supposedly "SoundBlaster (Pro/16) compatible" sound card doesn't work with the SoundBlaster driver. In this case you should try to find out the device your sound card is compatible with (by posting to the comp.os.linux.hardware newsgroup, for example).

Some sample sound files can be obtained from ftp://tsx-11.mit.edu/pub/linux/packages/sound/snd-data-0.1.tar.Z

Now you can verify sound recording. If you have sound input capability, you can do a quick test of this using commands such as the following:

```
# record 4 seconds of audio from microphone
EDT% dd bs=8k count=4 </dev/audio >sample.au
4+0 records in
4+0 records out
# play back sound
% cat sample.au >/dev/audio
```

Obviously for this to work you need a microphone connected to the sound card and you should speak into it. You may also need to obtain a mixer program to set the microphone as the input device and adjust the recording gain level.

If these tests pass, you can be reasonably confident that the sound D/A and A/D hardware and software are working. If you experience problems, refer to the next section of this document.

4.6 Troubleshooting

If you still encounter problems after following the instructions in the HOWTO, here are some things to check. The checks are listed in increasing order of complexity. If a check fails, solve the problem before moving to the next stage.

Step 1: Make sure you are really running the kernel you compiled.

You can check the date stamp on the kernel to see if you are running the one that you compiled with sound support. You can do this with the `uname` command:

```
% uname -a
Linux fizzbin 2.2.4 #1 Tue Mar 23 11:23:21 EST 1999 i586 unknown
```

or by displaying the file /proc/version:

```
% cat /proc/version
Linux version 2.2.4 (root@fizzbin) (gcc version 2.7.2.3) #1 Tue Mar 23 11:23:21 EST
1999
```

TOC

If the date stamp doesn't seem to match when you compiled the kernel, then you are running an old kernel. Did you really re
If you use LILO, did you re-install it (typically by running lilo)? If booting from floppy, did you create a new boot floppy ar
it when booting?

Step 2: Make sure the kernel sound drivers are compiled in.

The easiest way to do this is to check the output of `dev/sndstat` as described earlier. If the output is not as expected then something went wrong with the kernel configuration or build. Start the installation process again, beginning with configuration and building of the kernel.

Step 3: Did the kernel detect your sound card during booting?

Make sure that the sound card was detected when the kernel booted. You should have seen a message on bootup. If the messages scrolled off the screen, you can usually recall them using the `dmesg` command:

```
% dmesg
```

or

```
% tail /var/log/messages
```

If your sound card was not found then something is wrong. Make sure it really is installed. If the sound card works under DOS then you can be reasonably confident that the hardware is working, so it is likely a problem with the kernel configuration. Either you configured your sound card as the wrong type or wrong parameters, or your sound card is not compatible with any of the Linux kernel sound card drivers.

One possibility is that your sound card is one of the *compatible* type that requires initialization by the DOS driver. Try booting DOS and loading the vendor supplied sound card driver. Then soft boot Linux using `Control-Alt-Delete`. Make sure that card I/O address, DMA, and IRQ settings for Linux are the same as used under DOS. Read the Readme.cards file from the sound driver source distribution for hints on configuring your card type.

If your sound card is not listed in this document, it is possible that the Linux drivers do not support it. You can check with some of the references listed at the end of this document for assistance.

Step 4: Can you read data from the dsp device?

Try reading from the /dev/audio device using the `dd` command listed earlier in this document. The command should run without errors.

If it doesn't work, then chances are that the problem is an IRQ or DMA conflict or some kind of hardware incompatibility (the device is not supported by Linux or the driver is configured for a wrong device).

A remote possibility is broken hardware. Try testing the sound card under DOS, if possible, to eliminate that as a possibility.

When All Else Fails

If you still have problems, here are some final suggestions for things to try:

- carefully re-read this HOWTO document
- read the references listed at the end of this document and the relevant kernel source documentation files
- post a question to one of the `comp.os.linux` or other Usenet newsgroups (comp.os.linux.hardware is a good choice; because of the high level of traffic in these groups it helps to put the string "sound" in the subject header for the article so the right experts will see it)
- Using a web/Usenet search engine with an intelligently selected search criteria can give very good results quickly. One such choice is http://www.altavista.digital.com
- try using the latest Linux kernel (but only as a last resort, the latest development kernels can be unstable)
- send mail to the author of the sound driver

- send mail to the author of the Sound HOWTO
- fire up emacs and type `Esc-x doctor` :-)

5. Applications Supporting Sound

I give here a sample of the types of applications that you likely want if you have a sound card under Linux. You can check the Linux Software Map, Internet archive sites, and/or files on your Linux CD-ROM for more up to date information.

As a minimum, you will likely want to obtain the following sound applications:

- audio file format conversion utility (e.g. `sox`)
- mixer utility (e.g. `aumix` or `xmix`)
- digitized file player/recorder (e.g. `play` or `wavplay`)
- MOD file player (e.g. `tracker`)
- MIDI file player (e.g. `playmidi`)

There are text-based as well as GUI-based versions of most of these tools. There are also some more esoteric applications (e.g. speech synthesis and recognition) that you may wish to try.

6. Answers To Frequently Asked Questions

This section answers some of the questions that have been commonly asked on the Usenet news groups and mailing lists.

Answers to more questions can also be found at the OSS sound driver web page.

6.1 What are the various sound device files?

These are the most standard device file names, some Linux distributions may use slightly different names.

/dev/audio

normally a link to /dev/audio0

/dev/audio0

Sun workstation compatible audio device (only a partial implementation, does not support Sun ioctl interface, just u-law encoding)

/dev/audio1

second audio device (if supported by sound card or if more than one sound card installed)

/dev/dsp

normally a link to /dev/dsp0

/dev/dsp0

first digital sampling device

/dev/dsp1

second digital sampling device

/dev/mixer

normally a link to /dev/mixer0

/dev/mixer0

first sound mixer

/dev/mixer1

second sound mixer

/dev/music

high-level sequencer interface

/dev/sequencer

low level MIDI, FM, and GUS access

/dev/sequencer2

normally a link to /dev/music

/dev/midi00

TOC

1st raw MIDI port

/dev/midi01

2nd raw MIDI port

/dev/midi02

3rd raw MIDI port

/dev/midi03

4th raw MIDI port

/dev/sndstat

displays sound driver status when read (also available as /proc/sound)

The PC speaker driver provides the following devices:

/dev/pcaudio

equivalent to /dev/audio

/dev/pcsp

equivalent to /dev/dsp

/dev/pcmixer

equivalent to /dev/mixer

6.2 How can I play a sound sample?

Sun workstation (.au) sound files can be played by sending them to the /dev/audio device. Raw samples can be sent to /dev/dsp. This will generally give poor results though, and using a program such as `play` is preferable, as it will recognize most file types and set the sound card to the correct sampling rate, etc.

Programs like wavplay or vplay (in the snd-util package) will give best results with WAV files. However they don't recognize Microsoft ADPCM compressed WAV files. Also older versions of play (from the Lsox package) doesn't work well with 16 bit WAV files.

The splay command included in the snd-util package can be used to play most sound files if proper parameters are entered manually in the command line.

6.3 How can I record a sample?

Reading /dev/audio or /dev/dsp will return sampled data that can be redirected to a file. A program such as `vrec` makes it easier to control the sampling rate, duration, etc. You may also need a mixer program to select the appropriate input device.

6.4 Can I have more than one sound card?

With the current sound driver it's possible to have several SoundBlaster, SoundBlaster/Pro, SoundBlaster16, MPU-401 or MSS cards at the same time on the system. Installing two SoundBlasters is possible but requires defining the macros SB2_BASE, SB2_IRQ, SB2_DMA and (in some cases) SB2_DMA2 by editing local.h manually. It's also possible to have a SoundBlaster at the same time as a PAS16.

With the 2.0 and newer kernels that configure sound using make config, instead of local.h, you need to edit the file

/usr/include/linux/autoconf.h. After the section containing the lines:

```
#define SBC_BASE 0x220
#define SBC_IRQ (5)
#define SBC_DMA (1)
#define SB_DMA2 (5)
#define SB_MPU_BASE 0x0
#define SB_MPU_IRQ (-1)
```

add these lines (with values appropriate for your system):

```
#define SB2_BASE 0x330
#define SB2_IRQ (7)
#define SB2_DMA (2)
#define SB2_DMA2 (2)
```

The following drivers don't permit multiple instances:

- GUS (driver limitation)
- MAD16 (hardware limitation)
- AudioTrix Pro (hardware limitation)
- CS4232 (hardware limitation)

6.5 Error: No such file or directory for sound devices

You need to create the sound driver device files. See the section on creating device files. If you do have the device files, ensure that they have the correct major and minor device numbers (some older CD-ROM distributions of Linux may not create the correct device files during installation).

6.6 Error: No such device for sound devices

You have not booted with a kernel containing the sound driver or the I/O address configuration doesn't match your hardware. Check that you are running the newly compiled kernel and verify that the settings entered when configuring the sound driver match your hardware setup.

6.7 Error: No space left on device for sound devices

This can happen if you tried to record data to /dev/audio or /dev/dsp without creating the necessary device file. The sound device is now a regular file, and has filled up your disk partition. You need to run the script described in the *Creating the Device Files* section of this document.

This may also happen with Linux 2.0 and later if there is not enough free RAM on the system when the device is opened. The audio driver requires at least two pages (8k) of contiguous physical RAM for each DMA channel. This happens sometimes in machines with less than 16M of RAM or which have been running for very long time. It may be possible to free some RAM by compiling and running the following C program before trying to open the device again:

```
main() {
  int i;
  char mem[500000];
  for (i = 0; i < 500000; i++)
```

```
        mem[i] = 0;
      exit(0);
    }
```

6.8 Error: Device busy for sound devices

Only one process can open a given sound device at one time. Most likely some other process is using the device in question. One way to determine this is to use the `fuser` command:

```
% fuser -v /dev/dsp
/dev/dsp:             USER       PID ACCESS COMMAND
                      tranter    265 f....  tracker
```

In the above example, the `fuser` command showed that process 265 had the device open. Waiting for the process to complete or killing it will allow the sound device to be accessed once again. You should run the `fuser` command as root in order to report usage by users other than yourself.

On some systems you may need to be root when running the `fuser` command in order to see the processes of other users.

6.9 I still get device busy errors!

According to Brian Gough, for the SoundBlaster cards which use DMA channel 1 there is a potential conflict with the QIC-02 tape driver, which also uses DMA 1, causing "device busy" errors. If you are using FTAPE, you may have this driver enabled. According to the FTAPE-HOWTO the QIC-02 driver is not essential for the use of FTAPE; only the QIC-117 driver is required. Reconfiguring the kernel to use QIC-117 but not QIC-02 allows FTAPE and the sound-driver to coexist.

6.10 Partial playback of digitized sound file

The symptom is usually that a sound sample plays for about a second and then stops completely or reports an error message about "missing IRQ" or "DMA timeout". Most likely you have incorrect IRQ or DMA channel settings. Verify that the kernel configuration matches the sound card jumper settings and that they do not conflict with some other card.

Another symptom is sound samples that *loop*. This is usually caused by an IRQ conflict.

6.11 There are pauses when playing MOD files

Playing MOD files requires considerable CPU power. You may have too many processes running or your computer may be too slow to play in real time. Your options are to:

- try playing with a lower sampling rate or in mono mode
- eliminate other processes
- buy a faster computer
- buy a more powerful sound card (e.g. Gravis UltraSound)

If you have a Gravis UltraSound card, you should use one of the mod file players written specifically for the GUS (e.g. `gmod`).

6.12 Compile errors when compiling sound applications

The version 1.0c and earlier sound driver used a different and incompatible `ioctl()` scheme. Obtain newer source code or make the necessary changes to adapt it to the new sound driver. See the sound driver Readme file for details.

Also ensure that you have used the latest version of soundcard.h and ultrasound.h when compiling the application. See the installation instructions at beginning of this text.

6.13 SEGV when running sound binaries that worked previously

This is probably the same problem described in the previous question.

6.14 What known bugs or limitations are there in the sound driver?

See the files included with the sound driver kernel source.

6.15 Where are the sound driver ioctls() etc. documented?

Currently the best documentation, other than the source code, is available at the 4Front Technologies web site, http://www.opensound.com. Another source of information is the *Linux Multimedia Guide*, described in the references section.

6.16 What CPU resources are needed to play or record without pauses?

There is no easy answer to this question, as it depends on:

- whether using PCM sampling or FM synthesis
- sampling rate and sample size
- which application is used to play or record
- Sound Card hardware
- disk I/O rate, CPU clock speed, cache size, etc.

In general, any 386 machine or better should be able to play samples or FM synthesized music on an 8 bit sound card with ease.

Playing MOD files, however, requires considerable CPU resources. Some experimental measurements have shown that playing at 44kHz requires more than 40% of the speed of a 486/50 and a 386/25 can hardly play faster than 22 kHz (these are with an 8 bit card sound such as a SoundBlaster). A card such as the Gravis UltraSound card performs more functions in hardware, and will require less CPU resources.

These statements assume the computer is not performing any other CPU intensive tasks.

Converting sound files or adding effects using a utility such as `sox` is also much faster if you have a math coprocessor (or CPU with on board FPU). The kernel driver itself does not do any floating point calculations, though.

6.17 Problems with a PAS16 and an Adaptec 1542 SCSI host adaptor

TOC

(the following explanation was supplied by `seeker@indirect.com`)

Linux only recognizes the 1542 at address 330 (default) or 334, and the PAS only allows the MPU-401 emulation at 330. Even when you disable the MPU-401 under software, something still wants to conflict with the 1542 if it's at its preferred default address. Moving the 1542 to 334 makes everyone happy.

Additionally, both the 1542 and the PAS-16 do 16-bit DMA, so if you sample at 16-bit 44 KHz stereo and save the file to a SCSI drive hung on the 1542, you're about to have trouble. The DMAs overlap and there isn't enough time for RAM refresh, so you get the dread ``PARITY ERROR - SYSTEM HALTED'' message, with no clue to what caused it. It's made worse because a few second-party vendors with QIC-117 tape drives recommend setting the bus on/off times such that the 1542 is on even longer than normal. Get the SCSISEL.EXE program from Adaptec's BBS or several places on the internet, and reduce the BUS ON time or increase the BUS OFF time until the problem goes away, then move it one notch or more further. SCSISEL changes the EEPROM settings, so it's more permanent than a patch to the DOS driver line in CONFIG.SYS, and will work if you boot right into Linux (unlike the DOS patch). Next problem solved.

Last problem - the older Symphony chipsets drastically reduced the timing of the I/O cycles to speed up bus accesses. None of various boards I've played with had *any* problem with the reduced timing except for the PAS-16. Media Vision's BBS has SYMPFIX.EXE that's supposed to cure the problem by twiddling a diagnostic bit in Symphony's bus controller, but it's not a hard guarantee. You may need to:

- get the motherboard distributor to replace the older version bus chip,
- replace the motherboard, or
- buy a different brand of sound card.

Young Microsystems will upgrade the boards they import for around $30 (US); other vendors may be similar if you can figure out who made or imported the motherboard (good luck). The problem is in ProAudio's bus interface chip as far as I'm concerned; *nobody* buys a $120 sound card and sticks it in a 6MHz AT. Most of them wind up in 25-40MHz 386/486 boxes, and should be able to handle *at least* 12MHz bus rates if the chips are designed right. Exit soapbox (stage left).

The first problem depends on the chipset used on your motherboard, what bus speed and other BIOS settings, and the phase of the moon. The second problem depends on your refresh option setting (hidden or synchronous), the 1542 DMA rate and (possibly) the bus I/O rate. The third can be determined by calling Media Vision and asking which flavor of Symphony chip is incompatible with their slow design. Be warned, though - 3 of 4 techs I talked to were brain damaged. I would be very leery of trusting *anything* they said about someone else's hardware, since they didn't even know their own very well.

6.18 Is it possible to read and write samples simultaneously?

The drivers for some sound cards support full duplex mode. Check the documentation available from 4Front Technologies for information on how to use it.

6.19 My SB16 is set to IRQ 2, but configure does not allo this value.

TOC

On '286 and later machines, the IRQ 2 interrupt is cascaded to the second interrupt controller. It is equivalent to IRQ 9.

6.20 If I run Linux, then boot DOS, I get errors and/or sound applications do not work properly.

This happens after a soft reboot to DOS. Sometimes the error message misleadingly refers to a bad CONFIG.SYS file.

Most of the current sound cards have software programmable IRQ and DMA settings. If you use different settings between Linux and MS-DOS/Windows, this may cause problems. Some sound cards don't accept new parameters without a complete reset (i.e. cycle the power or use the hardware reset button).

The quick solution to this problem it to perform a full reboot using the reset button or power cycle rather than a soft reboot (e.g. Ctrl-Alt-Del).

The correct solution is to ensure that you use the same IRQ and DMA settings with MS-DOS and Linux (or not to use DOS :-).

6.21 Problems running DOOM under Linux

Users of the port of ID software's game DOOM for Linux may be interested in these notes.

For correct sound output you need version 2.90 or later of the sound driver; it has support for the real-time `DOOM mode`.

The sound samples are 16-bit. If you have an 8-bit sound card you can still get sound to work using one of several programs available in ftp://metalab.unc.edu/pub/Linux/games/doom.

If performance of DOOM is poor on your system, disabling sound (by renaming the file `sndserver`) may improve it.

By default DOOM does not support music (as in the DOS version). The program `musserver` will add support for music to DOOM under Linux. It can be found at ftp://pandora.st.hmc.edu/pub/linux/musserver.tgz.

6.22 How can I reduce noise picked up by my sound card?

Using good quality shielded cables and trying the sound card in different slots may help reduce noise. If the sound card has a volume control, you can try different settings (maximum is probably best). Using a mixer program you can make sure that undesired inputs (e.g. microphone) are set to zero gain.

Philipp Braunbeck reported that on his ESS-1868 sound card there was a jumper to turn off the built-in amplifier which helped reduce noise when enabled.

On one 386 system I found that the kernel command line option `no-hlt` reduced the noise level. This tells the kernel not to use the halt instruction when running the idle process loop. You can try this manually when booting, or set it up using the command `append="no-hlt"` in your LILO configuration file.

Some sound cards are simply not designed with good shielding and grounding and are prone to noise pickup.

TOC

6.23 I can play sounds, but not record.

If you can play sound but not record, try these steps:

- use a mixer program to select the appropriate device (e.g. microphone)
- use the mixer to set the input gains to maximum
- If you can, try to test sound card recording under MS-DOS to determine if there is a hardware problem

Sometimes a different DMA channel is used for recording than for playback. In this case the most probable reason is that the recording DMA is set up incorrectly.

6.24 My "compatible" sound card only works if I first initialize under MS-DOS.

In most cases a "SoundBlaster compatible" card will work better under Linux if configured with a driver other than the SoundBlaster one. Most sound cards claim to be compatible (e.g. "16 bit SB Pro compatible" or "SB compatible 16 bit") but usually this SoundBlaster mode is just a hack provided for DOS games compatibility. Most cards have a 16 bit native mode which is likely to be supported by recent Linux versions (2.0.1 and later).

Only with some (usually rather old) cards is it necessary to try to get them to work in the SoundBlaster mode. The only newer cards that are the exception to this rule are the Mwave-based cards.

6.25 My 16-bit SoundBlaster "compatible" sound card only works in 8-bit mode under Linux.

16-bit sound cards described as SoundBlaster compatible are really only compatible with the 8-bit SoundBlaster Pro. They typically have a 16-bit mode which is not compatible with the SoundBlaster 16 and not compatible with the Linux sound driver.

You may be able to get the card to work in 16-bit mode by using the MAD16 or MSS/WSS driver.

6.26 Where can I find sound applications for Linux?

Here are some good archive sites to search for Linux specific sound applications:

- ftp://metalab.unc.edu/pub/Linux/kernel/sound/
- ftp://metalab.unc.edu/pub/Linux/apps/sound/
- ftp://tsx-11.mit.edu/pub/linux/packages/sound/
- ftp://nic.funet.fi/pub/Linux/util/sound/
- ftp://nic.funet.fi/pub/Linux/xtra/snd-kit/
- ftp://nic.funet.fi/pub/Linux/ALPHA/sound/

Also see the References section of this document.

6.27 Can the sound driver be compiled as a loadable module?

TOC

With recent kernels the sound driver is supported as several kernel loadable modules.

See the files in /usr/src/linux/Documentation/sound, especially the files Introduction and README.modules.

6.28 Can I use a sound card to replace the system console beep?

Try the `oplbeep` program, found at ftp://metalab.unc.edu/pub/Linux/apps/sound/oplbeep-2.3.tar.gz

Another variant is the `beep` program found at ftp://metalab.unc.edu/pub/Linux/kernel/patches/misc/modreq_beep.tgz

The `modutils` package has an example program and kernel patch that supports calling an arbitrary external program to generate sounds when requested by the kernel.

Alternatively, with some sound cards you can connect the PC speaker output to the sound card so that all sounds come from the sound card speakers.

6.29 What is VoxWare?

The commercial version of the sound drivers sold by 4Front Technologies was previously known by other names such as *VoxWare*, *USS* (Unix Sound System), and even *TASD* (Temporarily Anonymous Sound Driver). It is now marketed as *OSS* (Open Sound System). The version included in the Linux kernel is sometimes referred to as *OSS/Free*.

For more information see the 4Front Technologies web page at http://www.opensound.com/. I wrote a review of OSS/Linux in the June 1997 issue of Linux Journal.

6.30 Sox/Play/Vplay reports "invalid block size 1024"

A change to the sound driver in version 1.3.67 broke some sound player programs which (incorrectly) checked that the result from the SNDCTL_DSP_GETBLKSIZE ioctl was greater than 4096. The latest sound driver versions have been fixed to avoid allocating fragments shorter than 4096 bytes which solves this problem with old utilities.

6.31 The mixer settings are reset whenever I load the sound driver module

You can build the sound driver as a loadable module and use `kerneld` to automatically load and unload it. This can present one problem - whenever the module is reloaded the mixer settings go back to their default values. For some sound cards this can be too loud (e.g. SoundBlaster16) or too quiet. Markus Gutschke (`gutschk@uni-muenster.de`) found this solution. Use a line in your /etc/conf.modules file such as the following:

```
options sound dma_buffsize=65536 && /usr/bin/setmixer igain 0 ogain 0 vol 75
```

This causes your mixer program (in this case `setmixer`) to be run immediately after the sound driver is loaded. The `dma_buffsize` parameter is just a dummy value needed because the option command requires a command line option. Change the line as needed to match your mixer program and gain settings.

If you have compiled the sound driver into your kernel and you want to set the mixer gains at boot time you can p
call to your mixer program in a system startup file such as /etc/rc.d/rc.local.

TOC

6.32 Only user root can record sound

By default the script in Readme.linux that creates the sound device files only allows the devices to be read by user `root`. This is to plug a potential security hole. In a networked environment, external users could conceivably log in remotely to a Linux PC with a sound card and microphone and eavesdrop. If you are not worried about this, you can change the permissions used in the script.

With the default setup, users can still play sound files. This is not a security risk but is a potential for nuisance.

6.33 Is the sound hardware on the IBM ThinkPad supported?

Information on how to use the mwave sound card on an IBM ThinkPad laptop computer can be found in the file /usr/src/linux/Documentation/sound/mwave, which is part of the kernel source distribution.

6.34 Applications fail because my sound card has no mixer

Some old 8-bit SoundBlaster cards have no mixer circuitry. Some sound applications insist on being able to open the mixer device, and fail with these cards. Jens Werner (werner@bert.emv.ing.tu-bs.de) reports a workaround for this is to link `/dev/mixer` to `/dev/null` and everything should work fine.

6.35 Problems with a SB16 CT4170

From Scott Manley (spm@star.arm.ac.uk):

> There seems to be a new type of Soundblaster - it was sold to us as a SB16 - the Model no. on the Card is CT4170. These Beasties only have one DMA channel so when you try to set them up then the kernel will have trouble accessing the 16 bit DMA. The solution is to set the second DMA to 1 so that the card will behave as advertised.

6.36 How to connect a MIDI keyboard to a soundcard

From Kim G. S. OEyhus (kim@pvv.ntnu.no):

> I looked all around the internet and in sound documentation on how to do something as simple as connecting the MIDI output from a master keyboard to the MIDI input on a sound card. I found nothing. The problem is that they both use the same device, /dev/midi, at least when using the OSS sound system. So I found a way to do it, which I want to share. This makes for a very simple synthesizer, with full MIDI support:
>
> CONNECTING A MIDI MASTER-KEYBOARD DIRECTLY TO A SOUNDCARD WITH MIDI
>
> A MIDI master-keyboard is a keyboard without any synthesizer, and with only a MIDI-out plug. This can be connected to the 15-pin D-SUB port on most sound-cards with a suitable cable.
>
> Such a keyboard can be used to control the MIDI synthesizer device for the card, thus making a simple keyboard controlled synthesizer.
>
> Compile the following program, say with 'gcc -o prog prog.c' and run it:

```
#include <fcntl.h>

main()
{
  int fil, a;
  char b[256];

  fil=open("/dev/midi", O_RDWR);
  for(;;)
    {
      a=read(fil, b, 256);
      write(fil, b, a);
    }
}
```

6.37 Problems with IRQ 15 and Ensoniq PCI 128

From Matthew Inger (mattinger@mindless.com):

Information on getting an Ensoniq PCI 128 card to work.

The problem that it was exhibiting was that it was trying to use interrupt 15 by default (Plug and Pray was responsible for this one). This interrupt is used by the secondary ide controller, and cannot be shared by other devices. You need to somehow force the es1370 to use another interrupt (should use interrupt 11 like it does under windows).

I figured this one out for myself believe it or not.

What I had to do was:

a) in the BIOS, you have to tell the computer that you don't have a Plug and Play OS. I believe this is under advanced options in my BIOS.

b) in the PCI settings in the BIOS, tell the computer to reserve interrupt 15 for legacy ISA devices. In my bios, under advanced options, there is a section for PCI settings. Under there, there is a Resource Exclusion area, and that's where to do this.

When you reboot into linux you will be able to use sound. (I don't remember if it shows up in the boot messages or not like it used to). To be safe, I ran sndconfig again so that it would play the test message, which sounded not great, but it was there. When I played a CD however, it sounded perfect.

Don't worry about windows, I tried both my cards: ISA Modem, and the Sound Card out, and they work without any hitches.

The odds are your BIOS will be different from mine, but you just have to figure out where the settings are for the above two items. Good luck.

6.38 Where can I get freely available MIDI patches to run SoftOSS?

SoftOSS is a software-based wavetable synthesizer included with the kernel sound driver that is compatible with the Gravis Utrasound card. To operate the driver needs GUS compatible MIDI patch files. The documentation mentions the "public domain MIDIA patchset available from several ftp sites".

As explained on the 4Front Technologies web page http://www.opensound.com/softoss.html they can be downloaded from ftp://archive.cs.umbc.edu/pub/midia/instruments.tar.gz.

7. References

If you have a sound card that supports a CD-ROM or SCSI interface, the Linux SCSI HOWTO and the Linux CD-ROM HOWTO have additional information that may be useful to you.

The Sound Playing HOWTO describes how to play various types of sound and music files under Linux.

The Linux SoundBlaster AWE32/64 Mini-HOWTO describes how to get a SoundBlaster 32 or 64 card working under Linux.

Programming information is available from the 4Front Technologies web site at http://www.opensound.com/pguide.

The following FAQs are regularly posted to the Usenet newsgroup news.announce as well as being archived at ftp://rtfm.mit.edu/pub/usenet/news.answers:

- PCsoundcards/generic-faq (Generic PC Soundcard FAQ)
- PCsoundcards/soundcard-faq (comp.sys.ibm.pc.soundcard FAQ)
- PCsoundcards/gravis-ultrasound/faq (Gravis UltraSound FAQ)
- audio-fmts/part1 (Audio file format descriptions)
- audio-fmts/part2 (Audio file format descriptions)

The FAQs also list several product specific mailing lists and archive sites. The following Usenet news groups discuss sound and/or music related issues:

- alt.binaries.sounds.* (various groups for posting sound files)
- alt.binaries.multimedia (for posting Multimedia files)
- alt.sb.programmer (Soundblaster programming topics)
- comp.multimedia (Multimedia topics)
- comp.music (Computer music theory and research)
- comp.sys.ibm.pc.soundcard.* (various IBM PC sound card groups)

A web site dedicated to multimedia can be found at http://viswiz.gmd.de/MultimediaInfo/. Another good site for Linux MIDI and sound applications is http://sound.condorow.net/. Creative Labs has a web site at http://www.creaf.com/. MediaTrix has a web site at http://www.mediatrix.com/.

The Linux mailing list has a number of "channels" dedicated to different topics, including sound. To find out how to join, send a mail message with the word "help" as the message body to majordomo@vger.rutgers.edu. These mailing lists are not recommended for questions on sound card setup etc., they are intended for development related discussion.

As mentioned several times before, the kernel sound driver includes a number of Readme files containing

useful information about the sound card driver. These can typically be found in the directory /usr/src/linux/drivers/sound.

Information on OSS, the commercial sound driver for Linux and other Unix compatible operating systems, can be found at the 4Front Technologies web page at http://www.opensound.com/.

The *Linux Software Map* (LSM) is an invaluable reference for locating Linux software. Searching the LSM for keywords such as *sound* is a good way to identify applications related to sound hardware. The LSM can be found on various anonymous FTP sites, including ftp://metalab.unc.edu/pub/Linux/docs/LSM/ (formerly known as sunsite). There are also various web sites that maintain databases of Linux applications. One such site is http://www.freshmeat.net.

The Linux Documentation Project has produced several books on Linux, including *Linux Installation and Getting Started*. These are freely available by anonymous FTP from major Linux archive sites or can be purchased in hardcopy format.

Finally, a shameless plug: If you want to learn a lot more about multimedia under Linux (especially CD-ROM and sound card applications and programming), check out my book *Linux Multimedia Guide*, ISBN 1-56592-219-0, published by O'Reilly and Associates. As well as the original English version, French and Japanese translations are now in print. For details, call 800-998-9938 in North America or check the web page http://www.ora.com/catalog/multilinux/noframes.html or my home page http://www.pobox.com/~tranter.

The Linux Sound Playing HOWTO

Yoo C. Chung, wacko@laplace.snu.ac.kr

v1.6, 11 August 1998

This document lists applications for Linux that play various sound formats.

1. Introduction

- 1.1 Copyright of this document
- 1.2 Copyright of the listed applications
- 1.3 Where to get this document
- 1.4 Feedback
- 1.5 Acknowledgments

2. Playing Various Sound Formats

- 2.1 MIDI
- 2.2 Modules
- 2.3 MPEG audio streams
- 2.4 WAV
- 2.5 Other stuff

3. Other useful sound utilities

- 3.1 volume
- 3.2 Sound Studio
- 3.3 Tickle Music

4. References

1. Introduction

This is the Sound Playing HOWTO. It lists the many sound formats and the applications that can be used to play them. It also lists some hacks and advice on using these applications. There are also some other interesting applications related to sound not directly related to playback. However, this document does *not* describe how one can setup a Linux system for sound support. Refer to the Linux Sound HOWTO by Jeff Tranter for instructions on setting up a Linux system for sound support and the supported sound hardware.

This deals with normal user sound applications. That is, it is only concerned about what the average user needs to know on the application side of sound, not exotic stuff like speech synthesis, or hardware stuff which is dealt in the Sound HOWTO.

1.1 Copyright of this document

This document can be freely distributed and modified (I would appreciate it if I were notified of any modifications), as long as this copyright notice is preserved. However, it cannot be placed under any further restrictions, and a modified document must have the same copyright as this one. Also, credit must be given where due.

1.2 Copyright of the listed applications

If there is no mention of any copyright, then the application is under the GNU General Public License.

1.3 Where to get this document

The most recent official version of this document can be obtained from the Linux Documentation Project. The most recent unofficial version of this document can be obtained from http://laplace.snu.ac.kr/~wacko/howto/.

A Korean version of this document (very outdated) is available at http://laplace.snu.ac.kr/~wacko/howto/Sound-Playing-HOWTO.ks.

A Japanese version of this document is available at http://jf.gee.kyoto-u.ac.jp/JF/JF-ftp/euc/Sound-Playing-HOWTO.euc.

TOC

1.4 Feedback

I am not omniscient, and I don't use all the applications in here (a few I can't even try), so there are bound to be mistakes. Also, programs usually continuously evolve, so documentation tends to get out of date. Therefore, if you find anything wrong, please send me any corrections. Suggestions or additions to this document are welcome, too.

1.5 Acknowledgments

All the authors of the applications in this HOWTO. Also, Hannu Savolainen for the great sound driver and Linus Torvalds for the great underlying OS.

I'd also like to thank Raymond Nijssen (`raymond@es.ele.tue.nl`), Jeroen Rutten (`jeroen@es.ele.tue.nl`), Antonio Perez (`aperez@arrakis.es`), Ian Jackson (`ijackson@gnu.org`), and Peter Amstutz (`amstpi@freenet.tlh.fl.us`) for their information and help.

---

2. Playing Various Sound Formats

There are many kinds of sound formats (WAV, MIDI, MPEG etc.). Below, we list the various formats and the applications that can be used to play them.

2.1 MIDI

MIDI stands for Musical Instrument Device Interface. MIDI files usually have the extension `.mid`. They contain sequencing information, that is, information on when to play what instrument in what way, etc. Depending on your hardware (and maybe the software you use to play them), the sound might be awesome, or it might be downright crappy.

adagio

This package includes `mp` (a command-line MIDI file player) and `xmp` (an XView based MIDI file player, not to be confused with the module player also called `xmp`). You will need the SlingShot extensions to use `xmp`. It also contains other programs for playing Adagio scores.

If you have a GUS, `mp` can also play MOD files (see section Modules for more information on modules).

One little annoying bug (as of version 0.5 on some hardware) is that the sound breaks at the end. Namely, instead of ending the sound the way the MIDI file specifies, it ends by playing the note right before the last one in a long interval. It hasn't stopped me from using `mp`, but it might prevent someone from using it for `real' work. It also starts up relatively slowly.

The package does not mention any copyright (at least none that I can find), so I assume it can be freely redistributed and modified. (By a strict interpretation of copyright law, nothing gives one the right to do these things, but I somehow doubt that this was the intention of the author.)

It is a port of the CMU MIDI Toolkit to Linux (though there was enough added to make this questionable) by Greg Lee (`lee@uhunix.uhcc.hawaii.edu`).

It can be obtained from ftp://tsx-11.mit.edu/pub/linux/packages/sound/adagio05.tar.gz. The binaries included here are in a.out format (linked with ancient libraries), and the `xmp` binary segfaults in a X11R6 environment (XFree86 3.1.1, libc 4.7.2). The `mp` binary works fine in an `a.out` environment.

You will need a bit of hackery to compile it. Actually, it's not much of a hackery. All you have to do is to include the `-lfl` switch at the end of `SHROBJ` and `XMPOBJ` in the Makefile. This is to link in the `flex` library, which is not linked in by default. Then follow the installation instructions. And don't forget to have XView and the SlingShot extensions installed if you want to compile `xmp`.

TOC

TiMidity

Some people recommend this *experimental* program because of good sound quality (which is very true: it's much better than `mp` on a Sound Blaster 16, though it probably won't be much different on soundcards with wavetable synthesis like the GUS). However, it suffers from high CPU loads. It plays MIDI by first converting MIDI to WAV and then plays the WAV (you can also convert a MIDI file to a WAV file without playing if you want). This is the reason for its CPU intensive nature.

It also has an optional ncurses, SLang, Tcl/Tk or Motif interface.

You need Gravis Ultrasound patch files to use this. Look into the FAQ included with TiMidity for more information.

The author is Tuukka Toivonen (`tt@cgs.fi`).

The latest version of TiMidity can be found at the TiMidity home page. This page also contains a link to a small library of GUS patches.

playmidi

This is a MIDI player that plays to FM, GUS, and external MIDI. It is supposed to have a faster startup time compared to other MIDI players. It is also able to play Creative Music Files, Microsoft RIFF files, and large MIDI archives from games such as Ultima 7.

It has an X interface and a SVGA interface. It also has an option for real time playback with tracking all the notes on each channel and the current playback clock (included automatically with `xplaymidi` and `splaymidi`).

You should do something like

```
$ splaymidi foo.mid; stty sane
```

if you are going to use the SVGA interface, since it doesn't reset the terminal tty mode properly. The SVGA interface may be removed in the near future.

It was written by Nathan Laredo (`laredo@gnu.org` or `laredo@ix.netcom.com`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/playmidi-2.3.tar.gz.

2.2 Modules

Modules (in computer music) are digital music files, made up of a set of samples and sequencing information, telling the player when to play which sample (instrument) on which track at what pitch, optionally performing an effect, like vibrato for example.

An advantage it has over MIDI is that it can include almost any kind of sound (including human voices). Another is that it sounds just about the same on any platform, because the samples are in the module. A

TOC

disadvantage it has is that it has a much larger file size compared to MIDI. Another one is that it has real standard format (the only `real' one is the ProTracker, which many modules aren't quite compati with). It originated on the Amiga.

The most common format has the extension `.mod`. There are many other extensions depending on what format they are in.

tracker

This very portable program (it has been ported to many platforms) plays Soundtracker and Protracker music modules. It uses 16 bit stereo output, and I consider the quality to be very good. If you need a simple way to reduce CPU load use the `-mono` option.

This is a giftware program (quoting the author). It is by Marc Espie (`Marc.Espie@ens.fr`).

A version of this with the Makefile already tweaked for Linux can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/tracker-4.3-linux.tar.gz.

gmod

This is a music module player for the Gravis Ultrasound card. 4/6/8 channel MOD, 8 channel 669, MultiTracker (MTM), UltraTracker (ULT), FastTracker (XM), and ScreamTracker III (S3M) are the supported formats.

It requires a version 3.0 or later sound driver. And a GUS, of course. You may need to modify the kernel to make volume control work the way you want.

This has an X interface. It uses the QT toolkit (needs version 0.99 or greater). Check the QT toolkit homepage for information on QT.

This can be freely distributed. It was originally written by Hannu Savolainen, and now maintained by Andrew J. Robinson (`robinson@cnj.digex.net`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/gmod-3.1.tar.gz.

MikMod

This portable module player plays XM, ULT, STM, S3M, MTM, MOD and UNI formats. (The UNI format is an internal format used by MikMod.) It has support for zipped module files. It uses 16 bit stereo for the sound output. Use the `-m` option (for mono output) if you need a simple way to lower the CPU load.

The Unix version can either use ncurses or Tcl/Tk for its interface. It can also be used as a library, not just an independent program.

It was originally written by Jean-Paul Mikkers (`mikmak@via.nl`). It is now maintained by Jake Stine (`dracoirs@epix.net`). This is shareware that has to be registered if you want to use it commercially. You also need permission to redistribute it commercially (non-commercial redistribution

does not need such permission).

This can be found at the MikMod home page.

xmp

This is a module player (not to be confused with Adagio's `xmp`) which can play MOD, S3M, MTM, PTM, PTR, STM, 669, and XM modules (other formats are also supported, but still experimental or incomplete). If you have soundcards with wavetable synthesis (GUS or SoundBlaster 32AWE), then you can use this feature of the soundcard to lower the load on the CPU. It also supports compressed modules.

An X frontend to `xmp` is also available.

This was written by Claudio Matsuoka (`claudio@pos.inf.ufpr.br`) and Hipolito Carraro Jr.

This can found at the xmp home page.

s3mod

This plays 4/6/8 track MOD modules and Scream Tracker 3 modules. It uses 8 bit mono output with a sampling rate of 22000 Hz by default. You can use the option `-s` to enable stereo, `-b` to enable 16 bit output, and `-f` to set the sampling frequency. However, the sound output is worse than tracker (some noise), so I recommend using `tracker` instead of `s3mod` for playing ordinary MOD files (unless you have an underpowered machine). It has a much smaller CPU load compared to tracker.

It is copyrighted by Daniel Marks and David Jeske (`jeske@uiuc.edu`), but you can do anything you want with it (except that you can't claim you wrote it).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/s3mod-v1.09.tar.gz.

mod

This *beta* program plays MODs (15/31-instrument, up to 32 voices), MTMs, ULTs and S3Ms on the Gravis Ultrasound card. It can also use packed modules if you have `gzip`, `lharc`, `unzip`, and `unarj` installed. It cannot play Powerpacked modules or modules packed with some Amiga composers ("PACK" signature).

This requires at least version 3.0 of the sound driver. It *won't* work with the 2.90-2 or earlier version of the sound driver. The text interface requires ncurses. There is also an X interface included, which uses Tcl/Tk.

It was written by Mikael Nordqvist (`mech@df.lth.se` or `d91mn@efd.lth.se`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/mod-v0.81.tgz.

TOC

nspmod

This is an *alpha* module player which can play MTM, S3M, and MOD modules. It is intended to be a module player for soundcards without a DSP (not to be confused with what Creative Labs calls a DSP). It has a CPU load somewhat similar compared to `tracker`.

It has a feature which lets modules loop if they want to. The number of loops can be limited by the `-l` option. It uses only 8 bit sound output (as of version 0.1).

This was written by Toru Egashira (`toru@jms.jeton.or.jp`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/nspmod-0.1.tar.gz.

yampmod

This *alpha* program was designed to play 4-channel modules using the minimum of CPU resources. It was *not* designed to produce high quality sound. So the only sound output it produces is 22 kHz mono output. Also, the output isn't as clean as it should be, reflecting its alpha status.

It was written by David Groves (`djg@djghome.demon.co.uk`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/yampmod-0.1.tar.gz.

2.3 MPEG audio streams

MPEG is a standard specifying the coding of video and the associated audio for digital storage. MPEG is usually associated with video, but the audio part of the standard can be used separately. The audio part of the MPEG standard defines three layers, layer I, II, and III. Players that can decode higher layers can also decode lower layers (e.g. layer III players can play layer II files). Layer I MPEG audio files usually have the extension `.mpg` (so if there is a file with this extension that can't be played by a MPEG video player, it's probably an audio stream), layer II usually have the extension `.mp2`, and layer III usually have the extension `.mp3`. The audio compression is pretty good. A two megabyte layer II MPEG audio file will probably take up 25 megabytes for a raw PCM sample file with the same quality.

mpg123

This *beta* program is an efficient MPEG audio stream player, which has support for layers I, II, and III. It is based on code from many sources. It is able to play in real time streams that are read by HTTP (i.e. one can play an MPEG audio stream directly over the World Wide Web).

The main author is Michael Hipp (`Michael.Hipp@student.uni-tuebingen.de`). It may be used and distributed in unmodified form freely for non-commercial purposes. Inclusion in a collection of free software (such as CD-ROM images of FTP servers) is explicitly allowed.

The latest version can be obtained from the mpg123 homepage.

TOC

maplay 1.2

This MPEG audio stream player only has support for layer I and layer II streams, and lacks support for layer III streams. It supports 16 bit sound cards on Linux.

It is pretty CPU intensive, taking up to about 55% CPU time on a 60MHz Pentium. The output is intolerable on a 66MHz 486 because the CPU just can't catch up with the sound. If this happens to you, try playing only one side of the audio stream (with the `-l` or `-r` option), instead of the default stereo.

A slight change in one of the files may be necessary in order to compile it. Namely, you may need to add the following line to the beginning of the file `configuration.sh`.

```
#! /bin/sh
```

The author is Tobias Bading (`bading@cs.tu-berlin.de`). `maplay` 1.2 can be obtained from ftp://ftp.cs.tu-berlin.de/pub/misc/maplay1.2/maplay1_2.tar.

maplay 1.3b

This is an unofficial modification (i.e. not by the original author) of `maplay` 1.2, so that it can run with a much lower load on the CPU. It accomplishes this mainly by making u-law output actually work on other platforms besides the SPARC. Note that it uses u-law output by default, so the sound quality is lower.

The modifications were made by Orlando Andico (`orly@gibson.eee.upd.edu.ph`).

This can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/maplay-1.3b-Linux.tar.gz.

maplay3

This is another derivative of `maplay` 1.2. It adds support for MPEG Layer 3 audio streams. Currently it seems to have some bugs in its playback (you may hear some screeching noises). You may have to twiddle with the options to solve this.

The modifications were made by Timo Jantunen (`timo.jantunen@hut.fi` or `jeti@cc.hut.fi`). It says that it can be used freely, but making money off of it is not allowed. However, I'm not entirely sure about the validity of this copyright, since the original `maplay` is under the GNU General Public License, which does not allow derivative works to have a different copyright.

This can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/maplay3.tar.gz.

splay

This *beta* player is another derivative of `maplay` 1.2 (actually, it is a derivative of `maplay` 1.2+, which is a MS Windows only derivative of `maplay` 1.2). It adds support for MPEG Layer 3 audio streams. It is also able to play WAV files. It can also play audio streams received over an HTTP connection.

Another feature of `splay` is that it can be used as a library (under the LGPL), so that it can be used in

TOC

other programs. It also tries to improve performance by using threading (you need `pthread` to use this feature) and a little inline assembly.

`splay` uses a command line interface and an optional X interface (which uses QT).

If after compiling it doesn't work (e.g. it segmentation faults), try compiling it again without threading.

This is by Jung Woo-jae (`jwj95@eve.kaist.ac.kr`).

It can be obtained from splay's home page.

Sajber Jukebox

This program is a MPEG audio player with a graphical user interface. It is based on `splay`, so it includes support for MPEG audio layers up to III. It is also able to play MPEG audio streams in real time with the stream being fed by HTTP. It is also easy to configure.

It uses the QT toolkit (at least version 1.2 is required). It also uses the LinuxThreads library (the included binary only works with version 0.5).

The author is Joel Lindholm (`wizball@kewl.campus.luth.se`).

The latest version can be obtained from ftp://kewl.campus.luth.se/pub/jukebox.

amp

This *beta* MPEG audio player only has support for MPEG Layer 3 audio streams. It is able to play directly to the soundcard, and it can output to raw PCM or WAV files. This also gives quite a load on the CPU (about 60% on a 133MHz Pentium).

This was written by Tomislav Uzelac (`tuzelac@rasip.fer.hr`). It can be freely used and distributed, as long as it is not sold commercially without permission (including it in CD-ROMs that contain free software is explicitly permitted, though).

It can be obtained from ftp://ftp.rasip.fer.hr/pub/mpeg/amp-0.7.3.tgz.

XAudio

This *alpha* library was written to be a fast implementation of an MPEG audio decoding library to be used by various GUI front-ends. It supports MPEG audio layers I, II, and III. It is capable of random access to bitstreams. A command-line interface is included. A Motif (Lesstif) front-end is also included in the Linux version.

This is by Gilles Boccon-Gibod, Alain Jobart and others. The front-ends to the libary can be freely downloaded. The library itself must be licensed to be used (a source and binary license is available).

The front-ends to the library can be obtained from the XAudio home page.

Layer 3 Shareware Encoder/Decoder

This is actually a converter that converts MPEG Layer 3 audio streams to WAV, AIFF, SND, AIFC, or just raw PCM sample files. The Linux version does not directly output the sound to the soundcard. One has to first convert it to some other format.

However, when you try to play a converted file using `sox`, you'll probably just get noise because the word order in the PCM samples is not right (at least on Intel platforms). You need to give `sox` the option `-x` to solve this problem. But there are some players that don't have to be told that the word order is wrong, so you might not have to worry about this.

If you have a really fast computer (probably at least a 100Mhz Pentium), then you can try to play MPEG Layer 3 streams directly without having to first convert the audio file to another format like in the following example (this example assumes that you're using `sox` and playing a 44.1 kHz stereo sample).

```
$ l3dec foo.mp3 -sto | play -t raw -x -u -w -c 2 -r 44100 -
```

The number after `-r` is the sample rate of the audio stream, and the number after `-c` depends on whether it is mono or stereo (or even quad). If this looks too complicated, you can use something like a shell script or an alias.

This is shareware copyrighted by Fraunhofer-IIS. A demo version for Linux on x86 systems can be obtained from ftp://ftp.fhg.de/pub/layer3. The demo version only converts layer III audio streams.

X11Amp

This *beta* software is an MPEG audio stream player with a graphical interface, similar to the one used by the Windows program `winamp`.

There is no copyright mentioned anywhere (I assume that it can be freely used for personal use). It is maintained by Mikael Alm (`psy@x11amp.bz.nu`), Thomas Nilsson (`fatal@x11amp.bz.nu`), and Olle Hallnas (`crocodile@x11amp.bz.nu`).

It can be obtained from X11Amp's homepage. Only binaries for Intel Linux and FreeBSD are made available here.

2.4 WAV

Quote from the `sox` man page:

> These appear to be very similar to IFF files, but not the same. They are the native sound file format of Windows 3.1. Obviously, Windows 3.1 is of such incredible importance to the computer industry that it just had to have its own sound file format.

These usually have the extension `.wav`.

Also see section sox and bplay for other WAV players besides the ones listed here.

TOC

wavplay

This program supports playing and recording with the WAV format. It uses locking so that only one sound may be played at a time. Its locking capabilities can also be used separately from its sound playing capabilities.

In addition to a command-line interface, it also has a Motif interface, which can be used with Lesstif.

It was originally written by Andre Fuechsel (`af1@irz.inf.tu-dresden.de`), but was evolved to the point of being completely rewritten by Warren W. Gay (`bx249@freenet.toronto.on.ca` or `wwg@ica.net`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/wavplay-1.0.tar.gz.

2.5 Other stuff

This section lists stuff that play sound formats that don't deserve a separate section (i.e. formats that have only one player available), or players that play more than one format.

sox

This program is actually a converter, that is, it converts one sound format to another. However, some versions of `sox`, when invoked as `play`, plays the sound (the `play` application in the Sound HOWTO probably refers to this). It supports raw (no header) binary and textual data, IRCAM Sound Files, Sound Blaster `.voc`, SPARC `.au` (w/header), Mac HCOM, PC/DOS `.sou`, Sndtool, and Sounder, NeXT `.snd`, Windows 3.1 RIFF/WAV, Turtle Beach `.smp`, CD-R, and Apple/SGI AIFF and 8SVX formats

Since somewhere in the 1.3.6x kernels, you might have to make a small change in one file to make it play the sound directly. Namely, you may have to change line 179 in `sbdsp.c` from

```
if (abuf_size < 4096 || abuf_size > 65536) {
```

to

```
if (abuf_size < 1 || abuf_size > 65536) {
```

But then again, you may not have to do this. But doing this won't break anything.

It is written and copyrighted by many people, and can be used for any purpose.

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/convert/Lsox-linux.tar.gz.

A more recent version by Chris Bagwell (`cbagwell@sprynet.com`) (based on the latest gamma version of the original `sox`, and includes the above fix) can be obtained from http://home.sprynet.com/sprynet/cbagwell/projects.html. In addition, this version supports MS ADPCM

and IMA ADPCM WAV formats.

bplay

This *beta* program plays raw audio, WAV, and VOC files. It's also able to record to these files. It uses a variety of techniques to get the highest speed possible so that it can run acceptably even on slow machines. One of these techniques require that the installed programs be setuid root. The paranoid hoping to use this may want to use the Debian package by Ian Jackson (`ijackson@gnu.org`), which disables the feature that needs the setuid bit.

The author is David Monro (`davidm@gh.cs.usyd.edu.au`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/bplay-0.96.tar.gz.

SIDPLAY

This program emulates the Sound Interface Device chip (MOS 6581, commonly called SID) and the Micro Processor Unit (MOS 6510) of the Commodore 64. Therefore it is able to load and execute C64 machine code programs which produce music or sound. In general these are independent fragments of code and data which have been ripped from games and demonstration programs and have been transferred directly from the C64.

It uses a command line interface by default. There are also Tk and QT interfaces available separately from the main package.

It is maintained by Michael Schwendt (`sidplay@geocities.com`).

It can be obtained from SIDPLAY's home page.

RealAudio Player

This lets you listen to sound, which is stored in a proprietary format, in real time over the Internet without downloading the whole sound file first. It could be used stand alone, but it is really intended to be used along with a web browser (the explicitly supported ones are Mosaic and Netscape). It cannot be used without X (you wouldn't be able to get it working with Lynx in a text console).

However, there exists a hack which allows one to run the RealAudio player from the text console. It requires the X virtual frame buffer (Xvfb) server to work. This hack can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/traplayer-0.5.tar.gz.

This is by Progressive Networks, Inc. This cannot be redistributed, modified etc. Look at the license for exact details on what you can do. It can be obtained by registering with no cost at the RealAudio home page.

cat

One might think what `cat`, the sometimes overused concatenating utility, has to do with playing sounds. I'll show a use of it through an example.

```
$ cat sample.voc > /dev/dsp
$ cat sample.wav > /dev/dsp
$ cat sample.au > /dev/audio
```

Doing a `cat` of an `.au` file to `/dev/audio` will usually work, and if you're lucky enough that the file has the correct byte order (for your platform) etc., a `cat` of a sound file that uses PCM samples (like `.wav` or `.voc`) to `/dev/dsp` might even sound right.

This isn't a totally useless use of `cat`. It might be useful, for example, if you have a sound file that none of your programs recognize, and you know that it uses PCM samples, then you might be able to get a very approximate idea on how it sounds like this way (if you're lucky).

---

3. Other useful sound utilities

This section has nothing to do with the actual playing of sound files. Rather, it is a collection of some sound utilities that one might find useful.

3.1 volume

This is a simple command line interface for controlling the volume (what else could it be?). It also has a separate program with a Tcl/Tk interface included in the package for controlling the volume and playing `.au` sound files. A very simple Tcl/Tk CD player is also included.

This is Freeware and it is written by Sam Lantinga (`slouken@cs.ucdavis.edu`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/soundcard/volume-2.1.tar.gz.

3.2 Sound Studio

This is a Tcl/Tk application that supports playback, recording, and editing of digital sound using `sox`. It includes `sox` in the distribution to avoid compatibility problems.

This was written by Paul Sharpe and N. J. Bailey (`N.J.Bailey@leeds.ac.uk`). It may be freely used and redistributed if a postcard is sent.

It can be found at Sound Studio's home page.

3.3 Tickle Music

This *beta* Tcl/Tk program is a music file browser that allows you to play various sound formats as long as an appropriate program to play it is on your system. By default `gmod` is used for playing MOD files and `mp` for playing MIDI files (you can change the source to use other programs).

It is written and copyrighted by Shannon Hendrix (`shendrix@pcs.cnu.edu` or `shendrix@escape.widomaker.com`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/tmusic-1.0.tar.gz.

4. References

1. The documentation included with the applications in this document.
2. The Linux Sound HOWTO. It can be found at the Linux Documentation Project.
3. Linux MIDI and Sound Applications
4. Programmer's Guide to OSS
5. SoX home page

Spanish Linux HOWTO

Gonzalo.Garcia-Agullo@jrc.es.

v0.1, 20 August 1996

This is the very first issue of the Spanish Linux HOWTO. The target audience of this document is the growing family of Linux users in Spain and Latin America, so this How-To is of course in Spanish. Some non-Spanish speakers have asked for information on Linux in the Spanish environment, for developing software with Spanish support or just to get in touch with the Linux community in Spain. You're welcome in any case...

1. Introducción

- 1.1 Empecemos por el principio...
- 1.2 ¿Qué es linux?
- 1.3 ¿Cómo conseguirlo?
- 1.4 ¡Socorro! ¿Dónde están los manuales?

2. Distribuciones

3. Ayuda a la instalación de la distribución Slackware

- 3.1 Instalación con UMSDOS
- 3.2 Instalación en partición nativa de Linux.

4. *Españolización* de Linux

- 4.1 Consola
- 4.2 Shell bash
- 4.3 Shell tcsh
- 4.4 Algunas aplicaciones

- 4.5 Xwindow
- 4.6 Emulador de DOS (DOSEMU)
- 4.7 Correo electrónico

5. Conectividad de Linux

- 5.1 Conectividad casera
- 5.2 Servidor de terminales
- 5.3 Red PPP
- 5.4 Servidor de modems
- 5.5 Conexion directa a Internet
- 5.6 Infovía: acceso a Internet desde casa

6. Direcciones útiles

- 6.1 Direcciones electrónicas
- 6.2 Direcciones de Firmas comerciales
- 6.3 Bibliografía.

7. Copyright

---

1. Introducción

1.1 Empecemos por el principio...

Este documento pretende ser el punto de entrada de los hispanohablantes al mundo Linux, intentando abarcar tanto a los principiantes de este apasionante sistema como a los ya iniciados que buscan información puntual. Se tratarán de forma genérica todos los temas que puedan interesar al usuario de Linux, desarrollándolos en castellano o dando referencias a otras fuentes de información en nuestro idioma, así como orientación cuando tenga que acudir a otras HOWTO y en general a cualquier documento en inglés.

Se distribuye integrado con el Linux Documentation Project en todos los foros de información Linux en Internet. Se actualiza periódicamente, pudiendo encontrar la última versión en Linux/España

Notas del autor: Aunque puede ser interesante de por sí, traducir sin más información ya editada en inglés no es el objetivo de este documento, sino más bien servir de base al usuario hispanohablante y divulgar el Linux en nuestro idioma. Una ambición de quien escribe es, desde estas líneas, poner su grano de arena para coordinar los esfuerzos (valiosos pero muy diseminados por la red) de tanto Linuxero como ya hay. Otro objetivo menos pretencioso de este documento es complementar todas las demas Howto's en aquellos aspectos concretos que afecten a los usuarios hispanos (configuración del sistema y de las aplicaciones para trabajar en nuestro idioma, teclados, correo electrónico en español, etc). Un tercer objetivo será servir de agenda actualizada de direcciones españolas dedicadas a Linux, tanto para difusión de las distribuciones originales de software Linux como de desarrollos propios en español que se vayan publicando. Vaya desde aquí mi agradecimiento en esta primera versión para Antonio L. Delgado, compañero de la Facultad de Informática de Sevilla, por prestarse a echar una mano redactando esta Howto en las calurosas tardes de Sevilla.

Cualquier comentario, sugerencia para nuevos contenidos, rectificación de errores, crítica, etc. es no sólo bienvenido sino también esperado. Contacta por correo electrónico para todo lo que quieras.

1.2 ¿Qué es linux?

Linux es una implementación del sistema operativo UNIX (uno más de entre los numerosos clónicos del histórico Unix), pero con la originalidad de ser gratuito y a la vez muy potente, que sale muy bien parado (no pocas veces victorioso) al compararlo con las versiones comerciales para sistemas de mayor envergadura y por tanto teóricamente superiores. Comenzó como proyecto personal del -entonces estudiante- Linus Torvalds, quien tomó como punto de partida otro viejo conocido, el Minix de Andy. S. Tanenbaum (profesor de sistemas operativos que creó su propio sistema operativo Unix en PCs XT para usarlo en su docencia). Actualmente Linus lo sigue desarrollando, pero a estas alturas el principal autor es la red Internet, desde donde una gigantesca familia de programadores y usuarios aportan diariamente su tiempo aumentando sus prestaciones y dando información y soporte técnico mútuo. La versión

original -y aun predominante- comenzó para PCs compatibles (Intel 386 y superiores), existiendo también en desarrollo versiones para prácticamente todo tipo de plataformas: PowerPC, Sparc, Alpha, Mips, etc. De todas ellas la más reciente en este momento es la versión para PowerMac (el PowerPC de Apple) basada en el microkernel Mach 3.0 y de la que ya hay una distribución para desarrolladores avalada directamente por Apple y OSF pero conservando el espíritu (gratuito, de libre distribución, etc) de la version original. Un servidor la acaba de probar hace unos días y se ha llevado una grata sorpresa (aún tendrá muuuchos fallos, pero para ser una primerísima versión y el poco tiempo que lleva en marcha, ha avanzado más de lo que me esperaba).

1.3 ¿Cómo conseguirlo?

La primera fuente para conseguir el sistema Linux es la propia red Internet, y es donde estarán siempre las últimas versiones y las aplicaciones más actualizadas en muchos servidores de FTP anónimo. Otra vía muy frecuente, de interés para principiantes y para quienes no deseen o no puedan permitirse copiar tanta cantidad de información a traves de la red, es mediante las versiones comercializadas en CDROM. Hay empresas que se dedican a elaborar CDROMs de bajo coste con recopilaciones de software, manuales, etc. El corazón del sistema es el mismo, aunque pueden tener externamente presentaciones y formas distintas de instalación. Hay revistas especializadas que también suelen incluir CDs con alguna versión de Linux. Hacia el final de este documento se relacionan diversas fuentes de Linux, tanto servidores públicos en la red como direcciones de empresas que lo comercializan.

Una ventaja (para muchos usuarios termina siendo un inconveniente) es la gran rapidez con la que se se desarrolla Linux. Constantemente llegan a los principales servidores Linux en la red actualizaciones del núcleo del sistema, de aplicaciones, utilidades, manuales y documentación, etc. Es bueno estar al día, seguir con atención su evolución y aprovechar las mejoras que se incorporen, pero en la mayoría de los casos no vale la pena estar reinstalando software por el simple hecho de ser una nueva versión, sino que hay que ser un poco selectivos, al menos con el software. En el caso de los manuales, How-To's, grupos de noticias y/o listas de correo sí que vale la pena estar "a la última", sobre todo porque es allí donde nos sacarán de apuros cuando agotemos nuestros propios recursos...

1.4 ¡Socorro! ¿Dónde están los manuales?

Como ocurre en todas las versiones de Unix, el primer sitio donde mirar cuando tenga una duda concreta sobre tal comando, fichero de configuración, etc. es la orden '`man`', que incluye la ayuda de referencia de Unix. Pero sólo es útil en ese ámbito, cuando ya se sabe más o menos lo se que busca, y sólo se necesita aclarar dudas concretas. Para todo lo demás la mejor fuente de información es, cómo no, la propia red. En ella están disponibles tanto libros completos (de los que algunas editoriales especializadas han publicado versiones en papel) como la colección "Linux HOWTO", de la que este documento forma parte. Hay una HOWTO para prácticamente cualquier tema. Su objetivo es cubrir, mediante manuales breves, concisos y específicos, cualquier duda que pueda surgir. Se actualizan permanentemente, y se distribuyen a través del denominado "Linux Documentation Project" (LDP) en Internet. La misma información se publica en muy diferentes formatos, orientados tanto a la búsqueda y consulta en línea como a su lectura convencional. Todas las versiones de Linux incluyen estos manuales (al menos la última edición disponible en el momento de recopilar el CDROM correspondiente), y al final de este

documento se aportan algunas de las muchas direcciones para consultar en la red.

2. Distribuciones

Extracto de la `Distributions-HOWTO` sobre distintas versiones de Linux. Acuda a ella para ampliar información

El único elemento común a todas las versiones Linux es su kernel -el núcleo del sistema operativo-, que se desarrolla de forma coordinada y con actualizaciones sistemáticas. Sin embargo todo sistema operativo necesita, junto al núcleo del sistema, todo un conjunto de utilidades y herramientas de instalación, configuración y uso. Ahí juegan su papel las diferentes distribuciones: algunos particulares, entidades y empresas se dedican a hacer determinadas recopilaciones de software que ellos mismos preparan para que sean facilmente instalables y configurables. Todas ellas facilitan el software junto a su código fuente, pero la flexible licencia GNU a la que se acojen permite tanto ofrecerlas gratuitamente como distribuirlas por canales comerciales (lo que se paga es el trabajo de recopilación, el software de cosecha propia que pueda aportar, una presentación más elaborada, gastos de distribución y soporte técnico al usuario).

Algunas de las distribuciones más conocidas son:

Slackware

Esta distribucion es quizas de las más extendidas en todo el mundo y la más conocida en España, ya que es la que han divulgado las publicaciones del sector. Incluye con creces todo el software que cualquier usuario medio o avanzado pueda necesitar, y su método de instalación es asequible aunque lejos de las típicas y vistosas instalaciones a las que el usuario neófito esta acostumbrado en los sistemas comerciales. Tiene diversos derivados comercializados bajo diferentes nombres (por ej. "Slackware Pro") que mejoran la instalación e incluyen manuales y bibliografía en un único lote. Esta en todos los FTPs principales, siendo http://www.cdrom.com/ su distribuidor de origen.

Red Hat

Creada por Red Hat Software, en Connecticut, EE.UU. Una de sus ventajas es el atractivo sistema de instalación (en modo gráfico) y el cómodo mantenimiento de componentes de software, lo que facilita enormemente las tan frecuentes actualizaciones. Se puede obtener tanto gratuitamente en la red como adquiriendo el CDROM correspondiente. Otras empresas comercializan también sistemas basados en Red Hat, como Caldera Inc. y Pacific Hi-Tech. Aún poco conocida en Espa–a pero pujante, sobre todo para principiantes. Sus creadores estan en http://www.redhat.com. El mencionado sistema de gestión de componentes de software es obra suya, pero lo han ofrecido con carácter abierto y gratuito a los demás desarrolladores bajo la licencia de GNU, por lo que es previsible que en el futuro otros muchos asuman este sistema en sus propias distribuciones, lo que facilitará enormemente las actualizaciones.

Debian

La Free Software Fundation (FSF) es bien conocida entre los acérrimos usuarios de software gratuito para Unix. Son los creadores del sistema **GNU**, su futuro Unix gratuito. Ya hay mucho

material pero no un sistema operativo completo, asi que mientras tanto ofrecen un Unix integr
por el nucleo de Linux y el software de GNU. http://www.debian.org.

3. Ayuda a la instalación de la distribución Slackware

Este es un extracto de la `Installation-HOWTO`, orientada fundamentalmente a la instalación de Linux Slackware. Mucha de la información sobre instalacion de Slackware es aplicable a cualquier versión de Linux, pero no necesariamente al pie de la letra.

Aparte de otras muchas variantes que se puedan considerar, a la hora de instalar Linux en nuestro disco duro hay básicamente dos formas: UMSDOS y Partición nativa. Cada una tiene sus ventajas e inconvenientes, pero como se verá es recomendable que todos los usuarios terminen usando el segundo método ya que es el más eficiente y el que mejor aprovecha las ventajas de Linux.

3.1 Instalación con UMSDOS

Consiste en aprovechar el espacio de disco que queda libre en los PCs con MSDOS, sin necesidad de tocar las particiones de disco. Linux es un sistema operativo, y como tal totalmente independiente, no requiriendo por tanto ningun otro sistema operativo en su ordenador (aunque puede convivir perfectamente con ellos). Sin embargo es un parque muy amplio el de usuarios que ya tienen instalado MSDOS y/o MS-Windows y que, al no considerase expertos, no se atreven en un primer momento a reparticionar el disco para sustituir total o parcialmente a su sistema actual. Para ellos, o simplemente para quien quiera ver y probar Linux sin ser tan drasticos formateando discos duros, la solución es UMSDOS. Permite crear, a partir de un directorio de MSDOS, toda la estructura que tendría un disco nativo de Linux. Cuando arranque el sistema, Linux emulará su disco a partir de la información en ese directorio. Es totalmente funcional, pero puede ser ligeramente más lento en ordenadores de gama media-baja y no aprovecha la gestión de disco nativo. Una gran ventaja es que si se libera disco en Linux, implícitamente estamos creando espacio que será aprovechable cuando arranquemos MSDOS, y viceversa. Además, el día que queramos deshacernos de Linux (¡poco probable!) o que simplemente nos decidamos a hacer una instalación completa, desinstalar ese Linux es tan sencillo como simplemente borrar sin contemplaciones el directorio `C:\LINUX` que creó desde MSDOS.

3.2 Instalación en partición nativa de Linux.

Salvo la fase inicial en la que se preparan las particiones nativas o el UMSDOS, el procedimiento restante de instalación es básicamente el mismo. Ya que puede ser el que provoque más dudas, centraremos estos consejos de instalación en la instalación con partición nativa. Pero recuerde que

---

```
ESTO NO ES UNA GUIA DE INSTALACION, SINO UN COMPLEMENTO A LAS
INTRUCCIONES DE INSTALACION DE SU VERSION DE LINUX.

LEA LA INSTALLATION-HOWTO Y/O EL MANUAL QUE ACOMPAñA A SU DISTRIBUCION
ANTES DE COMENZAR, O DE LO CONTRARIO PUEDE FALLAR LA INSTALACION E
INCLUSO INUTILIZAR EL SOFTWARE QUE PUEDA TENER YA INSTALADO EN SU
ORDENADOR.

LA DISTRIBUCION TOMADA COMO BASE ES LINUX SLACKWARE 3.0. EL PROCEDIMIENTO
```

```
PUEDE VARIAR DE UNAS VERSIONES A OTRAS. EN CASO DE DUDA ACUDA A LOS
MANUALES DE SU SISTEMA O CONSULTE A LOS CANALES DE AYUDA Y SOPORTE LI
HABITUALES (EN INTERNET O A SU DISTRIBUIDOR)

LOS AUTORES DE ESTA DOCUMENTACION NO SE RESPONSABILIZAN DE LOS POSIBLES
DESASTRES QUE PUEDA OCASIONAR. SI DECIDE CONTINUAR ES BAJO SU EXCLUSIVA
RESPONSABILIDAD. (Nos gusta dormir tranquilos ;^) )
```

Preliminares

En esta instalación es necesario que LINUX disponga de una partición distinta a la de MSDOS para trabajar. Si ya tiene algún sistema instalado, lo más frecuente es que sea una única partición DOS que ocupe todo el disco, por lo que reparticionar supone eliminar esa partición para volver a crearla asignándole menos disco para hacer sitio a Linux (o para dárselo totalmente, según la instalación escogida). Eso supone perder como norma general todos los datos en las particiones ya existentes.

Existe, sin embargo, una utilidad anexa a Linux que corre sobre MSDOS y que permite "encoger" una partición de DOS siempre que la unidad tenga espacio libre y haya sido totalmente defragmentada (con la utilidad "DEFRAG" de MSDOS 6.x o similares). Esto le permitirá crear nuevas particiones conservando sus datos en la partición MSDOS actual. Esta utilidad se llama `"fips"` y la encontrará en su servidor habitual en la red junto a otras utilidades de MSDOS para ayudar a la instalación de Linux.

LEA LA DOCUMENTACION DE FIPS ANTES DE UTILIZARLO.

Primero debe liberar en MSDOS el espacio que pretenda reservar para la nueva partición de Linux (y para el espacio de swap, como después se comentará). Después debe defragmentar la partición de DOS con la utilidad "defrag" que acompaña a las últimas versiones de DOS, o bien sus equivalentes de PCTools o Norton. Ahora puede correr el programa fips y crear una nueva partición con el espacio liberado. Lo habitual será crear dos particiones: una *Linux native* para el sistema, y una *Linux swap* con por ej. el doble que de RAM, pero en caso de necesidad incluso puede ahorrarse una partición específica para swap, y "robar" ese espacio de disco bien a MSDOS o al propio Linux.

Formateo de la partición.

Una vez añadida la nueva partición que destinaremos a Linux, hay que preparar la siguiente etapa, el arranque de Linux, ya que salvo el apartado anterior en el que salvaguardamos la instalación previa de MSDOS, toda la instalación se hará desde el propio Linux. Para arrancar necesitaremos dos disquetes (en algunas distribuciones como RedHat pueden ser tres), donde grabaremos tanto el núcleo del sistema operativo como los primeros programas con los que poder empezar a instalar. El primer disquete es el que etiquetaremos como "boot disk", y es donde irá el núcleo del sistema con el que arrancar. Como Linux es casi un sistema "a la carta", que puede llevar en el núcleo más o menos funcionalidades según nuestras necesidades, deberemos escoger entre los núcleos precompilados aquel que se adapte mejor a nuestro caso. En el futuro será recomendable que incluso recompilemos nuestro propio núcleo para personalizarlo aún más, añadir lo que falte y simplificar lo que sobre en nuestro caso. El siguiente disquete es el "root disk" o "ramdisk", que incluye el software mínimo que cargaremos en memoria para la instalación.

En la distribución de Linux tendremos una serie de ficheros que constituyen "imágenes" de esos disquetes. Es decir, una vez elegidas qué diquetes necesitaremos, los crearemos a partir de esas imágenes, bien desde otro sistema Linux ya funcionando (o incluso de cualquier Unix en general), bien desde MSDOS con otra utilidad complementaria de "fips": "rawrite".

Una vez hecho esto ya puedes arrancar con tus discos. Introduce el "boot disk", arranca el ordenador y sigue las instrucciones. Cuando finalice la fase de arranque Slackware te ofrecerá un símbolo para hacer "login" (entrar en el sistema), mientras que RedHat arrancará directamente la aplicación de instalación. Utiliza el nombre "root" como nombre de usuario (es el super-usuario que administra el sistema). La primera vez no necesitarás clave alguna para entrar.

Ya estas en el prompt de Linux, desde donde puedes crear tus particiones y tu espacio de swap (es una zona especial de disco que Linux usa como memoria virtual que complementa a la memoria RAM del ordenador). Al igual que en MSDOS, el comando para gestionar particiones se llama "fdisk". En algunas versiones hay también una "cfdisk" con un interfaz más asequible, aunque iguales en lo básico.

Si posees una partición de MSDOS, no te asustes, sigue adelante, no la vas a perder simplemente por entrar en fdisk, pero sí que debes tener precaución al añadir la partición linux sobre el espacio que le habíamos quitado a MSDOS, para no afectar a éste.

Dentro de fdisk usa la opción 'n' para añadir nuevas particiones. Dalas de alta como particiones primarias (si ya tienes una particion de arranque con MSDOS, ésta será la partición 1 y a partir de ahí añadirás particiones Linux). Es ahora el momento de añadir el espacio de swap, conveniente siempre, e incluso imprescindible si se dispone de memoria igual o inferior a 8Mb. Como regla orientativa puede servir tener el doble de swap que RAM (para 8Mb o menos de RAM), y 1.5 veces para memorias superiores, pero depende de la disponibilidad de disco y de memoria que tengamos, así como del uso que le vayamos a dar. En realidad será tu experiencia la que te irá diciendo en sucesivas instalaciones cuánto swap necesitas. En caso de disponer de más de un disco duro es muy recomendable tener separadas las particiones de linux y de swap (por ej. DOS+swap en uno y linux en el otro), lo que mejorará el rendimiento global.

Una vez creadas las particiones, hay que definir también en fdisk el uso que van a tener (tipo de partición), siendo obviamente "linux swap partition" (tipo 83) para el swap, y "Linux native partition" para el propio linux. Al final, y siguiendo con este ejemplo, tendrás tres particiones definidas:

- MSDOS, marcada como "activa", con el arranque.
- Linux swap.
- Linux Native, donde residirá el linux.

NO TENGAS MIEDO POR TOCAR Y PROBAR: HASTA QUE NO USES LA OPCION DE ESCRIBIR A DISCO, PUEDES HACER TODO TIPO DE PRUEBAS Y/O CANCELARLO TODO HASTA ESTAR SEGURO Y GRABAR LA NUEVA TABLA DE PARTICIONES.

Es probable que haya que rearrancar si modificamos las particiones, y continuar con el software de instalación de nuestro Linux. En el caso de Slackware, ese software de instalación ya necesita al menos 8Mb de memoria, por lo que si estamos justos de memoria tendremos que activar ya mismo el swap antes de ejecutarlo. Si ese es el caso, hay que hacerlo en dos pasos: formatear la partición de swap y luego activarla para que empiece a complementar la RAM. Los comandos serían

---

```
mkswap -c /dev/hdXX
swapon /dev/hdXX
```

---

donde `/dev/hdXX` sería el nombre de la partición que hemos definido como swap. Una vez hecho esto, Slackware ya aporta una utilidad de instalación que detectará las particiones creadas y procederá con el resto de la instalación con la aplicación "setup", que nos guiará durante toda la instalación con un sencillo interfaz de

menús, que no requiere muchas explicaciones. Durante una de las etapas de la instalación Setup intentará preparar el swap. Si lo hemos hecho ya a mano habrá que indicarle que se lo salte, mientras que en el resto de casos dejaremos que setup haga todo lo que tenga que hacer.

Instalación de los discos.

Slackware divide la distribución del sistema operativo en "series" temáticas, y a su vez cada una de ellas en directorios que se corresponden perfectamente con disquetes de 1.44Mb. Los discos esenciales serían los de la "Serie A", con los que el sistema ya es arrancable autónomamente desde el disco duro, aunque en la práctica lógicamente instalaremos también la "AP" (aplicaciones básicas), "N" (software de red, si vamos a conectarnos con cualquier modalidad), "D" (compiladores y librerías), "X" y "XAP" para el entorno gráfico XFree86.

Por su parte RedHat organiza el software también en "paquetes" instalables de forma independiente y también agrupados por temas, aunque no sigue ningún esquema traducible en disquetes, por lo que las únicas vías de instalación son CDROM o directamente de la red.

Configuración

Una vez cargados a disco los paquetes de software, la siguiente fase será la de configuración, que se realiza desde el mismo programa de setup. Como esta parte es muy variable y depende de cada caso. Nos centraremos en dos puntos típicamente problematicos: el arranque del sistema (incluida la posibilidad de arrancar múltiples sistemas operativos desde distintas particiones o distintos discos) y la configuración de red

Para instalar un método de arranque de Linux hay dos utilidades: "Lilo" y "Loadlin". La primera es un pequeño programa que se instala en el "Master boot record" del disco duro o en la partición que fdisk marcó como "activa", y que permite arrancar indistintamente Linux, DOS 6.x, Windows95, etc. La segunda es una utilidad de MSDOS que permite dejar en paz los arranques de particiones, haciendo en su lugar un "arranque en caliente" directamente desde DOS. Ambos funcionan bien, por lo que elegir una u otra es cuestión de preferencias de cada cual. En el segundo caso podrí darnos algún error si hubiera algún programa de DOS residente en memoria que pueda obstaculizar la carga de Linux. Eso es relativamente facil de solucionar jugando un poco con las configuraciones de DOS (incluso con los menus de arranque que incorpora el config.sys de MSDOS 6.x), pero incluso en ese caso podemos encontrarnos dos obstaculos mas: si tenemos instalado DoubleSpace/Drivespace hay que evitar que se cargue en memoria pulsando F8 durante el inicio de DOS para poder ejecutar Loadlin (no hay otra forma de pararlo ya que normalmente se carga incluso antes que config.sys), y ademas si usamos Windows95 hay que pulsar igualmente F8 para arrancar con solo el símbolo del sistema, evitando entrar en el modo gráfico.

Linux puede convivir con cualquier sistema operativo, pero unos dan mas dolores de cabeza que otros, por lo que en caso de duda acude a la documentación correspondiente (la de instalación de la propia distribución y la Booting-HOWTO. De todas formas, si una vez instalado Linux no conseguimos arrancarlo (sabemos que está ahí pero no conseguimos entrar en él), siempre queda un recurso asequible: introduce el "boot disk" y en vez de darle a INTRO cuando te aparezca un simbolo "boot", teclea `root=`dev/XXXX/ indicando que ya hay un Linux en la partición correspondiente, y que debe usarlo en vez de continuar con una hipotética instalación. Lo único diferente es que se estaría usando el núcleo (kernel) del disquete en vez del instalado en el disco duro, pero a partir de ahí montará la partición del disco duro y usará todo lo ya instalado allí.

Otro problema frecuente (solventado en la última version de RedHat pero no aún en Slackware) es que graba en el disco duro un núcleo genérico en vez del que hayamos usado en el "boot disk", que era precisamente el que habíamos elegido, por lo que es habitual que, si necesitamos algún controlador específico para algún dispositivo o cualquier otra funcionalidad, cuando arranquemos directamente desde el disco duro ya no nos lo reconocerá. Si estamos usando LILO haz lo siguiente para arreglarlo: entra como superusuario, copia la imagen que habias

usado para generar el "boot disk" sobre el fichero "/vmlinuz" del disco duro y ejecuta "lilo" para reinstalar el arranque con el núcleo correcto. Si estas usando LOADLIN, simplemente dale como parámetro el fichero de núcleo correcto (es facil tener varios núcleos y elegir en cada caso el que interese). Seguir arrancando con el disquete como en el caso anterior sería también otra posibilidad para no perder ese CDROM, esa tarjeta, etc.

4. *Españolización* de Linux

Uno de los aspectos más molestos es no poder escribir con naturalidad en nuestro idioma por falta de soporte de los caracteres especiales en el software. El juego de caracteres básico de ASCII es de 7 bits, y no incluye ni las vocales acentuadas ni nuestra maltratada 'ñ'. Hay una norma ISO-8859-1 (ó ISO-Latin1) que sí que abarca todos los caracteres pero que sigue dando la lata en muchas tareas habituales, como el correo electrónico (muchos sistemas solo respetan 7 bits, por lo que textos que podemos ver correctamente en nuestro ordenador probablemente se visualizarán de forma lamentable cuando los enviemos a otra persona. A continuación se mencionan algunas de las situaciones más frecuentes y cómo intentar solucionarlas.

4.1 Consola

...O cómo demonios poder usar acentos en nuestro propio ordenador

Mediante el comando loadkeys se cargan las tablas de traducción del teclado. Estas tablas traducen los códigos que se generan al pulsar las teclas a los que reciben los programas. Los ficheros que contienen estas tablas de traducción se encuentran en el directorio `/usr/lib/kbd/keytables/` y el fichero que contiene el mapeado del teclado español se denomina `es.map`. Su objetivo es que si pulsamos una determinada combinación de teclas, aparezca en pantalla el carácter que tenemos pintado en nuestro teclado, y no su equivalente en el teclado que se usa en cualquier otro país.

La llamada a este comando suele ser de la forma siguiente, que aparecerá en el arranque por defecto (en Slackware estaría en`/etc/rc.d/rc.keymap`):

```
loadkeys /usr/lib/kbd/keytables/es.map
```

Nota del autor: A igual configuración de teclado, más de una vez me han fallado los acentos por recompilar el núcleo, pero en el momento de escribir estas líneas aún no he visto la causa...

4.2 Shell bash

Para que la bash soporte el uso de vocales acentuadas asi como de la letra ñ basta con incluir las siguientes líneas en un fichero `.inputrc` (o al menos en `.bash_profile`) en tu $HOME:

```
set meta-flag on                # conservar bit 8 en entrada de teclado
set output-meta on      # conservar bit 8 en salida por terminal
set convert-meta off    # no convertir secuencias de escape

export LC_CTYPE=ISO_8859_1
```

4.3 Shell tcsh

Para que la tcsh soporte el uso de vocales acentuadas asi como de la letra ñ basta con incluir las siguientes líneas en un fichero `.tcshrc` (o al menos en `.login`) en tu $HOME:

```
setenv LANG C
setenv LC_CTYPE "iso_8859_1"
```

4.4 Algunas aplicaciones

Sería excesivamente largo detallar todas las posibilidades para trabajar con caracteres de 8 bits en todas las aplicaciones, utilidades, comandos del sistema, etc. Aquí van algunos ejemplos:

Editor joe

Joe tiene un parámetro `-asis` para que conserve los caracteres de 8 bits.

less

Por defecto muestra solamente los caracteres de 7 bits (los de 8 los representa por su código ASCII). Ello protege de ver en pantalla "basura" si por error intentamos leer ficheros binarios, de datos, etc., pero impide ver nuestros acentos. Se puede modificar momentáneamente con el modificador `-r` desde la propia utilidad, pero sería mejor tener definida la variable de entorno `LESSCHARSET=latin1`.

ls

Para que muestre los caracteres especiales en los nombres de los ficheros tiene las opciones `-8 -N`, que se pueden añadir a la variable `LS_OPTIONS` que probablemente ya tengas definida por defecto.

mc

Este clónico Norton tiene una opción *Full 8 bits* del submenú *Display bits...* bajo *Options*.

nroff, groff

Dispone de una opción `-Tlatin1`

4.5 Xwindow

El entorno gráfico XWindow usa para configurar el teclado un comando llamado `xmodmap` que tiene la misma funcion que `loadkeys` para la consola de texto. Ernest Artiaga (drw@portos.upc.es) preparó un fichero de configuracion para Xmodmap que genera los acentos con la tecla AltGr, que entre otros sitios puede encontrar en ftp://ftp.cs.us.es/pub/Linux/contrib/Xmodmap-es, y que podemos colocar como $HOME/.xmodmap para que se cargue como parte de nuestra configuración. Existe también una utilidad

`/usr/lib/kbd/keytables/mk_modmap` que genera un listado para Xmodmap a partir del ma de teclado usado por loadkeys.

De todas formas este método no es perfecto, ya que muchas aplicaciones bajo Xwindow tienen su propia forma de configuración o simplemente no tienen soporte para caracteres extendidos por lo que a veces es posible que podamos usar acentos en una ventana y no en la aplicación de la ventana de al lado.

4.6 Emulador de DOS (DOSEMU)

La linea de configuración del teclado en `/etc/dosemu.conf` deberá incluir '`layout es-latin1`' para poder usar la 'ñ' y las aperturas de interrogación y admiración. Una vez hecho esto, es posible cargar la configuración habitual con `keyb sp` desde el propio emulador de DOS, o bien ejecutar '`keybes2.com`, una pequeña utilidad para el propio emulador disponible en varias de las direcciones habituales.

4.7 Correo electrónico

Por último en relación con el uso del español en Linux, pero no menos importante, es poder leer y recibir por mensajería electrónica textos escritos en español sin deformar su contenido ni perder los acentos y símbolos propios.

El problema se reduce a lo de siempre: el uso de caracteres especiales no incluidos en la tipografía básica sajona requiere trabajar con 8 bits, pero aún hay muchos sistemas que siguen trabajando con 7 bits, lo que trunca o deforma mensajes escritos sin las debidas precauciones.

En el estándar RFC822 que define el formato usado en los mensajes de correo de Internet no se permite el uso de 8 bits en las *cabeceras* de los mensajes, dejando abierta la posibilidad de 8 bits en el *cuerpo* del mensaje. Pero tampoco podemos abusar de esa posibilidad ya que no tenemos garantías de que lo respete cualquier sistema que pueda recibir nuestros documentos. Una solución cada vez más extendida, y que permite a los usuarios escribir con libertad y enviar prácticamente cualquier información dejando todos esos detalles al software de gestión del correo (tanto los *agentes de usuario* como los *agentes de transporte* en los servidores de correo) es mediante el estandar MIME, que permite *codificar* nuestros mensajes de 8 bits y reducirlos a caracteres de 7 bits que puedan pasar a través de cualquier sistema sin peligro de perder información. El inconveniente (cada vez menor a medida que este sistema se extienda más y más) es que si intentamos leer un mensaje con codificación MIME sin un software que no implemente este sistema y no pueda interpretarlo, puede que veamos "basura" intercalada con nuestro mensaje, o que incluso no entendamos nada de nada.

MIME deja toda la responsabilidad al software, por lo que si ambos extremos (remitente y destinatario) usan software compatible con MIME, el usuario puede enviar junto a textos simples cualquier combinación de tipos de ficheros (documentos generados por procesadores de textos, gráficos, sonido, video, etc). Ni siquiera es necesario que ambos usen el mismo software. Basta con que ambos sean capaces de generar la codificación durante el envío, y de interpretarla en la recepción.

---

5. Conectividad de Linux

Uno de los puntos muy fuertes de Linux es su gran conectividad: incluso con recursos modestos en hardware puede ser un hábil servidor de red y convivir practicamente con cualquier configuracion que ya tengamos funcionando en nuestra red. A medida que mejoremos el equipamiento, Linux aprovechará todos los recursos convirtiéndose en un potente centro de servicios. Si no consigo abrirte el apetito de curiosidad, echa un vistazo a las varias How-To sobre el tema o lo lamentaras... :)

Como digno miembro de la familia Unix, el entorno nativo de red en el que ha nacido Linux es el TCP/IP, asi que nos moveremos en Internet y en las redes locales basadas en esta familia de protocolos como pez en el agua. Puede actuar tanto de simple cliente hasta como una potente estacion de trabajo de bajo coste, pasando por un XTerminal con todo su entorno grafico con absolutamente todo el software necesario, ya incluido o de más o menos fácil instalacion.

En el nivel físico (cacharreo para entendernos), Linux puede conectarse con otros Linux o con cualquier otro sistema usando casi cualquier cosa: cableado serie, paralelo, modems convencionales, tarjetas RDSI, Frame Relay, redes locales ethernet ó token ring, radiopaquete (AX.25 para radioaficionados), etc.

En cuanto a protocolos de red, lo que lleva en la sangre es el mencionado TCP/IP, pero puede acceder (como cliente o como servidor, segun los casos) a redes basadas en IPX (Novell), AppleTalk (Macintosh) y SMB (red LanManager para conectar con Windows para trabajo en grupo, Windows 95 y Windows NT). Por si sirve de pista, quien escribe ha hecho ya la prueba de poner un simple PC de gama media como nodo principal para dar servicios de disco, impresoras y, sobre todo, acceso a Internet a una LAN ethernet de 75 puestos de trabajo que usan indistintamente cualquiera de los mencionados protocolos compartiendo indistintamente sus recursos. Y lo mejor de todo es que nadie fue capaz de distinguir desde sus respectivos puestos a Linux del servidor propio que les atendía hasta entonces. De hecho lleva varios meses en uso ininterrumpido y sólo Linux ha sido capaz de hacer simultáneamente de servidor para usuarios Unix, Windows 3.11, Windows 95, NT, DOS, Netware y Mac sin colgarse ni una sola vez y con un coste ridículo. Nadie me garantiza que nunca me vaya a fallar, pero lo doy por modesta pero absolutamente victorioso en comparación con sistemas comerciales teóricamente superiores y con un coste mucho mayor.

Otra situación en la que actualmente uso Linux es como servidor de Web, mail, FTP anónimo (de servicio público a la red) y servicios de salida a Internet para una red local de 2.100 usuarios, también con muy buenos resultados a pesar de la fuerte carga de trabajo contínuo las 24 horas.

5.1 Conectividad casera

Algunas de las situaciones típicas en las que se puede aprovechar la capacidad de trabajo en red de Linux podrían ser:

TOC

5.2 Servidor de terminales

¿Quién no tiene en casa un PC (o algo peor) que se ha quedado antiguo pero que aún funciona? Cualquier ordenador con un puerto serie y algun software de emulación de terminal puede conectarse a un Linux, permitiendo varios usuarios trabajando simultaneamente. Para quien no lo haya hecho alguna vez, es la forma más simple de comprobar por sí mismo que Linux es un sistema 100% multiusuario y multitarea. Todos los recursos de memoria, disco, etc. estarán a la vez disponibles para compartirlos entre tantos usuarios como puertos serie tengamos.

5.3 Red PPP

Enlazando ordenadores por puertas serie, paralelo, modems, etc, es posible no solamente que un Linux sea el ordenador principal y otros usuarios accedan a través de simples terminales, sino que es posible aprovechar la capacidad de todos los equipos sumando recursos. Desde un equipo se puede acceder a impresoras, discos, etc. de cualquier otro equipo. El protocolo PPP permite crear una red TCP/IP entre todos ellos, usando como enlace tanto modems como simples cables. A medida que cambiemos la red física por algo mas "decente" como una LAN ethernet o análogo, mejorarán las prestaciones como es lógico, pero la idea es la misma y el software de usuario son idénticos (lo que varía, claro esta, es la configuración del propio sistema operativo).

5.4 Servidor de modems

Combinando los dos apartados anteriores, podemos unir la capacidad de servidor de puertos serie (a los que conectaremos modems) y el protocolo PPP (para dar a esas líneas serie la funcionalidad del TCP/IP). Así podemos dar servicio de acceso a Internet a los usuarios que se nos conecten por teléfono.

5.5 Conexion directa a Internet

Ciertamente no es lo normal en casa, pero es relativamente frecuente encontrarlo en universidades y de un tiempo para acá también es habitual en empresas. Si tenemos una red local enganchada a su vez a un proveedor (académico o comercial), entonces una simple tarjeta de red y una simple configuración del sistema serán suficientes para conectar Linux, que actuará como un ordenador Unix cualquiera, con todas sus ventajas. De hecho en la red encontraremos más de un servidor que esconde un Linux en sus "entrañas", o que precisamente quien nos esté haciendo de proveedor de acceso sea un Linux. Pero hay que tener en cuenta tambien sus inconvenientes de seguridad: igual que podemos conectarnos desde Linux, es perfectamente posible que se conecten a nosotros inadvertidamente, por lo que, salvo que necesitemos mantener abierto algun servicio en particular, es recomendable cerrar "todas las ventanas" para evitar problemas. Algunos buenos consejos para un Linux de uso personal serían:

- Crea cuentas para los usuarios que realmente vayan a usarlo, y asígnales claves dificilmente adivinables (no hace falta complicarse la vida con claves imposibles, pero tampoco debes ponérselo fácil a los intrusos)
- Hazte cargo de la clave del administrador `root` y procura mantenerla a salvo. Si la facilitas a

alguien más, mentalízale para que la use con precaución (por otra parte también es una forma de aprender nociones básicas de administración de sistemas...)

- Ponte como `root`, edita `/etc/inetd.conf` y ponle un signo de comentario (#) a todas las lineas excepto aquellas que abran servicios que necesites mantener accesibles; luego haz "killall -1 inetd". No te cortes y cierralo todo, que siempre estaras a tiempo de reabrir aisladamente aquello que eches en falta
- Aunque te consideres muy capaz de administrar un sistema de este tipo, no tengas la mala costumbre de usar `root` como usuario habitual con el que te conectes al sistema, o en caso de despiste podrías provocar algun destrozo. Es mas seguro que entres como un usuario "raso" y uses `su - root` cuando necesites hacer alguna operacion concreta como super-usuario (recuerda usar `exit` para volver a tu status normal cuando termines lo que tenías que hacer como super-usuario)
- El sistema va dejando muchos mensajes registrando la actividad de los usuarios y del propio sistema en múltiples ficheros de contabilidad. Muchos de ellos se centralizan a traves del servicio `syslogd`. En cuanto empieces a desenvolverte un poco con el manual y los comandos básicos para moverte por el sistema, aprende a recoger la información que genere ese servicio, eliminar la inútil, revisar la útil y limpiar la contabilidad de vez en cuando (de lo contrario te puedes dar un susto cuando veas ficheros gigantescos que devoran espacio de disco con información desfasada despues de tanto tiempo)

5.6 Infovía: acceso a Internet desde casa

¿Qué es infovía?

Infovía es una red creada por Telefónica a la que los usuarios acceden desde cualquier punto de la red telefónica española a precio de llamada local. Funciona como una intranet (una red basada en los mismos protocolos que Internet, pero independiente y sin integrarse en ella). No tiene por tanto acceso directo a Internet, pero es un buen intermediario ya que abarata costes tanto para el usuario como para el proveedor de acceso a internet, que aprovecha su infraestructura sin tener que invertir en una red propia. En este sentido es una vía interesante de acceso, pero para evitar confusiones hay que tener presente que Infovía NO es Internet, sino un producto comercial de Telefónica. Aún despues de habernos conectado a Infovía seguimos necesitando un proveedor de acceso a Internet para estar verdaderamente conectados a la *red de redes*.

Al funcionar igual que Internet (el software de Infovía no es más que un PPP y un navegador de web, que utiliza como interfaz de usuario) es perfectamente posible usar Linux para conectarse, pero surge el problema típico de configuración: Telefónica facilita su software gratuito para entornos comerciales típicos (Windows 3.11, Windows95 y Macintosh) que Linux no necesita (ya lo tiene). Pero Linux lo que necesita es conocer la configuracion de su red, para poder integrarse en ella.

¿Como conectar entonces?

El software básico es el `pppd`, que a través del modem creará un enlace TCP/IP entre nuestro ordenador e Infovia. Una vez creado, podremos contactar con nuestro "verdadero" proveedor, quien nos abrirá las puertas a Internet, y a partir de ese momento ya podremos usar cualquiera de nuestras propias

aplicaciones habituales desde Linux. Como complemento de `pppd` usaremos una utilidad `chat` que se encargara de dialogar via modem con el ordenador remoto al comienzo de la llamada, mientras se establece la conexión. También intervendrán varios ficheros de configuración (unos genéricos de Linux para cualquier conexión TCP/IP, otros característicos de las conexiones a través de PPP y modem, y por último alguno que usemos en el caso concreto de Infovía). Un detalle sobre versiones de software: un inconveniente en la evolución de Linux ha sido un cambio en la implementación del PPP que hace incompatible algunos `pppd` con algunas versiones del *kernel*. Si nuestro núcleo está entre 1.2.13 y 1.3.95 debemos usar un pppd 2.1.2d o superior, mientras que necesitaremos un pppd 2.2.0e o superior para los núcleos más recientes.

El listado siguiente es un ejemplo del fichero principal de configuración de `pppd`, en el que a su vez se hace referencia a varios ficheros que deberemos crear: identificación y clave facilitada por el proveedor de acceso, el guión de inicio de conexión, y el de cierre y cuelgue del modem. `/dev/modem` es un enlace (un *"symbolic link"*) que usaremos como nombre genérico para el modem, y que debe apuntar al nombre del dispositivo real en el que lo tengamos conectado.

/etc/ppp/options:

```
modem passive crtscts noipdefault defaultroute
lock lcp-echo-interval 15 lcp-echo-failure 5 mtu 1500
+ua /etc/ppp/infovia.pass
connect "/usr/sbin/chat -v -f /etc/ppp/infovia.chat"
disconnect "/usr/sbin/chat -v -f /etc/ppp/infovia-off.chat"
/dev/modem 38400
```

`/etc/ppp/infovia.pass` es un fichero con simplemente dos líneas, una con el login (identificación de usuario) y otra con la clave. Para acceder a Internet deberemos introducir en ambos casos lo que indique el proveedor, mientras que si sólo vamos a navegar por Infovía (sin salir a Internet) es suficiente con poner `infovia` como usuario genérico e `infovia` como clave.

`/etc/ppp/infovia.chat` contiene el guión que utilizará `chat` en su diálogo con el modem. Un ejemplo de su contenido podría ser:

```
TIMEOUT 6
ABORT '\nBUSY\r'
ABORT '\nNO ANSWER\r'
ABORT '\nNO CARRIER\r'
ABORT '\nRING\r\n\r\RING\r'
ABORT '\nNO DIALTONE\r'
''    ATDP055
'CONNECT'
```

En el ejemplo anterior se usa marcación por pulsos. Si fuera por tonos sustituir `ATDP` por `ATDT`.

Además, si al número 055 de llamada a Infovía hubiera que añadir algún prefijo de centralita o cualq otra marcación especial, no hay más que incluirlo en el listado. Por su parte el de cierre de conexión reseteo del modem sería algo así:

/etc/ppp/infovia-off.chat:

```
TIMEOUT 3
''     '\r'
'\r'   '+++\c'
'\r'   ATH0
OK     ATH0
OK
```

6. Direcciones útiles

6.1 Direcciones electrónicas

Esta seccion es la que previsiblemente cambiara con mas frecuencia, dado el fuerte auge de Linux y la consiguiente mayor disponibilidad de material para Linux en castellano. Esto es solo un ejemplo para empezar en esta primera edicion...

WWW

Linux/España

En Sevilla, el servidor más veterano, con mirrors de LDP/WWW, Slackware, Sunsite, Linux/PPC (la sede para Europa de linuxppc.org), MkLinux (Linux para Powermac), RedHat (para PC y Alpha), etc.

Grupo de Usuarios de Informática

En Valladolid, con mirrors de LDP/WWW, Slackware, Sunsite y manuales traducidos.

Página de Juan José Amor

En Madrid, traducciones de las How-To.

Linux Spanish Manpages

Traducción del manual del sistema.

FTP Anonimo

Grupos de noticias

Listas de correo

Aparte de las numerosísimas listas de correo en inglés (desde las más genéricas hasta las más especializadas en temas muy concretos), en España hay actualmente dos listas dedicadas a Linux en general:

- linux@hades.udg.es en Girona. Persona de contacto para mas información: Paco Culebras Amigo paco@hades.udg.es
- l-linux@cic.teleco.ulpgc.es en la escuela de Teleco en Las Palmas.

En Fidonet

Si Internet te queda muy lejos pero lo tuyo es Fidonet, entonces probablemente ya sabrás que existe un area R34.LINUX con bastante actividad (¡es una de las areas más animadas! ;^) Mucha de la información distribuida en Internet tiene su réplica en Fidonet, además del valioso soporte mútuo entre los propios usuarios.

- Europa III 2:346/207
- Elektra 2:345/422 (+34)(5) 416 4934
- La Voix 2:345/410 (+34)(5) 427 5321

6.2 Direcciones de Firmas comerciales

Poco a poco va creciendo el numero de empresas que prestan atencion al pujante Linux y desarrollan aplicaciones, ofrecen soporte tecnico y/o venden algunas de las distribuciones en CDROM que se mencionaban anteriormente.

En Madrid

- **ABC Analog** (91) 634 20 00
- **Sertec Informatica** (91) 364 23 88

En Barcelona

- **CD-Ware** (93) 302 04 35

En Bilbao

- **Kender** (94) 476 19 22

En Valencia

- **Encis-Net** (96) 351 25 88

6.3 Bibliografía.

- *Linux Edición Especial*, por Jack Tackett, David Gunter y Lance Brown. Editado por Prentice Hall Hispanoamericana, 1996. ISBN 968-880-580-7. Incluye un CD con Slackware 2.20.
- *The Linux programmer's guide*, por Sven Goldt, Sven van der Meer, Scott Burkett, Matt Welsh.

7. Copyright

Copyright (c) Gonzalo García-Agulló 1996. Elaborado a partir de experiencia propia y de información recopilada a través de Internet. Vayan mis agradecimientos para Paco Culebras Amigo y Antonio L. Delgado Gonzalez, buenos linuxeros alla donde los haya.

Este documento se publica bajo la Licencia General GNU, lo que permite su libre distribución gratuita siempre que se conserve en su integridad y se retenga el copyright original.

The Linux Tcl and Tk HOWTO

Luca Rossetti lukaros@tin.it

v0.2, 07 November 1998

This document describes the Linux approach to Tcl, a scripting langua ge. It is an easy to learn interpreted language that uses a typeless approach to achieve a higher level of programming and a rapid application development. The Tk toolkit is a programming environment for creating graphical user interf aces (GUI) under X Window System. Their capabilities include the possibility to extend and embed in other application, rapid development and ease of use. Toge ther, Tcl and Tk provide many benefits both to application developer and user. Tk-ba sed interfaces tend to be much more customizable and dynamic than those built with one of the C or C++ based toolkits. Tk implements the Motif look and feel. A great number of interesting X applications are implemented entirely in Tk, with no new application-specific commands at all.

1. Introduction

- 1.1 Purpose of this document
- 1.2 What you should know before reading
- 1.3 New versions of this document
- 1.4 Revision History
- 1.5 Feedback and other stuff
- 1.6 Credits
- 1.7 Distribution Policy

2. Tcl/Tk History

- 2.1 Cronology

3. What is Tcl/Tk

- 3.1 What is Tcl?
- 3.2 What is Tk?

- 3.3 Extensions
- 3.4 Supported Platforms

TOC

4. Installing and getting started with Tcl and Tk

- 4.1 Downloading the Core Distributions
- 4.2 Installation
- 4.3 The Contributed Archive
- 4.4 Mirror Sites
- 4.5 Which Releases to Use
- 4.6 Where to report problems, bugs, or enhancements

5. Just a little bit of Programming

- 5.1 The one-minute program in Tcl

6. Scripting Language: pros and cons

- 6.1 Some of the most common complaints about Tcl
- 6.2 Some of the most pros about Tcl

7. Most Famous Programs using Tcl/Tk

8. References

- 8.1 Books
- 8.2 Manual and On-line Tutorials
- 8.3 World Wide Web sites
- 8.4 Other documents & Frequently Asked Questions
- 8.5 Newsgroup

9. Tcl/Tk License Terms

1. Introduction

This is the Linux Tcl and Tk HOWTO. It is intended as a Linux reference covering everything you should know concerning installation, configuration and an introduction to development under Tcl and Tk. History and some pros and cons about Tcl and Tk under Linux are analized, and references are given to many other sources of information on a variety of topics related to this simple but powerful scripting language.

If you ever rebuilt your Linux kernel inside X using the command

---

```
make xconfig
```

---

you surely managed to face the strenght of this simple but powerful scripting language.

After executing the first step of kernel rebuilding, a script called kconfig.tk is executed via wish (the Tcl intepreter). The Linux Kernel Configuration wind ow appears. Instead of answering a series of questions, this X-based configuratio n utility allows you to use checkboxes to select which kernel options you want to enable.

The system stores your last configuration options so that every time you run it, you need only to add or remove some checks and you don't need to reent er the whole set of option. Moreover you can fill the whole (or part of the) list of kernel option the order you want. After this simplified step you can rebuil d your kernel in the traditional way.

There's actually another famous case. Have you ever used "printtool" ? (Printer Config Tool (C) Copyright 1994 by Red Hat Software - author: Michael Callahan). If you installed a Red Hat distribution you happily managed to use it to set up printing services . Well, printtool front-end is mainly a Tcl/Tk script.

For those who don't know Red Hat let me tell you how you can easily configure your printers just filling some textboxes and filling in some checkboxes.

The program itself takes care of setting up printing services through the creation of spooling directory, writes /etc/printcap file and the printer's filter, reloads lpd and tests your filter with ascii or postscript pages. It allows you manipulate ghostscript options (i.e. choose up to 8 pages per outpu t page and setting margins), has an help-on-line and many more features.

What is the difference compared to other service-printing implementation?

Everything is achieved by using Tcl/Tk as a "glue" between consolidated application and operating with normal Linux configuration files in a visual and interactive window under X-Window. No new application-specific commands were written at all.

TOC

1.1 Purpose of this document

Currently the purpose of the document is limited to giving initial references to Linux users; in future versions I will try to incorporate a small "Programm ing Tutorial". Let me state that again: this is not (and doesn't want to substitute) an omnicomprehensive "user manual" or "reference" for Tcl and Tk development and programming - it's just a starting point for Linux users.

The author's concept of reference manual coincides with the definition of man pages and many people learn Tcl/Tk from these basic source of informati on. These files are a part of the source code distribution and are installed on your Linux box. You will be able to access the Tcl/Tk manual pages through the man command.

Many structured and complete programming tutorials have been written in order to let the new user begin hacking with Tcl/Tk; tons of other material are available in the Internet. Interpreting Tcl/Tk philosophy I won't try to reinvent the wheel, I will manage to glue the huge amount of material already available.

I suggest you to have a look at the other documents listed in the References section in order to find where to retri eve specific information about Tcl and Tk.

1.2 What you should know before reading

In order to understand the abc of Tcl/Tk you shouldn't be a programming-guru, command sintax is very simple. Basically you have to be familiar with:

- simple programming concepts;
- using very common unix commands and/or utilities;
- having access to the Internet;
- using ftp.

1.3 New versions of this document

Newer versions of this document will be uploaded to ftp site: ftp://sunsite.unc.edu/ pub/Linux/HOWTO/ and will be available on all other mirrors.

Hypertext and other versions of this and other Linux HOWTOs are available mainly at sunsite.unc.edu/LDP/lpd.ht ml and www.linux-howto.com and on many other Mirror World-Wide-Web sites.

I will try to incorporate in my Web-Page the most recent html and sgml version of the document. Most Linux CD-ROM distributions include the HOWTOs, often in a subdir of /usr/doc/ directory, and you can also buy printed copies from several vendors.

Sometimes the HOWTOs available from CD-ROM vendors, ftp sites, and in hardcopy format are out of date. If the date on this HOWTO is more than six months in the past, then a newer copy is probably available on the Internet.

If you make a translation of this document into another language, drop me a line and I'll include a reference to it here.

TOC

1.4 Revision History

- **Version 0.1**: 28 October 1998 - first version;
- **Version 0.2**: 07 November 1998 - deep restyling and lifting.

1.5 Feedback and other stuff

I rely on you, the reader, to make this HOWTO useful. If you have any suggestio ns, corrections, or comments, please send them to me, (Luca Rossetti), and I will try to put them in the next revision.

If any of the links mentioned in this document becomes unavailable or changes, please notify me immediately so that I can update and/or remove the link.

I am willing to answer general questions about Tcl/Tk and Linux as best as I can. Before doing so, please read all of the information in this HOWTO, and send me detailed information about the problem.

If you publish this document on a CD-ROM or in hardcopy form, a complimentary copy would be really appreciated. E-mail me and I will send you back my postal address.

In many section I mention publishing companies or commercial URL sites: I really don't work for these guys.

1.6 Credits

Most of the information in this HOWTO comes from Dr. Ousterhout's Scriptics and Larry W. Virden comp.lang.tcl FAQs .

I would like to thank the PLUTO Italian Linux User Group and the whole volunteers of ILDP (Italian Linux Document ation Project) especially Eugenia Franzoni and Giovanni Bortolozzo for their feedbac k.

1.7 Distribution Policy

This document is Copyright 1998 by Luca Rossetti.

This document is distributed in the hope that it will be useful to the reader: of course it is considered to be without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. This HOWTO is free documentation; you can redistribute it and/or modify it under the terms of the LDP COPYRIGHT .

Please read the LDP Manifesto for more details.

---

2. Tcl/Tk History

Tcl/Tk originated with Dr. John Ousterhout (pronounced "Oh'-stir-howt") while teaching at University of California, Berkeley, Califor nia. He actually started implementing it when he got back to Berkeley in the spring of 1988; by summer of that year it was in use in some internal applications, but there was no Tk. Read about the history of Tcl/Tk directly from its author 's words at www.scripti cs.com/scripting/tclHistory.html.

2.1 Cronology

- **1989:** The first external releases of Tcl and beginning of Tk imp lemention;
- **1991**: First release of Tk;
- **1994:** Dr. Ousterhout was hired by Sun Microsystems, Inc.: he was a Distinguished Engineer and led the Tcl project.
- **April/May 1997:** the Sun research group responsible for Tcl devel opment were spun off into a Sun business group called SunScript. However, things chan ged again soon afterwards. You can read more about that evolution selecting "SunSc ript_story" at URL: Su nScript-Story .
- **August 1997:** a Tcl Consortium was formed. You can read more abou t it, including how to join, who currently serves on the board of directors, and so on by visi ting Tcl Consortium;
- **February 1998:** Dr. Ousterhout left Sun to create Scriptics, a company dedicated to scripting tools, applications, and services. According to www.scriptics.com /about/news/qa.html, core Tcl and Tk remain free, with the team at Sun continuing work right now on Tcl/Tk 8.1. After the next release, the intention is that work on the core will migrate from Sun to Scriptics, with the Sun team will focus more on Tcl extensions and applications.
- **April 23, 1998:** the Association for Computing Machinery ACM awarded the 1997 Software System Award to John Ousterhout and Scriptics (www.acm.org/awards/). This award is given to an institution or individual recognized for developing a software system that has had a lasting influence, reflected in contributions to concept s, in commercial acceptance, or both.

3. What is Tcl/Tk

3.1 What is Tcl?

Tcl is the acronym for "Tool Command Language" (it is pronounced "tickle"). Tcl is actually divided into two things: a language and a library.

Tcl is a simple textual programming language, intended for issuing commands to interactive programs such as text editors, debuggers and shells. It has a simple syntax and it is also programmable.

Tcl users can write command procedures to provide more powerful commands than those given in the built-in set.

Second, Tcl is a library package embeddable in applications. The Tcl library consists of a parser for the Tcl language, routines to implement the Tcl built -in commands, and procedures which allow each application to extend Tcl with addit ional commands specific to that application. The application program generates Tcl commands and passes them to the Tcl parser for execution.

Commands may be generated by reading characters from an input source, or by associating command strings with elements of the application's user interfa ce, such as menu entries, buttons, and other widgets. When the Tcl library receive s commands it parses them into component fields and executes built-in commands directly.

For commands implemented by the application, Tcl calls back to the application to execute the commands. In many cases commands will make recursive invocation s of the Tcl interpreter by passing in additional strings to execute (in fact procedures and conditional-looping commands all work in this way). An applicat ion program can obtain many advantages by using Tcl for its command language:

- Tcl provides a standard syntax: once users know Tcl, they will be able to issue commands easily to any Tcl-based application.
- Tcl succeeds to provides programmability. All a Tcl application needs to do is to implement a few application-specific low-level commands. Tcl provides many utility commands and a general programming interface for building up comp lex command procedures. By using Tcl, applications need not reimplement these feat ures.
- Extensions to Tcl, such as the Tk toolkit, provide mechanisms for communi cating between applications by sending Tcl commands back and forth. The common Tcl language framework makes it easier for applications to communicate with one another.

It is important to note that Tcl was designed thinking that the programmer should actually use two or more languages when designing large software system s. One for manipulating complex internal data structures, or where performance is important, and another, such as Tcl, for writing very small scripts that glue together the other pieces, providing hooks for the user to extend.

For the Tcl script writer, ease of learning, ease of programming and ease of gluing are more important

than performance or facilities for complex data structures and algorithms.

Tcl was designed to make it easy to drop into a lower language when you come across tasks that make more sense at a lower level. In this way, the basi c core functionality can remain small and one need only bring along pieces that one particular wants or needs.

One answer to "What is Tcl?" can be found at www.NeoSoft.com/tcl /whatistcl.html .

3.2 What is Tk?

Tk (pronounced "tee-kay") is an extension to Tcl which provides the programmer with an interface to the X11 windowing system . Note that Tk has been successf ully compiled under X11 R4, X11 R5, X11 R6, as well as Sun's NeWS/X11 environments.

Many users will encounter Tcl/Tk via the "wish" command. Wish is a simple windowing shell which permits the user to write Tcl/Tk applications in a proto typing environment.

At present Tcl/Tk cannot handle Japanese, Chinese, Korean, ... language fonts.

3.3 Extensions

Since Tcl is so easy to extend, many try to share extensions, including the popular itcl, [incr Tcl], ObjectTcl, TclX, Tix , and BLT.

These extensions, of course, require an extended Tcl interpreter. Moreover, many Tcl free applications require a particular Tcl extension to run.

One of the most popular extension is called Expect. It allows you to place a friendly front-end inside most command-line based UNIX applications, such as ftp, telnet, rlogin, passwd, fsck, and so on.

A complete list of Tcl/Tk extensions can be found at URL www.scr iptics.com/resource/software/extensions/.

3.4 Supported Platforms

This section contains information about Tcl 8.0 and Tk 8.0, the most recent version of Tcl/Tk. They were originally released on August 18, 1997 and the most recent patch releases (8.0.3) were made on September 3, 1998.

When you download Tcl and Tk you get two programs, wish and tclsh, supporting script libraries, and on-line reference documentation. These programs are gene ral purpose platforms for writing applications with Tcl. Wish includes the graphic al user interface toolkit Tk. The packages are ready to use after installation.

Tcl 8.0 and Tk 8.0 run on most releases of the following operating systems:

- Windows 95

TOC

- Windows NT
- Solaris and SunOS
- Linux
- HP-UX
- SGI
- IRIX
- Digital Unix
- AIX
- SCO Unix
- Most other Unix-like operating systems Macintosh (68K and Power Mac)
- Pre-compiled releases are available for different Linux distribution.

---

4. Installing and getting started with Tcl and Tk

Most modern distribution include Tcl and Tk. Rpm and deb packages with precompiled binaries are avalaible for Red Hat, SuSE and Debian distributions (that'll make installation easier).

A modern distribution standard installation will include Tcl/Tk as it is needed by many configuration tools running mainly under X.

Tcl and Tk are distributed freely in source form via the Internet. There are no restrictions on their use and no licenses or royalties are needed (see the license.terms section for complete information).

Many more Tcl/Tk scripts and extensions are also available freely.

4.1 Downloading the Core Distributions

The Tcl/Tk core consists of the Tcl and Tk libraries, plus the wish and tclsh applications, associated documentation, script libraries, and demonstrat ive applications. The primary FTP site for this information is ftp.scriptics.com/pub/tcl/.

The primary HTTP site is www.scriptics. com/software/download.html .

4.2 Installation

Unless already available for your Distribution in proprietary packages you'll want to download the source release. You'll need both Tcl and Tk source s. This procedure refers to the second case.

Choose between compressed tar and gzipped tar format.

Compressed Tar Files

Tcl sources (tcl8.0.3.tar.Z): compressed tar file (about 2.4 Mbytes). Tk sources (tk8.0.3.tar.Z): compressed tar file (about 3.3 Mbytes).

Gzipped Tar Files

Tcl sources (tcl8.0.3.tar.gz): gzipped tar file (about 1.5 Mbytes). Tk sources (tk8.0.3.tar.gz): gzip'ed tar file (about 2.1 Mbytes).

When you retrieve one of these files, you will get a compressed tar file with a name like tcl8.0.3.tar.gz or tcl8.0.3.tar.Z. The files are identical except for the technique used to compress them (.gz files are generally smalle r than .Z files).

To unpack the distribution, invoke shell commands like the following, depending on which version of the release you retrieved:

```
gunzip -c tcl8.0.3.tar.gz
```

or

```
tar xf - zcat tcl8.0.3.tar.Z
```

or

```
tar xf - unzip tcl80.3.zip
```

Each of these commands will create a directory named tcl8.0.3, which includes the sources for all platforms, documentation, and the script library for Tcl 8.0. To compile and install the distribution, follow the instructions in the README file in the distribution directory. Be sure to compile Tcl before Tk, since Tk depends on information in Tcl.

4.3 The Contributed Archive

There are many other freely available packages for Tcl and Tk, including both scripts written in Tcl and extensions written in C or C++. These packages include database applications and network access, a graphical user interface builder, the expect program, additional Tk widgets, and dozens of other things . The primary site for the Tcl/Tk archive is ftp.neosoft.com/pub/tcl.

4.4 Mirror Sites

Several other sites around the world mirror the whole or part of the material from the core site and the contributed archive; you may find more useful to retrieve information from a mirror site that is close to you.

Ftp file "0_mirror" at: ftp.scriptics.com/pub/tcl/ for a list of the mirror sites in your country.

4.5 Which Releases to Use

Always refer to newer recommended version in section "Tcl/Tk Core" of the

Scriptics Soft ware Central page.

At the time of this writing, recommended releases are the latest (Tcl 8.0.3 and Tk 8.0.3), which were released in September, 1998. Tcl 8.0 contains a new bytecode compiler that can speed up execution by a factor of 2-10x. It also provides namespaces, binary I/O, and several other new features.

Tk 8.0 is the first release to provide native look and feel on Macintoshes and PCs. Tk 8.0 also suppo application embedding and has a new portable font mechanism. Both Tcl 8.0 and Tk 8.0 provide additional features in the Safe-Tcl security model.

TOC

4.6 Where to report problems, bugs, or enhancements

Use comp.lang.tcl for public communicati ons.

The alternative would be to send problems, suggestions, new ideas, etc. directly to the author. Email to John Ousterhout will reach the author of Tcl and Tk.

When reporting problems or bugs, be sure to mention all the details needed for a correct diagnosis. Basically you will have to describe what hardware, operating system and version of Tcl/Tk you are using, if you have made any modification or add on and provide, if possible, either a small piece of code, or a URL to some code which demonstrates the problem.

If you have software from which you think the community might benefit (either a program, function, extension, or simple example), or you have a document, magazine or journal article, thesis, project, or even commercial advertisement , be sure to let the appropriate guys know.

There are FAQ maintainers for each of these areas as well as a comp.lang.tcl.announce newsgroup you can use.

It is always worthwhile to submit your contributions directly to the ftp site so more folk in the future can benefitthanks to your experience.

To make announcements to the comp.lang.tcl.announce newsgroup, send email with the details to tcl-announce. Also, feel free to just point folk at your own ftp site or WWW site if you have one which can be used .

---

5. Just a little bit of Programming

Since Tcl is an interpreted language, to run a Tcl program (also called a script), you normally pass the script file to the Tcl interpreter, wish, for example:

```
wish hello.tcl
```

You can also use wish in interactive mode and type in commands at the command line.

There's another standard Tcl interpreter, tclsh, which only understands the Tcl language. Tclsh does not have any of the Tk user interface commands, so you cannot create graphical programs in tclsh.

Some Tcl freeware applications extend the Tcl language by adding new commands written as C functions. If such is the case, you need to compile the applicati on instead of just passing its Tcl code to the wish interpreter. This application program, from a Tcl perspective, is really a new version of the wish interpret er, which the new C commands linked in. Of course, the application program may be a lot more than merely a Tcl interpreter. (Note: you can also use Tcl's auto-loading capability on systems that support it.)

5.1 The one-minute program in Tcl

Tcl has a simple structure. Each line starts out with a command, such as button and a number of arguments. Each command is implemented as if it was a C function. This function is responsible for handling all the arguments.

As a very standard example, the following is the Hello World program in Tcl/Tk:

```
# This is a comment
button .b -text "Hello World" -command exit
pack .b
```

In this case you have to type the commands interactively to tclsh or wish.

You can also place command into script files and invoke these just like shell scripts. To do this for the previous example, rewrite the Hello World program as follows:

```
#! /usr/local/bin wish -f
button .b -text "Hello World" -command exit
pack .b
```

Put the text inside a file called Hello and make sure that wish is installed in /usr/local/bin (otherwise you will have to change opportunely the path).

Make the file Hello executable issuing the command

```
chmod 775 Hello
```

and run it inside X.

You will see a button labelled Hello World inside a window: clicking it will close (exit) the window.

6. Scripting Language: pros and cons

To understand the importance and the future of Tcl/Tk I strongly suggest to point your web browser at the URL www.scriptics .com/people/john.ousterhout/ by John K. Ousterhout . You will read about the importance and the comparison between scripting (in langua ges such as Tcl) and system programming (in languages such as C and Java).

To read a document about comparisons see the comparison discussion.

Here's a summary of the most important pros and cons about Tcl/Tk.

6.1 Some of the most common complaints about Tcl

Tcl is interpreted

The data is primarily treated as strings, programs written in Tcl are slow. Tcl 8.x attempts to address this by doing some degree of compilation as well as permitting additional variable types.

Several characteristics are not intuitive

Comments are commands rather than traditional comments, numbers beginning with 0 are octal, proper use of quoting mechanisms, etc. These aspects are covered in the various FAQs.

6.2 Some of the most pros about Tcl

It is a high-level scripting language

You need to write a lot less code to get your job done, especially when compared to Motif or Win32 applications. In general, the number of Line Of Code (LOC) of a software project is one of the most important complexity index es.

Tcl is free

You can get the sources for free over the Internet from Scriptics Download Page or from the FTP site for Tcl. The software c ore site includes the source code version, as well as binary versions for Windows and Macintosh platforms; or, you can get Tcl on a number of CD-ROMs for a nominal cost.

Read about Tcl and Tk core free at www.scriptics.com/about/n ews/qa.html .

TOC

It runs on many platforms

Versions exist for UNIX (Linux... of course), Windows and Macintosh. Except for a few platform differences, your Tcl scripts will run the same way on all systems.

It is interpreted

You can execute your code directly, without compiling and linking (though Tcl compilers are available).

It is extensible

It's easy to add your own commands to extend the Tcl language. You can write your commands in C or Tcl.

It is embeddable in your applications

The Tcl interpreter is merely a set of C functions that you can call from your code. This means you can use Tcl as an application language, much like a macro language for a spreadsheet application.

Tcl/Tk is Year 2000 (Y2K) compliant

Read what the creator of the Tcl and Tk core tells about this topic www.scriptics.com/Y2K.html .

---

7. Most Famous Programs using Tcl/Tk

Apart from the two implementation described in the Introduction, there are many applications writte n in Tcl/Tk or a combination of Tcl and C. A complete list where to look for these implementation is the part 4 of the frequently-asked questions on Tcl/Tk (FAQs). I suggest you to visit

Scriptics' Softwar e Central.

Another good starting point is www.NeoSoft.com/tcl/.

One of the best Tcl applications running under Linux is called TkDesk a window manager and application launcher that works very well.

If you're a Tcl/Tk Linux developer, feel free to send me a URL (and a description of the work) that I can link to in here.

8. References

For many reasons people often like having a hard-copy manual as a reference or like to be helped by other folks online.

Here you can find a selection of reference books, tutorials, www-sites and newsgroup.

8.1 Books

Many books concerning Tcl/Tk were written and are to be published. I won't even try to list them all (**another howto woudn't be enough for that** :)). You can find much more information plus additional notes at:

- www.tcltk.com/consortium/resources/books.html
- www.amazon.com/exec/obidos/subst/categories/computer-programming/tcl-tk-article/002-8989352-4516417
- Page: tcl_books.html at URL starbase-neosoft-tcl-books

Here I will try to summarize with some notes the book I know concerning the subject, they're all at a basic-medium level. Again, people who know the subject, have enough information about where to find advanced level books.

Tcl and the Tk Toolkit

Author: John K. Ousterhout

WWW book information: cseng.aw.com/bookdetail.qry?ISBN=0%2D201%2D63337%2DX&ptype=0

Book's examples: ftp.script ics.com/pub/tcl/doc/book.examples.Z

Book suppliment: www.scriptics.com/doc/tk 4.0.ps

The book primarily covers Tcl 7.3 and Tk 3.6.

Practical Programming in Tcl and Tk, II ed.

Author: Brent Welch

WWW book information: www.beedub.com/book/

Book's table of contents: www.beedub.com/book/

Book promotion info at section_50000.html of the URL www.borders.com/sections/

This updated edition describes Tcl / Tk 8.0 as it was during the beta period. Along with the material from the first edition, it also covers sockets, packag es, namespaces, a great section describing the changes in Tcl 7.4, 7.5, 7.6, and 8.0 (and Tk as well), Safe Tk and the Plugin!

Tcl and Tk Reference Manual

Editors: Donald Barnes, Marc Ewing, Erik Troan

WWW book information: www.lsl.com/catalog/bo oks/tcltk/

The Visual TCL Handbook, 1/e

Author: David Young

WWW book information: www.amazon.com/exec/obidos/ISBN=013461674X/u/7141-5908756-107481

A comprehensive guide to Visual TCL. This book leads reader from basic graphical user interface development concepts to meaningful application develo pment. The book focuses on the TCLX and VT extensions, addressing many fundamental TCL topics. VT is a Motif based graphical interface, incompatible with Tk. The entire TCL language is documented in a separate Commands section. Comes with a CD-ROM that includes SGI, Solaris, HP-UX, AIX and Unixware versions of Visual Tcl.

Running LINUX

Author: Matt Welsh and Lar Kaufman

WWW book information: www.ora.com/ca talog/runux2/noframes.html

Running Linux is a really well written basic book. It has a chapter on programming using Tcl/Tk. (and Perl, C, C++).

Tcl/Tk for Dummies (For Dummies)

Author: Timothy Webster, with Alex Francis

WWW book information: www.dummies.com/

Another one of the series of the paperback programming books. This one focuses on the Tcl plugin as a programming environment.

Interactive Web Applications With Tcl/Tk

Authors: Michael Doyle Hattie Schroeder

WWW book information: www.eolas.com/tcl/

This is a learning by example book, for people who know a bit of programming, but are not experts. It covers developing applets as well as stand-alone appli cations and simple server applications. The book comes with the Spynergy toolkit, whic h adds a variety of pure Tcl/Tk procedures for distributed processing, URL retri eval, HTML rendering, database management and platform independent file managment, Ed, a Tcl editor and testing environment, an image conversion tool, a demo of Tk features, a client/server version of a rolodex application, a pure Tcl web server, a client/server push application, a tcl web browser,

TOC

8.2 Manual and On-line Tutorials

- John Ousterhout has written an engineering style guide that describes the coding, documentation, and testing conventions that will be used at Sun in the coding of the C code in the Tcl core and has made it available to other Tcl and Tk developers. It is located at ftp.script ics.com/pub/tcl/doc/engManual.tar.Z
- A second style guide, covering the writing of Tcl scripts, can be found at ftp.scr iptics.com/pub/tcl/docs/styleGuide.tar.gz. Other versions of it can be found at sunscript.su n.com/techcorner/styleGuide.ps.
- A brief introduction to TCL/TK by David Martland
- Another tutorial untitled User interfaces with Tcl/T kwas written by Fintan Culwin .
- Although you should have your Tcl/Tk manual pages on your system, here's another place where to look for TCL Manual Pa ges (from TCL7.4)-Tk Manual Pages (from TK4.0).
- The Tcl/Tk Cookbook offe rs a lot of getting-started information.

8.3 World Wide Web sites

There are a great number of WWW resources which provide additional information about many aspects of Tcl and its extensions.

- Refer to Tcl-FAQs (pa rt2) for a great number of pointers to Tcl/Tk documentation and web sites.
- Point your web browser at Tcl/Tk Information : a site with many links to TclTk resources on the web (Information, Extensions, Tools, Training and Events).
- Have a look at Tcl-Consortium : a non-profit organization which promotes Tcl/Tk and supports the development community.
- Point your Web browser at The Official Contributed Sources A rchive for the Tool Command Language (Tcl) and the Tk Toolkit, hosted by NeoSoft, Inc.
- For a discussion dealing with the pros and cons of the major scripting languages : article in SunWorld by Cameron Laird
- A web page that contains a variety of comparisons between Tcl/Tk and othe r similar systems. Most of them are taken from "comp.lang.tcl", the author would be happy to add any other important article that you folks want to send to him.: Comparison by Wayne Christopher.
- E.J. Friedman-Hill's Tcl/Tk Course: this document is available only in PowerPoint source form and in low-quality HTML form (in HTML format all the figures and some of the text is missing).

8.4 Other documents & Frequently Asked Questions

A lot of material is available on the Internet: introductory papers, white papers, tutorials, slides, postscript versions of published books in draft and many more.

For a complete reference please give a look at the excellent Tcl-FAQs .

8.5 Newsgroup

comp.lang.tcl is an unmoderated Usenet newsgroup, created for the discussion of the Tcl programming language and tool s that use some form of Tcl, such as the Tk toolkit for the X window system, Extended Tcl, and expect.

For Tcl/Tk related announcements always refer to comp.lang.tcl.announce : you will find release announcement, patches, new application and so on.

Again, faq could be retrieved at Tcl-FAQs.

---

9. Tcl/Tk License Terms

The following terms apply to the all versions of the core Tcl/Tk releases, the Tcl/Tk browser plug-in version 2.0, and TclBlend and Jacl version 1.0. Please note that the TclPro tools are under a different license agreement. This agreement is part of the standard Tcl/Tk distribution as the file named "license.terms".

TCL/TK LICENSE TERMS

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties. The followin g terms apply to all files associated with the software unless explicitly discla imed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behal f of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs. Notwithstanding the foregoing , the authors grant the U.S. Government and others acting in its behalf permissi on to use and distribute the software in accordance with the terms specified in this license.

The teTeX HOWTO: The Linux-teTeX Local Guide

Robert Kiesling

v3.7, 9 November 1998

This document covers the basic installation and usage of the teTeX TeX and LaTeX implementation under the major U.S. Linux distributions, and auxiliary packages like Ghostscript. Contents of the teTeX HOWTO: The Linux-teTeX Local Guide are Copyright (C) 1997, 1998 by Robert A. Kiesling. The exact terms of copying are given in the introduction and the appendices. Registered trademarks are the property of their respective owners. Please send all complaints, suggestions, errata, and any miscellany to kiesling@ix.netcom.com, so I can keep this document as complete and up to date as possible.

1. Introduction.

- 1.1 Copyright.
- 1.2 Software described in this document.

2. Using teTeX.

- 2.1 Printing the documentation.

3. TeX commands.

- 3.1 Command overview.
- 3.2 Font commands.
- 3.3 Paragraph styles and dimensions.
- 3.4 Page layout.
- 3.5 Page numbers, headers, and footers.
- 3.6 Titles and macros.

TOC

4. LaTeX commands.

- 4.1 Document structure.
- 4.2 Characters and type styles.
- 4.3 Margins and line spacing.
- 4.4 Document classes.

5. LaTeX extension packages and other resources.

6. Mixing text and graphics with `dvips`.

- 6.1 What if my printer isn't supported?

7. Using Postscript fonts.

8. Appendix A: CTAN site list.

9. Appendix B: Installing the CTAN teTeX distribution.

- 9.1 Installing the binary distribution.
- 9.2 Base system configuration.
- 9.3 Installing the CTAN source distribution.
- 9.4 Post-installation configuration details.

10. Appendix C: Distribution and Copyright.

- 10.1 Distribution.
- 10.2 GNU GENERAL PUBLIC LICENSE
- 10.3 How to Apply These Terms to Your New Programs

---

1. Introduction.

1.1 Copyright.

The teTeX-HOWTO is copyright (C) 1997, 1998 by Robert Kiesling. Permission is granted to make and distribute verbatim copies of this manual provided that the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided also that the sections entitled, ``Distribution," and, ``GNU General Public License," are included exactly as in the original, and provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions. except that the sections entitled, ``Distribution," and, ``GNU General Public License," may be included in a translation approved by the Free Software Foundation instead of in the original English. Please refer to Section Distribution and Copyright for terms of copying.

1.2 Software described in this document.

TeX handles only the formatting part of the document preparation. Generating output from TeX is like compiling source code into object code, which still needs to be linked. You prepare an input file with a text editor----what most people think of as ``word processing"--- and format the input file document with TeX to produce a device-independent output file, called a `.dvi` file.

You also need a program or two to translate TeX's `.dvi` output for your screen and printer. These programs are collectively known as ``dviware." For example, TeX itself only makes requests for fonts. It is up to the `.dvi` output translator to provide the actual font for the output regardless of whether the medium is a video screen or paper. This extra step may seem overly complicated, but the abstraction allows documents to display the same on different devices with little or no change to the original document.

teTeX.

TeX is implemented for practically every serious computer system in the world---and quite a few ``non-serious" ones---so implementors must provide the installation facilities for all of them. This accounts in part for teTeX's complexity, in addition to the inherent complexity of any TeX installation. It also accounts for the fact that installing the system yourself is a significant task, and unless you are already familiar with TeX, it is easy to get lost in the numerous executable programs, TeX files, documentation, and fonts.

TOC

Fortunately, teTeX is part of the GNU/Linux distribution. You can install the package much more ea using GNU/Linux installation tools. You may already have teTeX installed on your system. If so, yo can skip ahead to Section Using teTeX.

However, if you want to install the package, the archives necessary for a workable teTeX installation are on the CTAN archive network. There is a list of these sites in Section CTAN site list.

CTAN is the Comprehensive TeX Archive Network, a series of anonymous FTP sites that archive TeX programs, macros, fonts, and documentation. In the course of using TeX you'll probably become familiar with at least one CTAN site. In this document, a pathname like `~CTAN/contrib/pstricks` means ``look in the directory `contrib/pstricks` of your nearest CTAN site."

The installation of the generic teTeX distribution described in Section Installing the CTAN teTeX distribution concentrates on the Intel versions of Linux. Installing teTeX on other hardware should require only substituting the appropriate executable program archive in the installation process.

In addition to the executable programs, the distribution includes all of the TeX and LaTeX package, `metafont` and its sources, `bibtex, makeindex,` and *all* of the documentation... more than 4 megabytes' worth. The documentation covers everything you will forseeably need to know to get started. So, you should install all of the documents. Not only will you eventually read them, the documents themselves provide many examples of ``live" TeX and LaTeX code.

TeX was written by Professor Donald Knuth of Stanford University. It is a lower-level typesetting language for all of the higher-level packages like LaTeX. Essentially, LaTeX is a set of TeX macros that provide convenient, predefined document formats for end users. If you like the formats provided by LaTeX, you may never need to learn bare-bones TeX programming. The difference between the two languages is like the difference between assembly language and C. You can have the speed and flexibility of TeX, or the convenience of LaTeX.

By the way, the letters of the word ``TeX" are Greek, tau-epsilon-chi. It is not a fraternity, but the root of the Greek word, *techne*, which means art and/or science. ``TeX" is not pronounced like the first syllable in ``Texas." The *chi* has no English equivalent, but TeX is generally pronounced so that it rhymes with ``yecch," to use Professor Knuth's example from *The TeXBook*, which is one of the standard TeX references. When writing, ``TeX," on character devices, always use the standard capitalization, or the `\TeX{}` macro in typesetting.

Text editors.

Any of the editors that work under Linux---`jed, joe, jove, vi, vim, stevie,` Emacs, and microemacs---will work to prepare a TeX input file, as long as the editor reads and writes plain-vanilla ASCII text. My preference is GNU Emacs. There are several reasons for this:

- You can format, preview and print documents with Emacs's TeX and LaTeX modes.
- Emacs can automatically insert TeX-style, ``curly quotes," as you type, rather than the "ASCII-vanilla" kind.
- Emacs has integrated support for Texinfo, a hypertext documentation system.
- Emacs is widely supported. Versions 19.34 and later, for example, are included in the major U.S.

Linux distributions. The most recent version from the GNU archives is 20.3.

- Emacs does everything except butter the toast in the morning.
- Emacs is free.

dvips.

Tomas Rokicki's `dvips` generates Postscript from a `.dvi` file. In addition, it runs Metafont if necessary to generate the bit mapped fonts it needs or uses Postscript fonts for the output. It can also crop and resize pages and perform graphics translations from instructions in a TeX or LaTeX file,

The `dvips` program is part of the teTeX distribution. It is discussed fully in Section Mixing text and graphics with <tt>dvips</tt>

Fonts.

Much of TeX's, and therefore LaTeX's, complexity, arises from its implementation of various font systems, and the way these fonts are specified. A major improvement of LaTeX 2e over its predecessor was the way users specify fonts, the former New Font Selection Scheme. They're discussed in Section Characters and type styles, Section TeX Font Commands, and Section Using Postscript fonts.)

teTeX comes distributed with about a dozen standard fonts preloaded, which is enough to get you started. Also provided are the font metrics descriptions, in `.tfm` (TeX font metric) files. To generate the other fonts that you need, it is simply a matter of installing the `metafont` sources. teTeX's `.dvi` utilities will invoke `metafont` automatically and generate the Computer Modern fonts you need.

---

2. Using teTeX.

Theoretically, at least, everything is installed correctly and is ready to run. teTeX is a very large software package. As with any complex software package, you'll want to start by learning teTeX slowly, instead of being overwhelmed by its complexity.

At the same time, we want the software to do something useful. So instead of watching TeX typeset

```
``Hello, World!''
```

as Professor Knuth suggests, we'll produce a couple of teTeX's own documents in order to test it.

2.1 Printing the documentation.

You should be logged in as `root` the first few times you run teTeX. If you aren't, Metafont may not be able to create the necessary directories for its fonts. The `texconfig` program includes an option to make the font directories world-writable, but if you're working on a multi-user system, security considerations may make this option impractical or undesirable.

In either instance, if you don't have the appropriate permissions to write to the directories where the fonts are stored, Metafont will complain loudly because it can't make the directories. You won't see any output because you have a bunch of zero-length font characters. This is no problem. Simply log out, re-login as `root,` and repeat the offending operation.

The nice thing about teTeX is that, if you blow it, no real harm is done. It's not like a compiler, where, say, you will trash the root partition if a pointer goes astray. What, you haven't read the teTeX manual yet? Of course you haven't. It's still in the distribution, in source code form, waiting to be output.

So, without further delay, you will want to read the teTeX manual. It's located in the directory

```
/usr/lib/teTeX/texmf/doc/tetex.
```

The LaTeX source for the manual is called `TETEXDOC.tex`. (The `.tex` extension is used for both TeX and LaTeX files. Some editors, like Emacs, can tell the difference.) There is also a file `TETEXDOC.dvi` included with the distribution, which you might want to keep in a safe place---say, another directory ---in case you want to test your `.dvi` drivers later. With that out of the way, type

```
latex TETEXDOC.tex
```

LaTeX will print several warnings. The first,

```
LaTeX Warning: Label(s) may have changed. Rerun to get the
cross-references right.
```

is standard. It's common to build a document's Table of Contents by LaTeXing the document twice. repeat the command. The other warnings can be safely ignored. They simply are informing you that of the FTP paths mentioned in the documentation are too wide for their alloted spaces. Sections Paragraph styles and dimensions and Tolerances describe horizontal spacing in more detail.

teTeX will have generated several files from `TETEXDOC.tex.` The one that we're interested in is `TETEXDOC.dvi.` This is the device-independent output which you can send either to the screen or the printer. If you're running teTeX under the X Windows System, you can preview the document with `xdvi`.

For the present, let's assume that you have a HP LaserJet II. You would give the command

```
dvilj2 TETEXDOC.dvi
```

which writes a PCL output file from `TETEXDOC.dvi`, including soft fonts which will be downloaded to the LaserJet. This is *not* a feature of TeX or LaTeX, but a feature provided by `dvilj2`. Other `.dvi` drivers provide features that are relevant to the devices they support. `dvilj2` tries to fill the font requests which were made in the original LaTeX document with the the closest equivalents available on the system. In the case of a plain text document like `TETEXDOC.tex`, there isn't much difficulty. All of the fonts requested by `TETEXDOC.tex` will be generated by `metafont`, which is automatically invoked by `dvilj2`, if the fonts aren't already present. (If you're running `dvilj2` for the first time, the program may need to generate all of the fonts.) There are several options that control font generation via `dvilj2`. They're outlined in the manual page. At this point, you shouldn't need to operate `metafont` directly. If you do, then something has gone awry with your installation. All of the `.dvi` drivers will invoke `metafont` directly via the kpathsea path-searching library---the discussion of which is beyond the scope of this document---and you don't need to do any more work with `metafont` for the present---all of the `metafont` sources for the Computer Modern font library are provided.

You can print `TETEXDOC.lj` with the command

```
lpr TETEXDOC.lj
```

You may also need to install a printer filter that understands PCL.

The nine-page *teTeX Guide* provides some useful information for further configuring your system, some of which I have mentioned, much that this document doesn't cover.

Some of the information in the next section I haven't been able to test, because I have a non-Postscript HP Deskjet 400 color ink jet printer connected to the computer's parallel port. However, not owning a Postscript printer is no barrier to printing text and graphics from your text documents. Ghostscript is available in most Linux distributions and it could already be installed on your system.

---

3. TeX commands.

Preparing documents for TeX typesetting is easy. Make sure there's a blank line between the paragraphs of a plain text file, and run file through the TeX program with the command

```
tex your_text_file
```

The result will be a file of the same base name and the extension `.dvi`. TeX formats the text in 10-point, Computer Modern Roman, single-spaced, with justified left and right margins. If you receive error messages from special characters like dollar signs, escape them with a backslash character, `\`, and run TeX on the file again. You should be able to process the resulting file with the `.dvi` file translator of your choice (see above) to get printed output.

One peculiarity of TeX input is that you must use opening and closing quotes, which are denoted in the input file with the grave accent and single quote characters. Emacs' TeX mode does this for you automatically.

```
"These are ASCII-type quotes."
``These are `TeX-style' quotes.''
```

3.1 Command overview.

Commands in TeX start with a backslash (``\''). For example, the command to change the spacing between lines is

```
\baselineskip=24pt
```

The baseline is the bottom of the characters on a line, not counting descenders. The distance between the baseline of one line and the next is the `\baselineskip`, and is assigned a value of 24 points.

Measurements or dimensions in TeX are often given in the following units:

```
pt                  % Point        1/72 in.
pc                  % Pica:        12 pt.
in                  % Inch:        72.27 pt.
cm                  % Centimeter:  2.54 cm = 1 in.
mm                  % Millimeter:  10 mm = 1 cm.
```

Some commands do not take assignments. For example:

```
\smallskip          % Approximately 3 pt.
\medskip            % Two \smallskips.
```

```
\bigskip          % Two \medskips.
```

A `\smallskip` inserts a 3 pt. vertical space in the document. The measurements are approximate because TeX needs to adjust the dimensions for page breaks, section headings, and other units of vertical space. This is true for horizontal spacing as well.

```
\hsize=6.5in
```

This command sets the line length to a width of 6.5 inches. TeX tries to fill the line by adjusting the spacing between words, and some letters. If TeX cannot fill a line to within its tolerances, it produces a warning message, and adjusts the horizontal spacing within the line as best it can. Formatting tolerances are discussed in Section Tolerances.

There are many other commands that specify horizontal and vertical dimensions and tolerances, and the most commonly use commands are described below.

3.2 Font commands.

In TeX, the default font is 10 pt. Computer Modern Roman. To specify a typeface, like italic, bold, or monospaced, use the following commands.

```
\rm          % Roman (the default).
\it          % Italics.
\bf          % Bold.
\tt          % Monospaced (teletype).
\sl          % Oblique (slanted).
```

The commands change the typeface where they appear in the text, as in this example.

```
This text is Roman, \it and this text is italic.  \bf This text is
bold, and \rm this text is in Roman again.
```

To specify a font for your document, use the`\font` command.

```
\font\romantwelve=cmr12
```

This creates the font command `\romantwelve`, which, when used in the text, changes the font to Computer Modern Roman, 12 point.

```
\romantwelve
This is the Computer Modern Roman font at 12 points.
```

For information about the fonts in the teTeX distribution look at the file:

```
/usr/lib/teTeX/texmf/doc/fonts/fontname/fontname.dvi
```

If you want to print a sample of a font, TeX the file

```
/usr/lib/teTeX/texmf/tex/plain/base/fontchart.tex
```

and fill in the name of the font you want to print at the prompt.

You can also change the size of a font to get different effects. Font magnification is exponential, and specified with the `scaled \magstep` command, which is placed after the font specification.

```
\font\sfmedium=cmss12 scaled \magstep 1
```

This command will give you a sans serif font that is 120 percent the size of the 12-point Computer Modern sans serif font. Fonts can be magnified in steps from 0 to 5. Each step provides and additional 120 percent magnification.

3.3 Paragraph styles and dimensions.

As mentioned above, TeX typesets text in 10-point Computer Modern Roman by default. The length of a line is the value of `\hsize`, which defaults to 6.5 in. If you want to change the value of `\hsize` to 5.5 in. for example, use this command.

```
\hsize=5.5in
```

In TeX a *dimension* is an adjustable unit of length, either horizontal or vertical. The amount by which a dimension can be increased or decreased can be specified in its definition. Closely related to a dimension is a *skip*, which is a dimension that is placed in one of TeX's internal registers. Skips are defined with the `\newskip` command. The `\smallskip` dimension, as defined by TeX is:

```
\newskip\smallskipamount \smallskipamount=3pt plus 1pt minus 1pt
```

The `\smallskip` command is shorthand for:

```
\vskip\smallskipamount
```

There are a number of dimensions that control the page layout. They are summarized in Section Page layout.

TeX formats paragraphs with justified left and right margins. If you want the text to be left justified only, use this command:

```
\raggedright
```

To typeset a line that is justified to the right margin, use the `\rightline` command:

```
\rightline{This is the line to be typeset.}
```

The `\line` command typesets the text of its argument to fill the entire line.

```
\line{This text will be spaced to fit the entire line.}
```

The `\hfil` command adds space to fill out the line where it occurs. So, for example, the `\rightline` command is equivalent to:

```
\line{\hfilThis line will be right justified.}
```

To typeset a line that is centered, use the `\centerline` command.

```
\centerline{This is the line to be centered.}
```

To change the left margin, set the value of `\hoffset`, as in this example:

```
\hoffset=1.5in
```

The `\parindent` command specifies the amount that the first line of every paragraph is indented.

```
\parindent=.5in
```

Two other dimensions, `\leftskip` and `\rightskip`, will indent the right and left margins, respectively, of the paragraphs that come after them.

```
\leftskip=.5in
\rightskip=.5in
```

The control word `\narrower` is equivalent to:

```
\leftskip=\parindent
\rightskip=\parindent
```

That is, `\narrower` narrows the paragraph margins by the value of `\parindent`

As mentioned in the previous section, the `\baselineskip` specifies the distance between lines. The default is 12 pt. To approximate double-spaced text, use the following command.

```
\baselineskip=\baselineskip*1.6
```

The `\parskip` command specifies the distance in addition to `\baselineskip` between paragraphs. By default, no extra space is added, but the distance between paragraphs can stretch as much as 1 pt. to fill the page correctly. To put a blank line between paragraphs, use this command:

```
\parskip=\baselineskip
```

Tolerances. (What are those black rectangles after every line?)

TeX normally formats text to strict tolerances. If, for some reason, text cannot be formatted to within those tolerances, TeX produces a warning message and formats the text the best it can. If the text must be stretched too much to fit the line, TeX warns you that the `\hbox` is underfull. Text that must be

squeezed to fit in the line produces an overfull `\hbox` warning.

For each overfull `\hbox`, TeX places a *slug*, a black rectangle, after the line. The slug indicates that the line could not be formatted to within the specifications set by the `\hbadness` parameter.

The fit of the text within its specified dimensions is measured by its *badness*, which is a number between 0 and 10000. A badness of 0 is a perfect fit, and a badness of 10000 means that the line probably will never fit. The default value of `\hbadness` is 1000. If you set `\hbadness` to 10000, TeX does not report underfull lines.

Sometimes TeX allows a line to extend past the right margin. This is an aesthetic decision on the part of TeX's author. The amount is determined by the `\hfuzz` parameter, which defaults to 0.1 pt. If the text does not fit within the line, the `\tolerance` parameter determines how TeX will handle the overfull `\hbox`. The default value of `\tolerance` is 200. Setting `\tolerance` to 1000 suppresses overfull `\hbox` warnings and the printing of slugs.

3.4 Page layout.

In addition to the left margin and line length dimensions that are described in the previous section, TeX also lets you specify top and bottom margins, and vertical spacing.

Like the `\hsize` and `\hoffset` dimensions described in the previous section, TeX also provides the `\vsize` and `\voffset` commands. The default for `\vsize` is 8.9 in., and `\voffset` defaults to 0.

Normally, teTeX places the beginning of the first line of text 1 in. below the top of the paper and 1 in. from the left edge. You can start the text closer to the top of the page with the command:

```
\voffset=-0.5in
```

If you want to add vertical space in a document, the commands `\smallskip`, `\medskip`, and `\bigskip` will add approximately 3, 6, and 12 points of blank vertical space. These measurements are approximate; TeX will adjust them by as much as 1 pt. so the page is filled correctly.

The `\vfill` command adds an adjustable vertical space between paragraphs on a page. It is infinitely stretchable, so it will add vertical space to fill as much of the rest of the page as possible. If you want to specify a dimension, use `\vskip` as in:

```
\vskip 10pt
```

The commands `\hss` and `\vss` are similar to `\hfil` and `\vfill`, but they provide dimensions that are infinitely shrinkable as well as infinitely stretchable.

The `\vskip` and `\vfill` commands produce flexible lengths. They do not add space where no text exists; for example, at the top of a page. Use `\vglue` if you want to add an absolute space.

TeX fills the `\vsize` dimension with as much text as possible before it starts a new page. To force a page break, use the `\vfill \eject` sequence. If `\vfill` is not used, the text before the `\break` will be spaced to fill the page.

TOC

If you want TeX to be more flexible about its vertical page sizing, place the `\raggedbottom` command in your document. TeX will then adjust the bottom margin of each page slightly to make vertical spacing more consistent.

3.5 Page numbers, headers, and footers.

teTeX by default places the page number at the bottom center of the page. If you want to change the location and style of the page number, you can specify alternate headers and footers by changing definitions of `\headline` and `\footline`. The default value for `\footline` contains the `\folio` command, which prints the page number. The default value for `\headline` is `\hfil`, so a blank line is printed.

The `\pageno` command is a synonym for TeX's internal page counter. You can change the page number by changing the value of `\pageno`. If `\pageno` is negative, the numbers are printed as Roman numerals.

```
\pageno=10
\pageno=-1
```

The command `\nopagenumbers` is shorthand for:

```
\headline={\hfil}
\footline={\hfil}
```

The default footline also contains the font command `\tenrm`, which sets the page number's font to 10-point Roman. If you want to print the page number in 12-point Roman, for example, you would first define a 12-point Roman font, and use that in the definition of `\footline`. Font commands are discussed in Section Font commands.

```
\font\twelvrm=cmr12
\footline={\hss\twelvrm\folio\hss}
```

You can put a *rule*, a horizontal line, at the top of each page by redefining `\headline` as:

```
\headline={\hrulefill}
```

To specify different headers for even and odd pages use the `\ifodd`command, which has the form:

```
\ifodd[condition][true-action]\else[false-action
```

An example `\headline` that uses different headers for even and odd pages would be:

```
\headline={\ifodd\pageno odd-page-header \else even-page-header}
```

The `\ifodd` statement uses the first argument if the page number is odd, and the second argument otherwise.

3.6 Titles and macros.

TeX provides only the `\beginsection` macro for section headings. It leaves a space above its argument, prints the text of the heading in bold type, adds a `\smallskip` after the text of the heading, and starts the next paragraph with no indent.

The LaTeX chapter and section commands described below add section numbering, and will print the section names and numbers in the page headings, and automatically add the sections to the Table of Contents.

In plain TeX, you must write these functions yourself. The `\def` command allows you to define new commands. Suppose you want to print a chapter title. First you define the font that you want to use. A large, sans serif font for chapter titles would be defined like this:

```
\font\chapterfontsans=cmss12 scaled \magstep 4
```

You can use the `\chapterfontsans` command anywhere you want to switch to this font, which is approximately 24 points in height. However, in this example, it will be used primarily in the command `\chaptertitlesans`. Here is its definition:

```
\def\chaptertitlesans#1{\hbox{}\bigskip\bigskip
  \noindent{\leftline{\chapterfontsans#1}}
  \par\bigskip\bigskip\noindent}
```

The first line, `\hbox{}\bigskip`, anchors a 12-point space at the top of the page by placing an empty `\hbox{}` there. The line with the chapter title is not indented, nor is the paragraph which immediately follows it. If you place a blank line between the `\sschaptertitle` macro and the next paragraph, the final `\noindent` applies to the blank line, not the text of the following paragraph. To format correctly, use the `\sschaptertitle` as in this example:

The `#1` statement in the definition is replaced by the first argument to `\chaptertitlesans`; that is, the title of the chapter. Parameters TeX definitions are declared with `#1`, `#2`, `#3`, and so on. An example usage of `\chaptertitlesans` would be:

```
\chaptertitlesans{Chapter 1}
This is the starting text of the first paragraph of the chapter.
The paragraph will not be indented.  The chapter's title is
"Chapter 1."
```

4. LaTeX commands.

4.1 Document structure.

Documents formatted for LaTeX have a few more rules, but with complex documents, LaTeX can greatly simplify the formatting process.

Essentially, LaTeX is a document markup language which tries to separate the output style from the document's logical content. For example, formatting a section heading with TeX would require specifying 36 points of white space above the heading, then the heading itself set in bold, 24-point type, then copying the heading text and page number to the Table of Contents, then leaving 24 points of white space after the heading. By contrast, LaTeX has the `\section{}` command, which does all of the work for you. If you need to change the format of the section headings throughout your document, you can change the definition of `\section{}` instead of the text in the document. You can see where this would save hours of reformatting for documents of more than a dozen pages in length.

All LaTeX documents have three sections: a *preamble*, the *body* text, and a *postamble*. These terms are standard jargon and are widely used by TeXperts.

The preamble, at a minimum, specifies the type of document to be produced---the *document class*---and a statement which signals the beginning of the document's body text. For example:

```
\documentclass{article}
\begin{document}
```

The document's postamble is usually very simple. Except in specialized cases, it contains only the statement:

```
\end{document}
```

Note the `\begin{document}` and `\end{document}` pairing. In LaTeX, this is called an *environment*. All text must appear within an environment, and many commands are effective only in the environments in which they're called. The `document` environment is the only instance where LaTeX enforces this convention, however. That is, it's the only environment that is required in a document. (An exception is `letter` class, which also requires you to declare `\begin{letter}` and `\end{letter}`. See the section Letters.) However, many formatting features are specified as environments. They're described in the following sections.

The document classes can be called with arguments. For example, instead of the default, 10-point type used as the base point size, as in the previous example, we could have specified

```
\documentclass[12pt]{article}
```

to produce the document using 12 points as the base point size. The document class, *article*, makes the necessary adjustments.

There are a few document classes which are commonly used. They're described below. The *report* class is

similar to *article* class, but produces a title page and starts each section on a new page. The *letter* class includes special definitions for addresses, salutations, and closings, a few of which are described below.

TOC

You can include canned LaTeX code, commonly known as a *package*, with the `\usepackage{}` command.

```
\usepackage{fancyhdr}
```

The command above would include the LaTeX style file `fancyhdr.sty` from one of the `TEXINPUTS` directories, which you and teTeX specified during installation and setup processes.

```
\documentclass{article}
\usepackage{fancyhdr}
\begin{document}
```

Note that the `\usepackage{}` declarations are given before the `\begin{document}` statement; that is, in the document preamble.

`fancyhdr.sty` extends the `\pagestyle{}` command so that you can create custom headers and footers. Most LaTeX document classes provide headers and footers of the following standard page styles:

```
\pagestyle{plain}       % default pages style -- page number centered at
                        % the bottom of the page.
\pagestyle{empty}       % no headers or footers
\pagestyle{headings}    % print section number and page number at the
                        % top of the page.
\pagestyle{myheadings}  % print custom information in the page heading.
```

Everything on a line to the right of the percent sign is a comment.

The `\pagestyle{}` command doesn't take effect until the following page. To change the headers and footers on the current page, use the command

```
\thispagestyle{the_pagestyle}
```

4.2 Characters and type styles.

Character styles are partially a function of the fonts specified in the document. However, bold and italic character emphasis should be available for every font present on the system. Underlining, too, can be used, though its formatting presents special problems. See section LaTeX extension packages and other resources, below.

You can specify text to be emphasized in several ways. The most portable is the `\em` command. All text within its scope is italicized by default. For example:

```
This word will be {\em emphasized.}
```

If you have italicized text that runs into text which is not italicized, you can specify an italic correction factor to be used. The command for this is `\/`; that is, a backslash and a forward slash.

```
This example {\em will\/} print correctly.
```

```
This example will {\em not} print correctly.
```

Slightly less portable, but still acceptable in situations where they're used singly, are the commands `\it`, `\bf`, and `\tt`, which specify that the characters within their scope be printed using italic, bold, and monospaced (teletype) typefaces, respectively.

```
{\tt This text will be printed monospaced,}
{\it this text will be italic,} and
{\bf this text will be bold\dots} all in one paragraph.
```

The command `\dots` prints a series of three periods for ellipses, which will not break across a line.

The most recent version of LaTeX, which is what you have, includes commands which account for instances where one emphasis command would supersede another.

```
This is {\it not {\bf bold italic!}}
```

What happens is that teTeX formats the text with the italic typeface until it encounters the `\bf` command, at which point it switches to boldface type.

To get around this, the NFSS scheme of selecting font shapes requires three parameters for each typeface: shape, series, and family. Not all font sets will include all of these styles. LaTeX will print a warning, however, if it needs to substitute another font.

You can specify the following font shapes:

```
\textup{text}           % upright shape (the default)
\textit{text}           % italic
\textsl{text}           % slanted
\textsc{text}           % small caps
```

These are the two series that most fonts have:

```
\textmd{text}           % medium series (the default)
\textbf{text}           % boldface series.
```

There are generally three families of type available.

```
\textrm{text}           % Roman (the default)
\textsf{text}           % sans serif
\texttt{text}           % typewriter (monospaced, Courier-like)
```

Setting font styles using these parameters, you can combine effects.

```
\texttt{\textit{This example likely will result in a font
substitution, because many fonts don't include a typewriter italic
typeface.}}
```

The font family defaults to Computer Modern, which is a bit-mapped font. Other font families are usually

Postscript-format Type 1 fonts. See section Using PostScript fonts for details on how to specify them.

There are also many forms of accents and special characters which are available for typesetting. This is only a few of them. (Try typesetting these on your own printer.)

```
\'{o}   \`{e}   \^{o}   \"{u}   \={o}   \c{c}   `? `!
\copyright      \pounds         \dag
```

Finally, there are characters which are used as meta- or escape characters in TeX and LaTeX. One of them, the dollar sign, is mentioned above. The complete set of meta characters, which need to be escaped with a backslash to be used literally, is:

```
# $ % & _ { }
```

There are also different alphabets available, like Greek and Cyrillic. LaTeX provides many facilities for setting non-English text, which are covered by some of the other references mentioned here

4.3 Margins and line spacing.

Changing margins in a TeX or LaTeX document is not a straightforward task. A lot depends on the relative indent of the text you're trying to adjust the margin for. The placement of the margin-changing command is also significant.

For document-wide changes to LaTeX documents, the `\evensidemargin` and `\oddsidemargin` commands are available. They affect the left-hand margins of the even-numbered and odd-numbered pages, respectively. For example,

```
\evensidemargin=1in
\oddsidemargin=1in
```

adds on inch to the left-hand margin of the even and odd pages *in addition* to the standard one-inch, left-hand margin. These commands affect the entire document and will shift the entire body of the text right and left across a page, regardless of any local indent, so they're safe to use with LaTeX environments like `verse` and `list`.

Below is a set of margin-changing macros which I wrote. They have a different effect than the commands mentioned above. Because they use plain TeX commands, they're not guaranteed to honor the margins of any LaTeX environments which may be in effect, but you can place them anywhere in a document and change the margins from that point on.

```
%%  margins.sty -- v. 0.1   by Robert Kiesling
%%  Copies of this code may be freely distributed in verbatim form.
%%
%%  Some elementary plain TeX margin-changing commands. Lengths are
%%  in inches:
%%  \leftmargin{1}   %% sets the document's left margin in 1 inch.
%%  \leftindent{1}   %% sets the following paragraphs' indent in
%%                     1 inch.
```

```
%%  \rightindent{1}  %% sets the following paragraphs' right margins
%%                   %% in 1 inch.
%%  \llength{3}      %% sets the following lines' lengths to 3 inches.
%%
\message{Margins macros...}
\def\lmargin#1{\hoffset = #1 in}
\def\lindent#1{\leftskip = #1 in}
\def\rindent#1{\rightskip = #1 in}
\def\llength#1{\hsize = #1 in}
%%
%% (End of margins macros.}
```

Place this code in a file called `margins.sty` in your local `$TEXINPUTS` directory. The commands are explained in the commented section of the file. To include them in a document, use the command

```
\usepackage{margins}
```

in the document preamble.

While we're on the subject, if you don't want the right margin to be justified, which is the default, you can tell LaTeX to use ragged right margins by giving the command:

```
\raggedright
```

Setting line spacing also has its complexities.

The *baselineskip* measurement is the distance between lines of text. It is given as an absolute measurement. For example,

```
\baselineskip=24pt
```

or even better:

```
\setlength{\baselineskip}{24pt}
```

The difference between the two forms is that *setlength* will respect any scoping rules that may be in effect when you use the command.

The problem with using baselineskip is that it also affects the distance between section headings, footnotes, and the like. You need to take care that baselineskip is correct for whatever text elements you're formatting. There are, however, LaTeX macro packages, like `setspace.sty,` which will help you in these circumstances. See section LaTeX extension packages and other resources.

4.4 Document classes.

LaTeX provides document classes which provide standardized formats for documents. They provide environments to format lists, quotations, footnotes, and other text elements. Commonly used document classes are covered in the following sections.

Articles and reports.

As mentioned above, the `article` class and the `report` class are similar. The main differences are that the report class creates a title page by default and begins each section on a new page. Mostly, though, the two document classes are similar.

To create titles, abstracts, and bylines in these document classes, you can type, for example,

```
\title{The Breeding Habits of Cacti}
\author{John Q. Public}
\abstract{Description of how common desert cacti search
for appropriate watering holes to perform their breeding
rituals.}
```

in the document preamble. Then, the command

```
\maketitle
```

given at the start of the text, will generate either a title page in the report class, or the title and abstract at the top of the first page, in the article class.

Sections can be defined with commands that include the following:

```
\section
\subsection
\subsubsection
```

These commands will produce the standard, numbered sections used in technical documents. For unnumbered sections, use

```
\section*
\subsection*
\subsubsection*
```

and so on.

LaTeX provides many environments for formatting displayed material. You can include quoted text with the `quotation` environment.

```
\begin{quotation}
Start of paragraph to be quoted...

... end of paragraph.
\end{quotation}
```

For shorter quotes, you can use the `quote` environment.

To format verse, use the `verse` environment.

```
\begin{verse}
Because I could not stop for death\\
```

```
He kindly stopped for me
\end{verse}
```

Notice that you must use the double backslashes to break lines in the correct places. Otherwise, LaTeX fills the lines in a verse environment, just like any other environment.

Lists come in several flavors. To format a bulleted list, the `list` environment is used:

```
\begin{list}
\item
This is the first item of the list.
\item
This is the second item of the list...
\item
... and so on.
\end{list}
```

A numbered list uses the `enumerate` environment:

```
\begin{enumerate}
\item
Item No. 1.
\item
Item No. 2.
\item
\dots
\end{enumerate}
```

A descriptive list uses the `description` environment.

```
\begin{description}
\item{Oven} Dirty, needs new burner.
\item{Refrigerator}  Dirty.  Sorry.
\item{Sink and drainboard}  Stained, drippy, cold water faucet.
\end{description}
```

Letters.

The `letter` class uses special definitions to format business letters.

The `letter` environment takes one argument, the address of the letter's addressee. The `address` command, which must appear in the document preamble, defines the return address. The `signature` command defines the sender's name as it appears after the closing.

The LaTeX source of a simple business letter might look like this.

```
\documentclass[12pt]{letter}
\signature{John Q. Public}
\address{123 Main St.\\Los Angeles, CA.  96005\\Tel: 123/456-7890}
```

```
\begin{document}
\begin{letter}{ACME Brick Co.\\100 Ash St.\\San Diego, CA 96403}
\opening{Dear Sir/Madam:}

With regard to one of your bricks that I found on my living room
carpet surrounded by shards of my broken front window...

(Remainder of the body of the letter.)

\closing{Sincerely,}

\end{letter}
\end{document}
```

Note that the addresses include double backslashes, which specify where the line breaks should occur.

---

5. LaTeX extension packages and other resources.

We mentioned above that using underlining as a form of text emphasis presents special problems. Actually, TeX has no problem underlining text, because it is a convention of mathematical typesetting. In LaTeX, you can underline words with the command:

```
\underline{text to be underlined}
```

The problem is that underlining will not break across lines, and, in some circumstances, underlining can be uneven. However, there is a LaTeX macro package, ready-made, that makes underlining the default mode of text emphasis. It's called `ulem.sty`, and is one of the many contributed LaTeX packages that are freely available via the Internet.

To use `ulem.sty`, include the command:

```
\usepackage{ulem}
```

in the document preamble.

The packages which are available for LaTeX include:

ifthen

Include conditional statements in your documents.

initials

Defines a font for initial dropped capitals.

sanskrit

Font and preprocessor for producing documents in Sanskrit.

recipe

A LaTeX2e class to typeset recipes.

refman

Variant report and article styles.

To make the path given in the Catalogue into a fully-qualified URL, concatenate the path to the host name URL and top-level path of the CTAN archive you wish to contact. For example, the top-level CTAN directory of the site ftp.tex.ac.uk is `ctan/tex-archive`. The complete URL of the directory of the **refman** package would be:

```
ftp://ftp.tex.ac.uk/ctan/tex-archive/   +
macros/latex/contrib/supported/refman   =

ftp://ftp.tex.ac.uk/ctan/tex-archive/macros/latex/contrib/supported/refman/
```

Some packages have more than one file, so only the path to the package's directory is given.

When you have the URL in hand, you can retrieve the package from one of the CTAN archive sites listed i section Appendix A. You can download a complete list of the archive's contents as the file `FILES.byna` the archive's top-level directory. You can also search the archive on line for a keyword with the `ftp` comma

```
quote site index <keyword>
```

6. Mixing text and graphics with dvips.

In general, this section applies to any TeX or LaTeX document which mixes text and graphics. teTeX, like most other TeX distributions, is configured to request Computer Modern fonts by default. When printing documents with Type 1 scalable fonts or graphics, font and graphics imaging is the job of `dvips`. `dvips` can use either Computer Modern bit mapped fonts or Type 1 scalable fonts, or any combination of the two. First, let's concentrate on printing and previewing some graphics.

You will probably want to follow this procedure any time a LaTeX source document has the statement

```
\includepackage{graphics}
```

in the document preamble. This statement tells LaTeX to include the text of the `graphics.sty` package in the source document. There are other commands to perform graphics operations, and the statements in plain-TeX documents may not clue you in whether you need to use `dvips`. The difference will be apparent in the output, though, when the document is printed with missing figures and other graphics.

So, for now, we'll concentrate on printing documents which use the LaTeX `graphics.sty` package. You might want to take a look at the original TeX input. It isn't included in the teTeX distribution, but it is available at

```
~CTAN/macros/latex/packages/graphics/grfguide.tex.
```

What the teTeX distribution does include is the `.dvi` output file, and it is already TeXed for you. There is a reason for this, and it has to do with the necessity of including Type 1 fonts in the output in order for the document to print properly. If you want to LaTeX `grfguide.tex,` see the next section. For now, however, we'll work on getting usable output using `dvips`.

The file `grfguide.dvi` is located in the directory

```
texmf/doc/latex/graphics
```

The first step in outputting `grfguide.dvi` is to translate it to Postscript. The program `dvips` is used for this. It does just exactly what its name implies. There are many options available for invoking `dvips`, but the simplest (nearly) form is

```
dvips -f -r <grfguide.dvi >grfguide.ps
```

The `-f` command switch tells `dvips` to operate as a filter, reading from standard input and writing to standard output. `dvips` output can be configured so its output defaults to `lpr`.

If you can print Postscript directly to your printer via `lpr`, you can simply type

```
dvips -r grfguide.dvi
```

The `-r` option tells dvips to output the pages in reverse order so they stack correctly when they exit a printer. Use it or not, as appropriate for your output device.

TOC

Depending on whether you still have the fonts that `dvilj2` generated from the last document, `dvips` and metafont may or may not need to create new fonts needed by `grfguide.dvi.` Eventually, though, `dvips` will output a list of the pages translated to Postscript, and you will have your Postscript output ready to be rendered on whatever output device you have available.

If you're lucky (and rich), then you have a Postscript-capable printer already and will be able to print `grfguide.ps` directly. You can either spool the output to the printer using `lpr`. If for some reason your printer software doesn't work right with Postscript files, you can, in a pinch, simply dump the file to printer, with

```
cat grfguide.ps >/dev/lp0
```

or whichever port your printer is attached to, though this is not recommended for everyday use.

If you want or need to invoke Ghostscript manually, this is the standard procedure for its operation. The first thing you want to do is invoke Ghostscript to view its command line arguments, like this:

```
gs -help | less
```

You'll see a list of supported output devices and sundry other commands. Pick the output device which most nearly matches your printer. I generally produce black-and-white text and use the `cdjmono` driver, which drives a color Deskjet in monochrome (black and white) mode.

The command line I would use is:

```
gs -dNOPAUSE -sDEVICE=cdjmono -sOutputFile=/tmp/gs.out grfguide.ps -c quit
```

This will produce my HP-compatible output in the `/tmp` directory. It's a good idea to use a directory like `/tmp,` because `gs` can be particular about access permissions, and you can't (and shouldn't) always count on being logged in as `root` to perform these steps. Now you can print the file:

```
lpr /tmp/gs.out
```

Obviously, this can all go into a shell script. On my system, I have two simple scripts written, `pv` and `pr,` which simply outputs the Postscript file either to the display or the printer. Screen previewing is possible without X, but it's far from ideal. So, it's definitely worth the effort to install XFree86 to view the output on the screen..

The order of commands in a `gs` command line is significant, because some of the options tell Ghostscript to look for pieces of Postscript code from its library.

The important thing to remember is that `grfguide.dvi` makes requests for both Computer Modern bit mapped and Type 1 scaled fonts. If you can mix scalable and bit mapped fonts in a document, you're well on the way to becoming a TeXpert.

6.1 What if my printer isn't supported?

The teTeX distribution comes with only a limited selection of DVI output drivers: `dvips`, drivers for Hewlett Packard LaserJets, and nothing else. You have two options if you have a printer which isn't LaserJet-compatible: You can use `dvips` and Ghostscript, which I would recommend anyway, for reasons already mentioned, or you can investigate other dviware sources.

A limited number of DVI drivers have been ported to Linux and are available as pre-built binaries. They a located in the Linux archives at ftp://sunsite.unc.edu/pub/Linux/apps/tex/dvi/.

The master dviware libraries are maintained at the University of Utah archives. If you can't find a DVI driver there that supports your printer, chances are that it doesn't exist. You can also write your *own* DVI driver using the templates available there. The library's URL is ftp://ftp.math.utah.edu/pub/tex/dvi/.

---

7. Using Postscript fonts.

It used to be that public domain, Type 1 fonts were much poorer quality than Computer Modern bit mapped fonts. This situation has improved in the last several years, though, but matching the fonts is up to you. Having several different font systems on one machine can seem redundant and an unnecessary waste of disk space. And the Computer Modern fonts can seem, well, a little too *formal* to be suitable for everyday use. It reminds me sometimes of bringing out the good China to feed the dog. At least you don't need to spend a bundle on professional quality fonts any longer.

One of the major improvements of LaTeX2e over its predecessor was the inclusion of the New Font Selection Scheme. (It's now called PSNFSS.) Formerly, TeX authors would specify fonts with commands like

```
\font=bodyroman = cmr10 scaled \magstep 1
```

which provides precision but requires the skills of a type designer and mathematician to make good use of. Also, it's not very portable. If another system didn't have the font `cmr10` (this is TeX nomenclature for Computer Modern Roman, 10 point, with the default medium stroke weight), somebody would have to re-code the fonts specifications for the entire document. PSNFSS, however, allows you specify fonts by family (Computer Modern, URW Nimbus, Helvetica, Utopia, and so forth), weight (light, medium, bold), orientation (upright or oblique), face (Roman, Italic), and base point size. (See the section Characters and type styles for a description of the commands to specify typefaces.) Many fonts are packaged as families. For example, a Roman-type font may come packaged with a sans serif font, like Helvetica, and a monospaced font, like Courier. You, as the author of a LaTeX document, can specify an entire font family with one command.

There are, as I said, several high-quality font sets available in the public domain. One of them is Adobe Utopia. Another is Bitstream Charter. Both are commercial quality fonts which have been donated to the public domain.

These happen to be two of my favorites. If you look around one of the CTAN sites, you will find these and other fonts archived there. There are enough fonts around that you'll be able to design documents the way you want them to look, and not just English text, either. TeX was originally designed for mathematical typesetting, so there is a full range of mathematical fonts available, as well as Cyrillic, Greek, Kana, and other alphabets too numerous to mention.

The important thing to look for is files which have either the `.pfa` or `.pfb` extension. They indicate that these are the scalable fonts themselves, not simply the metrics files. Type 1 fonts use `.pfm` metric files, as opposed to the `.tfm` metric files which bit mapped fonts use. The two font sets I mentioned above are included in teTeX distributions, as well as separately.

What I said above, concerning the ease of font selection under PSNFSS, is true in this instance. If we want to use the Charter fonts in our document instead of Computer Modern bit mapped, all that is necessary is include the LaTeX statement

```
\renewcommand{\familydefault}{bch}
```

in the document preamble, where ``bch'' is the common designation for Bitstream Charter. The Charter fonts reside in the directory

```
/usr/lib/teTeX/texmf/fonts/type1/bitstrea/charter
```

There you'll see the `.pfb` files of the Charter fonts: `bchb8a.pfb` for Charter Bold, `bchr8a.pfb` for Charter Roman, `bchbi8a.pfb` for Charter Bold Italic. The ``8a'' in the font names indicates the character encoding. At this point you shouldn't need to worry much about them, because the encodings mostly differ for 8-bit characters, which have numeric values above 128 decimal. They mostly define accents and non-English characters. The Type 1 font encodings generally work well for Western alphabets because they conform to the ISO 8859 standards for international character sets, so this is an added benefit of using them.

To typeset a document which has Charter fonts selected, you would give the command

```
pslatex document.tex
```

`pslatex` is a variant of teTeX's standard `latex` command which defines the directories where the Type 1 fonts are, as well as some additional LaTeX code to load. You'll see the notice screen for `pslatex` followed by the status output of the TeX job itself. In a moment, you'll have a `.dvi` file which includes the Charter font requests. You can then print the file with `dvips`, and `gs` if necessary.

Installing a Type 1 font set is not difficult, as long as you follow a few basic steps. You should unpack the fonts in a subdirectory of the `/usr/lib/teTeX/texmf/fonts/type1` directory, where your other Type 1 fonts are located, and then run `texhash` to let the directory search routines know that the fonts have been added. Then you need to add the font descriptions to the file `psfonts.map` so `dvips` knows they're on the system. The format of the `psfonts.map` file is covered in a couple different places in the references mentioned above. Again, remember to run the `texhash` program to update the teTeX directory database.

It is definitely an advantage to use the X Windows System with teTeX--- XFree86 under Linux---because it allows for superior document previewing. It's not required, but in general, anything that allows for easier screen previewing is going to benefit your work, in terms of the quality of the output. However, there is a tradeoff with speed of editing, which is much quicker on character-mode displays.

---

8. Appendix A: CTAN site list.

This is the text of the file `CTAN.sites`, which is available in the top-level directory of each CTAN archive or mirror site.

```
In order to reduce network load, it is recommended that you use the
Comprehensive TeX Archive Network (CTAN) host which is located in the
closest network proximity to your site.  Alternatively, you may wish to
obtain a copy of the CTAN via CD-ROM (see help/CTAN.cdrom for details).

Known mirrors of the CTAN reside on (alphabetically):
  cis.utovrm.it (Italia)                 /TeX
  ctan.unsw.edu.au (NSW, Australia)      /tex-archive
  dongpo.math.ncu.edu.tw (Taiwan)        /tex-archive
  ftp.belnet.be (Belgium)                /packages/TeX
  ftp.ccu.edu.tw (Taiwan)                /pub/tex
  ftp.cdrom.com (West coast, USA)        /pub/tex/ctan
  ftp.comp.hkbu.edu.hk (Hong Kong)       /pub/TeX/CTAN
  ftp.cs.rmit.edu.au  (Australia)        /tex-archive
  ftp.cs.ruu.nl (The Netherlands)        /pub/tex-archive
  ftp.cstug.cz (The Czech Republic)      /pub/tex/CTAN
  ftp.duke.edu (North Carolina, USA)     /tex-archive
  ftp.funet.fi (Finland)                 /pub/TeX/CTAN
  ftp.gwdg.de (Deutschland)              /pub/dante
  ftp.jussieu.fr (France)                /pub4/TeX/CTAN
  ftp.kreonet.re.kr (Korea)              /pub/CTAN
  ftp.loria.fr (France)                  /pub/unix/tex/ctan
  ftp.mpi-sb.mpg.de (Deutschland)        /pub/tex/mirror/ftp.dante.de
  ftp.nada.kth.se (Sweden)               /pub/tex/ctan-mirror
  ftp.oleane.net (France)                /pub/mirrors/CTAN/
  ftp.rediris.es (Espa\~na)              /mirror/tex-archive
  ftp.rge.com (New York, USA)            /pub/tex
  ftp.riken.go.jp (Japan)                /pub/tex-archive
  ftp.tu-chemnitz.de (Deutschland)       /pub/tex
  ftp.u-aizu.ac.jp (Japan)               /pub/tex/CTAN
  ftp.uni-augsburg.de (Deutschland)      /tex-archive
  ftp.uni-bielefeld.de (Deutschland)     /pub/tex
  ftp.unina.it (Italia)                  /pub/TeX
  ftp.uni-stuttgart.de (Deutschland)     /tex-archive (/pub/tex)
  ftp.univie.ac.at (\"Osterreich)        /packages/tex
  ftp.ut.ee (Estonia)                    /tex-archive
  ftpserver.nus.sg (Singapore)           /pub/zi/TeX
  src.doc.ic.ac.uk (England)             /packages/tex/uk-tex
  sunsite.auc.dk (Denmark)               /pub/tex/ctan
  sunsite.cnlab-switch.ch (Switzerland) /mirror/tex
```

```
  sunsite.icm.edu.pl (Poland)           /pub/CTAN
  sunsite.unc.edu (North Carolina, USA) /pub/packages/TeX
  wuarchive.wustl.edu (Missouri, USA)   /packages/TeX

Known partial mirrors of the CTAN reside on (alphabetically):
  ftp.adfa.oz.au (Australia)            /pub/tex/ctan
  ftp.fcu.edu.tw (Taiwan)               /pub2/tex
  ftp.germany.eu.net (Deutschland)      /pub/packages/TeX
  ftp.gust.org.pl (Poland)              /pub/TeX
  ftp.jaist.ac.jp (Japan)               /pub/TeX/tex-archive
  ftp.uu.net (Virginia, USA)            /pub/text-processing/TeX
  nic.switch.ch (Switzerland)           /mirror/tex
  sunsite.dsi.unimi.it (Italia)         /pub/TeX
  sunsite.snu.ac.kr (Korea)             /shortcut/CTAN

Please send updates to this list to <ctan@urz.uni-heidelberg.de>.

The participating hosts in the Comprehensive TeX Archive Network are:
  ftp.dante.de  (Deutschland)
       -- anonymous ftp                 /tex-archive (/pub/tex /pub/archive)
       -- gopher on node gopher.dante.de
       -- e-mail via ftpmail@dante.de
       -- World Wide Web access on www.dante.de
       -- Administrator: <ftpmaint@dante.de>

  ftp.tex.ac.uk (England)
       -- anonymous ftp                 /tex-archive (/pub/tex /pub/archive)
       -- gopher on node gopher.tex.ac.uk
       -- NFS mountable from nfs.tex.ac.uk:/public/ctan/tex-archive
       -- World Wide Web access on www.tex.ac.uk
       -- Administrator: <ctan-uk@tex.ac.uk>
```

9. Appendix B: Installing the CTAN teTeX distribution.

The generic, teTeX distribution isn't any harder to install than the Linux packages. See section Generic CTAN distribution, below.

You should consider installing the generic teTeX distribution from the CTAN archives if:

- Your system isn't based on one of the standard Linux distributions.
- You don't have root privileges on your system.
- You want or need to have the very latest version of teTeX, or LaTeX.
- You don't have enough disk space available for a full installation.
- You want to install teTeX somewhere instead of the `/usr` file system.
- You would like to share your teTeX installation with other UNIX variants or platforms on a network. In this case, you should strongly consider installing from the *source* distribution. See section Installing the source distribution, below.
- You want the latest versions of teTeX's public domain Type 1 fonts, which are significantly better than the fonts included in earlier releases.

A complete installation of the binary distribution requires 40-50 Mb of disk space, and building the distribution from the source code takes about 75 Mb, so you should make sure that the disk space is available before you start. You don't need to have the GCC compiler or the X Windows System installed (although X certainly helps because it is much easier to preview documents on-screen). All you need is an editor that is capable of producing plain ASCII, text (see section 2). What could be simpler?

You can retrieve the files from one of the CTAN archives listed in section Appendix A. In the examples below, the files were retrieved from the CTAN archive at ftp.tex.ac.uk.

9.1 Installing the binary distribution.

Minimal installation.

First, FTP to ftp.tex.ac.uk and `cd` to the directory

```
ctan/tex-archive/systems/unix/teTeX/distrib/
```

Retrieve the files

```
INSTALL.bin
install.sh
```

and place them in the top-level directory where you want to install teTeX, for example, `/var/teTeX` if you plan to install teTeX in the `/var` file system.

Print out the `INSTALL.bin` file. Keep this file handy, because it describes how to install a minimal teTeX installation. The minimal installation requires only 10-15 MB of disk space, but it is recommended that you

TOC

install the complete teTeX package if at all possible. For a minimum installation, you'll need the files

```
ctan/tex-archive/systems/unix/teTeX/distrib/base/latex-base.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/base/tetex-base.tar.gz
```

You'll also need one of two archives which contain the executable teTeX programs. Retrieve the archive file

```
ctan/tex-archive/systems/unix/teTeX/distrib/binaries/i386-linux.tar.gz
```

if your system uses the Linux ELF shared libraries, `ld.so` of at least version 1.73, and clibs of at least version 5.09. If it doesn't, retrieve the archive

```
ctan/tex-archive/systems/unix/teTeX/distrib/binaries/i386-linuxaout.tar.gz
```

which is compiled for systems that use the older, a.out-format static libraries.

Then, following the instructions in the file `INSTALL.bin`, execute the command

```
sh ./install.sh
```

while in the top-level teTeX installation directory. (Make sure that the teTeX archives are located there, too.) After a few moments, the installation program will warn you that you are missing some of the teTeX packages. However, if you're planning only a minimal teTeX installation, you should ignore the warnings and proceed. To configure the basic teTeX system, see section Base system configuration, below.

To install the remaining packages, see the next section.

Complete installation.

To perform a complete teTeX installation, retrieve the archive files listed in the previous section, as well as the following files:

```
ctan/tex-archive/systems/unix/teTeX/distrib/doc/ams-doc.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/doc/bibtex-doc.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/doc/eplain-doc.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/doc/fonts-doc.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/doc/general-doc.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/doc/generic-doc.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/doc/latex-doc.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/doc/makeindex-doc.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/doc/metapost-doc.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/doc/programs-doc.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/fonts/ams-fonts.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/fonts/dc-fonts.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/fonts/ec-fonts.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/fonts/misc-fonts.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/fonts/postscript-fonts.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/fonts/sauter-fonts.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/goodies/amstex.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/goodies/bibtex.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/goodies/eplain.tar.gz
```

```
ctan/tex-archive/systems/unix/teTeX/distrib/goodies/latex-extra.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/goodies/metapost.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/goodies/pictex.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/goodies/pstricks.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/goodies/texdraw.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/goodies/xypic.tar.gz
```

All of these files should be placed in the top-level directory where you want teTeX to reside. As with the minimal installation, execute the command

```
sh ./install.sh
```

9.2 Base system configuration.

The `install.sh` script, after determining which teTeX archive series are present, will present you with a menu of options. The only setting you need to make at this point is to set the top-level directory where you want teTeX installed, by selecting the ``D'' option. You must, of course, choose a directory in whose parent directory you have write permissions. For example, if you are installing teTeX in your home directory, you would specify the teTeX installation directory as

```
/home/john.q.public/teTeX
```

and, after returning to the main menu, select ``I'' to proceed with the installation. Note that the directory must not exist already: the `install.sh` script must be able to create it.

An option which you should consider enabling, is setting an alternative directory for generated fonts. Even if you plan to use only Postscript-format, Type 1 scalable fonts, occasionally you'll process a file that requires the Computer Modern fonts. Enabling this option requires that you enter the directory to use. You must have write permissions for the parent directory. Following the example above, you could specify

```
/home/john.q.public/texfonts
```

or, if you want the generated fonts to be accessible by all users on the system, specify a directory like

```
/var/texfonts
```

I would recommend that you *not*, however, use the default `/var/tmp/texfonts` directory for this option, because the generated fonts could be deleted after the next reboot, and the fonts will need to be generated again the next time they're needed.

After you've selected the option ``I'', and `install.sh` has installed the archives, set various permissions, and generated its links and format files, the program will exit with a message telling you to add the teTeX binary directory to your `$PATH` environment variable, and the directories where the man pages and info files reside to your `$MANPATH` and `$INFOPATH` environment variables. For example, add the statements

```
export PATH=$PATH:"/home/john.q.public/teTeX/bin"
export MANPATH=$MANPATH":/home/john.q.public/teTeX/man"
export INFOPATH$=INFOPATH":/home/john.q.public/teTeX/info"
```

to your `~/.bash_profile` if you use `bash` as your shell, or to your `~/.profile` if you use another shell for logins.

Log out, and then log in again, so the environment variables are registered. Then, run the command

```
texconfig confall
```

to insure that the installation is correct.

Next, you can configure teTeX for you specific hardware. See section Post-installation configuration details, below.

9.3 Installing the CTAN source distribution.

To install teTeX V. 0.4 from the source code, `ftp` to a CTAN site like ftp://ftp.tex.ac.uk and retrieve the files

```
ctan/tex-archive/systems/unix/teTeX/distrib/INSTALL.src
ctan/tex-archive/systems/unix/teTeX/distrib/sources/README.texmf-src
ctan/tex-archive/systems/unix/teTeX/distrib/sources/teTeX-lib-0.4pl8.tar.gz
ctan/tex-archive/systems/unix/teTeX/distrib/sources/teTeX-src-0.4pl7.tar.gz
```

Read over the instructions in `INSTALL.src`, then `su` to root and unpack the files in a directory for which you have read-write-execute permissions.

Remember to use the `p` argument to `tar`, and also remember to unset the `noclobber` option of `bash`. You can do this with the counterintuitive command

```
set +o noclobber
```

Note that the argument `+o` to `set` *un*sets a variable, just exactly backwards from what you might expect.

The file `teTeX-lib-0.4pl8.tar.gz` will create the directory `./teTeX`. The file `teTeX-src-0.4pl7.tar.gz` will create the directory `teTeX-src-0.4` Print out the file `INSTALL.src` and keep it nearby for the following steps. `cd` to the `./teTeX-src-0.4` directory, and, per the instructions in the `INSTALL.src` file, edit `./Makefile`. You need to set the `TETEXDIR` variable to the absolute path of the parent teTeX directory. This will be the subdirectory `teTeX` of the directory where you unpacked the source and library archives. For example, if you unpacked the archives in your home directory, you would set `TETEXDIR` to

```
/home/john.q.public/teTeX
```

The rest of the `Makefile` options are pretty generic. With GCC version 2.7.2 and later, you should not need to make any further adjustments unless you have a non-standard compiler and library setup, or want the compiler to perform some further optimizations, or for some other reason. Check that the `USE_DIALOG`, `USE_NCURSES`, and `HAVE_NCURSES` variables are set correctly for your system, because the `dialog` program needs the ncurses library to be installed. A `ncurses` library is included in the source distribution, so the default values in the `Makefile` should work fine. If you can't get `ncurses` to compile or link, `texconfig` can also be run from the command line.

If you've done everything correctly up to this point, you should be able to type `make world` in the top-level source directory, and relax until the teTeX executables are built. This can take a few hours.

After the build has completed, set the environment variables `$PATH, $MANPATH`, and `$INFOPATH` to include the teTeX directories. The statements which would be added to the file `~/.bash_profile`, in the

example, above, would be

```
export PATH=$PATH":/home/john.q.public/teTeX/bin/i386-linux"
export MANPATH=$MANPATH":/home/john.q.public/teTeX/man"
export INFOPATH=$INFOPATH":/home/john.q.public/teTeX/info"
```

The `$PATH` variable is different in the source distribution than in the binary distribution. Note that here the path to the binaries is `teTeX/bin/i386-linux` instead of simply `teTeX/bin` as in the binary distribution.

At this point you can run `texconfig confall` to ensure that the paths have been set correctly, and then proceed to configure teTeX as in the binary distribution. See the section Post-installation configuration details, below.

9.4 Post-installation configuration details.

The first thing you want to do is look at Thomas Esser's `README` file. It contains a lot of hints on how to configure teTeX for your output device (i.e., printer). The `README` file is located in the directory

```
/usr/lib/teTeX/texmf/doc/tetex
```

Read the file over with the command (the path in the following examples is that of the Slackware distribution):

```
less /usr/lib/teTeX/texmf/doc/tetex/README
```

or, print it out with the command

```
cat /usr/lib/teTeX/texmf/doc/tetex/README >/dev/lp0
```

assuming that your printer is connected to `/dev/lp0`. Substitute the device driver file that your printer is connected to, as appropriate.

Or, better still, print it using the `lpr` command:

```
lpr /usr/lib/teTeX/texmf/doc/tetex/README
```

You should have installed the printer daemon that is included with your distribution of Linux. If not, do that now, per the instructions that come with the package.

Print out the `teTeX-FAQ.` Keep the FAQ handy because it contains useful hints for configuring teTeX's output drivers for your printer. We'll get to that in a moment. In more recent releases of teTeX, the `teTeX-FAQ` is viewable via the `texconfig` utility.

Next, you want to define a directory to store your own TeX format files. teTeX searches the directories listed by the `$TEXINPUTS` environment variable for local TeX input files:

```
export TEXINPUTS=".:~/texinputs:"
```

to the system-wide `/etc/profile` file. Individual users can set their own local `$TEXINPUTS` directory, by adding the line in their `~/.profile` or `~/.bash_profile` if `bash` is the default shell. The `$TEXINPUTS` environment variable tells teTeX to look for users' individual TeX style files in the `~/texinputs` directories under each user's home directory. It is *critical* that a colon appear before and after this directory. teTeX is going to append its own directory searches to your own. You want to have teTeX search

the local format files first, so it uses the local versions of any of the standard files you have edited.

Add the `/usr/lib/teTeX/bin` directory to the system-wide path if you're installing teTeX as root. Again, if you're installing a personal copy of teTeX, add the directory where the teTeX binaries are located to *the front* your `$PATH` with the following line in your `~/.profile` or `~/.bash_profile`:

```
export PATH="~/tetex/bin:"$PATH
```

Now, log in as `root` and run `texconfig` per the instructions in the `teTeX-FAQ` and choose the printer that is attached to your system. Make sure that you configure teTeX for both the correct printer and printer resolution.

Finally, run the `texhash` program. This ensures that teTeX's internal database is up to date. The database is actually a `ls-lR` file. You *must* run `texhash` every time you change the system configuration, or teTeX will not be able to locate your changes.

---

10. Appendix C: Distribution and Copyright.

10.1 Distribution.

teTeX is *free software;* this means everyone is free to use the software and free to redistribute it on certain conditions. The package is not in the public domain. It is copyrighted and there are restrictions on its distribution, but these restrictions are designed to permit everything that a good cooperating citizen would want to do. What is not allowed is to try to prevent others from further sharing any version of free software that they might get from you. The precise conditions are found in the GNU General Public License that comes with many of the software packages and also appears following this section.

One way to get a copy of the package is from someone else who has it. You need not ask for our permission to do so, or tell any one else; just copy it. If you have access to the Internet, you can get the latest distribution versions by anonymous FTP. See the chapter ``Sources'' for more information.

You may also receive the software when you buy a computer. Computer manufacturers are free to distribute copies on the same terms that apply to everyone else. These terms require them to give you the full sources, including whatever changes they may have made, and to permit you to redistribute these packages received from them under the usual terms of the General Public License. In other words, the program must be free for you when you get it, not just free for the manufacturer.

You can also order copies of GNU software from the Free Software Foundation on CD-ROM. This is a convenient and reliable way to get a copy; it is also a good way to help fund our work. (The Foundation has always received most of its funds in this way.) An order form is included many distribution, and on our web site in http://www.gnu.ai.mit.edu/order/order.html. For further information, write to

```
Free Software Foundation
59 Temple Place, Suite 330
Boston, MA  02111-1307 USA
USA
```

The income from distribution fees goes to support the foundation's purpose: the development of new free software, and improvements to our existing programs.

If you use GNU software at your workplace, please suggest that the company make a donation. If company policy is unsympathetic to the idea of donating to charity, you might instead suggest ordering a CD-ROM from the Foundation occasionally, or subscribing to periodic updates.

10.2 GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA

02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble.

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software---to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

- This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The ``Program'' below, refers to any such program or work, and a ``work based on the Program'' means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term,``modification.'') Each licensee is addressed as ``you.'' Activities other than copying, distribution and modification are not covered by

this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

- You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program. You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
- You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions: 1. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change. 2. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License. 3. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.) These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program. In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.
- You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following: 1. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, 2. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, 3. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the

program in object code or executable form with such an offer, in accord with Subsection b abo The source code for a work means the preferred form of the work for making modifications to For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable. If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

- You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
- You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
- Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
- If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program. If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances. It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice. This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

TOC

- If the distribution and/or use of the Program is restricted in certain countries either by patents or copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
- The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and ``any later version," you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
- If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally. **NO WARRANTY**
- BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM ``AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
- IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

10.3 How to Apply These Terms to Your New Programs

TOC

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the ``copyright'' line and a pointer to where the full notice is found.

```
[one line to give the program's name and an idea of what it does.
Copyright (C) 19[yy]  [name of author]

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19[yy] [name of author]
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'.  This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands ```show w`'' and ```show c`'' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ```show w`'' and ```show c`''; they could even be mouse-clicks or menu items---whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a ``copyright disclaimer'' for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
```

```
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.

[signature of Ty Coon] 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

The Linux Text-Terminal-HOWTO

David S. Lawyer mailto:bf347@lafn.org

v1.07, August 1999

This document explains what text terminals are, how they work, how to install and configure them, and provides some info on how to repair them. It may be of some use even if you don't have a terminal manual. While it's written for real terminals on a Linux system, some of it is also applicable to terminal emulation and may be helpful for non-Linux systems.

1. Introduction

- 1.1 Copyright, Trademarks, Disclaimer, & Credits
- 1.2 Future Plans: You Can Help
- 1.3 New Versions of this HOWTO
- 1.4 Related HOWTO's
- 1.5 Terminology Used in this Document
- 1.6 What is a Terminal ?

2. Types of Terminals

- 2.1 Dumb Terminals
- 2.2 Text Terminals
- 2.3 Graphics Terminals
- 2.4 Network Computers (NCs)
- 2.5 Emulation on a PC

3. Quick Install

TOC

4. Why Use a Terminal ?

- 4.1 Intro to Why Use a Terminal
- 4.2 Lower Hardware Costs ?
- 4.3 Control of Software
- 4.4 Hardware Upgrades
- 4.5 Other Advantages of Terminals
- 4.6 Major Disadvantages of Terminals
- 4.7 Are Text Terminals Obsolete ?

5. Overview of How Terminals Work (in Linux)

- 5.1 Device Names
- 5.2 Login/Logout
- 5.3 Half/Full Duplex
- 5.4 Terminal Memory
- 5.5 Commands for the Terminal
- 5.6 Lack of Standardization Solved by Terminfo
- 5.7 The Interface
- 5.8 Emulation
- 5.9 The Console

6. Terminal Special Files such as /dev/tty

- 6.1 Serial Port Terminals
- 6.2 Pseudo Terminals
- 6.3 The Controlling Terminal /dev/tty
- 6.4 /dev/ttyIN "Terminals"
- 6.5 The Console: /dev/ttyN
- 6.6 Creating a Device with "mknod"

TOC

7. Some Details on How Terminals Work

- 7.1 Terminal Memory Details
- 7.2 Early Terminals
- 7.3 Escape Sequences and Control Codes (intro)
- 7.4 Display Attributes & Magic Cookies

8. Special Features of Some Terminals

- 8.1 Color
- 8.2 Multiple Sessions
- 8.3 Printer/Auxiliary Port
- 8.4 Pages
- 8.5 Character-Sets
- 8.6 Fonts
- 8.7 Keyboards & Special Keys

9. Terminal Emulation; the Console

- 9.1 Terminal Emulation
- 9.2 Testing Terminal Emulation
- 9.3 The Linux Console

10. Flow Control (Handshaking)

- 10.1 Why Is Flow Control Needed ?
- 10.2 Padding
- 10.3 Overrunning a Serial Port
- 10.4 Stop Sending
- 10.5 Keyboard Lock
- 10.6 Resume Sending
- 10.7 Hardware Flow Control (RTS/CTS etc.)
- 10.8 Is Hardware Flow Control Done by Hardware ?
- 10.9 Obsolete ?? ETX/ACK or ENQ/ACK Flow Control

TOC

11. Physical Connection

- 11.1 Multiport I/O Cards (Adapters)
- 11.2 Direct Cable Connection.
- 11.3 Modem Connection
- 11.4 Terminal Server Connection
- 11.5 Connector and Adapter Types
- 11.6 Making or Modifying a Cable

12. Set-Up (Configure) in General

- 12.1 Intro to Set-Up
- 12.2 Terminal Set-Up (Configure) Overview
- 12.3 Computer Set-Up (Configure) Overview
- 12.4 Many Options
- 12.5 Communication Interface Options
- 12.6 Quick Attempt

13. Terminal Set-Up (Configure) Details

- 13.1 Send Escape Sequences to the Terminal
- 13.2 Older Terminals Set-Up
- 13.3 Getting Into Set-Up (Configuration) Mode
- 13.4 Communication Options
- 13.5 Saving the Set-up
- 13.6 Set-Up Options/Parameters
- 13.7 Emulation {Personality} {{Terminal Modes}}
- 13.8 Display Options
- 13.9 Page Related Options
- 13.10 Reporting and Answerback
- 13.11 Keyboard Options
- 13.12 Meaning of Received Control Codes
- 13.13 Where New Text Goes
- 13.14 Function Keys

- 13.15 Block Mode Options
- 13.16 Locks
- 13.17 Screen Saver {Scrn Saver}
- 13.18 Printer

14. Computer Set-Up (Configure) Details

- 14.1 Getty (in /etc/inittab)
- 14.2 Stty & Setserial
- 14.3 Setserial
- 14.4 Stty
- 14.5 Terminfo & Termcap (brief)
- 14.6 Setting TERM and TERMINFO
- 14.7 Rarely Needed /etc/ttytype File
- 14.8 Login Restrictions
- 14.9 Run Command Only If TERM=my_term_type

15. Terminfo and Termcap (detailed)

- 15.1 Intro to Terminfo
- 15.2 Terminfo Database
- 15.3 Bugs in Existing Terminfo Files (and Hardware)
- 15.4 Modifying Terminfo Files
- 15.5 Init String
- 15.6 TERM Variable
- 15.7 Terminfo/Termcap Documents

16. Using the Terminal

- 16.1 Intro to Using Terminal
- 16.2 Starting Up the Terminal
- 16.3 Terminal (Serial) Device Driver
- 16.4 Problems with Editors
- 16.5 Color ls Corruption
- 16.6 Display Freezes (hung terminal)

- 16.7 Corrupted Terminal Interface
- 16.8 Special (Control) Characters
- 16.9 Viewing Latin-1 Files on a 7-bit Terminal
- 16.10 Inspecting the Interface
- 16.11 Changing the Terminal Settings
- 16.12 Make a Terminal the Console
- 16.13 Multiple Sessions
- 16.14 Logging Out
- 16.15 Chatting between Terminals, Spying

17. Trouble-Shooting (software)

- 17.1 Terminal Was Working OK
- 17.2 Terminal Newly Installed
- 17.3 Is the Terminal OK ?
- 17.4 Missing Text
- 17.5 Getty Respawning Too Rapidly
- 17.6 Fails Just After Login
- 17.7 Can't Login
- 17.8 Garbled Login Prompt
- 17.9 No Sign of any Login Prompt
- 17.10 Serial Monitoring/Diagnostics
- 17.11 Local Mode
- 17.12 Serial Electrical Test Equipment

18. Repair & Diagnose

- 18.1 Repair Books & Websites
- 18.2 Safety
- 18.3 Appearance of Display
- 18.4 Diagnose
- 18.5 Error Messages on the Screen
- 18.6 Capacitors
- 18.7 Keyboards

TOC

19. Appendix A: General

- 19.1 List of Linux Terminal Commands
- 19.2 The Internet and Books
- 19.3 Non-Linux OS's

20. Appendix B: Escape Sequence Commands Terminology

- 20.1 Esc Sequence List
- 20.2 8-bit Control Codes
- 20.3 Printer Esc
- 20.4 Reports
- 20.5 Cursor Movements
- 20.6 Pages (definition)

21. Appendix C: Serial Communications on EIA-232 (RS-232)

- 21.1 Intro to Serial Communication
- 21.2 Voltages
- 21.3 Parity Explained
- 21.4 Forming a Byte (Framing)
- 21.5 Limitations of EIA-232
- 21.6 Synchronization & Synchronous
- 21.7 Block Mode
- 21.8 EIA-232 (RS-232) Books
- 21.9 Serial Software

22. Appendix D: Notes by Brand Name

- 22.1 CIT
- 22.2 IBM Terminals
- 22.3 Teletypes

- 22.4 VT (DEC)
- 22.5 Wyse

1. Introduction

For a quick attempt to install a terminal see Quick Install.

1.1 Copyright, Trademarks, Disclaimer, & Credits

Copyright

Copyright 1998-9 by David S. Lawyer. Please freely copy and distribute (sell or give away) this document. For corrections and minor changes contact the maintainer. Otherwise you may create derivative works and distribute them provided you:

1. Discuss it with the maintainer (if there is one). 2. Put the derivative work at an appropriate Internet site for free downloading and supply the url to both LDP and the HOWTO maintainer. 3. License the work in the spirit of this license or use GPL. 4. Give due credit to previous authors and major contributors.

Disclaimer

While I haven't intentionally tried to mislead you, there are likely a number of errors in this document. Please let me know about them. Since this is free documentation, it should be obvious that I cannot be held legally responsible for any errors.

Trademarks

If certain words are trademarks, the context should make it clear to whom they belong. For example "MS Windows NT" implies that "Windows NT" belongs to Microsoft (MS). Mac is by Apple Computer. Trademarks belong to their respective owners.

Credits

Much of the section "Physical Connection" is from Serial-HOWTO v. 1.11 by Greg Hankins. His "How Do I Set Up A Terminal Connected To My PC?" was incorporated into version 1.00 at various places. The portions not incorporated were on topics already covered in my previous version v0.05.

1.2 Future Plans: You Can Help

Please let me know of any errors in facts, opinions, logic, spelling, grammar, clarity, links, etc. But first, if the date is over a few months old, check to see that you have the latest version. Please send me any info that you think belongs in this document.

Starting with version 1.00, a first attempt was made to help people set up terminals without recourse to a

terminal manual. Much more is needed in this respect. One way to solve this problem would be if terminal manufacturers put their manuals on the Internet. I suggest that you encourage them to do so. task of providing information on how to configure most terminals in this HOWTO is daunting. There are so many different terminals, but there are far fewer models than there used to be in the 1980,s so the task is not totally infeasible.

Please send me any surplus terminal manuals which you may have, especially on terminals made within the past 10 years (but I'll accept older ones also). Also, you might want to write up something on a certain terminal to put in the Appendix D: Notes by Brand Name.

1.3 New Versions of this HOWTO

New versions of the Text-Terminal-HOWTO will be available to browse and/or download at LDP mirror sites. For a list of mirror sites see: http://metalab.unc.edu/LDP/mirrors.html. Various formats are available. If you only want to quickly check the date of the latest version look at: http://metalab.unc.edu/LDP/HOWTO/Text-Terminal-HOWTO.html.

1.4 Related HOWTO's

Go to the web-sites shown above to get these.

- Serial-HOWTO has info on Multiport Serial Cards used for both terminals and banks of modems. It has general technical info on the serial port including troubleshooting it.
- Modem-HOWTO
- Serial-Programming-HOWTO
- Xterminal-HOWTO (unmaintained). It's at http://sunsite.unc.edu/pub/Linux/docs/HOWTO/unmaintained/mini/Xterminal

1.5 Terminology Used in this Document

Configuration means the same as set-up. While Linux commands take options (using - symbols), options in a broader sense include various other types of choices. Install in the broad sense includes setting up (configuring) software and hardware. A statement that I suspect is true (but may not be) ends with 2 question marks: ?? If you know for sure, let me know.

1.6 What is a Terminal ?

A terminal consists of a screen and keyboard that one uses to communicate remotely with a (host) computer. One uses it just like it was a personal computer but the terminal is remote from the host computer (on the other side of the room or even on the other side of the world). Programs execute on the host computer but the results display on the terminal screen. Its computational ability is relatively low (otherwise it would be a computer and not a terminal). This computational ability is generally limited to the ability to display what is sent to it (possibly including full-screen graphics) and the ability to send to

TOC

the host what is typed at the keyboard.

In the days of mainframes from the mid 1970's to the mid 1980's, most people used terminals to communicate with computers. They typed in programs, ran programs, wrote documents, issued printing commands, etc. A cable connected the terminal to the computer (often indirectly). It was called a terminal since it was located at the terminal end of this cable.

If you've been using Linux (except for X-Window use) with a computer monitor and keyboard you already know what a terminal is because you have been using one (or more precisely a "virtual terminal"). The monitor (along with the keyboard) is called the console, but it emulates a terminal. In X-Windows: xterm, rxvt, and zterm emulate terminals.

A real terminal is different from a monitor because it's a different electronic setup. A text terminal is often connected to a serial port of the computer via a long cable. Thus, in contrast to a monitor which is usually located right next to the computer, a terminal may be quite a distance away from its host computer. The video card inside a computer stores the video image seen on the monitor screen. For a terminal, the equivalent of this video card is built right into the terminal but since text terminals are often monochrome without much graphics, the capabilities of its "video card" are rather weak. Also, most text terminals do not have mice.

In network client-server terminology, one might think that the terminal is the client and that the host computer is the server. The terminal has been called a "thin client" by some. But it is not actually a "client" nor is the host a "server". The only "service" the host provides is to receive every letter typed at the keyboard and react to this just like a computer would. The terminal is like a window into the computer just like a monitor (and keyboard) are. You may have already used virtual terminals in Linux (by pressing Alt-F2, etc.). A real terminal is just like running such a virtual terminal but you run it on the terminal screen instead of having to share the monitor screen. In contrast to using a virtual terminal at the console (monitor), this allows another person to sit at the real terminal and use the computer simultaneously with others.

---

2. Types of Terminals

2.1 Dumb Terminals

There are various conflicting definitions of "dumb terminal" but as time goes by, more and more terminals are called dumb. This document mainly covers text terminals which display only text on the screen. It might be titled "Dumb-Terminal-HOWTO" but in some magazines articles any terminal, no matter how smart, including ones which present a full graphical user interface (GUI), are called dumb. If all terminals are "dumb" then there is no point of prefixing the word "dumb" to terminal (except as a sales pitch to sell computers or the like in place of "smart" terminals). Due to the ambiguous meaning of "dumb terminal" it is not classified here as a type of terminal.

2.2 Text Terminals

For a text terminal, a 2-way flow of information between the computer and the terminal takes place over the cable that connects them together. This flow is in ASCII bytes where each byte usually represents a character. Bytes typed at the keyboard go to the computer and most bytes from the computer are displayed on the terminal screen. Special bytes (or sequences of bytes) from the computer tell the terminal where to move the cursor to, what to erase, where to begin and end underlining and/or blinking and/or bold, etc. There are often hundreds of such special commands and many terminals can even change fonts.

The communication uses characters (letters) encoded using a code chart for the character set being used. Usually, the first 128 bytes out of 256 possible bytes use ASCII codes. Terminals for Unix-like systems, normally connect to computers via a cable running between the asynchronous serial ports (RS-232-C = EIA-232-D) of the host computer and terminal. Sometimes the connection is via modem or terminal server, etc.

Other names for text terminals are "serial terminal", "character-cell terminal", "ASCII/ANSI terminal", "asynchronous terminal", "data terminal", "video terminal" and "video display terminal" (VDT). In olden days "video display unit" (VDU) was used for terminals but strictly speaking, it excludes the keyboard.

"Block mode" was used exclusively by old IBM mainframe terminals but many modern terminals also have this capability (which is not used much). The characters you type are temporarily retained in the terminal memory (and may possibly be edited by a built-in editor at the terminal). Then when the send key (or the like) is pressed, a block of characters (sometimes just a line of characters) is sent to the computer all at once. Block mode (as of late 1998) is not supported by Linux. Block mode makes the terminal a block device (and not a character device). See section Block Mode.

TOC

2.3 Graphics Terminals

To a limited degree some ASCII symbols can provide graphics on text terminals. One may form arrows <--- and draw boxes with _ and |. With special graphic character sets, even more is possible. None of these are really graphics terminals. However, the term "graphics terminal" is sometimes applied to all text-only terminals since text is a limited form of graphics.

There are two basic types of graphics displays: raster and vector (rarely used). Raster graphics (bit-mapped) puts dots on the screen by horizontal scan lines drawn by an electron beam (or by activating pixels or dots on a flat screen). Vector graphic displays are usually for monochrome screens that don't have any dots. They use smart electronics to draw lines and curves with an electron beam that can move in any direction (at any angle and location). Vector graphics draws high quality lines without zig-zags but is both rare and expensive. Raster graphics is almost universally used today. For PC's, images encoded in vector graphic format are sometimes used but they are translated to raster graphics format for display (with a drop in image quality).

Serial Line Graphics Terminals

Most of this document also applies to these. Most of these can also function as text terminals. The protocols for such graphics include: Tektronix Vector Graphics, ReGIS (DEC), Sixel (DEC), and NAPLPS (North American Presentation Level Protocol Syntax).

Fast Graphics Terminals (often known by other names)

None of these covered in this document. A terminal that deserves to be called smart is a graphics terminal which can rapidly display full-screen graphics just like a PC monitor. It will also have a mouse. Bytes sent to it often represent bit-maps for pictures (and other graphics). It will often use a high-speed connection to its host computer using twisted pair or coax cable. X-Window terminals are such devices. See the link to Xterminal-HOWTO at Related HOWTO's

For displaying a MS-Windows GUI there are at various types of interfaces and terminals: Winterm using WinFrame software from Citrix is one. Another (based in part on Citrix's code) is Hydra (code name) by Microsoft, also known as "Windows Terminal Server" which works with versions 4 or higher of MS Windows NT. Citrix uses its ICA protocol and has created an add-on to Hydra known as pICAsso so that WinFrame (ICA) based terminals can use the Hydra system. Hydra is also multi-user. There is also the "MultiConsole Personal Terminal" by Unbounded Technologies and Tektronix had its own multi-user interface but will now support Hydra. A magazine article in 1997 called Winterm a "dumb terminal" but it's really pretty smart. Such terminals are often called "thin clients", but some thin clients are more that just terminals as they can execute Java code sent to them, etc.

2.4 Network Computers (NCs)

These are neither true computers nor terminals but are something in-between. One type of network computer (NC's) is a computer with a CPU but no hard Disk. They are full-graphics and connect to a server computer. They are different from terminals since the programs they run execute on their own

CPU chips. Java code may be sent to them for execution. IBM calls this a "Network Station". They should work on IP networks and might work under a server running Linux. Wintel established a "Ne which, unlike the above, is almost a PC computer. However, it has no removable disks so users can't install their own software or obtain copies of anything.

Although the promoters of NCs and related Window-Terminals projected that they would replace millions of PCs, it hasn't happened. A major reason is that PCs have come down in price in recent years so that they are often even cheaper than NCs, etc. Thus for terminals, the Text-Terminal still predominates.

2.5 Emulation on a PC

Since a PC has a screen and keyboard (as does a terminal) but also has much more computing power, it's easy to use some of this computing power to make the PC computer behave like a text terminal. This is called "terminal emulation". They usually emulate text-terminals. See Terminal Emulation

---

3. Quick Install

This is a quick procedure to install a terminal without going through a Setup procedure for both the terminal and the host computer. It probably will not work right if the terminal happens to have been set up incompatible with the computer. If you don't understand some of it you'll need to consult other parts of this document for more info.

To install a terminal, first look in `/etc/termcap` or `terminfo.src` to find an entry for it (see Terminfo and Termcap (detailed)). Figure out what serial port you'll connect it to and what the tty designation is for that port (e.g. `ttyS1`, see Device Names). As the root user, edit `/etc/inittab` and add a getty command next to the other getty commands. The format of the getty command depends on which getty program you use. `agetty` (called just `getty` in the Debian distribution) is the easiest (no configuration file). See the "`info`" or "`man` re `getty`. For getty parameters use the terminfo (or termcap) name (such as vt100) for your terminal. Type in a baud-rate that the terminal supports. But if you set the baud too high you may need to use (See Flow Control).

Then physically connect the main serial port of the terminal to the chosen serial port of the computer with a null-modem cable and turn on the terminal. Don't expect most ready-made cables to be wired correctly for hardware flow control. Make sure the baud-rate of the terminal is set the same as you gave to getty and that its "data bits" is 8. Then at the computer console type "init q" to apply the changes you made to the inittab file. You should now see a login prompt at the terminal. If you don't, tap the terminal's return key. If this doesn't work read more of this document and/or see Trouble-Shooting.

4. Why Use a Terminal ?

4.1 Intro to Why Use a Terminal

PC's are so powerful today that just one PC can often support several persons using it at once, especially if they are doing low-load tasks such as text editing, data entry, etc. One way to do this is to connect a number of terminals to a single PC (or other host computer) by modems or direct cable connection. To do this, one needs a multi-user operating system such as Linux. This has been called "time sharing" but it's not good terminology today since "distributed" computing over a network is also a type of time sharing. It might be better described as "centralized" computing. But the central computer may be connected to the rest of the world via a network so that terminal users may send email, browse the Internet with the "lynx" browser, etc. So it's not exactly "centralized" either.

Terminals have seldom been used with PC's because the popular operating systems used for them (Windows, DOS, and Mac) were not multiuser until 1998 (available for MS Windows NT) and previously could not support terminals. Now that Linux, a multiuser operating system, is freely available for PC's, the use of terminals with PC's becomes more feasible. The drawback is that text terminals are not smart enough to support the type of graphical user interface (GUI) that many computer users today normally expect.

4.2 Lower Hardware Costs ?

When Computers (including PCs) were quite expensive, lower hardware costs was a significant advantage of using terminals. Today with cheap PCs, the cost savings is problematical. Here's what I wrote years ago when PCs were more expensive. It's still true today but of less significance.

If several people use the same computer as the same time, there is a reduction in the amount of hardware needed for the same level of service. One type of savings is due to code sharing. The application files on hard disks are shared as well as shared libraries in memory (even when people are running different programs provided they use some of the same functions in their code). Another type of savings is due to reduction of peak load. The hardware of a single PC may be idle most of the time as people slowly type in information, think, talk, or are away from their desks. Having several people on the same computer at once makes good use of much of this idle time which would otherwise be wasted.

These savings are substantial. One may roughly estimate (using statistical theory) that for 9 persons (8 terminals & 1 console) the shared PC only needs only about 3 times as much capacity (in memory, disk storage, CPU power, etc.) as a single PC in order to provide the same level of service per person. Thus the computational hardware for such a shared system should only cost about 1/3 as much per user. However, the cost of the display hardware (CRT's, keyboards, video electronics, etc.) is about the same for both cases. The terminals have the added cost of requiring additional serial ports at the host computer.

For a fair comparison with PC's, the terminals should have the same capabilities as the PC monitors.

Unfortunately, color graphic terminals for Linux (X-windows) with high speed communication is a r
market with high prices so in this case there is not likely to be any savings in hardware costs. But for
terminals there will be some savings, especially if the terminals are obtained used at low cost.

TOC

4.3 Control of Software

For centralized computing, software (and the updates to software) only need be installed on one host computer instead of several. The person in charge of this computer may control the software which is installed on it. This is advantageous if the person controlling the host computer does an excellent job and knows about the needs and preferences of the other users. Users can be prevented from playing games or surfing the Internet by not installing the software (or by otherwise restricting access to it). Whether or not centralized control is desirable depends on the situation.

4.4 Hardware Upgrades

With terminals, the computer hardware upgrades take place on only one computer instead of many. This saves installation labor effort. While the cost of the hardware for the host computer upgrade will be more than that for a single PC (since the host needs more computing power than a PC), the cost will be significantly less than upgrading a number of PC's being used instead of terminals.

4.5 Other Advantages of Terminals

- The elimination of noise from fans and disk drives provided the terminals are not close to the computer.
- The users of the terminals can share data and files and send e-mail to each other. It's similar to a local network.

4.6 Major Disadvantages of Terminals

- Text terminals have no high-speed graphic display (or high resolution graphics) although they can often use graphic character sets to draw boxes, etc. This lack limits the software that may be used on it.
- If the host computer goes down, then no one can use the terminals either (unless there is a "standby" host computer to connect to).

4.7 Are Text Terminals Obsolete ?

Text terminals are technologically obsolete because for a slightly higher cost of hardware, one could build a smart terminal (with the same quality of display). This wasn't always the the case since around 1980 memory cost thousands of dollars per megabyte. Today with low costs for memory and processors, one could make a text terminal smart for only about a 10% or 20% increase in hardware cost.

The reasons that text terminals are not yet obsolete are:

- There is no satisfactory standard interface for smart graphics terminals. The MS Hydra system is for MS Windows NT, while X-Windows is not as efficient as it should be (and X-Windows terminals are too costly).
- Many people don't need full screen graphics.
- Text terminals are low in cost and allegedly take longer to become obsolete, yet can give access to a much newer (and powerful) computer.
- Since running a text-terminal (in contrast to a full-graphics terminal) doesn't consume much of a modern PC's resources, a large number of terminals may be efficiently run from one PC.

5. Overview of How Terminals Work (in Linux)

See also section Some Details on How Terminals Work

5.1 Device Names

Each terminal is connected to a serial port on the host computer (often just a PC). The ports have names: ttyS0, ttyS1, ttyS2 etc. These are represented by special files found in the /dev (device) directory. /dev/ttyS0 corresponds to COM1 in DOS or Windows. ttyS1 is COM2, etc. See Terminal Special Files for details on these and related "devices" such as cua.

5.2 Login/Logout

When the host computer starts up it runs the program getty (see Serial-HOWTO 4.1 & 7.2) on each serial port which has a terminal on it (as specified in the file etc/inittab). The getty program runs the "login" program to log people in. A "login:" prompt appears on the screen. People at the terminals log in (after giving their passwords) and then have access to the computer. When it's time to shut the terminal down, one normally logs out and turns the terminal off. See Login Restrictions regarding restricting logins (including allowing the root user to log in at terminal).

5.3 Half/Full Duplex

If one watches someone typing at a terminal, the letters one types simultaneously appear on the screen. A naive person might think that what one types is being sent directly from the keyboard to the screen with a copy going to the computer (half-duplex like, see next paragraph). What is usually going on is that what is typed at the keyboard is directly sent only to the host computer which in turn echoes back to the terminal each character it receives (called full-duplex). In some cases (such as passwords or terse editor commands) the typed letters are not echoed back.

Full-duplex means that there are two (dual) one-way communication links. Full-duplex is the norm for terminals. Half-duplex is half of a duplex, meaning that there is only a single one-way communication link. This link must be shared by communications going in both directions and only one direction may be used at a time. In this case the computer would not be able to echo the characters you type (and send to it) so the terminal would need to also send each character you type directly to the terminal screen. Some terminals have a half-duplex mode of operation which is seldom used.

TOC

5.4 Terminal Memory

The image on a CRT tube will fade away almost instantly unless it is frequently redrawn on the screen by a beam of electrons shot onto the face of the tube. Since text sent to a terminal needs to stay on the screen, the image on the screen must be stored in the memory chips of the terminal and the electron beam must repeatedly scan the screen (say 60 times per second) to maintain the image. See Terminal Memory Details for more details.

5.5 Commands for the Terminal

The terminal is under the control of the computer. The computer not only sends the terminal text to display on the screen but also sends the terminal commands which are acted on. These are Control Codes (bytes) and escape sequences. For example, the CR (carriage return) control code moves the cursor the the left hand edge of the screen. A certain escape sequence (several bytes where the first byte is the "escape" control code) can move the cursor to the location on the screen specified by parameters placed inside the escape sequence.

The first terminals had only a few such commands but modern terminals have hundreds of them. The appearance of the display may be changed for certain regions: such as bright, dim, underline, blink, and reverse video. A speaker in a terminal can "click" when any key is pressed or beep if a mistake has occurred. Function keys may be programmed for special meanings. Various fonts may exist. The display may be scrolled up or down. Specified parts of the screen may be erased. Various types of flow control may be used to stop the flow of data when bytes are being sent to the terminal faster than the terminal can handle them. There are many more as you will see from looking over an advanced terminal manual or from the Internet links Esc Sequence List

5.6 Lack of Standardization Solved by Terminfo

While terminals made for the US all used the same ASCII code for the alphabet (except for IBM terminals which used EBCDIC), they unfortunately did not all use the same escape sequences. This happened even after various ANSI (and ISO) standards were established since these standards were never quite advanced enough. Furthermore, older terminals often lacked the capabilities of newer terminals. This might cause problems. For example, the computer might send a terminal an escape sequence telling it to split the screen up into two windows of specified size, not realizing that the terminal was incapable of doing this.

To overcome these problems a database called "termcap" (now "terminfo") was established. This database resides in certain files on the computer and has a section of it (sometimes an entire file) for each model of terminal. For each model (such as VT100) a list of capabilities is provided including a list of certain escape sequences available and what they do. See Section Termcap and Terminfo (detailed) for more details. Application programs may utilize this database by calling certain C-Library programs. One large set of such programs (over 200) is named "ncurses" and are listed in the manual page for ncurses.

TOC

5.7 The Interface

The environment variable TERM is the type of terminal Linux thinks you are using. Some application programs use this to look up the capabilities in the terminfo database so TERM needs to be set correctly. But there is more to a correct interface than the computer knowing about the capabilities of the terminal.

For bytes to flow from the computer to the terminal the terminal must be set to receive the bytes at the same baud rate (bits per second) as they are sent out from the terminal. If the terminal is set to receive at 19,200 baud and the computer sends out characters at 9600 baud, only garbage (or perhaps nothing) will be seen on the screen. One selects the baud rate for a terminal (as well as many other features) from the terminals "set-up" menus at the terminal. Most terminals have a large number of options in their "set-up" menus (see Terminal Set-Up (Configure) Details). The computer serial port has options also and these options must be set up in a compatible way (see Computer Set-Up (Configure) Details.

5.8 Emulation

Most terminals today have more than one emulation (personality or "terminal mode"). The terminal model numbers of terminals formerly made by DEC (Digital Equipment Corporation now Compaq) start with VT (e.g. VT100). Many other terminals which are not VT100 may be set up to emulate a VT100. Wyse is a major terminal manufacturer and most of their terminals can emulate various DEC terminals such at VT100 and VT220. Thus if you want to, say, use a VT320 terminal you may either use a real VT320 in "native" personality or possibly use some other terminal capable of emulating a VT320. The "native" personalities usually have more capabilities so, other things being equal, "native" is usually the best to use.

The most common type of emulation is to use a PC like it was a vt100 terminal (or the like). Programs loaded into the PC's memory permits this. In Linux (unless you're in X-windows) the PC monitor (called the console) emulates a terminal of type "Linux" (close to vt100). Even certain windows within X-windows emulate terminals. See Terminal Emulation.

5.9 The Console

On a PC, the monitor is known as the console. It emulates a terminal of type "Linux". One logs on to it as a virtual terminal. See The Console: /dev/tty?. It receives messages from the kernel regarding booting and shutdown progress. One may have the messages that normally go to the console, go to the terminal. To get this you must manually patch the kernel, except that for kernel 2.2 (or higher) it is a "make config" option. See Make a Terminal the Console.

---

6. Terminal Special Files such as /dev/tty

"tty" is an abbreviation for "Teletype". The first terminals were Teletypes (like remotely controlled typewriters). See subsection Teletypes. A list of Linux devices (the stuff in the /dev directory) may be found in "Linux Allocated Devices" which should be included with kernel sources. It "describes" what each device used for in only a word or two but doesn't tell you how to use them.

6.1 Serial Port Terminals

The computer considers each serial port to be a "device". It's sometimes called a terminal device since at one time terminals were the common use for the serial port. For each such serial port there is a special file in the /dev (device) directory. /dev/ttyS0 is the special file for the serial port known as COM1 in the DOS/Windows world. To send text to a terminal you may redirect standard output of some command-line command to the appropriate special file. For example typing "echo test > /dev/ttyS1" at the command prompt should send the word "test" to the terminal on ttyS1 (COM2) provided you have write permission on /dev/ttyS1. Similarly, typing "cat my_file > /dev/ttyS0" will send the contents of the file my_file to COM1 (ttyS0).

In addition to ttyS0 (/dev/ttyS0), ttyS1, ttyS2, etc. (the "S" stands for Serial port) there is also a "cua" series: cua0, cua1, cua2, etc. cua0 is the same port as ttyS0, etc. The "cu" of cua stands for CalloUt. The ttyS series are Posix compliant while using cua may permit the opening of a port that the modem control lines say is not ready. Starting with kernel version 2.2 cua is obsolete and a warning message is issued when you attempt to use it (although it still works). For the past few years it has only been included with Linux for backwards compatibility. A programmer can arrange things so that ttyS can behave just like cua, so cua is not really needed.

6.2 Pseudo Terminals

Pseudo terminals have no unique physical connector on the computer. They are used to emulate a serial port. For example, if someone connects via telnet to your computer over a network, they may wind up connected to the device /dev/ptyp2 (a pseudo terminal port). In X-Windows, the terminal emulator program, xterm (or rxvt), uses pseudo terminals. Ham radio programs under Linux also use them. Using certain application software it is possible to have 2 or more pseudo terminals attached to the same physical serial port.

Pseudo terminals come in pairs such as ttyp3 and ptyp3. The pty... is the master or controlling terminal and the tty... is the slave. ttyq5 is also a pseudo terminal as is ttysc (c is a hexadecimal digit). More precisely, pseudo master terminals are /dev/pty[p-s]N and the corresponding slaves are /dev/tty[p-s]N where N is a hexadecimal digit 0-f.

Unix98 doesn't use the above but instead uses a "pty master" which is /dev/ptmx. This can supply a pty

on demand. Details on using Pseudo terminals are not to be found in this HOWTO.

6.3 The Controlling Terminal /dev/tty

/dev/tty stands for the controlling terminal (if any) for the current process (the process that uses "/dev/tty" in a command). To find out which tty's are attached to which processes use the "ps -a" command at the shell prompt (command line). Look at the "tty" column. For the shell process you're in, /dev/tty is the terminal you are now using. Type "tty" at the shell prompt to see what it is (see manual pg. tty(1)). /dev/tty is something like a link to the actually terminal device name with some additional features for C-programmers: see the manual page tty(4).

6.4 /dev/ttyIN "Terminals"

N stands for an integer. One use of these in Linux is with the ISDN driver package: isdn4linux. The ttyIN is something like ttySN. There is also a cuiN which is something like cuaN. The ttyI and cui emulate modems and may be given modem commands.

6.5 The Console: /dev/ttyN

In Linux the PC monitor is usually called the console and has several device special files associated with it: tty0, tty1, tty2, etc. When you log in you are on tty1. To go to tty2 press Alt-F2. tty1, tty2, etc. are "virtual terminals" (sometimes called "virtual consoles"). You may log in to different virtual terminals and thus have a few different sessions with the computer going on at the same time. You switch between them using the Alt-F? key where "?" is the virtual-terminal number you want. The console is also known as /dev/tty0 and system messages may go to that device and display on your console. Only the system or the root user may write to /dev/tty0 to which /dev/console is sometimes linked. System messages may also be written directly to the hardware address of the console thus bypassing /dev/tty0. For more info on the console see The Linux Console.

6.6 Creating a Device with "mknod"

The /dev directory comes supplied with many device special files. If you need something that's not there you may try to create it with the "mknod" command. See the manual page ttys(4) for how to do this for serial ports. To use `mknod` you must know the major and minor device numbers. You might be able to infer the numbers you need by using the "`ls -l`" command in the /dev directory. It will display the major and minor numbers of existing special files.

---

7. Some Details on How Terminals Work

If you know almost nothing about terminals, it's suggested that you first read Introduction and also read Overview of How Terminals Work.

7.1 Terminal Memory Details

The terminal screen refreshes itself at perhaps 60 times per second from an image stored in the memory of the terminal. For a PC the monitor's image is stored on the video card inside the computer but for a terminal, the equivalent of the video card is inside the terminal. For a text terminal the storage of the image uses little memory. Instead of putting every dot (pixel) on the screen into memory and requiring the storage of about a quarter-million dots, a much more efficient method of storage is used.

A screen-full of text may be represented inside the terminal memory by ASCII bytes, one for each character on the screen. An entire screen only takes about 2K ASCII bytes. To display these characters, the terminal must also know the bit-map (the shape) of each of the almost 100 printable ASCII characters. With a bit-map of a character using say 15 bytes, only about 1.5K of memory is needed for the bit-maps of all the ASCII characters (the font). This ASCII text and font memory is scanned so that the resulting image is put on the screen about 60 times each second. This is a form of shared memory where a single bit-map of a letter such as the letter e, is shared by all of the many letter e's which appear on a screen-full of text. Low memory requirements meant low costs to produce monitors in the early 1980's when the cost of memory was several thousand times higher than it is today (costing then several dollars per kilobyte).

7.2 Early Terminals

The first terminals were something like remotely controlled typewriters which could only "display" (print on paper) the character stream sent to them from the computer. The earliest models were called Teletypes. The name "tty" is just an abbreviation for "Teletype". Early terminals could do a line feed and a carriage return just like a typewriter and ring a bell when a bell character was received. Due to the lack of significant capabilities this was the first type of terminal to be labeled "dumb". This type of terminal interface (using a terminal type called "dumb") is sometimes used today when the computer can't figure out what kind of a terminal it is communicating with.

7.3 Escape Sequences and Control Codes (intro)

Terminals have many capabilities some of which are always present and some of which require commands from the computer to change or activate. To exercise all these capabilities under the control of the computer requires that special codes be established so that the computer can tell the terminal what to do. There are two major type of such codes: escape sequences and control codes (control characters).

There are many times more escape sequences than control codes.

Control Codes

The control codes (or control characters) consist of the first 32 bytes of the ASCII alphabet. They include the following: carriage-return (cursor to far left), line-feed (cursor down one line), backspace, escape-character, tab, and bell. They do not normally show on the screen. There is usually a command which you may give to your terminal which will result in them being displayed when they are received by the terminal. It's called something like "Display Controls" or "Monitor". If you do this then the display may look a mess since escape sequences, which all start with the ESC (escape) control character, are no longer executed. Words which should appear at the top or bottom of the screen will show up in other locations. The escape sequences to reposition the cursor display on the screen but the cursor doesn't move to where the escape sequence says.

Escape Sequences

Since there are not nearly enough control codes to do everything (and for some reason, not all of them are utilized) many escape sequences are used. They consist of the "escape" (ESC) control character followed by a sequence of ordinary characters. Upon receiving an escape character, the terminal examines the characters following it so that it may interpret the sequence and carry out the intended command from the computer. Once it recognizes the end of a valid sequence, further characters received just display on the screen (unless they are control codes or more escape sequences). Some escape sequences may take parameters (or arguments) such as the coordinates on the screen to move the cursor to. The parameters become a part of the escape sequence. An Esc Sequence List is on the web for some terminals, but it's terse.

A list of the escape sequences for your terminal should be in the "programmers manual" for the terminal. Except for very old terminals, there may be two or three hundred such sequences. If you don't have a such manual it's not easy to find them. Some of the sequences are available on the Internet. One link is Esc Sequence List. By searching the Internet for one sequence (such as ESC[5m) you may come across a long list of them.

Another way to determine some of them is to find the terminfo entry (termcap) for the terminal and mentally decode it. See Terminfo and Termcap (detailed) in this document and/or the Termcap Manual on the Internet. Unfortunately, the terminfo (termcap) for a terminal often does not list all of the escape sequences which the terminal has available for use, but fortunately, the most important ones are usually there.

7.4 Display Attributes & Magic Cookies

Terminals have various methods of generating character attributes such as bold, reverse-video, underlining, etc. There should be no need for the user to worry about how how this is done, except that it creates problems for some old terminals and there is sometimes an option for this in the set-up menu of newer terminals.

The magic cookie method is obsolete. It's the simplest (and worst) method of defining attributes: Use a

certain byte for the start of an attribute and another to end that attribute. For example, a "start underlining" magic cookie byte is placed just before the first word to be underlined. These extra bytes are put into the memory of the screen page, just like character bytes that display as characters. But this might foul up the count of the number of characters per line since non-printable magic cookie characters are intermingled with other printable characters. This sometimes causes problems.

A better method which uses more memory is to assign an attribute byte (or half=byte, etc.) to each displayed character. This method is used by PC video cards (for text) for the common PC monitor.

---

8. Special Features of Some Terminals

8.1 Color

While the common monochrome terminal is not a color terminal it may have a fixed "color" display other than white such as green or amber. All terminals have black (electron beam turned off = zero brightness). A real color terminal can change the color of the text and background to many different colors while a monochrome terminal can only change the brightness of a fixed color.

However, changing the brightness, etc. gives a lot of possibilities. For example, a black and white (monochrome) terminal can have white, grey, and black by varying the brightness. Some words can be black on a light grey background while other are highlighted by black on white. In addition there is white on black, underlining, and blinking.

Color works like the color on a computer monitor or TV screen. The CRT has three colors of dots on it with each color controlled by its own electron beam (3 beams). Monochrome has inherently better resolution since it doesn't depend on dots permanently fixed on the screen. For text terminals the only use of color is to differentiate text and this advantage is not always worth the cost of worse resolution. Thus monochrome may be better since it also costs less.

8.2 Multiple Sessions

For dual sessions the terminal has two serial ports of equal status. Each port is connected to a serial port on a different computer. Thus one may log in to two different computers with each session displaying in a split-screen window. Alternatively, each session may run full-screen with a "hot" key (or the like) to switch between sessions. One could also connect to two different serial ports on the same computer and log in twice (similar to "virtual terminals" at the console). The program "screen" will make any ordinary terminal (single session) connected to a single computer run two or more "sessions".

8.3 Printer/Auxiliary Port

Many terminals have a connector on the rear for such a port. It may be labeled as "Aux" or "Printer", etc. Some printer ports are for parallel printers while others are for serial printers. If a printer is connected to the printer or auxiliary port, then pressing certain keys will print the screen. One may also have everything that displays on the screen go also to the printer. If the port is an auxiliary port, one may connect this to another computer and almost have dual sessions as above. However, the video memory inside the terminal may not retain both sessions so you may need to refresh the screen when switching to the other session. There will likely not be a hot key either but possibly a programmable function key may be programmed to do this. There exists various key combinations and escape sequences for controlling such a port. See Printer Esc.

There is a program called `vtprint` which is designed to send a print job to your terminal to be pri on a printer attached to the terminal. It's homepage is `http://people.qualcomm.com/garrett/vtprint` . It's also a included (as of 1998) in the Debian distribution of Linux. `xprt` (also in Debian) seems to do something similar, but only for X-Window terminals ??

8.4 Pages

Many terminals permit the storage of more than one page in their video memory. Sometimes the page size is the same as the screen, but sometimes it is larger so that scrolling will reveal unseen parts of a page. So when one looks at a screen, there may be hidden text on the same page above or below the display. In addition, if there is more than just one page, there may be hidden text on these other pages. One use for pages is on terminals that support dual sessions. Each session may have its own page and one may switch back and forth between them.

Even if you only have a one-page-terminal with the page sized equal to what is displayed on the screen, you will still see other pages of a file (etc.) as the host sends more data to the terminal. One advantage to having additional pages stored in the terminal memory is so that you can jump to them instantly without waiting a second or so for them to be transmitted from the host.

I know of no Linux application program that exploits the use of multiple pages. Let me know if you know of any. There is a commercial program called "Multiscreen" which supports this but probably not for Linux ?? Multiscreen is reported to be part of SCO and is something like the virtual terminals on a Linux PC console. The Linux program "screen" makes it look like you have multiple pages but they are stored in the computer and but you can have only one page-like window for each running program.

8.5 Character-Sets

A character-set is normally represented by a list (or table or chart) of characters along with the byte code assigned to each character. The codes for a byte range from 0 to 255 (00 to FF in hexadecimal). In MS-DOS, character-set tables are called "code-pages". You should examine such a table if you're not familiar with them. They are sometimes included in printer and terminal manuals but may be difficult to find.

ASCII is one of the most common character-sets used on text terminals. It is a 7-bit code but can be made into 8-bit if the first bit (high order bit) is always set to 0. Other character-sets are usually available (except on very old terminals where the only choice is ASCII). The first half of most character-sets are the conventional ASCII characters and the second half (the characters with the high-order bit set to 1) belong to a wide variety of character-sets. Character sets are often ISO standards. To get specialized character sets on a terminal, you may need to download a soft-font for that character-set into the memory of the terminal.

Besides ASCII, there are some other common character-sets, all 8-bit. CP stands for Code Page character sets invented by IBM: CP-437 (DOS ECS), CP-850 (Multilingual Latin 1 --not the same as ISO Latin-1), ISO-8859-1 (Latin-1), ANSI (derived from Latin-1). MS Windows uses ANSI while the Internet often uses Latin-1. There are several ISO-8859 character sets in addition to Latin-1. These include Greek (-7),

Arabic (-6), Eastern European (-2), and Russian (-5). There are many others. For example, KOI8-R i more commonly used for Russian than IS0-8859-5. Unicode is a very large character-set where each character is represented by 2 bytes instead on just one byte.

More info re character-sets are:

- Manual pages: ASCII and latin1
- HOWTO's for various languages (likely written in that language). See "Cyrillic" for Russian.
- http://www.cc.columbia.edu/kermit/charsets.html for a short listing of various character-set names.
- http://www.pku.edu.cn/on_line/w3html/International/Overview.html for info on character-sets and the Internet.
- Languages, Countries and Character Sets
- http://vancouver-webpages.com/multilingual/ name="Using Multiple Languages in HTML">

Once you've found the character set name (or number) you are interested in, you may search for more info about it on the Internet.

8.6 Fonts

Most terminals made after the mid 1980's can accept downloaded soft-font. This means that they can display almost any character set provided that you can find the soft-font for it. If you can't find the needed soft-font, you can always create your own. A free font editor for this is called BitFontEdit (written by the author of this document) and (in 1998) was at
Europe: http://www.funet.fi/pub/culture/russian/comp/cyril-term/
N. America: ftp://cs.utk.edu/pub/shuford/terminal/BitFontEdit.tar.gz

8.7 Keyboards & Special Keys

Terminal keyboards often have a number of keys that one doesn't find on a PC keyboard. Few (if any) actual terminals will have all of these keys and most will have additional keys not listed here. Some have a large number of special purpose keys such as terminals made for use with cash registers. There are often many more key meanings than shown here since these keys often have extended meanings when used in conjunction with other keys (such as shift and control).

- BREAK sends a very long 0 bit (space = +12 V) of duration 300 to 700 milliseconds to the host. The host may interpret this as an interrupt if stty has set brkint or ignore it if ignbrk is set.
- NO SCROLL stops the screen from scrolling like ^S does. Depressing it again resumes scrolling. Uses flow control signals to do this.
- REPEAT if held down with an other key, forces repeated output of that other key even if the auto-repeat option is set to off.
- LINE FEED sends the line feed character ^J to the host. Seldom used.
- SET-UP allows the manual configuration of the terminal via menus. Sometimes purposely disabled by putting a block under it so it can't be pressed down. Sometimes another key such as shift or control must be pressed at the same time. See Getting Into Set-Up (Configuration) Mode.

- LOCAL disconnects the terminal from the host. In local, what one types goes directly to the screen. Useful for testing.
- RETURN is the same as the "enter" key on a PC. It usually sends a carriage return to the host which normally get translated to a new-line character by the host's device driver. On some terminals it may be set up to send something else.
- F1, F2, ... or PF1, PF2, ... are function keys which usually may be programmed to send out a sequence of bytes (characters). See Function Keys

---

9. Terminal Emulation; the Console

9.1 Terminal Emulation

Intro to Terminal Emulation

Since a PC has a screen and keyboard (as does a terminal) but also has much more computing power, it's easy to use some of this computing power to make the PC computer behave like a text terminal. This is one type of terminal emulation. Another type of terminal emulation is where you set up a real terminal to emulate another brand/model of terminal. To do this you select the emulation you want (called "personality" in Wyse jargon) from the terminal's set-up menu. This section is about the first type of emulation: emulating a terminal on a PC.

Emulation software is available for MS Windows and comes built-in with recent versions of MS Windows. Most Linux software can only emulate a VT100, VT102, or VT100/ANSI. If you find out about any others, let me know. Since most PC's have color monitors but VT100 and VT102 were designed for a monochrome monitor, the emulation usually adds color capabilities (and a choice of colors). Sometimes the emulation is not 100% perfect but this usually causes few problems. For using a Mac computer to emulate a terminal see the mini-howto: Mac-Terminal.

Don't Use TERM For Emulation

Some have erroneously thought that they could create an emulator at a Linux console (monitor) by setting the environment variable TERM to the type of terminal they would like to emulate. This does not work. The value of TERM only tells an application program what terminal you are using. This way it doesn't need to interactively ask you this question. If you're at the PC monitor it's a terminal of type "Linux" and your can't change this. So you must set TERM to "Linux".

If you set it to something else you are fibbing to application programs. As a result they will incorrectly interpret certain escape sequences from the console reulting in a corrupted interface. Since the Linux console behaves almost like a vt100 terminal, it could still work almost OK if you falsely claimed it was a vt100 (or some other terminal which is something like a vt100).

Communication (Dialing) programs

An emulation program is often combined with a modem dialing program (such as Minicom, Seyon, or Kermit) so that one may (for example) dial up public libraries to use their catalogs and indexes, (or even read magazine articles). Seyon is only for use with X-windows and can emulate Tektronix 4014 terminals. Emulators exist under DOS such as `telix` and `procomm` work just as well. The terminal emulated is often the old VT100, VT102, or ANSI (like VT100).

TOC

Emulation under X-Windows

Xterm (obsolete ??) may be run under X-Windows which can emulate a VT102, VT220, or Tektronix 4014. There is also an xterm emulation (although there is no real terminal named "xterm"). If you don't need the Tektronix 4014 emulation (a vector graphics terminal; see Graphics Terminals) you may use `eterm`. Predecessors to `eterm` are `rxvt` and `xvt`. `eterm` supports pixmaps.

For non-Latin alphabets, kterm is for Kanji terminal emulation (or for other non-Latin alphabets) while xcin is for Chinese. There is also 9term emulation. This seems to be more than just an emulator as it has a built-in editor and scroll-bars. It was designed for Plan 9, a Unix-like operating system from AT&T.

Real Terminals Better

Unless you are using X-Windows with a large display, a real terminal is often nicer to use than emulating one. It usually costs less, has better resolution for text, and has no disk drives to make annoying noises.

9.2 Testing Terminal Emulation

For the VT series terminals there is a test program: `vttest` to help determine if a terminal behaves correctly like a vt53, vt100, vt102, vt220, vt320, vt420 etc. There is no documentation but it has menus and is easy to use. To compile it run the configure script and then type "make". It may be downloaded from: ftp://ftp.clark/net:/pub/dickey/vttest/. An alternate download site is: http://sunsite.unc.edu/pub/Linux/utils/console/

9.3 The Linux Console

The console for a PC Linux system is the computer monitor It emulates a terminal of type "Linux". There is no way (unless you want to spend days rewriting the kernel code) to get it to emulate anything else. Setting the TERM environment variable to type of terminal other than "Linux" will not result in emulating that other terminal. It will only result in a corrupted interface since you have falsely declared (via the TERM variable) that your "terminal" is of a type different from what it is. See Don't Use TERM For Emulation

The "Linux" emulation is flexible and has features which go well beyond those of the vt102 terminal which it was intended to emulate. These include the ability to use custom fonts and easily re-map the keyboard (without patching the source code and recompiling the kernel as is required for the case of a real terminal). These extra features reside in the console driver software and not in the emulation software but the results are like it was part of the emulation.

Many commands exist (see Keyboard-and-Console-HOWTO) to utilize these added features. Real terminals, which use neither scan codes nor VGA cards, unfortunately can't use most of these features. One may recompile Linux to make a terminal receive the messages which normally go to the console. See Make a Terminal the Console.

---

10. Flow Control (Handshaking)

Flow control (= handshaking = pacing) is to prevent too fast of a flow of bytes from overrunning a terminal, computer, modem or other device. Overrunning is when a device can't process what it is receiving quickly enough and thus loses bytes and/or makes other serious errors. What flow control does is to halt the flow of bytes until the terminal (for example) is ready for some more bytes. Flow control sends its signal to halt the flow in a direction opposite to the flow of bytes it wants to stop. Flow control must both be set at the terminal and at the computer.

There are 2 types of flow control: hardware and software (Xon/Xoff or DC1/DC3). Hardware flow control uses dedicated signal wires such as RTS/CTS or DTR/DSR while software flow control signals by sending DC1 or DC3 control bytes in the normal data wires. For hardware flow control, the cable must be correctly wired.

The flow of data bytes in the cable between 2 serial ports is bi-directional so there are 2 different flows (and wires) to consider:

1. Byte flow from the computer to the terminal
2. Byte flow from the terminal keyboard to the computer.

10.1 Why Is Flow Control Needed ?

You might ask: "Why not send at a speed slow enough so that the device will not be overrun and then flow control is not needed?" This is possible but it's usually significantly slower than sending faster and using flow control. One reason for this is that one can't just set the serial port baud rate at any desired speed such as 14,500, since only a discrete number of choices are available. The best choice is to select a rate that is a little higher than the device can keep up with but then use flow control to make things work right.

If one decides to not use flow control, then the speed must be set low enough to cope with the worst case situation. For a terminal, this is when one sends escape sequences to it to do complex tasks that take more time than normal. In the case of a modem (with data compression but no flow control) the speed from the computer to the modem must be slow enough so that this same speed is usable on the phone line, since in the worst case the data is random and can't be compressed. If one failed to use flow control, the speed (with data compression turned on) would be no faster than without using any compression at all.

Buffers are of some help in handling worst case situations of short duration. The buffer stores bytes that come in too fast to be processed at once, and saves them for processing later.

TOC

10.2 Padding

Another way to handle a "worst case" situation (without using flow control or buffers) is to add a bunch of nulls (bytes of value zero) to escape sequences. Sometimes DEL's are used instead provided they have no other function. See Recognize Del.

The escape sequence starts the terminal doing something, and while the terminal is busy doing it, it receives a bunch of nulls which it ignores. When it gets the last null, it has completed its task and is ready for the next command. This is called null padding. These nulls formerly were called "fill characters". These nulls are added just to "waste" time, but it's not all wasted since the terminal is usually kept busy doing something else while the nulls are being received. It was much used in the past before flow control became popular. To be efficient, just the right amount of nulls should be added and figuring out this is tedious. It was often done by trial and error since terminal manuals are of little or no help. If flow control doesn't work right or is not implemented, padding is one solution. Some of the options to the `stty` command involve padding.

10.3 Overrunning a Serial Port

One might wonder how overrunning is possible at a serial port since both the sending and receiving serial ports involved in a transmission of data bytes are set for the same speed (in bits/sec) such as 19,200. The reason is that although the receiving serial port electronics can handle the incoming flow rate, the hardware/software that fetches and processes the bytes from the serial port sometimes can't cope with the high flow rate.

One cause of this is that the serial port's hardware buffer is quite small. Older serial ports had a hardware buffer size of only one byte (inside the UART chip). If that one received byte of data in the buffer is not removed (fetched) by CPU instructions before the next byte arrives, that byte is lost (the buffer is overrun). Newer UART's, namely most 16550's, have 16-byte buffers (but may be set to emulate a one-byte buffer) and are less likely to overrun. It may be set to issue an interrupt when the number of bytes in its buffer reaches 1, 4, 8, or 14 bytes. It's the job of another computer chip (usually the main CPU chip for a computer) to take these incoming bytes out of this small hardware buffer and process them (as well as perform other tasks).

When contents of this small hardware receive buffer reaches the specified limit (one byte for old UART'S) an interrupt is issued. Then the computer interrupts what it was doing and software checks to find out what happened. It finally determines that it needs to fetch a byte (or more) from the serial port's buffer. It takes these byte(s) and puts them into a larger buffer (also a serial port buffer) that the kernel maintains in main memory. For the transmit buffer, the serial hardware issues an interrupt when the buffer is empty (or nearly so) to tell the CPU to put some more bytes into it to send out.

Terminals also have serial ports and buffers similar to the computer. Since the flow rate of bytes to the terminal is usually much greater than the flow in the reverse direction from the keyboard to the host computer, it's the terminal that is most likely to suffer overrunning. Of course, if you're using a computer as a terminal (by emulation), then it is likewise subject to overrunning.

Risky situations where overrunning is more likely are: 1. When another process has disabled interrupts

(for a computer). 2. When the serial port buffer in main (or terminal) memory is about to overflow.

TOC

10.4 Stop Sending

When its appears that the receiver is about to be overwhelmed by incoming bytes, it sends a signal to the sender to stop sending. That is flow control and the flow control signals are always sent in a direction opposite to the flow of data which they control (although not in the same channel or wire). This signal may either be a control character (^S = DC3 = Xoff) sent as an ordinary data byte on the data wire (in-band signalling), or a voltage transition from positive to negative in the dtr-to-cts (or other) signal wire (out-of-band signalling). Using Xoff is called "software flow control" and using the voltage transition in a dedicated signal wire (inside the cable) is called hardware flow control.

10.5 Keyboard Lock

When a terminal is told to stop sending, the terminal "locks" its keyboard. This seldom happens but when it does, a message or light should inform you that the keyboard is locked. Anything you type at a locked keyboard is ignored. The term "locked" is also used when the computer is told to stop sending to a terminal. The keyboard is not locked so that whatever you type goes to the computer. Since the computer can't send anything back to you, characters you type don't display on the screen and it may seem like the keyboard is locked but it's not.

10.6 Resume Sending

When the receiver has caught up with its processing and is ready to receive more data bytes it signals the sender. For software flow control this signal is the control character ^Q = DC1 = Xon which is sent on the regular data line. For hardware flow control the voltage in a signal line goes from negative (negated) to positive (asserted). If a terminal is told to resume sending the keyboard is then unlocked and ready to use.

10.7 Hardware Flow Control (RTS/CTS etc.)

Some older terminals have no hardware flow control while others used a wide assortment of different pins on the serial port for this. The most popular today seems to be the DTR pin (or both the DTR pin and the DSR pin).

RTS/CTS, DTR, and DTR/DSR Flow Control

Linux PC's use RTS/CTS flow control, but DTR/DSR flow control (used by some terminals) behaves similarly. DTR flow control (in one direction only and also used by some terminals) is only the DTR part of DTR/DSR flow control.

RTS/CTS uses the pins RTS and CTS on the serial (EIA-232) connector. RTS means "Request To Send". When this pin stays asserted (positive voltage) at the receiver it means: keep sending data to me. If RTS

TOC

is negated (voltage goes negative) it negates "Request To Send" which means: request not to send to (stop sending). When the receiver is ready for more input, it asserts RTS requesting the other side to resume sending. For computers and terminals (both DTE type equipment) the RTS pin sends the flow control signal to the CTS pin (Clear To Send) on the other end of the cable. That is, the RTS pin on one end of the cable is connected to the CTS pin at the other end.

For a modem (DCE equipment) it's a different scheme since the modem's RTS pin receives the signal and its CTS pin sends. While this may seem confusing, there are valid historical reasons for this which are too involved to discuss here.

Terminals usually have either DTR or DTR/DSR flow control. DTR flow control is the same as DTR/DSR flow control but it's only one-way and the DSR pin is not used. For DTR/DSR flow control at a terminal, the DTR signal is like the signal sent from the RTS pin and the DSR pin is just like the CTS pin.

Connecting Up DTR or DTR/DSR Flow Control

Some terminals use only DTR flow control. This is only one-way flow control to keep the terminal from being overrun. It doesn't protect the computer from someone typing too fast for the computer to handle it. In a standard null modem cable the DTR pin at the terminal is connected to the DSR pin at the computer. But Linux doesn't support DTR/DSR flow control (although drivers for some multiport boards may support DTR/DSR flow control.) A way around this problem is to simply wire the DTR pin at the terminal to connect to the CTS pin at the computer and set RTS/CTS flow control (stty crtscts). The fact that it's only one way will not affect anything so long as the host doesn't get overwhelmed by your typing speed and drop RTS in a vain attempt to lock your keyboard. See Keyboard Lock. For DTR/DSR flow control (if your terminal supports this two-way flow control) you do the above. But you also connect the DSR pin at the terminal to the RTS pin at the computer. Then you are protected if you type too fast.

Old RTS/CTS Handshaking Is Different

What is confusing is that there is the original use of RTS where it means about the opposite of the previous explanation above. This original meaning is: I Request To Send to you. This request was intended to be sent from a terminal (or computer) to a modem which, if it decided to grant the request, would send back an asserted CTS from its CTS pin to the CTS pin of the computer: You are Cleared To Send to me. Note that in contrast to the modern RTS/CTS bi-directional flow control, this only protects the flow in one direction: from the computer (or terminal) to the modem.

For older terminals, RTS may have this meaning and goes high when the terminal has data to send out. The above use is a form of flow control since if the modem wants the computer to stop sending it drops CTS (connected to CTS at the computer) and the computer stops sending.

Reverse Channel

Old hard-copy terminals may have a reverse channel pin (such as pin 19) which behaves like the RTS pin in RTS/CTS flow control. This pin but will also be negated if paper or ribbon runs out. It's often feasible to connect this pin to the CTS pin of the host computer. There may be a dip switch to set the polarity of this signal.

10.8 Is Hardware Flow Control Done by Hardware

TOC

Some think that hardware flow control is done by hardware but (unless you are using an intelligent serial card with several serial ports) it's actually done by your operating system software. UART chips and associated hardware usually know nothing at all about hardware flow control. When a hardware flow control signal is received, the signal wire flips polarity and the hardware gives an electrical interrupt signal to the CPU. However, the hardware has no idea what this interrupt means. The CPU stops what it was doing and jumps to a table in main memory that tells the CPU where to go to find a program which will find out what happened and what to do about it.

It's this program (part of the serial device driver) that stops (or resumes) the sending. This program checks the contents of registers in the UART chip to find out that a certain wire has changed polarity. Then the software realizes that a flow control signal has been received and goes about stopping (or starting) the flow. However, if it's a stop signal that has been received, the flow stops almost instantly when the signal arrives because the interrupt has stopped whatever the CPU was doing (including the program that was sending the data and putting it in the serial ports hardware buffers for transmission). However any bytes (up to 16) which were already in the serial port's hardware transmit buffer will still get transmitted ?? Thus the hardware almost instantly stops the flow only because it's reaction to a hardware signal is to interrupt and stop everything the CPU was doing.

10.9 Obsolete ?? ETX/ACK or ENQ/ACK Flow Control

This is also software flow control and requires a device driver that knows about it. Bytes are sent in packets (via the async serial port) with each packet terminated by an ETX (End of Text) control character. When the terminal gets an ETX it waits till it is ready to receive the next packet and then returns an ACK (Acknowledge). When the computer gets the ACK, it then send the next packet. And so on. This is not supported by Linux ?? Some HP terminals use the same scheme but use ENQ instead of ETX.

---

11. Physical Connection

Multiport boards allow many terminals (or modems) to be connected to one PC computer. A terminal may be connected to its host computer either by a direct cable connection, via a modem, or via a terminal server.

11.1 Multiport I/O Cards (Adapters)

Additional serial cards may be purchased which have many serial ports on them called "multiport boards". These boards are not covered in this HOWTO but there is quite a lot of coverage in the Serial-HOWTO One company which had (in 1998) below average prices is ByteRunner.

11.2 Direct Cable Connection.

The simplest way to connect a terminal to a host computer is via a direct connection to a serial port on the computer. Most PC's come with a couple of serial ports, but one is usually used by a mouse. For the EIA-232 port, you need a null modem cable that crosses over the transmit and receive wires. If you want hardware flow control, you will probably use the DTR pin (or both the DTR and DSR pins).

Make sure you have the right kind of cable. A null modem cable bought at a computer store may do it (if it's long enough), but it probably will not work for hardware flow control. Such a cable may be labeled as a serial printer cable. See this to help determine if you should Buy or Make your own cable. Make sure you are connecting to your PC's serial port at the male DB25 or the DB9, and not your parallel port (female DB25).

Null Modem Cable Pin-out (3, 4, or 5 conductor)

If you only have DTR flow control (one-way) you may eliminate the RTS-to-DSR wire. If you have no hardware flow control, then you may also eliminate the CTS-to-DTR wire. Then if you have 2@ twisted pairs, you may then use 2 wires for signal ground per A Kludge using Twisted-Pair Cable. For a DB25 connector on your PC, you need:

```
PC male DB25                                Terminal DB25
  TxD   Transmit Data        2 --> 3          RxD   Receive Data
  RxD   Receive Data         3 <-- 2          TxD   Transmit Data
  SG    Signal Ground        7 --- 7          SG    Signal Ground
  CTS   Clear To Send        5 <--20          DTR   Data Terminal Ready
  RTS   Request To Send      4 --> 6          DSR   Data Set Ready
```

If you have a DB9 connector on your serial port, try the following:

TOC

```
        PC DB9                                  Terminal DB25
    RxD   Receive Data        2 <-- 2             TxD   Transmit Data
    TxD   Transmit Data       3 --> 3        RxD   Receive Data
    SG    Signal Ground       5 --- 7        SG    Signal Ground
    CTS   Clear To Send       8 <--20        DTR   Data Terminal Ready
    RTS   Request To Send     7 --> 6        DSR   Data Set Ready   **
```

The above don't have modem control lines so be sure to give a "local" option to getty (which is equivalent to "stty clocal"). Also if you need hardware flow control it must be enabled at your computer (use a -h flag with agetty) (equivalent to "stty crtscts").

Standard Null Modem Cable Pin-out (7 conductor)

The following diagrams show full "standard" null modem cables. One that you purchase is apt to be wired this way. They will work for terminals using software (Xon/Xoff) flow control (or no flow control). However, they don't work for hardware flow control since most terminals support DTR or DTR/DSR flow control (handshaking) but Linux doesn't.

```
PC male DB25                                Terminal DB25
TxD   Transmit Data         2 --> 3         RxD   Receive Data
RxD   Receive Data          3 <-- 2         TxD   Transmit Data
RTS   Request To Send       4 --> 5         CTS   Clear To Send
CTS   Clear To Send         5 <-- 4         RTS   Request To Send
DSR   Data Set Ready        6
                            |
DCD   Carrier Detect        8 <-- 20        DTR   Data Terminal Ready
SG    Signal Ground         7 --- 7         SG    Signal Ground
                                  6         DSR   Data Set Ready
                                  |
DTR   Data Terminal Ready  20 --> 8         DCD   Carrier Detect
```

Alternatively, a full DB9-DB25 null modem cable (will not work with terminal hardware handshaking; see above):

```
PC DB9                                      Terminal DB25
RxD   Receive Data          2 <-- 2         TxD   Transmit Data
TxD   Transmit Data         3 --> 3         RxD   Receive Data
                                  6         DSR   Data Set Ready
                                  |
DTR   Data Terminal Ready   4 --> 8         DCD   Carrier Detect
GND   Signal Ground         5 --- 7         GND   Signal Ground
DCD   Carrier Detect        1
                            |
DSR   Data Set Ready        6 <-- 20        DTR   Data Terminal Ready
RTS   Request To Send       7 --> 5         CTS   Clear To Send
```

```
CTS   Clear To Send          8 <-- 4     RTS   Request To Send
(RI   Ring Indicator         9 not needed)
```

(Yes, the pins 2 and 3 *really do* have the opposite meanings in DB9 connectors than in DB25 connectors!)

Using the above 2 connections provide full modem control signals and seemingly allow one to set "stty -clocal". Then one must turn on the terminal first (asserts DTR) before the port may be opened in a normal manner by getty, etc. But there is likely to be trouble if you fail to turn on the terminal first (see Getty Respawning Too Rapidly). For this reason one should use "stty clocal" which is the default (ignores modem control lines) and the additional wires in these cables then serve no useful purpose.

In olden days when it may not have been this easy to ignore modem control signals etc, the following "trick" was done for cables that lacked conductors for modem control: on your computer side of the connector, connect RTS and CTS together, and also connect DSR, DCD and DTR together. This way, when the computer needs a certain handshaking signal to proceed, it will get it (falsely) from itself.

Length Limitations

A cable longer than a 50 feet or so may not work properly at high speed. Much longer lengths sometimes work OK, especially if the speed is low and/or the cable is a special low-capacitance type and/or the electronics of the receiving end are extra sensitive. It is claimed that under ideal conditions at 9600 baud, 1000 feet works OK. One way to cover long distances is to install 2@ line drivers near each serial port so as to convert unbalanced to balanced (and conversely) and then use twisted pair cabling. But line drivers are expensive.

Hardware Flow Control Cables

If you expect to use hardware flow control (handshaking) you will likely need to make up your own cable (or order one made). Of course, if the connecters on the ends of a used cable remove, you might rewire it. See Installing DB Connectors. You will need to determine whether or not the terminal uses the DTR pin for this, and if not, what pin (or pins) it uses. The set-up menus may give you a clue on this since there may be an option for enabling "DTR handshaking" (or flow control) which of course implies that it uses the DTR pin. It may also use the DSR pin. See Hardware Flow Control for a detailed explanation of it. Older terminals may have no provision for hardware flow control.

Cable Tips

The normal "straight thru" cable will not work unless you are using it as an extension cable in conjunction with either a null modem cable or a null modem adapter. Make sure that the connectors on the cable ends will mate with the connectors on the hardware. One may use telephone cable which is at least 4-conductor (and possibly twisted pair). Shielded, special low-capacitance cable computer cable is best.

A Kludge using Twisted-Pair Cable

Although none of the EIA-232 signals are balanced for twisted pair one may attempt to use twisted-pair cable with it. Use one pair for transmit and another for receive. To do this connect signal ground to one wire in each of these 2 pair. Only part of the signal ground current flows in the desired wire but it may

TOC

help. Due to the lower inductance of the twisted pair circuit (as compared to ground return current by some other path) more return (ground) current will confine itself to the desired twisted pair than one would expect from only resistance calculations. This is especially true at higher frequencies since inductive impedance increases with frequency. The rectangular wave of the serial port contains high frequency harmonics.

Cable Grounding

Pin 1 (of a DB25) should be chassis ground (also earth ground) but on cheap serial ports it may not even be connected to anything. A 9-pin connector doesn't even have a chassis ground. The signal ground is pin 7 and is usually grounded to chassis ground. This means that part of the signal current will flow thru the ground wires of the building wiring (undesirable). Cable shields are supposed to be only grounded at one end of the cable, but it may be better to ground both ends since it's better to have current in the shield than in the building wiring ??

11.3 Modem Connection

Using a terminal-modem combination (without a computer) one may connect to BBS's. Some BBS's (such a free-nets) permit Internet access via the text browser lynx which will work on text terminals. Thus with an old terminal and external modem, one may connect to the Internet. If one connects to a host computer on which one has an account, then one may sometimes store ones work (or downloads) there.

Dialing Out From a Terminal

Instead of connecting a terminal (or computer emulating a terminal) directly to a host computer using a cable it may be connected to the host via a telephone line (or dedicated private line) with a modem at each end of the line. The terminal (or computer) will usually dial out on a phone line to a host computer.

Most people use a PC and modem for dialing out. The PC could have a terminal connected to a serial port and the person at the terminal may dial out using the PC. Connecting a real terminal directly to an external modem is more difficult since the real terminal isn't very intelligent and doesn't give as much feedback to the user. For dialing out, many terminals can store one or more telephone numbers as messages which may be "set-up" into them and are sent out to the modem by pressing certain function keys. Many modems can also store phone numbers. The modem initiation sequence must precede the telephone number. The host computer may use "mgetty" (a getty for use with modems) to log in someone who has dialed in to the host.

Terminal Gets Dialed Into

It's common for a computer running Linux to get dialed into. The caller gets a login prompt and logs in. At first glance, it may seem strange how a dumb terminal (not connected to any computer) could accept an incoming call, but it can. One possible reason for doing this is to save on phone bills where rates are not symmetric. Your terminal needs to be set up for dial-in: Set the modem at your terminal for automatic answer (Register S0 set to 2 will answer on the 2nd ring). You turn on the terminal and modem before you expect a call and when the call comes in you get a login prompt and log in.

TOC

The host computer that dials out to your terminal needs to do something quite unusual. As soon as yo modem answers, it needs to run login (getty). A host may do this by running the Linux program "callback" sometimes named "cb". Callback is for having computer A call computer B, and then B hangs up and calls A back. This is what you want if you are using computer A to emulate a terminal. For the case of a real terminal this may be too complex a task so the host may utilize only the "back" part of the callback program. The setup file for callback must be properly configured at the host. Callback makes the call to the terminal and then has mgetty run a login on that port. Mgetty by itself (as of early 1998) is only for dial-in calls but there is work being done to incorporate callback features into it and thus make it able to dial-out. As of early 1999 it didn't seem to have been done.

11.4 Terminal Server Connection

A terminal server is something like an intelligent switch that can connect many terminals (or modems) to one or more computers. It's not a mechanical switch so it may change the speeds and protocols of the streams of data that go thru it. A number of companies make terminal servers: Xyplex, Cisco, 3Com, Computone, Livingston, etc. There are many different types and capabilities. Another HOWTO is needed to compare and describe them (including the possibility of creating your own terminal server with a Linux PC). Most are used for modem connections rather than directly connected terminals.

One use for them is to connect many terminals (or modems) to a high speed network which connects to host computers. Of course the terminal server must have the computing power and software to run network protocols so it is in some ways like a computer. The terminal server may interact with the user and ask what computer to connect to, etc. or it may connect without asking. One may sometimes send jobs to a printer thru a terminal server.

A PC today has enough computing power to act like a terminal server for text terminals except that each serial port should have its own hardware interrupt. PC's only have a few spare interrupts for this purpose and since they are hard-wired you can't create more by software. A solution is to use an advanced multiport serial card which has its own system of interrupts (or on lower cost models, shares one of the PC's interrupts between a number of ports). See Serial-HOWTO for more info about such cards. If such a PC runs Linux with getty running on many serial ports it might be thought of as a terminal server. It is in effect a terminal server if it is linked to other PC's over a network and if its job is mainly to pass thru data and handle the serial port interrupts every 14 (or so) bytes. Software called "radius" is sometimes used.

Today real terminal servers serve more than just terminals. They also serve PC's which emulate terminals, and are sometimes connected to a bank of modems connected to phone lines. Some even include built-in modems. If a terminal (or PC emulating one) is connected directly to a modem, the modem at the other end of the line could be connected to a terminal server. In some cases the terminal server by default expects the callers to use PPP packets, something that real text terminals don't generate.

11.5 Connector and Adapter Types

A connector is more-or-less permanently attached to the end of a cable or to a hardware unit. There are two basic types of connectors used in serial communications: 1. DBxx with pins (such as DB25) and 2. modular telephone-style connectors.

An adapter looks about like a connector but it has two ends. It is just like a cable that is so short that there is no cable part left at all --just different connectors on each end is all that remains. The adapter just plugs in on each side. It allows two incompatible connectors to mate with each other by going in between them. Sometimes the purpose of the adapter is to interchange wires. Obviously, one may use a special cable (perhaps homemade) as a substitute for an adapter.

Sex of Connector/Adapters

Connectors (or one side of adapters) are either male or female. The connectors that have pins are male and the ones that have sockets (sometimes also called pins) are female. For modular connectors, the ones with exposed contacts are plugs while the ones with internal contacts (not easy to see) are jacks. Plugs are male; jacks are female.

Types of Adapters

There are three basic types of adapters: null modem, gender changers and port adapters. Some adapters perform more than one of these three functions.

- null modem adapter: Reroutes wires. Like a null modem cable.
- gender changer: Changes the sex of a cable end. Two connectors of the same sex can now connect (mate) with each other.
- port adapter: Goes from one type of connector to another (DB9 to DB 25, etc.)

DB Connectors

(For how to install a DB connector on the ends of a cable see Installing DB Connectors.) These come in 9 or 25 pins. The EIA-232 specs. call for 25 pins but since most of these pins are not used on ordinary serial ports, 9 pins is sufficient. See DB9-DB25 for the pin-out. The pins are usually numbered if you look closely enough or use a magnifying glass.

RJ Modular Connectors

These look like modern telephone connectors but are sometimes not compatible with telephone connectors. See also Installing RJ Connectors. They may be 6, 8, or 10 conductor. RJ11/14 is a 4-6 conductor telephone plug. A look-alike is a MMJ connector (6-conductor) used on later model VT (and other) terminals. MMJ has an offset tab and is not compatible with RJ11/14. However, some connectors have been made that are compatible with both MMJ or RJ11/14. The MMJ pin-out is: 1-DTR, 2-TXD, 3-TXD GND, 4-RXD GND, 5-RXD, 6-DSR.

A null-modem cable with MMJ (or RJ11/14) connectors will connect: 1-6, 2-5, and 3-4. Note that such a cable support DTR/DSR flow control which is not supported (yet) by Linux. Making up your own 6-conductor null-modem cable is very simple if you understand that the ordinary 4-conductor telephone cable from the wall to your telephone, used in hundreds of millions of homes, is also a null-modem cable. Find one and wire your cable the same way.

If you lay such a cable (or your terminal null-modem cable) flat on the floor (with no twists) you will note

that both plugs on the ends have their gold contacts facing up (or both facing down). Although it's symmetrical, it is also null- modem if you think about it a bit. One may put a few such cables togethe with inline couplers and everything works OK because each inline coupler is also a null-modem adapter. Two null-modem devices in series result in a straight-thru connection.

RJ45 and RJ48 are 8-conductor (or 10-conductor for some computer applications) telephone plugs. The 10-conductor ones are allegedly wider and will not mate with 8-conductor ones. They are used for both flat telephone cable and round twisted pair cable. The cable end of the connector may be different for round and flat cable and both RJ45 and RJ48 may be 8 or 10 conductor so make sure you get the right one. RJ48 has an extra tab so that a RJ48 plug will not push into a RJ45 jack (but a RJ45 plug will mate with a RJ48 jack). They're used on some multiport serial cards and networks. Heres the pin numbers for an 8-conductor:

```
Plug                          Jack
(Looking at the end           (Looking at the cavity
 end of a cable)               in a wall)
   .___________.                 .___________.
   | 87654321  |                 | 12345678  |
   |__.    .__|                 |__.    .__|
      |____|                        |____|
```

11.6 Making or Modifying a Cable

Buy or Make ?

You may try to buy a short, null modem cable. They are often labeled as serial printer cables (but serial printers are not very popular today and neither are the cables). Unfortunately, they will probably not work for hardware flow control. Make sure the connectors on the cable ends will fit the connectors on your computer and terminal.

But if you need longer cables to connect up terminals or need hardware flow control, how do you get the right cables? The right ready-made cables may be difficult to find (you might find them by searching the Internet), especially if you want to use a minimum (say 4) of conductors. One option is to get them custom made, which is likely to be fairly expensive although you might find someone to make them at prices not too much higher than ready-made cable (I did). Another alternative is to make your own. This may require special tools. If you can find it, used cable can be a real bargain but you will probably need to rewire the connectors. Most connectors that come with short cables are permanently molded to the cable and can't be rewired but most custom-made and homemade cables have connectors that can be rewired. One advantage of making your own cable is that the skills you learn will come in handy if a cable breaks (or goes bad) or if you need to make up another cable in a hurry.

Pin Numbers

The numbers of the pins should be engraved in the plastic of the connector. Each pin should have a number next to it. You may need a magnifying glass to read them.

Installing DB Connectors on Cable Ends

TOC

See DB Connectors for a brief description of them. Unfortunately, most cables one purchases today have molded connectors on each end and can't be modified. Others have connectors which unscrew and can be rewired. If you are making up cable or modifying an existing one then you need to know about pins. There are two types: soldered and crimped.

The crimped pins require a special crimping tool and also need an "insertion/extraction" tool. But once you have these tools, making up and modifying cable may be faster than soldering. If you are connecting two wires to one pin (also needed if you want to jumper one connected pin to another pin) then soldering is faster (for these pins). This is because the crimped pins can only take one wire each while the soldered ones can accept more than one wire per pin.

To insert crimped pins just push them in by hand or with the tool. Removing a pin with the tool is a little tricky. These directions can be best understood if you have both the tool and wires in front of you. You must put the tool tip around a wire at the the back of a pin and insert it as far as it will go into the hole (about 1 1/2 cm.). Some tools have a mark (such as a tiny hole) on them to indicate how far to insert it. The tool tip should have a tapered gap so that you may get the tip around the wire by starting it in where the gap is wider than the wire. The tool may have 2 tips. The one that is the most difficult to get around the wire is also the one that removes the wire the easiest since it almost completely envelops the wire.

With the tip properly inserted pull on both the tool and the wire with a gentle pull. If it doesn't come out, the tool was likely not inserted correctly so either push it in more or twist it to a different position (or both). Perhaps you should have used the other tip that goes more around the pin. Using this tool, one may readily convert a straight-thru cable to a null-modem cable, etc.

There can be problems using the "insertion/extraction" tool. If the tools will not insert on the back of the pin, it could be that the pin was not neatly crimped to the wire and is sort of square where it should be round, etc. If a pin starts to come out but will not pull out all the way, the pin may be bent. Look at it under a magnifying glass. Straightening a pin with needle-nose pliers may damage the gold plating. Sometimes a stuck pin may be pushed out with a thick screwdriver blade tip (or the like) but if you push too hard you may gouge the plastic hole or bend the pin:.

Don't try soldering unless you know what you're doing or have read about how to do it.

Installing RJ Connectors

These are telephone modular connecters one type of which is used for most ordinary telephones. But there are many different types (see RJ Modular Connectors).

These are not easy to reuse. You might be able to pull the wires out, push in something wedged that would lift up the gold-colored contacts and reuse the connector. There are special crimping tools used to install them; a different tool for each type.

If you don't have a crimping tool, installation is still possible (but difficult) using a small screwdriver (and possibly a hammer). Push in the cable wires and then push each gold-colored contact down hard with a small screwdriver that will just fit between the insulating ridges between the contacts. You may damage it if you fail to use a screwdriver with a head almost the same thickness as the contacts or if the screwdriver

slips off the contact as you are pushing it down. You may also use a small hammer to pound on the screwdriver (push first by hand).

Be sure to not hurt the "remove lever" on the connecter when you push in the contacts. Don't just set it down on a table and push in the contacts. Instead, put a shim (about 1 mm thick) that fits snugly in the crevice between the lever and the body. For such a shim you may use thick cardboard, several calling cards, or wood. Since the bottom of the connector (that you will put on the table) isn't level (due to the "remove lever), make sure that the table top has something a little soft on it (like a sheet of cardboard) to help support the non-level connector. Even better would be to put another 1mm shim under the first 6mm of the connector, supporting it just under where you see the contacts. A soft tabletop wouldn't hurt either. Another method (I've never done this) is to hold the connector in a vice but be careful not to break the connector.

As compared to using a crimping tool, installing it per above takes a lot longer and is much more prone to errors and failure but it's sometimes more expedient and a lot cheaper than buying a special tool if you only have one or two connectors to install.

---

12. Set-Up (Configure) in General

12.1 Intro to Set-Up

Configuring (Set-Up) involves both storing a configuration in the non-volatile memory of the terminal, and putting commands in start-up files (on you hard disk) that will run each time the computer is powered on (or possibly only when the run-level changes). This section gives an overview of configuring and covers the configuring of the essential communication options for both the terminal and the computer. The next two major sections cover in detail the configuration of the terminal (see Terminal Set-Up and the computer (see Computer Set-Up (Configure) Details.

12.2 Terminal Set-Up (Configure) Overview

When a terminal is installed it's necessary to configure the physical terminal by saving (in its non-volatile memory which is not lost when the terminal is powered off) the characteristics it will have when it is powered on. You might be lucky and have a terminal that has already been set-up correctly for your installation so that little or no terminal configuration is required.

There are two basic ways of configuring a terminal. One is to sit at the terminal and go thru a series of set-up menus. Another is to send escape sequences to it from the host computer. Before you can send anything to the terminal (such as the above escape sequences), its Communication Interface) options such as the baud rate must be set up to match those of the computer. This can only be done by sitting at the terminal since the communications must be set up right before the computer and the terminal can "talk" to each other. See Terminal Set-Up.

12.3 Computer Set-Up (Configure) Overview

Besides possibly sending escape sequences from the computer to configure the terminal, there is the configuring of the computer itself to handle the terminal. If your lucky, all you need to do is to put a "getty" command in the /etc/inittab file so that a "login:" prompt will be sent to the terminal when the computer starts up. See Getty for details.

The computer communicates with the terminal using the device driver software (part of the kernel). The serial device driver has a default configuration and is also partly (sometimes fully) configured by the getty program before running "login" at each terminal. However, additional configuration is sometimes needed using programs named "stty" and "setserial". These programs (if needed) must be run each time the computer starts up since this configuration is lost each time the computer powers down. See Computer Set-Up (Configure) Details.

TOC

12.4 Many Options

There are a great many configuration options for you to choose from. The communication options must be set right or the terminal will not work at all. Other options may be set wrong, but will cause no problem since the features they set may not be used. For example, if you don't have a printer connected to the terminal it makes no difference how the printer configuration parameters are set inside the terminal. This last statement is not 100% correct. Suppose that you have no printer but the compute (by mistake) sends the terminal a command to redirect all characters (data) from the computer to the printer only. Then nothing will display on the screen and your terminal will be dead. Some terminals have a configuration option to inform the terminal that no printer is attached. In this case the terminal will ignore any command to redirect output to the "printer" and the above problem will never happen. However, this doesn't help much since there are many other erroneous commands that can be sent to your terminal that will really foul things up. This is likely to happen if you send the terminal a binary file by accident.

In some cases a wrong setting will not cause any problem until you happen to run a rare application program that expects the terminal to be set a certain way. Other options govern only the appearance of the display and the terminal will work fine if they are set wrong but may not be as pleasant to look at.

Some options concern only the terminal and do not need to be set at the computer. For example: Do you want black letters on a light background? This is easier on the eyes than a black background. Should a key repeat when held down? Should the screen wrap when a line runs off the right end of the screen? Should keys click?

12.5 Communication Interface Options

Some of these communication settings (options) are for both the terminal and the computer and they must be set exactly the same for both: speed, parity, bits/character, and flow control. Other communication options are only set at the terminal (and only a couple of these are essential to establish communications). Still others such as the address and interrupt (IRQ) of the physical port ttyS2 are set only at the computer using the "setserial" command. Until all of the above essential options are compatibly set up there can be no satisfactory serial communication (and likely no communication at all) between the terminal and the computer. For the terminal, one must set these options manually by menus at each terminal (or by using some sort of special cartridge at each terminal). The host computer is configured by running commands each time the computer is powered up (or when people log in). Sometimes the getty program (found in the /etc/inittab file) which starts the login process will take care of this for the computer. See Getty (in /etc/inittab)

The settings for both the computer and the terminal are:

- Speed (bits/second)
- Parity
- Bits per Character
- Flow Control

TOC

Some essential settings for the terminal alone are:

- Port Select
- Set communication to full duplex (=FDX on Wyse terminals)

If the Getty (in /etc/inittab) program can't set up the computer side the way you want, then you may need to use one (or both) of the Stty & Setserial commands.

Speed

These must be set the same on both the terminal and the computer. The speed is the bits/sec (bps or baud rate). Use the highest speed that works without errors. Enabling flow control may make higher speeds possible. There may be two speeds to set at the terminal: Transmit and Receive, sometimes abbreviated T and R. Usually they are both set the same since stty in Linux doesn't seem to have the option yet of setting them differently. (There is an option to do this with the "stty" command but it seems to actually set them both the same.) Common speeds are 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, ... The slower speeds (like 600) are for printers and hard-copy terminals.

Parity & Should You Use It ?

For a definition see Parity Explained. Parity-disabled is often the default. To enable parity, you must both enable it and then select either even or odd parity. It probably makes no difference if it's odd or even. For terminals there are sometimes settings for both transmit and receive parity. You should set both of these the same since stty at the computer doesn't permit setting them differently. Some terminal are unable to set receive parity and will simply always ignore received parity bits. On some older terminals if you use 8-data-bits per byte then parity will not work since there is no room in the hardware for the extra parity bit.

Should you use parity at all? Parity, while not really necessary, is nice to have. If you don't have parity, then you may get an incorrect letter here and there and wind up trying to correct spelling errors that don't really exist. However parity comes at a cost. First, it's more complicated to set up since the default is usually no parity. Secondly, parity will slow down the speed with which bytes travel over the serial cable since there will be one more bit per byte. This may or may not slow down the effective speed.

For example, a hard-copy terminal is usually limited by the mechanics of the printing process. Increasing the bytes/sec when the computer (its UART chip) is transmitting only results in more flow-control "halt" signals to allow the mechanical printing to catch up. Due to more flow-control waits the effective speed is no better without parity than with it. The situation is similar for some terminals: After you implement parity there may be fewer flow-control waits per unit time resulting in more bits/sec (average). However, due to the added parity bits the bytes/sec (average) stays the same.

One option is to install terminals with no parity. Then if parity errors are noticed, it can be implemented later. To spot possible errors with no parity, look for any spelling errors you don't think you made. If you spot such an error, refresh the screen (retransmit from the computer). If the error goes away, then it's likely a parity error. If too many such errors happen (such as more than one every few hundred screens) then corrective action is needed such as: Enable parity and/or reduce speed, and/or use a shorter/better cable. Enabling parity will not reduce the number of errors but it will tell you when an error has

happened.

Just the opposite policy is to initially enable parity. Then if no parity errors (error symbols on the CRT) are ever seen (over a reasonable period of time, say a month or two) it may be safely disabled.

Bits/Character

This is the character size (the number of data bits per character excluding any parity bit). For ASCII it's 7, but it's 8 for ISO character sets. If you are only going to use ASCII characters, then select 7-bits since it's faster to transmit 7 bits than 8. Some older terminals will only display 7-bit characters.

Which Flow Control (Handshaking) ?

The choice is between "hardware" (for example dtr/cts) or "software" (Xon/Xoff) flow control. (The Adds terminal menu incorrectly use "Xon/Xoff" to mean any kind of flow control.) While hardware flow control may be faster (if the one or two extra wires for it are available in the cable and if the terminal supports it) in most cases Xon/Xoff should work OK. Some people report that they solved disturbing problems (see below) by converting to hardware flow control but software flow control has worked fine at other installations (and for me personally).

If you use software (Xon/Xoff) flow control and have users who don't know about it, then they may accidentally send an Xoff to the host and lock up their terminal. While it's locked, they may type frantically in a vain attempt to unlock it. Then when Xon is finally sent to restore communication, all that was typed in haste gets executed, perhaps with unexpected results. They can't do this with hardware flow control. See Flow Control for an explanation of flow control.

Port Select

Since most terminals have two or more connectors on the back, it is usually possible to assign one of these connecters to connect to the host computer and assign another connector to be the printer port. The connector may have a name next to it (inspect it) and this name (such as Aux, Serial 2, or Modem) may be assigned to either be the main host connection or the printer connection (or the like).

12.6 Quick Attempt

While all the above may seem overly complex, to get a terminal working is often fairly simple. The Quick Install section describes a simple way to try to do this. But if that doesn't work or if you want to make the display look better and perform better, more reading will be needed.

---

13. Terminal Set-Up (Configure) Details

Except for the next subsection on sending escape sequences to the terminal, this section mainly presents the details of setting up the terminal manually by sitting at the terminal and going thru menus. If you haven't already done so, you should read Terminal Set-Up (Configure) Overview. It's best if you have a terminal manual, but even it you don't there is information here on many of the options which you might possibly need to set.

The communication parameters such as its baud rate must always be set up at the terminal since if this is not done there can be no communication with the terminal. Once communication is established you have two choices for doing the rest the terminal configuration. You may continue to configure manually at the terminal and save the results in the terminal's non-volatile memory or you may do this by sending escape sequences to the terminal from the computer each time the terminal is powered on (or the like).

If you know how to set up and save a good configuration inside the terminal it may be the best way. If you don't, you might want to just send the init string from terminfo to your terminal each time you use the terminal. Perhaps doing nothing will still give you a usable terminal. You (or an application program) can always change things by sending certain escape sequences to the terminal.

13.1 Send Escape Sequences to the Terminal

Once the communication interface is established, the rest of the configuration of the terminals may sometimes be done by sending escape sequences to the terminals from the computer. If you have a large number of terminals, it may be worthwhile to write (or locate) a shell script to automatically do this. There may (or may not) be a command you can send to a terminal to tell it to save its current set-up in its non-volatile memory so that it will be present the next time the terminal is powered on.

There is an simple way to send these escape sequences and a complex way. Using the simple way, you never look up escape sequences but issue commands that automatically find an appropriate escape sequence in the terminfo database and send that. Unfortunately, not all the escape sequences which you might want to send are always in the terminfo database. Thus the more complex (but possibly better) way is to directly send escape sequences.

For this complex method you'll need an advanced manual. Old terminal manuals once included a detailed list of escape sequences but newer ones usually don't. To find them you may need to purchase another manual called the "programmers manual" (or the like) which is not supplied with the terminal. A Esc Sequence List for some terminals is on the Internet but it's terse and likely incomplete.

Even without a manual or the like, you may still send commands to configure the terminal by using the programs "tput" and "setterm". See Changing the Terminal Settings. You could just send the terminal an init string from the terminfo entry if the init string sets up the terminal the way want it. See Init String. Unless you plan to have these sequences sent from the computer to the terminal each time the terminal is

powered on, you must somehow save the settings in the non-volatile memory of the terminal.

13.2 Older Terminals Set-Up

On older terminals look at the keyboard for labels just above the top row of numeric keys. If they exist, these labels may be what these keys do in set-up mode. Some older terminals may have only one "set-up" menu. Still older ones have physical switches. In some cases not all the switches are well labeled but they may be well concealed. Of course, if you set something with a switch, it's "saved" and there is no need to save the setting in non-volatile memory.

13.3 Getting Into Set-Up (Configuration) Mode

To select options (configure) at the terminal, you must first enter "set-up" mode and then select options (i.e. configure) using menus stored inside the terminal and displayed on the screen. To do this, the terminal does not even need to be connected to a computer. How to get into set-up mode is covered in the terminal's manual, but here's some hints:

If there's a "set-up" key try pressing it. Also try it shifted.

- Wyse: First try the shifted "Select" key; then substitute Ctrl for shifted in all of the above.
- VT, Dorio: F3 may be the set-up key. On VT420 and later models this key may have been programmed to do something else so turn off the power. When you turn on the power again, hit the F3 key as soon as you get an initial screen message.
- IBM: 3151: Ctrl-ScrollLock. 3153: Ctrl-Minus_on_Keypad (or like 3151)

To move around in the set-up menus, try the arrow keys. Use Return, Space, or a special key ("toggle" on old terminals) to select. To exit set-up mode select exit from a menu (or on some older terminals press the set-up key again).

13.4 Communication Options

For the terminal to work at all, speed, parity, bits/character, and communication mode must be set correctly. Incorrect flow control may cause loss and/or corruption of data seen on the screen. The essential communication options were dealt with (for both the terminal and computer) in another section: See Communication Interface. The following list provides some links to that section, as well as some additional communication options set only at the terminal.

- Speed (bits/second) (baud rate): 9600, 19200, etc.
- Parity none, even, odd, mark, space
- Bits per Character {Data}: 7 or 8
- Flow Control: or Handshake {Hndshk}: none, Xon-Xoff, or hardware (DTR, etc).
    - Receiver Handshake {Rcv Hndshk} protects data being Received by the terminal by transmitting flow-control signals to the host.
    - Transmitter Handshake {Xmt Hndshk} is protection of data being Transmitted by the

terminal. The terminal receives flow-control signals (and locks/unlocks the keyboard). Includes "Incoming Xon/Xoff".

- number of stop bits: 1 or 2. See Voltage Sequence for a Byte
- Flow control level {Rcv Hndshk Level} {{Xoff at ...}}: Flow control will send "stop" when this number of bytes in the terminal's buffer is exceeded.
- Communication Mode {Comm}: Full Duplex {FDX}, Half Duplex {HDX} {{Local Echo}}, Local Mode {{Online/Local}}
- Transmit Rate (Speed) Limit {Xmt Lim}: limits the transmit rate to the specified cps (chars/sec) even though the baud rate setting may be at a higher speed.
- Function-Key Rate Limit: as above but for function key messages.
- Port Select: Which physical connecter is for the host {Host Port} ?

13.5 Saving the Set-up

Your set-up must be saved in the non-volatile memory of the terminal so that it will be effective the next time you turn on the terminal. If you fail to save it, then the new settings will be lost when you turn off the terminal. Before you go to the trouble of setting up a terminal, make sure that you know how to save the settings. For modern terminals the save command is done via a menu. In some older terminals, only the manual tells how to save. For many of these you press Ctrl-S to save.

13.6 Set-Up Options/Parameters

See the Teemworld's Set-Up for a description of many of these parameter as used in terminal emulation. Emulation is often a little different than an actual terminal.

What follows in this section describes some of the options which are available in the set-up menus of many terminals. Options are also called parameters or features. Many options may be called "modes". Setting options is often called "configuring". Many of these options may also be set by sending certain escape sequences to the terminal. Different models and brands of terminals have various options and the same option may be called by different names (not all of which are given here) Terse names used by Wyse are enclosed in {...}. Names used mostly for VT terminals are enclosed in {{...}}.

13.7 Emulation {Personality} {{Terminal Modes}}

Most modern terminals can emulate several other terminals. The terminal can likely do more if it is set to emulate itself (actually no emulation) {native personality}. Sometimes there are 2 different emulations for the same model of terminal. For example VT220-7 emulates a VT200 with 7-bits/byte while VT220-8 emulates a VT220 with 8-bits/byte (256 possible characters).

Older models of terminals usually have fewer features than newer models. Suppose one wanted to emulate an old terminal but also wanted some of the advanced capabilities of the later model terminal they are sitting at. This is sometimes possible (to some degree). This feature is sometimes called

{Enhance} (or Enhanced ??).

13.8 Display Options

Character Cell Size {Char Cell}

This is the size of the cell in which a character fits. It is measured in pixels (=tiny dots). The more dots, the better the resolution. 10x16 is 10 dots wide by 16 dots high (16 rows and 10 columns). Note the notation is inverted as compared to the notation for matrix dimensions which gives rows (height) first.. Also, the character cell includes rows and columns of pixels allocated for the space between adjacent characters so the cell size which defines the boundaries of an actual character may be smaller.

Columns/Lines

Usually 80 columns and 24 or 25 lines. This means that there may be up to 80 characters in a row (line) on the screen. Many terminals have a 132 column option but unless you have a large screen, the tiny characters may be hard to read. {{Set 132 column mode}}

Cursor

The cursor may be set to appear as a rectangle (= block) {Blk}. Other options are underline {Line} or blinking. I prefer non-blinking {Steady} block since it's big enough to find quickly but there is no distractive blinking. If you set it invisible (an option on some terminals) it will disappear but new letters will appear on the screen as you type at the invisible cursor.

Display Attributes (Magic Cookies)

Display Attributes may either be magic cookies or be attribute bytes assigned to each character. For magic cookies, there is a limit to their extent: Are they in effect to the end of the line or to the end of the page? It's best to use attribute bytes (which could actually be half-bytes = nibbles).

Display Control Characters {Monitor}

May be called various names such as "Display Controls". When off (normal) it's "Interpret Controls". When set on, you see the escape sequences from the host (which you normally never see on the screen). So that these sequences may be viewed in sequence on a line, they are not acted upon (interpreted) by the terminal. Except that a CR LF sequence creates a new line. See Control Codes.

Double Width/Height

Some terminals can have their characters double width and/or double height. This feature is seldom needed. When changing a line to double width (DW) the right half (RH) is pushed off the screen and there is the question of whether or not to delete (erase) it. "Preserve" means to keep the RH of DW lines. When in double height mode, it may be necessary to send each such line twice (the 2nd time down one row) in order to get a double-height line on the screen.

TOC

Reverse Video {Display} (Background Light/Dark)

Normal video is light (white, green, amber) letters (foreground) on a dark (black) background. Reverse video {Display Light} is the opposite: black text on a light background. This is easier on the eyes (unless the room is dark).

Status Line

A status line is a line at the top or bottom of the screen that displays info about the application program you are running. It's often highlighted in some way. With a status line enabled, an application can send the terminal a special escape sequence which means that the text that follows is for the status line. However, many applications don't use this feature but instead only simulate a real status line by direct cursor positioning. The ordinary user looking at it doesn't know the difference.

Upon 80/132 Change: Clear or Preserve?

When switching the number of columns from 80 to 132 (or conversely) should the data displayed in the old format be erased (cleared) or preserved? {80/132 Clr} {{Screen Width Change}}. It should make no difference how you set this option since if an application program uses 132 columns, it should set this option appropriately via a control sequence.

13.9 Page Related Options

For a Wyse terminal to be able to access multiple pages of display memory {Multipage} must be set to on.

Page Size

The terminal memory may be divided up into a number of pages. See Pages and Pages (definition) for explanations of pages. You may partition the page memory into a number of pages of selected length. Linux applications don't seem to use pages at present so it shouldn't make much difference how you set this up.

Coupling (of cursor & display)

The terminal memory may be divided up into a number of pages. See Pages and Pages for explanations of pages. When the cursor is moved to a location in video memory not currently displayed (such as another page, or on the same page but to a location not displayed on the screen) should the display change to let one view the new cursor location? If so, this is called "Coupling". For cursor movement within the same page there is "Vertical Coupling" and "Horizontal Coupling". For movement to another page there is "Page Coupling".

TOC

13.10 Reporting and Answerback

The terminal will identify itself and its state, or send out a pre-recorded message in response to certain escape sequences.

Answerback Message (String)

You may write a short message during set-up which may optionally be sent to the host at power-up or be sent to the host in response to a request from the host (perhaps the ENQ (inquire) control character).

Auto Answerback

If set, sends the answerback message to the host at power-on without the host asking for it. Do any "getty" processes look for this ??

Answerback Concealed

If set, will never let anyone see the answerback message (except of course the host computer). If it needs to be changed, deselect "answerback concealed" and the formerly concealed message will be destroyed so you then may enter a new message (but you don't get to see the old one).

Terminal ID {ANSI ID}

The terminal send this reply in answer to a request for identity.

13.11 Keyboard Options

Keyclick

When set, pressing any key makes a click (broadcast by a tiny loudspeaker in the keyboard). These clicks annoy some people and I think it's best to set keyclick off.

Caps Lock {Keylock}

When the Caps-Lock key is down, should only the alphabetic keys generate shifted characters? If set to {Caps} or upper-case-only then hitting a number key with the Caps-Lock on will type the number. To get the symbol above the number one must manually hold down the shift key. This is the normal mode. If set to {Shift} then all keys type the shifted character when Caps-Lock is on (hitting the 5 key should type % without holding down Shift, etc.).

TOC

Auto Repeat {Repeat}

If a key is held down then that key is repeatedly "typed". This is handy for repeatedly typing the same character to create a line across the page.

Margin Bell

When the cursor is 8 columns away from the right side of the display, a bell is rung (like on an old typewriter). Almost all editors will automatically create a new line if needed (no need to hit the Return key) so this feature is seldom needed.

Remapping the Keys

The code sent to the host when a key is pressed is normally the ASCII code for that key (depends also on Shift and Control key). On some terminals you may make any key send any code you wish. That is, you may completely remap the keyboard by setting up the terminal that way. This may be useful for some foreign languages and Dvorak keyboard layouts, etc. which permit one to type faster.

Corner Key (for Wyse only)

Wyse terminals have a key near the lower left corner which may be set to assume various functions. Its either labelled "Funct", "Compose Character" or "Alt". When set to {Hold} No-Scroll it stops the flow of data (using flow control) to the terminal. Hitting the key again restores normal flow. When set to {Compose} it permits one to generate a limited number of pre-defined non-Latin characters. When set to Meta, it makes it a meta shift key which sets the high-order bit on each byte. When set to {Funct} (and pressed) any alphanumeric key pressed gets a header (SOH) and trailer (CR) byte framing the ASCII byte code. When set to {Kpd Compose} (and pressed) then typing a decimal number on the numeric keys (followed by "enter") sends out the same number in hexadecimal ??

Numeric Keypad or Arrow Keys Sends

The numeric keypad (the rectangle of mostly numeric keys to the right of the main part of the keyboard) can be set to send special codes which will do special things in certain application programs. Ditto for the arrow keys. There is thus a "normal" mode where they send what is shown on the keycap (or the normal code sequence for an arrow-key) and an "application" mode where a special escape sequence is sent. In some cases there is a "hex" numeric mode which is almost like normal numeric mode except that 6 non-numeric keys send the letters A-F. Thus one may type for example "B36F" on the numeric keypad.

What does shifted-del and shifted-bs send?

Depending on how they're set up shifted-del sometimes sends the control character CAN and shifted backspace sometimes sends DEL.

TOC

PC Scan Codes

Newer terminals can emulate a PC keyboard by sending PC scan codes (see Keyboard-and-Console-HOWTO) instead of ASCII codes. This would be used if you were directly connected to a PC running Dos/Windows. Set {Keycode} to {Scan}. Emulating the Dec "PCTerm" should do this and more. A serial port under Linux can't cope with such scan codes.

Alternate Characters

Some keys may have alternative letters on them. When keys is set to "Typewriter" they send what they would normally send on a typewriter. When keys is set to something else the alternative characters are sent.

13.12 Meaning of Received Control Codes

Auto New Line {Newline}

In this case "New Line" means a new line starting at the left margin below the current line. In Linux and C "new line" (NL) may have a different meaning: the line-feed control character LF also called new-line or NL. This is because in Linux text files, the LF character means a "new line starts here" so it's labeled NL. Normally, a LF (NL) sent to a terminal only results in the cursor jumping down one line below where is was and does not result in a new line.

If Auto New Line is set, the above "normal" situation is canceled and a physical new line is created on the display upon receiving a LF from the host. This is exactly what one wants in Linux. Except that (when Auto New Line is set) the Return (or Enter) key sends a CR LF sequence to the host (for Wyse and VT100, but for VT420 ??). Since Linux uses LF as a "new line" marker in files, Linux would like only a LF to be sent (and not a CR LF). Thus the "New Line" option is seldom used. Instead, the required translations are made by the serial port device driver by default. It is as if one gave the command "stty onlcr icrnl" but you don't need to do this since it's the default.

Auto Line Feed {Rcv CR}

This is just another type of "Auto New Line". When a CR (carriage return) character is received, a LF (line feed) action is added resulting in a new line being displayed. Since Linux marks the end of lines with LF, this option is not used.

Recognize Del (Wyse Only ??) or Null

If off, the DEL character received by the terminal is ignored. If on the DEL performs a destructive backspace. Null characters are usually ignored in any case. Both DEL and NULL are sometimes used for padding. See Padding

13.13 Where New Text Goes

TOC

Line Wrap

Also called "Auto Wrap(around)". What happens when the right edge of the screen is reached (col. 80, etc) and no return character (or the like) has been sent from the host? If Line Wrap is set, then the rest of the line displays on the line below, etc. Otherwise, the rest of the line is lost and is not seen on the screen. Any good application should provide "return" characters so that wrap happens before the end of the screen is reached. But a raw copy command (and other situations) may not do this so it's usually best to set line wrap.

For an 80 col. screen, most terminals only wrap if the 81st character from the host is a graphic (printable) character. This allows for the case where 81st character from the host might be "return" or a "newline" (non-graphic characters) which means that the application is handing the wrapping OK and intervention by the terminal is not needed.

Scrolling

Scrolling {Scrl} is where all the lines on the screen move up or down. Its also called "panning" which includes movement sideways. In ordinary scrolling lines roll off the bottom or top of the screen and disappear, and new lines from the host appear at the opposite edge (top or bottom). There are 3 types of this: smooth, jump, or burst. Burst is not really scrolling since its an instant replacement of an old screenfull by a new one (although some lines on the new screen may be from the old screen). Jump is where new lines jump into view one at a time. Smooth {Smth} is where the text moves at a steady speed upward or downward. If the smooth scroll rate is slow enough, one may read the newly visible lines when they are still scrolling (in motion).

Smooth scrolling on slow terminals was once useful since one could continue reading as the display was scrolling. But with higher baud rates, jump scroll is so fast that little time is lost as the new display appears. Since it takes a little longer to read scrolling text than fixed text, it may actually waste more time if smooth scrolling is selected.

If (auto)scrolling {Autoscrl} is disabled, then new text from the host must go somewhere so it is put at the top of the display. If the old text is not erased, the new text merges (nonsensically) into the old. If the old text is erased, then the new text is out of context. So keep (auto)scrolling enabled.

New Page?

See Pages and Pages for explanations of pages. When the current page is full (the last line is finished) should the page scroll, or should a new page be created (leaving the previous page stored in the terminal's display memory). If {Autopage} is set, then a new page is created. Since you are probably not using pages, you should probably set this to off.

TOC

13.14 Function Keys

These are the keys labeled F1, F2, etc. On older terminals they may be labeled PF1, PF2, etc. where the P stands for Programmable. Some keyboards have both. One may program (redefine) these keys to send out a string of user-defined bytes. They may often be easily "programmed" using a certain set-up menu {FKey}. On some terminals, one may also specify where this string is sent to when the key is pressed. In "normal" mode, pressing the key is just like typing the string at the keyboard. In "local" mode pressing the key sends it to the terminal (just like if the terminal was in local mode). This may be used to send escape sequences to the terminal so as to configure it in a special way. In "remote" mode, the string is always sent out the serial port to the host computer (even if the terminal is in local mode).

13.15 Block Mode Options

Some options are only for the case of Block Mode. This option is powerful since it provides forms and takes load off the host computer by transmitting in bursts. But it's more complicated to set up and is thus not used too much.

Forms Display

In block mode some regions of the screen are for the text of forms and are thus write-protected "Prot" {WPRT}. Options may set the characters in these regions to appear dim, reverse video {WPRT Rev}, and/or underlined {WPRT Undrln}. {WPRT Intensity} may be set to dim, normal, or even blank (invisible)

Send Entire Block ?

Should write-protected text (the original text in the form) be sent to the host upon transmission of a block: {Send All} or is write-protected text also read-protected: {Send Erasable}

Region to Send

Should the entire screen be sent or just the scrolling region? {Send Area}. Should the sending stop when the current cursor position is reached? If {Xfer Term} is set to Cursor, only the data on the screen up to the cursor is sent.

Block/Page terminator

What is the termination symbol to be appended to a block of data? {Blk End} or at the end of a page {Send Term}ination.

TOC

13.16 Locks

There are various types of Locks. One is the Locked keyboard due to flow control. See Keyboard Lock Another lock {Feature Lock} is that which prohibits the host computer from changing the terminal set-up by sending certain escape sequences to the terminal. Placing such a lock may result in unexpected behavior as application programs send escape sequences to the terminals that are ignored. Not all set-up parameters lock. Unless you have a good reason to do so, you should not enable such locking.

A Function Key lock will prohibit the computer from redefining what a programmable function key sends. You may want to use this if you have something important programmed into the function keys.

13.17 Screen Saver {Scrn Saver}

Also called "CRT Saver". This turns off (or dims) the screen after the terminal is not used for a period of time. It prolongs the life of the screen and may save some energy. Hitting any key will usually restore the screen and may "execute" that key so it's best to hit the shift-key, etc.

13.18 Printer

For Wyse, if there is no {Printer Attached} set it to Off. It's not essential to do this, but if you do it any escape sequence to send text to the printer (instead of the terminal) will be ignored.

Setting up the printer port is about the same (usually simpler) as setting up the communications on the main port. There are a couple of options specific to the printer. Is the printer a serial or parallel printer? If it's parallel it should be designated as such in setup and connected to the parallel port on the terminal (if there is one). Should a FF (form feed) be sent to the printer at the end of a print job? If {Print Term} is set to FF, this will happen.

---

14. Computer Set-Up (Configure) Details

There are various files to edit to set up the computer for terminals. If you're lucky, you'll only need to edit /etc/inittab. One does this by editing at the console (or from any working terminal).

14.1 Getty (in /etc/inittab)

In order to have a login process run on a serial port when the computer starts up (or switches run levels) a getty command must be put into the /etc/inittab file. Getty GETs a TTY (a terminal) going. Each terminal needs its own getty command. There is also at least one getty command for the console in every /etc/inittab file. Find this and put the getty commands for the real terminals next to it. This file may contain sample getty lines for text terminals that are commented out so that all you need to do is to uncomment them (remove the leading #) and change a few arguments.

The arguments which are permitted depend on which getty you use:
Two gettys best for directly connected terminals are:

1. agetty (sometimes just called getty): Very easy to set up. No config files. See Agetty
2. See getty (part of getty_ps)

Two gettys best for modems (avoid for terminals) are:

1. mgetty: the best one for modems; works for terminals too but inferior
2. uugetty: for modems only; part of the getty_ps package

A simple getty to use only for console login is:

1. mingetty: for consoles only

If you don't have the getty you want check other distributions and the `alien` program to convert between RPM and Debian packages. The source code may be downloaded from Serial Software.

If you are not using modem control lines (for example if you only use the minimum number of 3 conductors: transmit, receive, and common signal ground) you should let getty know this by using a "local" flag. The format of this depends on which getty you use.

Agetty (may be named getty)

An example line in /etc/inittab:

```
S1:23:respawn:/sbin/getty -L 19200 ttyS1 vt102
```

S1 is from ttyS1. 23 means that getty is run upon entering run levels 2 or 3. respawn means that if getty is killed, it will automatically start up (respawn) again. /sbin/getty is the getty command. The -L means Local (ignore modem control signals). -h (not shown in the example) enables hardware flow control (same as stty crtscts). 19200 is the baud rate. ttyS1 means /dev/ttyS1 (COM2 in MS-DOS). vt102 is the type of terminal and this getty will set the environment variable TERM to this value. There are no configuration files. Type "init q" on the command line after editing getty and you should see a login prompt.

The `agetty` program will auto-detect any parity set inside the terminal. If you use `stty` to set parity, `agetty` will automatically unset it since it wants the parity bit to come thru as if it was a data bit. This is because it needs to

TOC

get the last bit (possibly a parity bit) as you type your login-name so that it can auto-detect parity. Thus if you u parity, enable it only at the terminals and let `agetty` auto-detect it and set it at the computer. If your terminal supports received parity, the login prompt will look garbled until you type something so that getty can detect the parity. The garbled prompt will deter visitors, etc. from trying to login. That could be just what you want.

There is sometimes a problem with this garbled prompt if you don't have the terminal powered on when getty starts to run. The `agetty` program detects the parity OK but if the first login attempt fails then the `login` program is run. This `login` program can't detect parity. Normally, parity has already been detected so all is OK. But if the terminal was off, login may fail without you typing anything. It seems that you can't login, but if you keep slowly pressing the return key then eventually the login prompt will be supplied by `agetty` again and you may finally login. For terminals that only support transmitted parity, you will not see any garbling but you will not be able to login until the `agetty` supplies the login prompt.

getty (part of getty_ps)

(This is from the old Serial-HOWTO by Greg Hankins)
Add entries for `getty` to use for your terminal in the configuration file `/etc/gettydefs` if there they aren't already there:

```
# 38400 bps Dumb Terminal entry
DT38400# B38400 CS8 CLOCAL # B38400 SANE -ISTRIP CLOCAL #@S @L login: #DT38400

# 19200 bps Dumb Terminal entry
DT19200# B19200 CS8 CLOCAL # B19200 SANE -ISTRIP CLOCAL #@S @L login: #DT19200

# 9600 bps Dumb Terminal entry
DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S @L login: #DT9600
```

If you want, you can make `getty` print interesting things in the login banner. In my examples, I have the system name and the serial line printed. You can add other things:

```
@B    The current (evaluated at the time the @B is seen) bps rate.
@D    The current date, in MM/DD/YY.
@L    The serial line to which getty is attached.
@S    The system name.
@T    The current time, in HH:MM:SS (24-hour).
@U    The number of currently signed-on users.  This is  a
      count of the number of entries in the /etc/utmp file
      that have a non-null ut_name field.
@V    The value of VERSION, as given in the defaults file.
To display a single '@' character, use either '\@' or '@@'.
```

When you are done editing `/etc/gettydefs`, you can verify that the syntax is correct by doing:

```
linux# getty -c /etc/gettydefs
```

Make sure there is no `getty` or `uugetty` config file for the serial port that your terminal is attached to (`/etc/default/{uu}getty.ttyS`*N* or `/etc/conf.{uu}getty.ttyS`*N*), as this will probably interfere with running `getty` on a terminal. Remove the file if it exits.

Edit your `/etc/inittab` file to run `getty` on the serial port (substituting in the correct information for your

TOC

environment - port, speed, and default terminal type):

```
S1:23:respawn:/sbin/getty ttyS1 DT9600 vt100
```

Restart `init`:

```
linux# init q
```

At this point, you should see a login prompt on your terminal. You may have to hit return to get the terminal's attention.

mgetty

The "m" stands for modem. This program is primarily for modems and as of mid 1999 doesn't always work very well for text-terminals. It's poorly documented for terminals and you may need to wade thru much documentation for modems in order to figure out how to use it for terminals. Look at the last lines of /etc/mgetty/mgetty.config for an example of configuring it for a terminal. It will only support software (Xon/Xoff) flow control (used by many terminals) if you recompile it. This will be hopefully be fixed in the future. It would be nice to use the same getty for terminals as for modems but mgetty needs a little fixing to fill the bill.

14.2 Stty & Setserial

There is both a "stty" command and a "setserial" command for setting up the serial ports. Some (or all) of the needed stty settings can be done via getty and there may be no need to use setserial so you may not need to use either command. These two commands (stty and setserial) set up different aspects of the serial port. Stty does the most while setserial configures the low-level stuff such as interrupts and port addresses. To "save" the settings, these commands must be written in certain files (shell scripts) which run each time the computer starts up. Distributions of Linux often supply a shell script which runs `setserial` but seldom supply one which runs `stty`.

14.3 Setserial

Introduction

Don't ever use `setserial` with Laptops (PCMCIA). `setserial` is a program which allows you to tell the device driver software the I/O address of the serial port, which interrupt (IRQ) is set in the port's hardware, what type of UART you have, etc. It can also show how the driver is currently set. In addition, it can probe the hardware (if certain options are given). Besides the manual for `setserial`, check out info in `/usr/doc/setserial.../` or the like. It should tell you how setserial is handled in your distribution of Linux.

`Setserial` is often run automatically at boot-time by a start-up shell-script. It will only work if the serial module is loaded. If you should for some reason unload the serial module later on, the changes previously made by `setserial` will be forgotten. So `setserial` must be run again to reestablish them. In addition to running via a start-up script, something akin to `setserial` runs when the serial module is loaded. Thus when you watch the start-up messages on the screen it may look like it ran twice, and in fact it has.

m4_define TorS

If your serial port is Plug-and-Play you may need to consult other HOWTOs such as Plug-and-Play or Serial.

With appropriate options, `setserial` can probe (at a given I/O address) for a serial port but you must guess the I/O address. If you ask it to probe for /dev/ttyS2 for example, it will only probe at the address it thinks ttyS2 is at. If

TOC

you tell setserial that ttyS2 is at a different address, then it will probe at that address, etc. See Probing

Setserial does not set either IRQ's nor I/O addresses in the serial port hardware itself. You must tell setserial the identical values that have been set in the hardware. It's set in the hardware either by jumpers or by plug-and-play. Do not just invent some values that you think would be nice to use and tell them to setserial. However, if you know the I/O address but don't know the IRQ you may command setserial to attempt to determine the IRQ.

You can see a list of possible commands to use (but not the one-letter options such as -v for verbose --which you should normally use when troubleshooting) by just typing `setserial` with no arguments. Note that setserial calls an I/O address a "port". If you type:

```
setserial-g /dev/ttyS*
```

you'll see some info about how that device driver is configured for your ports. Add a "v" to the option "-g" to see more. But this doesn't tell you if the hardware actually has these values set in it. If fact, you can run setserial and assign a purely fictitious I/O address, any IRQ, and whatever uart type you would like to have. Then the next time you type "setserial ..." it will display these bogus values without complaint. Note that assignments made by setserial are lost when the PC is powered down so it is usually run automatically somewhere each time that Linux is booted.

Probing

In order to try to find out if you have a certain piece of serial hardware you must first know (or guess) its I/O address (or the device driver must have an I/O address for it, likely previously set by setserial). To try to detect the physical hardware use the -v (verbose) and `autoconfig` command to `setserial`. If the resulting message shows a uart type such as 16550A, then you're OK. If instead it shows "`unknown`" for the uart type, then there is likely no serial port at all at that I/O address. Some cheap serial ports don't identify themselves correctly so if you see "`unknown`" you still might have something there.

Besides auto-probing for uart type, setserial can auto-probe for IRQ's but this doesn't always work right either. In versions of setserial >= 2.15, your last probe test may be saved and put into the configuration file `/etc/serial.conf` which will be used next time you start Linux. The script that runs `setserial` at boot-time does not usually probe, but you could change it so that it does. See the next section.

Can Linux Configure The Serial Devices Automagically?

Yes, but ... You may set Linux up to detect and set up the serial devices automatically on startup (probing). It's easy to do for setserial < 2.15. Just add some lines to the file that runs setserial on start-up. See Old configuration method: edit a script For example, for ttyS3 you would add:

```
/sbin/setserial /dev/ttyS3 auto_irq skip_test autoconfig
```

to the file that runs setserial on startup. Do this for every serial port you want to auto configure. Be sure to give a device name that really does exist on your machine.

For versions >= 2.15 (provided your distribution implemented the change, Redhat didn't) it's much harder to do since the file that runs setserial on startup, /etc/init.d/setserial or the like was not intended to be edited by the user. There may be no helpful comments in it like there were in earlier versions.

Boot-time Configuration

When the kernel loads the serial module (or if the "module" is built into the kernel) then only `ttyS{0-3}` are auto-detected and the driver is set to IRQs 4 and 3 (regardless of what the hardware is actually set at). You see this

TOC

as a boot-time message just like as if `setserial` had been run.. If you use 3 or more ports, this may result in conflicts.

To fix such conflicts by telling setserial the true IRQs (or for other reasons) there may be a file somewhere that runs `setserial` again. This happens early at boot-time before any process uses the serial port. In fact, your distribution may have set things up so that the setserial program runs automatically from a start-up script at boot-time. More info about how to handle this situation should be found in /usr/doc/setserial.../ or the like.

New configuration method using /etc/serial.conf

Starting with version 2.15 (1999) of `setserial` the shell-script that runs from a file at boot-time gets its data from a configuration file: /etc/serial.conf. But serial.conf is normally never edited. Instead just use `setserial` on the command line. Normally, what you changed with the setserial command is saved to the configuration file (serial.conf) when you shutdown (normally) or reboot. This only works if "###AUTOSAVE### or the like is on the first line of serial.conf. If you should use `setserial` experimentally and it doesn't work out right, then don't forget to redo it so that the experimental settings don't get saved by mistake. The file most commonly used to run setserial at boot-time (in conformance with the configuration file) is now /etc/init.d/setserial (Debian) or /etc/init.d/serial (Redhat), or etc., but it also should not normally be edited.

To disable a port, use `setserial` to set it to "uart none". The format of /etc/serial.conf appears to be just like that of the parameters placed after "setserial" on the command line with one line for each port. If you don't use autosave, you may edit /etc/serial.conf manually. For 2.15, the Debian distribution installs the system with autosave enabled, but Redhat 6.0 just had a file /usr/doc/setserial-2.15/rc.serial which you have to move to /etc/init.d/.

BUG: As of July 1999 there is a bug/problem since with ###AUTOSAVE### only the setserial parameters displayed by "setserial -G /dev/ttyS?" (where ? = 0, 1, 2, ...) get saved but the other parameters don't get saved. This will only affect a small minority of users since the parameters not saved are seldom used anyway. It's been reported as a bug and may be fixed by now.

In order to force the current settings set by setserial to be saved to the configuration file (serial.conf) without shutting down, do what normally happens when you shutdown: Run the shell-script `/etc/init.d/{set}serial stop`. The "stop" command will save the current configuration but the serial ports still keep working OK.

Old configuration method: edit a script

Prior to 2.15 (1999), or if your distribution chose not to use /etc/serial.conf then you need to look for a file that runs setserial at boot-time. If it doesn't exist, you need to create one (or place the commands in a file that runs early at boot-time). If such a file is currently being used it's likely somewhere in the /etc directory-tree. But Redhat has supplied it in /usr/doc/setserial/ although you need to move it to the /etc tree before using it. You might use "locate" to try to find such a file. For example, you could type: locate "*serial*".

What you are looking for could be named rc.serial, or 0setserial (Debian). If such a file is supplied, it should contain a number of commented-out examples. By uncommenting some of these and/or modifying them, you should be able to set things up correctly. Make sure that you are using a valid path for `setserial`, and a valid device name. You could do a test by executing this file manually (just type its name as the super-user) to see if it works right. Testing like this is a lot faster than doing repeated reboots to get it right. Of course you can also test a single `setserial` command by just typing it on the command line.

The script `/etc/rc.d/rc.serial` was commonly used in the past. The Debian distribution used `/etc/rc.boot/0setserial`. Another file once used is `/etc/rc.d/rc.local` but it's not a good idea to use this since it may not be run early enough. It's been reported that other processes may try to open the serial port before rc.local runs resulting in serial communication failure.

TOC

IRQs

By default, both ttyS0 and ttyS2 share IRQ 4, while ttyS0 and ttyS3 share IRQ 3. But sharing serial interrupts is not permitted unless you: 1. have kernel 2.2 or better, and 2. you've complied in support for this, and 3. your serial hardware supports it. See Serial-HOWTO: Interrupt sharing and Kernels 2.2+.

If you only have two serial ports, ttyS0 and ttyS1, you're still OK since IRQ sharing conflicts don't exist for non-existent devices.

If you add an internal modem and retain ttyS0 and ttyS1, then you should attempt to find an unused IRQ and set it both on your serial port (or modem card) and then use setserial to assign it to your device driver. If IRQ 5 is not being used for a sound card, this may be one you can use for a modem. To set the IRQ in hardware you may need to use isapnp, a PnP BIOS, or patch Linux to make it PnP. To help you determine which spare IRQ's you might have, type "man setserial" and search for say: "IRQ 11".

----------From TT-HOWTO----------

Setserial tells the serial port driver both the interrupt (IRQ) number and the port IO address. It can tell you what type of UART chips you have if you use the autoconfig parameter. For the serial port to work, the serial module (for the Linux kernel) must be loaded. If you use more than 2 serial ports and want to assign unique IRQ numbers (or use more than 4 serial ports), then you need to use setserial. If your serial port is plug-and-play you should look in the Serial-HOWTO. In addition to the man page for it, but sure to look at /usr/doc/setserial/ if it exists. It may tell you how setserial is run automatically at boot time and how to alter the configuration. Don't ever use setserial with Laptops (PCMCIA bus).

starting with version 2.15 (1999) of `setserial` changes made to setserial on the command line are sometimes automatically saved so that they will remain after you shut off the computer.

Where to Run Setserial ?

Starting with version 2.15 of `setserial` the method of configuration changed. Before 2.15 you configured it by editing a shell-script that ran `setserial`. After 2.15 there's a configuration file /etc/serial.conf which is automatically edited to conform with whatever you do with the `setserial` command.

You may find you have both the old and new way of configuring installed. If the shell-script is named "...pre-2.15" or the like, it's probably not being used anymore so everything should be OK if you keep it.

Prior to version 2.15

Prior to 2.15 there was no configuration file for `setserial`. One needs to edit the shell-script that runs setserial at boot-time. You need to find it or create one if it doesn't exist. Where it is depends on your distribution. It might be in a file named "rc.serial" . In the Debian distribution it was the 0setserial file in /etc/rc.boot. One way to try to find it is to search for it with the "locate" command.

Version 2.15 and later

You neither edit the file that runs setserial nor the configuration file. Changes made by using `setserial` on the command line are *sometimes* saved in /etc/serial.conf during normal shutdown. This feature is called autosave. It only works if the first line of serial.conf contains ###AUTOSAVE### or the like.

There's a bug I've reported in 2.15 and it's been fixed (but you could still have it): With the bug, seldom used changes made by setserial don't get saved. Only the most important parameters that you see if you give the

command "setserial -g /dev/ttyS?" do get saved.

If autosave is enabled, then each time you boot, setserial is run by /etc/init.d/setserial (or "serial", etc.) using the saved configuration.

14.4 Stty

Introduction

`stty` does much of the configuration of the serial port but since application programs (and the getty program) often handle it, you may not need to use it much. It's handy if your having problems or want to see how the port is set up. Try typing ``stty -a'' at your terminal/console to see how it's now set. Also try typing it without the -a (all) for a short listing which shows how it's set different than normal. Don't try to learn all the setting unless you want to become a serial guru. Most of the defaults should work OK and some of the settings are needed only for certain obsolete dumb terminals made in the 1970's. Dumb terminals made after about 1980 don't use them.

Whereas `setserial` only deals with actual serial ports, stty is used both for serial ports and for virtual terminals such as the standard Linux text interface at a PC monitor. For the PC monitor, many of the stty settings are meaningless. Changing the baud rate, etc. doesn't appear to actually do anything.

Here are some of the items stty configures: speed (bits/sec), parity, bits/byte, # of stop bits, strip 8th bit?, modem control signals, flow control, break signal, end-of-line markers, change case, padding, beep if buffer overrun?, echo, allow background tasks to write to terminal?, define special (control) characters (such as what key to press for interrupt). See the `stty` man or info page for more details. Also see the man page: `termios` which covers the same options set by stty but (as of mid 1999) covers features which the stty man page fails to mention. For use of some special characters see Special (Control) Characters

With some implementations of getty (getty_ps package), the commands that one would normally give to stty are typed into a getty configuration file: /etc/gettydefs. Even without this configuration file, the getty command line may be sufficient to set things up so that you don't need stty.''')

One may write C programs which change the stty configuration, etc. Looking at some of the documentation for this may help one better understand the use of the stty command (and its many possible arguments). Serial-Programming-HOWTO is useful. The manual page: termios contains a description of the C-language structure (of type termios) which stores the stty configuration in computer memory. Many of the flag names in this C-structure are almost the same (and do the same thing) as the arguments to the stty command.

Using stty for another port

Using `stty` to inspect or configure another terminal or serial port is tricky. For example, let's say you are at the PC monitor (tty1) and want to use `stty` for the serial port ttyS2. You need to use the redirection operator <. First, be warned that if there is a terminal on ttyS2 and a shell running on that terminal, then what you see will likely be deceptive and trying to set it will not work. See Two Interfaces at a Terminal to understand it.

Type ``stty -a < /dev/ttyS2'' to look at the settings of ttyS2. Use the same redirection operator < to set ttyS2. Some people try to use ``stty ... > /dev/ttyS2 to set the terminal. This will not do it. Instead, it takes the message normal displayed by the stty command for the terminal you are on (tty1) and sends this message to ttyS2 but it doesn't change any settings for ttyS2.

Versions starting with 1.17 (still not released as of mid 1999) will no longer need to use redirection (<) but instead will use ``stty ... -F /dev/ttyS2'' (or --file instead of F) etc.

TOC

Two interfaces at a terminal

When using a shell (such as bash) with command-line-editing enabled there are two different terminal interfaces (what you see when you type stty -a). When you type at the command line you have a temporary "raw" interface (or raw mode) where each character is read by the command-line-editor as you type it. Once you hit the <return> key, the command-line-editor is exited and the terminal interface is changed to the nominal "cooked" interface (cooked mode) for the terminal. This "cooked" mode lasts until the next prompt is sent to the terminal. Note that one never types anything to this "cooked" mode but it does have a purpose.

When a prompt is sent to the terminal the terminal goes from "cooked" to "raw" mode (just like it does when you start an editor since you are starting the command-line editor). The settings for the "raw" mode are based only on the basic settings taken from the "cooked" mode. Raw mode keeps these setting but changes several other settings in order to change the mode to "raw". It is not at all based on the settings used in the previous "raw" mode. Thus if one uses stty to change settings for the raw mode, such settings will be lost as soon as one hits the <return> key at the terminal that has supposedly been "set".

Now when one types stty to look at the terminal interface, one may either get a view of the cooked mode or the raw mode. You need to figure out which one you're looking at. It you use stty from another terminal to deal with a terminal that is displaying a command line, then the view is that of the raw mode. Any changes made will only be made to the raw mode and will be lost when someone presses <return> at the terminal you tried to "set". But if you type a stty command at your terminal (without using < for redirection) and then hit <return> it's a different story. The <return> puts the terminal in cooked mode. Your changes are saved and will still be there when the terminal goes back into raw mode (unless of course it's a setting not allowed in raw mode).

This situation can create problems. For example, suppose you corrupt your terminal interface and to restore it you go to another terminal and "stty sane <dev/ttyS1" to restore it. It will not work! Of course you can try to type "stty sane ..." at the terminal that is corrupted but you can't see what you typed. All the above not only applies to dumb terminals but to virtual terminals used on a PC Monitor as well as to the terminal windows in X. In other words, it applies to almost everyone who uses Linux. Luckily, a file that runs stty at boot-time will likely deal with a terminal (or serial port with no terminal) that has no shell running on it so there's no problem.

Where to put the stty command ?

Should you need to have `stty` set up the serial interface each time the computer starts up then you need to put the `stty` command in a file that will be executed each time the computer is started up (Linux boots). It should be run before the serial port is used (including running getty on the port). There are many possible places to put it. If it gets put in more than one place and you only know about (or remember) one of those places, then a conflict is likely. So make sure to document what you do.

One good place to put it would be in the same file that runs setserial when the system is booted. See Where to Run Setserial?. It would seem best to put it after the setserial command so that the low level stuff is done first.

14.5 Terminfo & Termcap (brief)

See Terminfo and Termcap (detailed) for a more detailed discussion of termcap. Many application programs that you run use the terminfo (formerly termcap) data base. This has an entry (or file) for each model or type (such as vt100) of terminal and tells what the terminal can do, what codes to send for various actions, and what codes to send to the terminal to initialize it.

Since many terminals (and PC's also) can emulate other terminals and have various "modes" of operation, there may be several terminfo entries from which to choose for a given physical terminal. They usually will have similar

TOC

names. The last parameter of getty (for both agetty and getty_ps) should be the terminfo name of the terminal (or terminal emulation) that you are using (such as vt100).

The terminfo does more than just specify what the terminal is capable of doing and disclose what codes to send to the terminal to get it to do those things. It also specifies what "bold" will look like (will it be reverse video or will it be high intensity, etc.), what the cursor will look like, if the letters will be black, white, or some other color, etc. In PC terminology these are called "preferences". It also specifies initialization codes to send to the terminal (analogous to the init strings sent to modems). Such strings are not automatically sent to the terminal by Linux. See Init String. If you don't like the way the display on the screen looks and behaves you may need to edit (and then update) the terminfo (or termcap) file. See Terminfo Compiler (tic) for how to update.

14.6 Setting TERM and TERMINFO

These are two environment variables for terminals: TERM and TERMINFO, but you may not need to do anything about them. TERM must always be set to the name of the terminal you are using. If you don't know the name see What is the terminfo name of my terminal ?. TERMINFO contains the path to the terminfo data base, but may not be needed if the database is in a default location (or TERMINFO could be set automatically by a file that comes with your distribution of Linux). You may want to look at Where is the compiled database located ?.

Fortunately, the getty program usually sets TERM for you just before login. This permits application programs to find the name of your terminal and then look up the terminal capabilities in the terminfo data base. See TERM Variable for more details on TERM.

If your terminfo data base can't be found you may see an error message about it on your terminal. If this happens it's time to check out where terminfo resides and set TERMINFO if needed. You may find out where the terminfo database is by searching for a common terminfo file such as "vt100" using the "locate" command. Make sure that your terminal is in this database. An example of setting TERMINFO is: export TERMINFO=/usr/share/terminfo (put this in /etc/profile or the like). If the data for your terminal in this data base is not to your liking, you may need to edit it. See Terminfo & Termcap (brief).

What is the terminfo name of my terminal ?

You need the exact name in order to set the TERM environment variable or to give to `getty`. The same name should be used by both the termcap and terminfo databases so you only need to find it once. A terminal usually has alias names but if more than one name is shown, use the first one.

To find it, try looking at the /etc/termcap... file (if you have it). If not, then either look at the terminfo trees (see Where is the compiled database located ?) or try to find the terminfo source code file (see Source code database locations.

14.7 Rarely Needed /etc/ttytype File

The configuration file /etc/ttytype is used to map /dev/ttySn's to terminal names per terminfo. tset uses it, but if the TERM environment variable is already set correctly, then this file is not needed. Since the Linux getty sets TERM for each tty, you don't need this file. In other Unix-like systems such as FreeBSD, the file /etc/ttys maps ttys to much more, such as the appropriate getty command, and the category of connection (such as "dialup"). An example line of Linux ttytype: vt220 ttyS1

TOC

14.8 Login Restrictions

By default, the root user may not login from a terminal. To permit this you must create (or edit) the file /etc/securetty per the manual page "securetty". To restrict logins of certain users and/or certain terminals, etc. edit /etc/login.access (this replaces the old /etc/usertty file ??). /etc/login.def determines if /etc/securetty is to be used and could be edited so as to make /etc/securetty not needed (or not used). /etc/porttime restricts the times at which certain ttys and users may use the computer. If there are too many failed login attempt by a user, that user may be prohibited from ever logging in again. See the man page "faillog" for how to control this.

14.9 Run Command Only If TERM=my_term_type

Sometimes there are commands that one wants to execute at start-up only for a certain type of terminal. To do this for the stty command is no problem since one uses the redirection operator < to specify which terminal the command is for. But what about shell aliases or functions? You may want to make a function for the ls command so it will color-code the listing of directories only on color terminals or consoles. For monochrome terminals you want the same function name (but a different function body) which will use symbols as a substitute for color-coding. Where to put such function definitions that are to be different for different terminals?

You may put them inside an "if" statement in /etc/profile which runs at startup each time one logs on. The conditional "if" statement defines certain functions, etc. only if the terminal is of a specified type.

Example for ls Function

While much of what this if statement does could be done in the configuration file for dircolors, here's an example for the case of the bash shell:

```
if [ $TERM = linux ]; then
    eval `dircolors`;
elif [ $TERM = vt220 ]; then
    ls () { command ls -F $* ; }# to export the function ls():
    declare -xf ls
else echo "From /etc/profile: Unknown terminal type $TERM"
fi
```

---

15. Terminfo and Termcap (detailed)

15.1 Intro to Terminfo

Terminfo (formerly Termcap) is a database of terminal capabilities and more. For every (well almost) model of terminal it tells application programs what the terminal is capable of doing. It tells what escape sequences (or control characters) to send to the terminal in order to do things such as move the cursor to a new location, erase part of the screen, scroll the screen, change modes, change appearance (colors, brightness, blinking, underlining, reverse video etc.). After about 1980, many terminals supported over a hundred different commands (some of which take numeric parameters).

The terminfo abbreviations are usually longer than those of termcap and thus it's easier to guess what they mean. The manual pages for terminfo are more detailed (and include the old termcap abbreviations). Thus, unless you are already committed to working with termcap, it's suggested you use terminfo.

15.2 Terminfo Database

Introduction

The terminfo database is compiled and thus has a source part and a compiled part. The old termcap database has only a source part but this source can, by a single command, be both converted to terminfo source and then compiled. Thus you may get by without having any terminfo source since the termcap source can create the compiled terminfo database. To see a display of the database for the terminal you're now using (including a PC monitor) type "infocmp" and you should see the source terminfo "file" for it.

To see if your terminal (say vt100) is in the terminfo data base type "locate vt100". If you don't know what your terminal name is, explore the listing of files in the compiled database or see What is the terminfo name of my terminal ?

Where is the compiled database located ?

Compiled code database locations

Typing "locate vt100" may show /usr/lib/terminfo/v/vt100, /usr/share/terminfo/v/vt100, /home/you/.terminfo/v/vt100, and/or /etc/terminfo/v/vt100. All these are possible locations of the compiled terminfo files. Only older programs may expect it to be in the /etc/terminfo directory. Unless this terminfo data base happens to be located where every program that needs it expects to find it, then the environment variable TERMINFO should be set to the path to this database. Example: TERMINFO=/usr/share/terminfo

If the compiled terminfo is in more than one location, everything is usually OK until someone gets a new

terminfo file(s) (from a newer distribution, from the net, by editing the old one, etc.) The newer term needs to be put in all the existing locations (or redundant locations need to be abolished). If you don't ensure this gets done, then some application programs may wind up still finding and using the old (and buggy) terminfo data that sill exists in a "usual" location.

Source code database locations

The source code you use may reside in /etc/termcap and/or in terminfo.src (or another name). See the man pages: terminfo(5) or termcap(5) for the format required to create (or modify) these source files. The file terminfo.src may be in various locations on your computer or it may not be included with your linux distribution. Use the locate command to try to find it. It is available for downloading (under the name termtypes.ti) from http://sagan.earthspace.net/terminfo

Terminfo Compiler (tic)

The data in the source files is compiled with the "tic" program which is capable of converting between termcap format and terminfo format. Thus you can create a compiled terminfo data base from termcap source. The installation program which was used to install Linux probably installed the compiled files on your hard disk so you don't need to compile anything unless you modify /etc/termcap (or terminfo.src). "tic" will automatically install the resulting compiled files into a terminfo directory ready to be used by application programs.

Look at Your Terminfo

It's a good idea to take a look at the terminfo entry for the terminal you are using (source code of course) and read the comments. A quick way to inspect it without comments is to just type "infocmp". But the comments may tell you something special about the terminal such as how you need to set it up so that it will work correctly with the terminfo database.

Deleting Data Not Needed

In order to save disk space, one may delete all of the database except for the terminals types that you have (or might need in the future). Don't delete any of the termcaps for a "Linux terminal" (the console) or the xterm ones if you use X-Windows. The terminal type "dumb" may be needed when an application program can't figure out what type of terminal you are using. It would save disk space if install programs only installed the terminfo for the terminals that you have and if you could get a termcap for a newly installed terminal over the Internet in a few seconds.

15.3 Bugs in Existing Terminfo Files (and Hardware)

Unfortunately, there are a number of bugs in the terminfo and termcap files. In addition, many of these definitions are incomplete and do not define certain features available on the terminals. Sometimes you can get by without modifying the terminfo but in other cases you need to modify it or possibly use

another emulation that has a good terminfo.

The sad state of the supplied terminfo files is due to a number of reasons. One is that during the 1980's when many of them were written (often in termcap format), application programs did not utilize more advanced terminal features. Thus if such feature were not in the termcap (or terminfo) file, no one complained. Today, programs such as vim use "context highlighting" and minicom uses the terminal's graphics character set. These often need more definitions to be added to the old termcap. This may (or may not) have already been done.

Most terminals had hardware bugs (in their firmware) and sometimes they were "fixed" by modifying the termcap. Then the manufacturer might send out replacement chips which would fix the bug. Not all owners would bother to get the replacement chips. Thus there may be 2 or more terminfos for your terminal, depending on what firmware chips it has in it. This situation was often not noted in the termcap and only one termcap may be supplied with Linux. Some hardware bugs which existed for features that were almost never used in the past likely never did get fixed. Also, some reported hardware bugs may never have been fixed since they were not of much significance at the time or because the company went out of business, etc.

15.4 Modifying Terminfo Files

To do this you need a manual for your terminal showing what escape sequences it uses. Newer manuals from the 1990's often don't show this. You also need a terminfo manual (or the like). For example, in order to add graphic capability you must assign values to the terminfo variables: enacs, rmacs, and smacs by editing a source file. Then by using "tic" you may compile it. "tic" should automatically put the compiled terminfo file in the correct directory reserved for it.

If you would like to find a better terminfo entry for a certain terminal than the one supplied, you might try searching the Internet (but what you find may be worse). If your new terminfo entry is better than the old one and it seems stable (you've used it for a while with no problems) then you should send a copy to the maintainer of terminfo as noted at the start of the source file for terminfo (or termcap).

15.5 Init String

Included in the terminfo are often a couple of initialization strings which may be sent to the terminal to initialize it. This may change the appearance of the screen, change what mode the terminal is in, and/or make the terminal emulate another terminal. An initialization string is not automatically sent to the terminal to initialize it. One might expect that the getty program should do this but if it did, one could make a change to the set-up at the terminal and this change wouldn't be implemented because the init string would automatically cancel it. You must use a command given on the command line (or in a shell script) to send the init string such. Such commands are: "tset", "tput init", or "setterm -initialize". Sometimes there is no need to send the init string since the terminal may set itself up correctly when it is powered on (using options/preferences one has set up and saved in non-volatile memory of the terminal).

15.6 TERM Variable

The Environment variable TERM should be set to the name of terminal which you are using. If TERM hasn't been set yet and you don't know the name of your terminal see What is the terminfo name of my terminal ?. It is normally set by the terminal_type parameter passed to the getty program (look at it in the /etc/inittab file). This name must be in the Terminfo data base. Just type "set" at the command line to see what TERM is set to (or type: tset -q). At a console (monitor) TERM is set to "linux" which is the PC monitor emulating a fictitious terminal model named "linux". Since "linux" is close to a vt100 terminal and many text terminals are also, the "linux" designation will sometimes work as a temporary expedient with a text terminal.

If more than one type of terminal may be connected to the same port (/dev/tty...) (for example, if there is a switch to permit different terminal types to use the same serial port, or if the port is connected to a modem to which people call in from different types of terminals) then TERM needs to be set each time someone connects to the serial port. There is often a query escape sequence so that the computer may ask the terminal what type it is. Another way is to ask the user to type in (or select) the type of terminal s/he is using. You may need to use tset for this or write a short shell script to handle this.

One way to do this is to use "tset" (see the manual page). tset tries to determine the terminal name of the terminal you are using. Then it looks up the data in terminfo and sends your terminal an init string. It can also set the value of TERM. For example, a user dials in and logs in. The .profile login script is executed which contains within it the following statement: eval `tset -s ?vt100`. This results in: The user is asked if s/he is using a vt100. The user either responds yes or types in the actual terminal type s/he is using. Then tset sends the init string and sets TERM to this terminal name (type).

15.7 Terminfo/Termcap Documents

-
- manual pages for terminfo(5) (best) and/or termcap(5). The Termcap Manual (2nd ed.) by Richard M. Stallman is a GNU manual which is somewhat obsolete. Although dated 1992, it fails to even mention terminfo.
- the files: terminfo.src and /etc/termcap have info about various versions of termcap files, naming conventions for terminals, and special capabilities code named u6-u9. If you don't have one, go to http://sagan.earthspace.net/terminfo
- "Termcap and Terminfo" is a book published by O'Reilly in 1988.

---

16. Using the Terminal

16.1 Intro to Using Terminal

This section is about controlling the terminal-computer interface and/or changing the terminal set-up while using the terminal. It explains (or points to explanations of) how the user of a terminal can control and inspect the interface and how to use various commands provided by the device driver. It does not explain how to use the many application programs, shells or most Linux utilities. Two commands commonly used at the terminal are:

- clear (to clear the screen)
- reset (to reset the terminal)

16.2 Starting Up the Terminal

Of course the power must be on for the terminal to work. If you don't see a login prompt hit the "return" (or "enter") key a few times. Then type your account name (followed by a return/enter) and your password when prompted for it (also followed by return/enter). Make sure not to type all capital letters. If you do, the computer may think that you have an old terminal that can't send lowercase letters and the serial driver may set itself up to send only capital letters to the terminal.

If nothing happens, make sure that both the host computer and the terminal are OK. If the host computer is shut down (no power) what you type at the terminal keyboard may appear on the screen since the transmit and receive pins at the computer may be connected together resulting in echoing of characters by an "off" computer. If you can't log in when the host computer is running, see Trouble-Shooting.

16.3 Terminal (Serial) Device Driver

When typing at the command line, the shell (such as the Bash shell) is reading what you type and reacting to it. What you type first passes thru the terminal driver part of your operating system. This driver may translate certain characters (such as changing the "return" character generated by the "return" key into a "new-line" character for Linux files). It also recognizes certain control codes which you may type at the keyboard such as ^C to interrupt the execution of a program. It also normally echoes what you type back to the display. Stty may be used to configure how this terminal driver behaves, including disabling some (or all) of its functionality.

16.4 Problems with Editors

There may be some problems with using both emacs and vi on some terminals.

emacs and ^Q

If software flow control exists, then the ^Q command in emacs will freeze the display. The fix is to map this to another key-press by configuring emacs that way.

vi and Cursor-Keys

Vi uses the esc-key as a command to exit insert mode. If one hits an arrow-key (cursor-key) an escape sequence (starting with the ESC character) is sent to the host. Vi must distinguish between these two meanings of ESC. A smart vi (such as vim) should be able to detect the difference by noting the key that follows the esc-key.

On VT terminals the left-arrow-key may be either set to send ESC [D or ESC O D. The other arrow keys are similar but use A, B, and C instead of D. If you're having problems, choose ESC [D since the "O" in the other alternative could be interpreted by vi as a command to "Open a line". The "[" should be interpreted by vi to mean that an arrow-key has been pressed. ESC [D will be sent provided "Cursor Key Application Mode" has not been set. ESC [D is normally the default so everything is seemingly OK. Except that many termcaps contain a string (not the init string) which sets what you want to avoid: "Application Mode". Editors may send this string to the terminal when the editor starts up. Now you are in trouble.

This string has the termcap code "ks" (smkx in terminfo) meaning enable the function (and related) keys (including the arrow keys). An application enables these keys by sending the "ks" string to the terminal. Whoever wrote the termcap reasoned that if an application wants to enable these keys, then they should be put into "Application Mode" since this is an "application", but you don't want this.

The Linux console has no "ks" string so you can't fall into this trap at the console. For other terminals you may need to edit the termcap (or terminfo) or use another termcap entry. You need to change not only the "ks" string but also the termcap definitions of what they send: kd, kl, kr, ku. Then run tic to install it.

For vim (vi iMproved) there is a way to set it up to work OK with ESC O D (so you don't need to edit termcap): See vim help for "vt100-cursor-keys". You may run "gitkeys" and then press your cursor keys to see what they send but they may be set to send something different when you're in an editor.

16.5 Color ls Corruption

If `ls` is corrupting your terminal emulation with the color feature, turn it off. `ls --color`, and `ls --colour` all use the color feature. Some installations have `ls` set to use color by default. Check `/etc/profile`, etc. for `ls` aliases. See Example for ls Function for how to have `ls` do color for the console and do monochrome for terminals.

16.6 Display Freezes (hung terminal)

TOC

The symptom of a hung terminal is where what you type doesn't display on the terminal (or in some cases displays but doesn't do anything). If what you type is invisible (or does nothing) type ^Q to restart flow (if flow control stopped it). Hanging may also be due to:
Sent Terminal Binary or Abnormally Exited a Program
If you didn't do any of these two, then your program could by buggy or you interaction with it fatally illegal.

If you want to quit the program you were running and you can't do it by the usual methods (some programs have special keys you must hit to exit) then try killing it from another terminal using "top" or "kill". If the process refuses to die, you may try sending it signal 9 from top which should force it to exit. The "9" type of forced exit may leave some temporary files lying around as well as a corrupted interface. Killing the login shell should result in a startup of getty with a new login prompt.

People new to Linux may unintentionally press Ctrl-S (^S) (or the "No Scroll" key) which mysteriously freezes the screen (although that is what this key is supposed to do if you use software flow-control). To restore normal screen interaction, press Ctrl-Q (^Q). Note that everything typed during the "freeze" gets executed but you don't see any report of this until you hit ^Q. Thus when it's frozen, don't type anything drastic that might destroy files, etc. One argument for using hardware flow-control is to prevent such freezes.

16.7 Corrupted Terminal Interface

This includes the case of a "frozen display" = "hung terminal" of the previous section.

Symptoms

When the display doesn't look right, or when what you type doesn't display correctly (if at all), or nothing happens when you type a command, you may have a corrupted terminal interface. In rare cases when the serial port hardware gets itself corrupted, the only fix may be to cycle power (turn off the PC and reboot). The corruption may be due to things such as bug in the program you're using, a hardware failure (including an obscure hardware defect that you can normally live with), or possibly an incorrect configuration. If everything was working normally but it suddenly goes bad, it may be that the interface got corrupted by something you did. Three mistakes you might have made to corrupt the interface are:

- Sent Terminal Binary
- Abnormally Exited a Program
- Typed ctrl-S by mistake

Sent Terminal Binary Characters

Your terminal will change its characteristics if sent certain escape sequences or control characters. It you inadvertently try to display a binary file, it might by chance contain such sequences which may put your terminal into some strange mode of operation or even make it unusable. Always view or edit a binary file

TOC

with programs designed for that purpose so that this doesn't happen. Most editors and pagers will ha binary OK so as not to corrupt the interface. Some may display a message telling you that they can't binary. But using "cat ...." or "cp .... /dev/tty.." where .... is a binary file, will send the binary to the terminal and likely corrupt things.

Corruption it can also happen when using a communications program where a remote computer may send binary to your screen. There are numerous other ways it can happen so be prepared for it. Even a supposed ASCII file may contain unwanted control codes.

To fix this problem reset the terminal. You may try typing either "reset" or "setterm -reset" (although you may not be able to see what you're typing). This will send the reset string from the terminfo entry to the terminal. If the correct set-up has been saved inside the terminal, then pressing a special key(s) (perhaps in setup mode) may restore this setting. Then you might still need to use "reset" to send the init string if you use it to set up your terminal.

Abnormally Exited a Program

Large application programs (such as editors) often use the stty command (or the like) in their code to temporarily change the stty configuration when you are running the program. This may put the device driver into "raw" mode so that every character you type goes directly thru to the application program. Echoing by the driver is disabled so that everything you see on the screen comes directly from the application program. Thus many control commands (such as ^C) may not work within such applications.

When you tell such an application to quit, the application program first restores the stty settings to what they were before the application program started. If you abnormally exit the program (you may guess this has happened when what you type no longer displays on the screen) then you may still be in "raw mode" on the command line.

To get out of raw mode and restore the normal stty settings type "stty sane". However, you must type this just after a "return" and end it with a "return". But hitting the "return" key doesn't do the job since the "return" code no longer gets translated to the new-line characters that the shell is waiting for. So just type new-line (^J) instead of "return". The "sane" terminal interface may not be exactly the same as the normal one but it usually works. "stty sane" may also be useful to get out of a corrupted interface due to other causes.

16.8 Special (Control) Characters

A number of control characters which you may type at the keyboard are "caught" by the terminal driver and perform various tasks. To see these control commands type: stty -a and look at lines 2-4. They are tersely explained in the stty manual pages. They may be changed to different control characters or disabled using the stty command. Thus your control characters might be different than those described below. They are used for command-line editing, interrupting, scrolling, and to pass the next character thru transparently.

TOC

Command-Line Editing

While the terminal driver has a few commands for command-line editing, some shells have a built-in real editor (such as "readline" in the Bash shell). Such an editor is normally on by default so you don't need to do anything to enable it. If it's available you don't need to learn many of the following commands although they often still work along with the command-line editor. The most important to learn are ^C (interrupt), ^D, and how to stop scrolling.

- Delete-key (shown by stty as ^?) erases the last character
- ^U kills (deletes) the line
- ^W deletes a word backwards
- ^R reprints the line. Useful mainly on hard copy terminals ??

Interrupting (& Quit, Suspend, EOF, Flush)

- ^C interrupts. Exits the program and returns you to the command-line prompt.
- ^/ quits. Same as interrupt ^C but weaker. Also dumps a "core" file (which you likely have no use for) into your working directory.
- ^Z suspends. Stops the program and puts it in the background. Type fg to restart it.
- ^D end of file. If typed at the command-line prompt, exits the shell and goes to where you were before the shell started.
- ^O flush. Not implemented in Linux. Sends output to /dev/null.

Stop/Start Scrolling

If what you want to see scrolls off the bottom of the screen, you may prevent this by sending a "stop" signal (^S or Xoff) to the host (provided Xon-Xoff Flow Control is enabled). Send a "start signal to resume (^Q or Xon). Some terminals have a "No Scroll" key which will alternately send Xoff and Xon or possibly send the hardware flow control signals ?? Here's what ctrl-S (^S) and ctrl-Q (^Q) do:

- ^S stops scrolling (Xoff)
- ^Q resume scrolling (Xon)

If you want to both stop scrolling and quit, use ^C. If you want to stop scrolling to do something else but want to keep the program that was generating the output in memory so you can resume scrolling later, use ^Z suspend.

An alternative scrolling method is to pipe the output thru a pager such as more, less, or most. However, the output might not be standard output but could be error output which the pager doesn't recognize. To fix this you may need to use redirection "2>&1" to get the pager to work OK. It is often simpler to just use ^S and ^Q unless you need to scroll backwards.

At a PC console (emulating a terminal) you may scroll backwards by using Shift-PageUp. This is frequently needed since the scrolling is often too fast to stop using ^S. Once you've scrolled backwards Shift-PageDown will scroll forward again.

Take Next Character Literally

^V sends the next character typed (usually a control character) directly thru the device driver with no action or interpretation. Echoed back are two ASCII characters such as ^C.

16.9 Viewing Latin-1 Files on a 7-bit Terminal

Some "text" files are 8-bit Latin1 (see Character-Sets). If you have a terminal that will not display Latin1 (or don't have the Latin1 character set selected), then a bullet symbol will display as a 7, etc. When viewing manual pages (they are Latin1) you may give the option -7 to man so as to translate the 7's, etc. to something close to a bullet (in ASCII). Are there some pagers that make these translations ??

16.10 Inspecting the Interface

These utility programs will provide information about the terminal interface:

- gitkeys: shows what byte(s) each key sends to the host.
- tty: shows what tty port you are connected to.
- set: shows the value of TERM (the terminfo entry name)
- stty -a: shows all stty settings.
- setserial -g /dev/tty?? (you fill in ??) shows UART type, port address and IRQ number.
- infocmp: shows the current terminfo entry (less comments)

16.11 Changing the Terminal Settings

The terminal settings are normally set once when the terminal is installed using the setup procedures in the terminal manual. However, some settings may be changed when the terminal is in use. You normally would not give any "stty" of "setserial" commands when the terminal is in use as they are likely to corrupt the terminal interface. However, there are changes you may make to the appearance of the terminal screen or to its behavior without destroying the integrity of the interface. Sometimes these changes are made automatically by application programs so you may not need to deal with them.

One direct method of making such changes is to use the setup key (or the like) at the terminal and then use menus (or the like) to make the changes. To do this you may need to be familiar with the terminal. The other 3 methods send an escape sequence from the computer to the terminal to make the changes. These 3 examples show different methods of doing this to set reverse video:

1. setterm -reverse
2. tput -rev
3. echo ^[[7m

TOC

setterm

This is the easiest command to use. It uses long options (but doesn't use the normal -- before them). It consults the terminfo database to determine what code to send. You may change the color, brightness, linewrap, keyboard repeat, cursor appearance, etc.

tput

The "tput" command is similar to "setterm" but instead of using ordinary words as arguments, you must use the abbreviations used by terminfo. Many of the abbreviations are quite terse and hard to remember.

echo

In the example "echo ^[[7m" to set reverse video, the ^[is the escape character. To type it type ^V^[(or ^V followed by the escape key). To use this "echo" method you must find out what code to use from a terminal manual or from terminfo or termcap. It's simpler to use setterm or tput if you are typing on the command line. Since "echo ..." will execute faster (since it doesn't do any lookups) it's good for using in shell scripts which run at start-up, etc.

Saving Changes

When you turn off the terminal the changes you made will be lost (unless you saved them in non-volatile terminal memory by going into set-up mode and saving it). If you want to use them again without having to retype them, put the commands in a shell script or make it a shell function. Then run it when you want to make the changes. One way to make the changes semi-permanent is to put the commands in a file that runs each time you login or start up the computer.

16.12 Make a Terminal the Console

This is also called a "serial console". Many messages from the system are normally only sent to the console. Some of the messages sent to the console at boot-time may also be seen on a terminal after the boot succeeds by typing the command: dmesg. If the boot fails this will not be of any use. It's possible to modify the Linux kernel so as to make a terminal serve as the console and receive all the messages from Linux intended for the console. Unfortunately, the messages from the BIOS will be lost as they will not show on the terminal. The "console" is now located on a serial port and thus called a "serial console". Of course it will not have the colors, fonts, graphical capabilities, and keyboard remapping possibilities of the real PC console.

Prior to kernel 2.2 you had to manually patch the kernel. Starting with kernel 2.2, support is built into the kernel provided the kernel has been compiled with a "make config" option to do this. These two cases are covered in the next two sub-sections:

For Kernels 2.2 or higher

The instructions for creating a serial-console are included with source code documentation in the file: serial-console.txt. Normally, the device /dev/console is linked to tty0 (the PC console). For a serial-console you create a new /dev/console which is a true device (and not linked to something else). You must also put a statement regarding the serial-console into /etc/lilo.conf and then run lilo. This is because the equivalent of "setserial" needs to be run to set up your serial-console before the kernel is loaded. See the above mentioned documentation for details.

For Kernels before 2.2

The Linux Journal in April 1997 had an article on patching the Linux kernel. You add a couple of #defines at the start of src/linux/drivers/char/console.c:

```
#define CONFIG_SERIAL_ECHO
#define SERIAL_ECHO_PORT 0x2f8  /* Serial port address  */

The following was not in the Linux Journal article.
In kernel 2.+ (and earlier ??) you need to also set the baud
rate (unless 9600 is OK).  Find these 2 lines:

serial_echo_outb(0x00, UART_DLM); /* 9600 baud */
serial_echo_outb(0x0c, UART_DLL);

Change 0x0c in the line above (depending on the baud rate you want):

115200 baud: 0x01       19200 baud: 0x06        2400 baud: 0x30
 57600 baud: 0x02        9600 baud: 0x0c        1200 baud: 0x60
 38400 baud: 0x03        4800 baud: 0x18
```

If you currently use the console to select which operating system to boot (using LILO), but would like to do this from a terminal, then you need to add a line to the /etc/lilo.conf file. See the manual page for lilo.conf and search for "serial=".

Can I Run Linux without a Monitor (PC Console) ?

Yes, you use a terminal and make it behave like the console per above. You will likely still need a video card since most BIOSs require one to get the PC started. Your BIOS may also require a keyboard to get started or it may have an option where you can set the BIOS not to require a keyboard.

16.13 Multiple Sessions

The "`screen`" package runs multiple sessions something like virtual terminals on the console: See The Console: /dev/tty?. However, this is not like "pages" (Pages) since the image of the pages are stored in the host computer and not inside the terminal as they are with "pages".

TOC

16.14 Logging Out

To log out type either "logout" or "exit". Under some circumstances your request will be refused, but you should be told why. One reason for refusal is if you are not in the same shell into which you logged into. Another way to log out is to press ^D. Since ^D is also used for other purposes, you may not want it to log you out. If you set IGNOREEOF in the Bash shell then ^D will no longer log you out.

16.15 Chatting between Terminals, Spying

If two persons logged into terminals on the same host computer want to chat with each other they may use the "write" or the "talk" program. On the internet, chatting may be done using the "lynx" browser.

For spying on what someone else is doing at their terminal use the "ttysnoop" program. In "ttysnoop" the two terminals have the same status and anything typed on either keyboard appears on both screens (in the same location). So if you're spying and don't want to be detected, you shouldn't type anything. ttysnoop can be used for training with instructor and trainee sitting side-by-side, each at their own terminal. The instructor may watch what the trainee is typing and can correct any mistakes by typing it correctly. The trainee can watch what the instructor types and then repeat it on there own. It's just like they used one terminal with both people having there hands on the keyboard at the same time.

---

17. Trouble-Shooting (software)

If you suspect that the problem is a hardware problem, see the Repair and Diagnose section. If the problem involves the serial port itself see the Serial-HOWTO.

Here is a list of possible problems:

- Is the Terminal OK ? Suspect the terminal is defective.
- Missing Text Either skips over some text or displays some text OK and hangs.
- Getty Respawning Too Rapidly (console error message)
- Fails Just After Login
- Can't Login but login prompt is OK.
- Garbled Login Prompt
- No Sign of any Login Prompt

There are two cases where the terminal goes bad. One is when it's been recently working OK and suddenly goes bad. This is discussed in the next sub-section. The other case is where things don't work right just after you install a terminal. For this case you may skip over the next section.

17.1 Terminal Was Working OK

When a formerly working terminal suddenly goes bad it is often easy to find the problem. That's because if you think about what recently happened it will likely give a clue to the cause of the problem.

The problem may be obvious such as an error message when the terminal is first turned on. If it makes a noise it likely needs repair. See Repair & Diagnose. First, think about what has been done or changed recently as it's likely the cause of the problem. Did the problem happen just after new system software was installed or after a change in the configuration?

If the terminal isn't responding correctly (if at all) to what you type to it, you may have a Corrupted Terminal Interface.

17.2 Terminal Newly Installed

If you've just connected up a terminal to a computer per instructions and it doesn't work this section is for you. If a terminal that formerly worked OK doesn't work now then see Terminal Was Working OK If you suspect that the serial port on your computer may be defective you might try running a diagnostic test program on it. At present (June 1998) it seems that Linux doesn't yet have such a diagnostic program so you may need to run diagnostics under MS DOS/Windows. There are some programs to monitor the various serial lines such at DTR, CTS, etc. and this may help. See Serial Monitoring/Diagnostics

One approach is to first see if the the terminal will work by trying to copy a file to the terminal (cp my_file /dev/ttyS?) under the most simple situation. This means with the modem control lines disabled and at a show speed that doesn't need flow control (make sure that any hardware flow control is disabled). If this copy works, then make the situation a little more complicated and see if it still works, etc., etc. When the trouble appears just after you made a change, then that change is likely the source of the trouble. Actually, its more efficient (but more complex) to jump from the simple situation to about half way to the final configuration so that the test eliminates about half of the remaining possible causes of the problem. Then repeat this methodology for the next test. This way it would only take about 10 tests to find the cause out of a thousand possible causes. You should deviate a little from this method based on hunches and clues.

17.3 Is the Terminal OK ?

A good terminal will usually start up with some words on the screen. If these words convey no error message, its probably OK. If there is no sign of power (no lights on, etc.) re-plug in the computer power cord at both ends. Make sure there is power at the wall jack (or at the extension cord end). Try another power cord if you have one. Make sure the terminal is turned on and that its fuse is not blown. A blank (or dim) screen may sometimes be fixed by just turning up the brightness and contrast using knobs or a keyboard key in set-up mode. If it still won't work See Repair & Diagnose for tips on repairing it.

If the terminal starts up OK, but you suspect that something may be wrong with it, go into "local mode" where is works like a typewriter and try typing on it. See Local Mode.

17.4 Missing Text

If some text displays on the terminal OK and then it stops without finishing (in the middle of a word, etc.) or if chunks of text are missing, you likely have a problem with flow control. If you can't figure out right away what's causing it, decrease the speed. If that fixes it, it's likely a flow control problem. It may be that flow control is not working at all due to failure to configure it correctly, or due to incorrect cable wiring (for hardware flow control). See Flow Control

If single characters are missing, perhaps the serial port is being overrun by too fast a speed. Try a slower baud rate.

If you are using a baud rate under 1200 (very slow, mostly used for old hard-copy terminals and printers) and the text gets truncated, then the problem may be in the serial device driver. See Printing-HOWTO under "Serial devices" on how to fix this.

17.5 Getty Respawning Too Rapidly

TOC

Serial Module Not Loaded

Use the "lsmod" command to see if the serial module is loaded.

No Modem Control Voltage

If getty can't open and/or use a port because of the lack of a positive modem control voltage on one of the pins, then getty might be killed. Then, per the instructions in inittab, getty respawns and tries again, only to be killed again, etc., etc. You may see an error message indicating that due to getty respawning too rapidly it has been temporarily disabled. Try using the "local" option with getty and/or check the modem control settings and voltages.

Key Shorted

Another possible cause of getty respawning is if a keyboard key is shorted, giving the same result as if the key was continuously held down. With auto-repeat enabled, this "types" thousands characters to the login prompt. Look for a screen filled with all the same character (in some cases with 2 or more different characters).

17.6 Fails Just After Login

If you can login OK but then you can't use the terminal it may be because the starting of the login shell has reconfigured the terminal (to an incorrect setting) by a command which someone put into one of the files that are run when you login and a shell starts. These files include /etc/profile and /.bashrc. Look for a command starting with "stty" or "setserial" and make sure that it's correct. Even if it's done OK in one initialization file, it may be reset incorrectly in another initialization file that you are not aware of. Ways to get into the systems to fix it are to use another terminal or console, use a rescue diskette, or type: "linux single" at the lilo prompt which puts you into single user mode without running startup files.

17.7 Can't Login

If you get a login prompt but get no response (or perhaps a garbled response) to your login attempts a possible cause is that the communication is bad one-way from the terminal to the computer. It could be a bad or mis-wired cable/connector. If you're not already using the "local" option with getty, do so to disable the modem control lines. See Getty (in /etc/inittab). You might also disable hardware flow control (stty -crtscts) if it was enabled. If it now works OK then your modem control lines are likely either not wired correctly or there's a mistake in your set-up. Some terminals allow setting different values (such as baud rate) for send and receive so the receive could be OK but the send bad.

You should also (at the console) try "stty < /dev/ttyS1" (if you use ttyS1) to see that it's set up correctly. It will often be in raw mode (and this is probably OK) with -icanon and -echo etc. If the terminal incorrectly set at half-duplex (HDX), then one set of the characters you see when you type are coming from the terminal itself. If the characters are doubled, then the echos from the computer are OK and you may switch to full-duplex to fix this. But if half-duplex is set and you only see what looks like normal

"echos", then they are not coming from the computer as they should be.

If you get a message saying something like "login failed" then if there is no error in typing or in the password, there may be some restrictions on logins which will not allow you to log in. Unfortunately, this message, may not tell you why it failed. See Login Restrictions

17.8 Garbled Login Prompt

This may be due to using the wrong character set, transmission errors due to too high of a baud rate, incompatible baud rates, incompatible parity, or the wrong number of bits per byte. If it's a variety of strange characters you have the wrong character set or a high order bit is being set by mistake. If words are misspelled, try a lower baud rate. For baud, parity, or bits/character incompatibilities you see a lot of the same "error character" which represents a real character that can't be displayed correctly due to an error in parity or baud rate.

If you are using agetty (often just named getty), the agetty program will detect and set parity and/or bits/character if you type something. Try it with a return to see if it fixes it.

17.9 No Sign of any Login Prompt

This is when nothing at all happens at the terminal, but the terminal seems to be working OK. One of the first things to do is to make sure that all cable connections are tight and connected to the correct connector on both the computer and terminal. Other causes include defective hardware or cables (must be null modem), getty not running, terminal setup at wrong baud-rate, terminal in local mode, etc.. At this point two different avenues of approach are (you may pursue more than one at a time).

- Diagnose Problem from the Console
- Measure Voltages Measure Voltages

Diagnose Problem from the Console

At the console (or another working terminal), use "top" or "ps -al" to see if getty is running on the port. Don't confuse it with getty programs running on other ports or on the virtual consoles. You will not get a login prompt unless getty runs.

One test is to try to copy a short file to the terminal (It might be a good idea to try this near the start of the installation process before setting up getty). Use the Linux copy command such as: cp file_name /dev/ttyS1. If it doesn't work, use stty to make the interface as simple as possible with everything disabled (such as hardware flow control: -crtscts; parity, and modem control signals: clocal). Be sure the baud rates and the bits/byte are the same. If nothing happens verify that the port is alive with a voltmeter per the next section.

TOC

Measure Voltages

If you have a voltmeter handy check for a negative voltage (-4v to -15v) at pin 3 (receive data) at the terminal side of the null modem cable. The positive lead of the meter should be connected to a good ground (the metal connectors on the ends of cables are often not grounded). If there is no such negative voltage then check for it at the transmit pin (TxD) on the computer (see DB9-DB25 for the pin-out). If it's present there but not at the receive pin (RxD) at the terminal, then the cable is bad (loose connection, broken wire, or not a null modem). If voltage is absent, then the serial port on the computer is dead. Test it with a software diagnostic test or replace it.

If the serial port is alive, you may want to send a file to it (with modem controls disabled) and see if anything gets to it. To check for a transmitted signal with an analog voltmeter, look at the needle at -12 V when the line is idle. Then start sending a file (or start getty). You should see the needle dropping to 0 and fluttering about 0 as it measures short-run averages of the bit stream. You can see this also on the AC scale provided that your meter has a capacitor to block out DC voltages when on the AC scale. If it doesn't, then the idle DC of -12 V will cause a high false AC reading. Without a meter, you could connect a known good device (such as another terminal or an external modem) to the serial port and see if it works OK.

17.10 Serial Monitoring/Diagnostics

A few Linux programs will monitor the modem control lines and indicate if they are positive (1) or negative (0).

- statserial (in Debian distribution)
- serialmon (doesn't monitor RTS, CTS, DSR but logs other functions)
- modemstat (only works on Linux PC consoles. Will coexist with the command line)

You may already have them. If not, go to Serial Software. When using these, bear in mind that what you see is the state of the lines at the host computer. The situation at the terminal will be different since some wires are often missing from cables while other wires cross over. As of June 1998, I know of no diagnostic program in Linux for the serial port.

17.11 Local Mode

In local mode, the terminal disconnects from the computer and behaves like a typewriter (only it doesn't type on paper but on the screen). Going back into on-line reconnects to the computer allowing you to resume activities at the same point where you left off when you went into "local". This is useful both for testing the terminal and for educational purposes. When in local mode you may type escape sequences (starting with the ESC key) and observe what they do. If the terminal doesn't work correctly in local mode, it's unlikely that it will work correctly when connected to the computer. If you're not exactly sure what an escape sequence does, you can try it out in local mode. You might also use it for trying out a terminal that is for sale. To get into local mode you first enter set-up mode and then select "local" from a menu (or press a certain key). See Getting Into Set-Up (Configuration) Mode.

17.12 Serial Electrical Test Equipment

TOC

Breakout Gadgets, etc.

While a multimeter (used as a voltmeter) may be all that you need for just a few terminals, simple special test equipment has been made for testing serial port lines. Some are called "breakout ... " where breakout means to break out conductors from a cable. These gadgets have a couple of connectors on them and insert into the serial cable. Some have test points for connecting a voltmeter. Others have LED lamps which light when certain modem control lines are asserted (turned on). Still others have jumpers so that you can connect any wire to any wire. Some have switches.

Radio Shack sells (in 1998) a "RS-232 Troubleshooter" or "RS-232 Line Tester" which checks TD, RD, CD, RTS, CTS, DTR, and DSR. A green light means on (+12 v) while red means off (-12 v). They also sell a "RS-232 Serial Jumper Box" which permits connecting the pins anyway you choose.

Measuring Voltages

Any voltmeter or multimeter, even the cheapest that sells for about $10, should work fine. Trying to use other methods for checking voltage is tricky. Don't use a LED unless it has a series resistor to reduce the voltage across the LED. A 470 ohm resistor is used for a 20 ma LED (but not all LED's are 20 ma). The LED will only light for a certain polarity so you may test for + or - voltages. Does anyone make such a gadget for automotive circuit testing?? Logic probes may be damaged if you try to use them since the TTL voltages for which they are designed are only 5 volts. Trying to use a 12 V incandescent light bulb is not a good idea. It won't show polarity and due to limited output current of the UART it probably will not even light up.

To measure voltage on a female connector you may plug in a bent paper clip into the desired opening. The paper clip's diameter should be no larger than the pins so that it doesn't damage the contact. Clip an alligator clip (or the like) to the paper clip to connect up.

Taste Voltage

As a last resort, if you have no test equipment and are willing to risk getting shocked (or even electrocuted) you can always taste the voltage. Before touching one of the test leads with your tongue, test them to make sure that there is no high voltage on them. Touch both leads (at the same time) to one hand to see if they shock you. Then if no shock, wet the skin contact points by licking and repeat. If this test gives you a shock, you certainly don't want to use your tongue.

For the test for 12 V, Lick a finger and hold one test lead in it. Put the other test lead on your tongue. If the lead on your tongue is positive, there will be a noticeable taste. You might try this with flashlight batteries first so you will know what taste to expect.

---

18. Repair & Diagnose

Repairing a terminal has much in common with repairing a monitor and/or keyboard. Sometimes the built-in diagnostics of the terminal will tell you what is wrong on the screen. If not, then by the symptoms, one may often isolate the trouble to one of the following: bad keyboard, CRT dead, terminal digital electronics failure. It's best to have a service manual, but even if you don't have one, many terminals may still be repaired.

18.1 Repair Books & Websites

Books

Bigelow, Stephen J.: Troubleshooting & Repairing Computer Monitors, 2nd edition, McGraw-Hill, 1997. Doesn't cover the character generation electronics nor the keyboard.

Websites

The FAQ http://www.repairfaq.org for the newsgroup: sci.electronics.repair is long and comprehensive, although it doesn't cover terminals per se. See the section "Computer and Video Monitors". Much of this information is applicable to terminals as are the sections: "Testing Capacitors", "Testing Flyback Transformers", etc. Perhaps in the future, the "info" on repair in this HOWTO will consist mainly of links to the above FAQ (or the like). Shuford's repair archive of newsgroup postings on terminal repair is another source of info.

18.2 Safety

CRT's use high voltage of up to 30,000 volts for color (less for monochrome). Be careful not to touch this voltage if the set is on and the cover off. It probably won't kill you even if you do since the amount of current it can supply is limited. But it is likely to badly burn and shock you, etc. High voltage can jump across air gaps and go thru cracked insulation so keep your hands a safe distance from it. You should notice the well-insulated high voltage cable connected to one side of the picture tube. Even when the set is off, there is still enough residual voltage on the picture tube cable connection to give you quite a shock. To discharge this voltage when the set is unplugged use a screwdriver (insulated handle) with the metal blade grounded to the picture tube ground cable with a jumper wire. Don't use chassis ground.

The lower voltages (of hundreds of volts) can be even more dangerous since they are not current limited. It is even more dangerous if your hands are wet or if you are wearing a metal watchband, ring or the like. In rare cases people have been killed by it so be careful. The lowest voltages of only several volts on digital circuitry are fairly safe but don't touch anything (except with a well insulated tool) unless you know for sure.

18.3 Appearance of Display

TOC

If the display is too dim, turn up the brightness and/or contrast. using knobs on the exterior of the unit (if they exist). If the width, height or centering is incorrect, there are often control knobs for these. For some older terminals one must press an arrow key (or the like) in set-up mode.

You may need to remove the cover to make adjustments, especially on older models. You could arrange things so that a large mirror is in front of the terminal so as to view the display in the mirror while making adjustments. The adjustments to turn may be on a printed circuit board. While a screwdriver (possibly Phillips-head) may be all that's needed, inductors may require special TV alignment tools (plastic hex wrenches, etc.). The abbreviated name of the adjustment should be printed on the circuit board. For example, here are some such names:

- V-Size adjusts the Vertical height (Size)
- H-Size adjusts the Horizontal width (Size). It may be an inductor.
- V-Pos adjusts the Vertical Position
- H-Pos adjusts the Horizontal Position
- V-Lin adjusts Vertical Linearity (Use if width of scan lines differs at the top and bottom of the screen)
- V-Hold adjusts Vertical Hold (Use if screen is uncontrollable scrolling)
- Bright adjusts brightness (an external knob may also exist)
- Sub-Bright adjusts brightness of subdued intensity mode (often the normal mode: dimmer than bold or bright mode).

Changing linearity may change the size so that it will need to be readjusted. A terminal that has been stored for some time may have a small display rectangle on the screen surrounded by a large black border. If it's difficult to adjust, wait a while before adjusting it since it will likely recover some with use (the black borders will shrink).

18.4 Diagnose

Terminal Made a Noise

If the terminal made some noise just before it failed (or when you turn it on after it failed) that noise is a clue to what is wrong. If you hear a sparking noise or see/smell smoke, immediately turn the terminal off to prevent further damage. The problem is likely in the high voltage power supply of several thousand volts. Remove the cover and if the bad spot is not evident, turn it on again for a short time in a dimly lit room to look for arcing. The high voltage cable (runs between the flyback transformer and the side of the picture tube) may have broken insulation that arcs to ground. Fix it with high-voltage insulating dope, or special electrical tape designed say for 10,000 volts.

The flyback transformer (high voltage) may make only a faint clicking or sparking noise if it fails. You may not hear it until you turn the terminal off for a while to rest and then turn it back on again. To track down the noise you may use a piece of small rubber tubing (such as used in automobiles) as a

TOC

stethoscope to listen to it. But while you are listening for the noise, the terminal is suffering more da so try find it fast (but not so fast as to risk getting shocked).

A short in the power supply may cause a fuse to blow with a pop. Replacing a blown fuse may not solve the problem as the same short may blow the fuse again. Inspect for any darkened spots due to high heat and test those components. Shorted power transistors may cause the fuse to blow. They may be tested with a transistor checker or even with an ohm-meter. Use the low ohm scale on an ohm-meter so that the voltage applied by the meter is low. This will reduce the possible damage to good components caused by this test voltage.

If the terminal has been exposed to dampness such as being stored in a damp place or near a kitchen with steam from cooking, a fix may be to dry out the unit. Heating a "failed" flyback transformer with a blow dryer for several minutes may restore it.

Terminal Made No Noise

A blank screen may be due to someone turning the brightness control to the lowest level or to aging. The next thing to do is to check the cables for loose or broken connections. If there is no sign of power, substitute a new power cord after making sure that the power outlet on the wall is "hot".

If the keyboard is suspected, try it on another terminal of the same type or substitute a good keyboard. Wiggle the keyboard cable ends and the plug. Wires inside cables may break, especially near their ends. If the break is verified by wiggling it (having the problem go on and off in synchronization with the wiggles), then one may either get a new cable or cut into the cable and re-solder the breaks, etc.

One of the first things to do if the keyboard works is to put the terminal into Local Mode. If it works OK in local, then the problem is likely in the connection to the host computer (or incorrect interface) or in the UART chips of the terminal.

By carefully inspecting the circuitry, one may often find the cause of the problem. Look for discoloration, cracks, etc. An intermittent problem may sometimes be found by tapping on components with a ball-point pen (not the metal tip of course). A break in the conductor of a printed circuit board may sometimes be revealed by flexing the board. Solder that looks like it formed a drop or a solder joint with little solder may need re-soldering. Soldering may heat up transistors (and other components) and damage them so use a heat sink if feasible.

If you have a common brand of terminal, you may be able to search newsgroup postings on the Internet to find out what the most frequent types of problems are for your terminal and perhaps information on how to fix them.

To see if the digital electronics work, try (using a good keyboard) typing at the bad terminal. Try to read this typing at a good terminal (or the console) using the copy command or with a terminal communication program such as Minicom. You may need to hit the return key at the terminal in order to send a line. One may ask the bad terminal for its identity etc. from another terminal. This will show if two-way communication works.

TOC

18.5 Error Messages on the Screen

You are in luck if you see an error message on the screen. This usually happens when you first turn the terminal on.

Keyboard Error

This usually means that the keyboard is not plugged in, or that the connection is loose. For more serious problems see Keyboards

Checksum Error in NVR

NVR is "Non-Volatile RAM". This means that the NVR where the set-up information is stored has become corrupted. The terminal will likely still work but the configuration that was last saved when someone last configured the terminal has likely been lost. Try configuring again and then save it. It might work. On very old terminals (early 1980's) there was a battery-powered CMOS to save the configuration so in this case the problem could be just a dead battery. Sometimes the EEPROM chip (no battery needed) goes bad after too many saves. It may be hard to find. If you can't fix it you are either stuck with the default configuration or you may have escape sequences sent to the terminal when you start it up to try to configure it.

18.6 Capacitors

Electrolytic capacitors have a metal shell and are may become weak or fail if they set for years without being used. Sometimes just leaving the terminal on for a while will help partially restore them. If you can, exercise any terminals you have in storage by turning them on for a while every year or two.

18.7 Keyboards

Interchangeability

The keyboards for terminals are not the same as keyboards for PC's. The difference is not only in the key layout but in the codes generated when a key is pressed. Also, keyboards for various brands and models of terminals are not always interchangeable with each other. Sometimes one get an "incompatible" keyboard to partially work on a terminal. All the ASCII keys will work OK, but special keys such as set-up and break will not work correctly.

How They Work

Most keyboards just make a simple contact between two conductors when you press a key. Electronics inside a chip in the keyboard converts this contact closure into a code sent over the keyboard's external cable. Instead of having a separate wire (or conductor) going from each key to the chip, the following type scheme is used. Number the conductors say from 1-10 and A-J. For example: conductor 3 goes to

TOC

several keys and conductor B goes to several keys, but only one key has both conductors 3 and B going to it. When that key is pressed, a short circuit is established between 3 and B. The chip senses this short and knows what key has been pressed. Such a scheme reduces the number of conductors needed (and reduces the number of pins needed on the chip). It's a similar scheme to what is called a "crossbar switch".

One Press Types 2 Different Characters

If, due to a defect, conductors 3 and 4 become shorted together then pressing the 3-B key will also short 4 and B and the chip will think that both keys 3-B and 4-B have been pressed. This is likely to type 2 different characters when all you wanted was one character.

Modern vs Old Keyboards

While the modern keyboard and the old fashioned type look about the same, the mechanics of operation are different. The old ones have individual key switches under the key-caps with each switch enclosed in a hard plastic case. The modern ones use large flexible plastic sheets (membrane) the size of the keyboard. A plastic sheet with holes in it is sandwiched between two other plastic sheets containing printed circuits (including contact points). When you press a key, the two "printed" sheets are pressed together at a certain point, closing the contacts printed on the sheets at that point.

Keyboard Doesn't Work At All

If none of the keys work try another keyboard (if you have one) to verify that the keyboard is the problem. The most likely cause is a broken wire inside the cord (cable) that connects it to the terminal. The most likely location of the break is at either end of the cord. Try wigging the ends of the cord while tapping on a key to see if it works intermittently. If you find a bad spot, you may carefully cut into the cord with a knife at the bad spot and splice the broken conductor. Sometimes just a drop of solder will splice it. Seal up the cord with electrical tape, glue, or caulk.

Typing b Displays bb, etc. (doubled)

If all characters appear double there is likely nothing wrong with the keyboard. Instead, your terminal has likely been incorrectly set up for half-duplex (HDX or local echo=on) and every character you type is echoed back both from the electronics inside your terminal and from your host computer. If the two characters are not the same, there may be a short circuit inside your keyboard. See One Press Types 2 Different Characters

The Keyboard Types By Itself

If a key is shorted out it is likely to type out a large number of the same character if auto-repeat is enabled. If more than one key is shorted out then repeating sequences of a few characters will be typed. This may cause getty to respawn too fast if it happens at the login prompt. See Key Shorted. The fix is to clean the contacts per Cleaning Keyboard Contacts.

TOC

Liquid Spilled on the Keyboard

If water or watery liquid has been spilled on the keyboard (or if it was exposed to rain, heavy dew, or dampness) some keys may not work right. The dampness may cause a key to short out (like it was pressed down all the time) and you may see the screen fill up with that letter if auto-repeat is enabled. If it's gotten wet and then partially (or fully) dried out, certain keys may not work due to deposits on the contact surfaces. For the modern type of keyboard, one may readily take apart the plastic sheets inside and dry/clean them. For the old type one may let it dry out in the sun or oven (low temp.). When it's dry it may still need contact cleaner on some keys as explained below.

Cleaning Keyboard Contacts

Keyboards with Membranes

On some newer keyboards, the plastic sheets (membranes) are easy to remove for inspection and cleaning if needed. You only need to remove several screws to take apart the keyboard and get to the sheets. On some old IBM keyboards the sheets can't be removed without breaking off many plastic tabs which will need to be repaired with glue to put back (probably not worthwhile to repair). Such a keyboard may sometimes be made to work by flexing, twisting, and/or pounding the assembly containing the plastic sheets.

Keyboards with Individual Switches

What follows is for older keyboards that have separate hard plastic switches for each key. Before going to all the work of cleaning electrical contacts first try turning the keyboard upside-down and working the bad keys. This may help dislodge dirt, especially if you press the key hard and fast to set up vibration. Pressing the key down and wiggling it from side to side, etc. often helps.

Often the key-caps may be removed by prying them upward using a small screwdriver as a lever while preventing excessive tilting with a finger. There exists a special tool known as keycap puller but you can get by without it. (Warning: Key-caps on modern keyboards don't pry up.) The key-cap may tilt a bit and wobble as it comes loose. It may even fly up and onto the floor. Then you have two choices on how to clean the contacts: Use contact cleaner spray directly on top of the key switch, or take the key switch apart and clean it. Still another choice is to replace the key switch with a new or used one.

Directly spraying contact cleaner or the like (obtained at an electronics store) into the top of the key switch is the fastest method but may not work and it also might damage the plastic. Before spraying, clean the area next to the switch rods. With the keyboard live (or with the key contacts connected to an ohm-meter) use the tube which came with the spray to squirt cleaner so it will get inside the key switch. Don't let the cleaning liquid get under nearby keys where it may pick up dust and then seep (with the dust) into adjacent key switches. If you make this mistake you may fix one key but damage nearby keys. If this should happen, immediately work the affected nearby keys until they work OK.

You might tilt the keyboard so that the cleaner flows into the contacts. For the CIT101e terminal with an Alps keyboard, this means tilting the digit row up toward the ceiling. Work the key switch up and down with a pen or small screwdriver handle to avoid getting the toxic cleaner liquid on your skin (or wear

gloves). Finally turn the keyboard upside-down while working the key to drain off remaining cleaner. The more cleaner you squirt in the more likely it will fix it but it is also more likely to do more damage to the plastic or contaminate adjacent keys, so use what you think is just enough to do the job. Once the key works OK, work it up and down a little more and test it a half minute later, etc. to make sure it will still work OK.

Sometimes a key works fine when the contacts inside are saturated with contact cleaner liquid, but when the liquid dries a few minutes later then the resulting scale on the contacts prevents good contact and the key works erratically (if at all). Operating the key when the liquid is drying inside may help. Some switches have the contacts nearly sealed inside so little if any contact cleaner reaches the contacts. The cleaner that does get to the contacts may carry contamination with it (cleaning around the tops before spraying helps minimize this).

If you need to disassemble the key switch, first inspect it to see how it is installed and comes apart. Sometimes one may remove the cover of the switch without removing the switch from the keyboard. To do this pry up (or pull up) the top of the key switch after prying apart thin plastic tabs that retain it. Don't pry too hard or you may break the thin plastic. If this can't be done, you may have to unsolder the switch and remove it in order to take it apart (or replace it). Once the switch has been taken apart you still may not be able to see the contacts if the contact surfaces are sandwiched together (nearly touching). You may get contact cleaner on the contacts by slightly prying apart the conducting surfaces and squirting cleaner between them. There may be some kind of clip holding the contact surfaces together which needs to be removed before prying these surfaces apart. With cleaner on the contacts, work them. Tilting the keyboard or inverting it may help. Take care not to loose small parts as they may fly up into the air when taking apart a key switch.

---

19. Appendix A: General

19.1 List of Linux Terminal Commands

Sending a Command to the Terminal

- setterm: long options
- tput: terse options
- tset: initializes only
- clear: clears screen
- reset: sends reset string

Configuring the Terminal Device Driver

- Setserial:
- Stty

Terminfo

- Terminfo Compiler (tic) terminfo compiler & translator
- toe: shows list of terminals for which you have terminfo files
- infocmp compares or displays terminfo entries

Other

- gitkeys: shows what bytes each key sends to the host.
- tty: shows what tty port you are connected to.
- set (or tset -q): shows the value of TERM, the terminfo entry name
- tset: sets TERM interactively and initializes

19.2 The Internet and Books

Terminal Info on the Internet

- Shuford's Website at the University of Tennessee has a great deal of useful information about text terminals.

TOC

- Boundless purchased the VT and Dorio terminal business from DEC. To get Specs select either ADDS, VT, or DORIO links. Then select a "data sheet" link. Then on the data sheet select the "Go to Specs" link.
- Wyse text-terminals kbase is a major manufacturer of terminals. For new models see Wyse terminal See also Old Wyse terminal Specs
- Escape Seqs.; N. America or Escape Seqs.; Europe is a list of escape sequences (and control codes) for some terminal emulations (including VT 100, 300, 420, and Wyse).
- comp.terminals is the newsgroup for terminals

Books Related to Terminals

- EIA-232 serial port see EIA-232 (RS-232) Books.
- Repair see Repair Books & Websites.
- Terminfo database see Termcap Documents

Entire Books on Terminals

As far as I know, there is no satisfactory book on text terminals (unless you are interested in antique terminals of the 1970's).

- Handbook of Interactive Computer Terminals by Duane E. Sharp; Reston Publishing Co. 1977. (mostly obsolete)
- Communicating with Display Terminals by Roger K. deBry; McGraw-Hill 1985. (mostly on IBM synchronous terminals)

The "HANDBOOK ... " presents brief specifications of over 100 different models of antique terminals made in the early 1970's by over 60 different companies. It also explains how they work physically but incorrectly shows a diagram for a CRT which uses electrostatic deflection of the electron beam (p. 36). Terminals actually used magnetic deflection (even in the 1970's). This book explains a number of advanced technical concepts such as "random scan" and "color penetration principle".

The "COMMUNICATING ... " book in contrast to the "Handbook ... " ignores the physical and electronic details of terminals. It has an entire chapter explaining binary numbers (which is not needed in a book on terminals since this information is widely available elsewhere). It seems to mostly cover old IBM terminals (mainly the 3270) in block and synchronous modes of operation. It's of little use for the commonly used ANSI terminals used today on Unix-like systems. Although it does discuss them a little it doesn't show the various wiring schemes used to connect them to serial ports.

Books with Chapters on Terminals

These chapters cover almost nothing about the terminals themselves and their capabilities. Rather, these chapters are mostly about how to set up the computer (and its terminal driver) to work with terminals. Due to the differences of different Unix-like systems, much of the information does not not apply to Linux.

- Unix Power Tools by Jerry Peck et. al. O'Reilly 1998. Ch. 5 Setting Up Your Terminal, Ch. 41: Terminal and Serial Line Settings, Ch. 42: Problems With Terminals
- Advanced Programming in the Unix Environment by W. Richard Stevens Addison-Wesley, 1993. Ch. 11: Terminal I/O, Ch. 19: Pseudo Terminals
- Essential System Administration by Aleen Frisch, 2nd ed. O'Reilly, 1998. Ch. 11: Terminals and Modems.

The "UNIX POWER TOOLS" book has 3 short chapters on text terminals. It covers less ground than this HOWTO but gives more examples to help you.

The "ADVANCED PROGRAMMING ... " Chapter 11 covers only the device driver included in the operating system to deal with terminals. It explains the parameters one gives to the stty command to configure the terminal.

The "ESSENTIAL SYSTEM ..." book's chapter has more about terminals than modems. It seems well written.

19.3 Non-Linux OS's

The configuration of the host computer for terminals for non-Linux operating systems is usually significantly different than for Linux. Here are some links to on-line manuals for Unix-like systems.

- SCO's OpenServer Adding Serial Terminals in SCO OpenServer Handbook.
- Hewlett-Packard's HP-UX Configuring Terminals and Modems

---

20. Appendix B: Escape Sequence Commands Terminology

These are sometimes called "control sequences". This section of Text-Terminal-HOWTO is incomplete (and may never be complete as there are such a huge number of control sequences). This section is for reference and perhaps really belongs in something that would be called "Text-Terminal-Programming-HOWTO".

An example of an ANSI standard escape sequence is ESC[5B which moves the cursor down 5 lines. ESC is the Escape character. The parameter 5 is included in the sequence. If it were 7 the cursor would move down 7 lines, etc. A listing for this sequence as "move cursor down x lines: ESC[xB" is easy to to understand. But command jargon such as: "tertiary device attribute request" is less comprehensible. This section will try to explain some of the more arcane jargon used for escape sequence commands. A full listing (including the escape sequence codes for the ANSI standard) is a "wish list" project. Since many escape sequences do the same thing as is done when setting up the terminal with Set-Up Options, such escape sequences options will not be repeated here.

20.1 Esc Sequence List

For a list of many (but not all) escape sequences for various terminals see Escape Seqs.; N. America or Escape Seqs.; Europe. These are used for terminal emulation and are not always the same as on the corresponding real terminal.. A list for VT (not maintained) may be found at Emulators FAQ. Search for "VT".

20.2 8-bit Control Codes

Table of 8-bit DEC control codes (in hexadecimal). Work on VT2xx or later. CSI is the most common.

```
ACRONYM  FULL_NAME                          HEX     REPLACES
IND Index (down one line)                   84      ESC D
NEL Next Line                               85      ESC E
RI  Reverse Index (one line up)             8D      ESC M
SS2 Single Shift 2                          8E      ESC N
SS3 Single Shift 3                          8F      ESC O
DCS Device Control String                   90      ESC P
CSI Control Sequence Introducer)            9B      ESC [
ST  String Terminator                       9C      ESC \
```

TOC

20.3 Printer Esc

- Auto Print on/off: When on, data from the host is also teed (sent) to the printer port of the terminal (and also shows on the terminal screen).
- Print Controller on/off: When on, data from the host is sent only to the printer (nothing shows on the terminal screen).

20.4 Reports

These sequences are usually a request sent from the host to request a report from the terminal. The terminal responds by sending a report (actually another escape sequence) to the host which has embedded in it certain values telling the host about the current state of the terminal. In some cases a report may be sent to the host even if it wasn't asked for. This sometimes happens when set-up is exited. By default no unsolicited reports should be sent.

- Request for Status (Report Operating Status): Meaning of replies for VT100 is either "I'm OK" or "I'm not OK"
- Request for Device Attributes: The "device" is usually the printer. Is there a printer? Is it ready?
- Reqest for Tertiary Device Attributes (VT): Reply is report that was entered during set-up. The tertiary device is the 3rd device (the printer or auxiliary port device ??). The 1st device may be the host computer and the 2nd device the terminal.
- Request for Terminal Parameters: What is the parity, baud rate, byte width, etc. This request doesn't seem to make much sense, since if the host didn't already know this it couldn't communicate with the terminal or send a reply.

20.5 Cursor Movements

The cursor is where the next character received from the host will be displayed. Most of the cursor movements are self-explanatory. "index cursor" means to move the cursor down one line. Cursor movements may be relative to the current position such as "move 4 spaces left" or absolute such as "move to row 3, column 39". Absolute is called "Direct Cursor Positioning" or "Direct Cursor Addressing".

The home position is row 1 col. 1 (index origin is 1). But where this home position is on the physical screen is not completely clear. If "Cursor Origin Mode" = "Relative Origin Mode" is set, then home is at the top of the scrolling region (not necessarily the top of the screen) at the left edge of the screen. If "Absolute Origin Mode" is set (the same as unsetting any of the two modes in the previous sentence) then home is at the upper left corner of the screen. On some old terminals if "Cursor Origin Mode" is set it means that it's relative.

20.6 Pages (definition)

TOC

See Pages for an explanation of pages. There are a number of escape sequences to deal with pages. Text may be copied from one page to another and one may move the cursor from page to page. Switching pages may or may not be automatic: when the screen becomes full (page 1) then more data from the host goes to page 2. The cursor may only be on one page at a time and characters which are sent to the terminal go there. If that page is not being displayed, new text will be received by the terminal and go into display memory, but you will not see it (until the terminal is switched to that page).

---

21. Appendix C: Serial Communications on EIA-232 (RS-232)

21.1 Intro to Serial Communication

(Much of this section is now found in Serial-HOWTO.) Text terminals on Unix-like systems (and on PC's) are usually connected to an asynchronous 232 serial port of a computer. It's usually a RS-232-C, EIA-232-D, or EIA-232-E. These three are almost the same thing. The original RS prefix became EIA (Electronics Industries Association) and later EIA/TIA after EIA merged with TIA (Telecommunications Industries Association). The EIA-232 spec provides also for synchronous (sync) communication but the hardware to support sync is almost always missing on PC's. The RS designation is obsolete but is still in use. EIA will be used in this article.

The serial port is more than just a physical connector on the back of a computer or terminal. It includes the associated electronics which must produce signals conforming to the EIA-232 specification. The standard connector has 25 pins, most of which are unused. An alternative connector has only 9 pins. One pin is used to send out data bytes and another to receive data bytes. Another pin is a common signal ground. The other "useful" pins are used mainly for signalling purposes with a steady negative voltage meaning "off" and a steady positive voltage meaning "on".

The UART (Universal Asynchronous Receiver-Transmitter) chip does most of the work. Today, the functionality of this chip is usually built into another chip.

21.2 Voltages

Voltage for a Bit

At the EIA-232 serial port, voltages are bipolar (positive or negative with respect to ground) and should be about 12 volts in magnitude (some are 5 or 10 volts). For the transmit and receive pins +12 volts is a 0-bit (sometimes called "space") and -12 volts is a 1-bit (sometimes called "mark"). This is known as inverted logic since normally a 0-bit is both false and negative while a one is normally both true and positive. Although the receive and transmit pins are inverted logic, other pins (modem control lines) are normal logic with a positive voltage being true (or "on" or "asserted") and a negative voltage being false (or "off" or "negated"). Zero voltage has no meaning (except it usually means that the unit is powered off).

A range of voltages is allowed. The specs say the magnitude of a transmitted signal should be between 5 and 15 volts but must never exceed 25 V. Any voltage received under 3 V is undefined (but some terminals will accept a lower voltage as valid). One sometimes sees erroneous claims that the voltage is commonly 5 volts (or even 3 volts) but it's usually 11-12 volts. If you are using a EIA-422 port on a Mac computer as an EIA-232 (requires a special cable) or EIA-423 then the voltage will actually be only 5 V.

TOC

The discussion here assumes 12 V. There is much confusion about voltages on the Internet.

Note that normal computer logic normally is just a few volts (5 volts was once the standard) so that if you try to use test equipment designed for testing 3-5 volt computer logic (TTL) on the 12 volts of a serial port, it may damage the test equipment.

Voltage Sequence for a Byte

The transmit pin (TxD) is held at -12 V (mark) at idle when nothing is being sent. To start a byte it jumps to +12 V (space) for the start bit and remains at +12 V for the duration (period) of the start bit. Next comes the low-order bit of the data byte. If it's a 0-bit nothing changes and the line remains at +12 V for another bit-period. Then comes the next bit, etc. Finally, a parity bit may be sent and then a -12 V (mark) stop bit. The line remains at -12 V (idle) until the next start bit. Note that there is no return to 0 volts and thus there is no simple way (except by a synchronizing signal) to tell where one bit ends and the next one begins for the case where 2 consecutive bits are the same polarity (both zero or both one).

A 2nd stop bit would also be -12 V, just the same as the first stop bit. Since there is no signal to mark the boundaries between these bits, the only effect of the 2nd stop bit is that the line must remain at -12 V idle twice as long. The receiver has no way of detecting the difference between a 2nd stop bit and a longer idle time between bytes. Thus communications works OK if one end uses one stop bit and the other end uses 2 stop bits, but using only one stop bit is obviously faster. In rare cases 1 1/2 stop bits are used. This means that the line is kept at -12 V for 1 1/2 time periods (like a stop bit 50% wider than normal).

21.3 Parity Explained

Characters are normally transmitted with either 7 or 8 bits (of data). An additional parity bit may (or may not) be appended to this resulting in a byte length of 7, 8 or 9 bits. Some terminal emulators and older terminals do not allow 9 bits. Some prohibit 9 bits if 2 stop bits are used (since this would make the total number of bits too large: 12 bits total).

The parity may be set to odd, even or none (mark and space parity may be options on some terminals). With odd parity, the parity bit is selected so that the number of 1-bits in a byte, including the parity bit, is odd. If a such a byte gets corrupted by a bit being flipped, the result is an illegal byte of even parity. This error will be detected and if it's an incoming byte to the terminal an error-character symbol will appear on the screen. Even parity works in a similar manner with all legal bytes (including the parity bit) having an even number of 1-bits. During set-up, the number of bits per character usually means only the number of data bits per byte (7 for true ASCII and 8 for various ISO character sets).

A "mark" is a 1-bit (or logic 1) and a "space" is a 0-bit (or logic 0). For mark parity, the parity bit is always a one-bit. For space parity it's always a zero-bit. Mark or space parity only wastes bandwidth and should be avoided when feasible. "No parity" means that no parity bit is added. For terminals that don't permit 9 bit bytes, "no parity" must be selected when using 8 bit character sets since there is no room for a parity bit.

TOC

21.4 Forming a Byte (Framing)

In serial transmission of bytes via EIA-232 ports, the low-order bit is always sent first. Serial ports on PC's use asynchronous communication where there is a start bit and a stop bit to mark the beginning and end of a byte. This is called framing and the framed byte is sometimes called a frame. As a result a total of 9, 10, or 11 bits are sent per byte with 10 being the most common. 8-N-1 means 8 data bits, No parity, 1 stop bit. This adds up to 10 bits total when one counts the start bit. One stop bit is almost universally used. At 110 bits/sec (and sometimes at 300 bits/sec) 2 stop bits were once used but today the 2nd stop bit is used only in very unusual situations (or by mistake since it seemingly still works OK that way).

21.5 Limitations of EIA-232

Low Speed & Short Distance

The conventional EIA-232 serial port is inherently low speed and is severely limited in distance. Ads often read "high speed" but it can only work at high speed over very short distances such as to a modem located right next to the computer. All of the wires use a common ground return so that twisted-pair technology (needed for high speeds) can't be used without additional hardware. However some computers have more modern interfaces. See Successors to EIA-232.

It is somewhat tragic that the RS-232 standard from 1969 did not use twisted pair technology which could operate about a hundred times faster. Twisted pairs have been used in telephone cables since the late 1800's. In 1888 (over 110 years ago) the "Cable Conference" reported its support of twisted-pair (for telephone systems) and pointed out its advantages. But over 80 years after this approval by the "Cable Conference", RS-232 failed to utilize it. Since RS-232 was originally designed for connecting a terminal to a low speed modem located nearby, the need for high speed and longer distance transmission was apparently not recognized.

Successors to EIA-232

A number of EIA standards have been established for higher speeds and longer distances using twisted-pair (balanced) technology. Balanced transmission can sometimes be a hundred times faster than unbalanced EIA-232. For a given speed, the distance (maximum cable length) may be many times longer with twisted pair. But PC-s keep being made with the "obsolete" EIA-232 since it works OK with modems connected to slow telephone lines, and it works OK with mice.

One exception is Apple's Mac computer with its EIA-232/EIA-422 GeoPort which provides twisted-pairs (balanced) for transmit and receive. It uses a small round "mini-DIN" connector. It also provides conventional EIA-232 but only at 5 volts (which is still legal EIA-232). However, due to the fact that Macs cost more than PC's, they are seldom used as a host computer for terminals. Some newer terminals use EIA-423 but this is just like the unbalanced EIA-232 and can be connected to a EIA-232 port. This EIA-423 is only 5 volts, but the specs call for higher speeds than the EIA-232 (which will be of no help on a long run where it's the unbalance that causes interference).

EIA-530-A (balanced but can also be used unbalanced) at 2Mbits/s (balanced) was intended to be a

TOC

replacement for EIA-232 but few have been installed. It uses the same 25-pin connector as EIA-232. High Speed Serial Interface (HSSI = EIA-612/613) uses a 50-pin connector and goes up to about 50 Mbits/s but the distance is limited to only several meters. The Universal Serial Bus (USB) is being built into PCI chips. It is 12 Mbits/s over a twisted pair with a 4-pin connector (2 wires are power supply) but it also is limited to short distances of at most 5 meters (depends on configuration).

Line Drivers

For a text terminal, the EIA-232 speeds are fast enough but the usable cable length is often too short. Balanced technology could fix this. The common method of obtaining balanced communication with a text terminal is to install 2@ line drivers in the serial line to convert unbalanced to balanced (and conversely). They are a specialty item and are expensive if purchased new.

21.6 Synchronization & Synchronous

How "Asynchronous" is Synchronized

Per EIA-232 there are only two states of the transmit (or receive) wire: mark (-12 V) or space (+12 V). There is no state of 0 V. Thus a sequence of 1-bits is transmitted by just a steady -12 V with no markers of any kind between bits. For the receiver to detect individual bits it must always have a clock signal which is in synchronization with the transmitter clock. Such clocks generate a "tick" in synchronization with each transmitted (or received) bit.

For asynchronous transmission, synchronization is achieved by framing each byte with a start bit and a stop bit (done by hardware). The receiver listens on the line for a start bit and when it detects one it starts its clock ticking. It uses this clock tick to time the reading of the next 7, 8 or 9 bits. (It actually is a little more complex than this since several samples of a bit are often taken and this requires additional timing ticks.) Then the stop bit is read, the clock stops and the receiver waits for the next start bit. Thus async is actually synchronized during the reception of a single byte but there is no synchronization between one byte and the next byte.

Defining Asynchronous vs Synchronous

Asynchronous (async) means "not synchronous". In practice, an async signal is what the async serial port sends and receives which is a stream of bytes each delimited by a start and stop bit. Synchronous (sync) is most everything else. But this doesn't explain the basic concepts.

In theory, synchronous means that bytes are sent out at a constant rate one after another in step with a clock signal tick. There is often a separate wire or channel for sending the clock signal. Asynchronous bytes may be sent out erratically with various time intervals between bytes (like someone typing characters at a keyboard).

There are borderline situations that need to be classified as either sync or async. The async serial port often sends out bytes in a steady stream which would make this a synchronous case but since they still have the start/stop bits (which makes it possible to send them out erratically) its called async. Another case is where data bytes (without any start-stop bits) are put into packets with possible erratic spacing

between one packet and the next. This is called sync since the bytes within each packet must be transmitted synchronously.

Synchronous Communication

Did you ever wonder what all the unused pins are for on a 25-pin connector for the serial port? Most of them are for use in synchronous communication which is seldom implemented on PC's. There are pins for sync timing signals as well as for a sync reverse channel. The EIA-232 spec provides for both sync and async but PC's use a UART (Universal Asynchronous Receiver/Transmitter) chip such as a 16450, 16550A, or 16650 and can't deal with sync. For sync one needs a USART chip or the equivalent where the "S" stands for Synchronous. Since sync is a niche market, a sync serial port is likely to be quite expensive.

Besides the sync part of the EIA-232, there are various other EIA synchronous standards. For EIA-232, 3 pins of the connector are reserved for clock (or timing) signals. Sometimes it's a modem's task to generate some timing signals making it impossible to use synchronous communications without a synchronous modem (or without a device called a "synchronous modem eliminator" which provides the timing signals).

Although few serial ports are sync, synchronous communication does often take place over telephone lines using modems which use V.42 error correction. This strips off the start/stop bits and puts the date bytes in packets resulting in synchronous operation over the phone line.

21.7 Block Mode

Intro to Block Mode

Block mode is seldom used with Linux. In block mode when one types at a terminal, the results are saved in the terminal memory and are not sent just yet to the host computer. Such terminals often have built-in editing capabilities. When the user presses certain keys (such as the send key) what has been saved in the terminal memory is sent to the host computer. Now the Linux editors vi and emacs, react instantly to pressing certain keys but in the above situation such keys will be pressed and nothing will happen since nothing is sent when a key is pressed. Thus using a block mode terminal will not allow the use of such interactive programs. The old IBM mainframe interface uses block mode (see IBM Terminals so many IBM terminals are block-mode only and also synchronous (see Section Synchronization & Synchronous).

Types of Block Modes, Forms

Block mode may itself have various sub-modes such as "page" (a page at a time) and "line" (a line at a time). Some terminals have both block transmission modes and conventional character modes and may be switched from one mode to another. Async terminals which have block modes include HP2622A, VT130, VT131, VT330, VT340, and Visual500. Many later model terminals can emulate block mode. Block modes may include a forms capability where the host computer sends a form to the terminal. Then the user fills it out and hits the send key which sends only the data in the form back to the host computer. The form itself (not the data) is displayed on the screen in protected fields which don't get transmitted to

the host.

Efficiency

Block mode takes a great deal of load off the host computer, especially if the host computer's hardware is designed for block modes (as IBM mainframes were). In character mode every character typed is sent immediately to the serial port and usually causes an interrupt at the host computer. The host that receives the byte must stop whatever it is doing and fetch that character from the port hardware. Even with UART's that have FIFO hardware buffers, the hardware timeout is normally only the transmission time of 3 bytes so that an interrupt is usually issued for every character typed.

In true block mode a long block of characters is received using only one interrupt. If block mode is used with conventional async FIFO serial ports, an interrupt is needed only every 14 bytes since they have 16-byte hardware buffers. Thus much of the load and overhead of interrupt handling is eliminated and the computer has more time to do other tasks when block mode is used.

A significant savings for block mode occurs if the terminal is connected to its host via a network. Without block mode, every character (byte) typed is sent in its own packet including all the overhead bytes (40 in a TCP/IP packet as used on the Internet). With block mode, a large number of characters are sent in a single packet.

21.8 EIA-232 (RS-232) Books

(Note: The first book covers much more than just EIA-232.)

- Black, Uyless D.: Physical Layer Interfaces & Protocols, IEEE Computer Society Press, Los Alamitos, CA, 1996.
- Campbell, Joe: The RS-232 Solution, 2nd ed., Sybex, 1982.
- Putnam, Byron W.: RS-232 Simplified, Prentice Hall, 1987.
- Seyer, Martin D.: RS-232 Made Easy, 2nd ed., Prentice Hall, 1991.

21.9 Serial Software

See Serial Software for Linux software for the serial ports including getty and port monitors.

---

22. Appendix D: Notes by Brand Name

Here are notes by brand name that were too specific to a certain terminal to be put elsewhere in this HOWTO. If you have some info to contribute on a certain terminal that is not covered elsewhere, it could go here. Various models and brands often have much in common which only need be written about in one place. It would be nice to have for each terminal model, a large set of links linking to the documentation relevant to that model (including escape codes). There are so many models of terminals that such a task would be quite onerous and I, David Lawyer (as of 1998), have no intention of attempting this. If terminal manufacturers would only make their manuals available on the net, then all this might not be needed.

22.1 CIT

CIT terminals were made in Japan in the 1980's for CIE Terminals. They ceased to be imported in the late 1980's. The company, CIE, still makes CItoh printers (in 1997) but has no parts for its abandoned terminals. Ernie at (714) 453-9555 in Irvine CA sells (in 1997) some parts for models 224, 326, etc. but has nothing for the 80 and 101. (The document you are now reading was written mostly on the 101e.)

To save the Set-Up parameters press ^S when in Set-Up mode. cit80: Contrast: knob on rear of terminal, cit101e: Brightness: use up/down arrow keys in Set-Up mode.

22.2 IBM Terminals

Don't confuse IBM terminals with IBM PC monitors. Many IBM terminals don't use ASCII but instead use an 8-bit EBCDIC code. It's claimed that in EBCDIC the bit order of transmission is reversed from normal with the high-order bit going first. The IBM mainframe communication standards are a type of synchronous communication in block mode (sends large packets of characters). Two standards are "BISYNC" and "SNA" (which includes networking standards). Many of their terminals connect with coax cable (RG62A/U) and naive persons may think the "BNC" connecter on the terminal is for ethernet (but it's not).

While this IBM system is actually more efficient than what is normally used in Linux, terminals meeting this IBM standard will not currently work with Linux. However, some IBM terminals are asynchronous ASCII terminals and should work with Linux on PC's. The numbers 31xx may work with the exception that 317x and 319x are not ASCII terminals. Before getting an IBM terminal, make sure there is a termcap (terminfo) for it. If their isn't, it likely will not work with Linux. Even if there is a terminfo, it may not work. For example, there is a termcap for 327x but the 3270 is an EBCDIC synchronous terminal.

The 3270 series includes the 3278 (late 1970's), 3279 with color and graphics, and the 3274 terminal controller (something like the 3174). They may be used for both BISYNC and SNA. The 3290 has a split screen (splits into quarters).

The synchronous IBM terminals don't connect directly to the IBM mainframe, but connect to a "terminal controller" (sometimes called "cluster controller" or "communication controller"). Some of these controllers can convert a synchronous signal to asynchronous so that in this case a synchronous terminal could indirectly connect to a Unix-like host computer via its serial port. But there is still a major problem and that is block transmission. See section Block Mode.

IBM 3153

It's claimed that the Aux port is DCE and uses a straight-thru cable.

22.3 Teletypes

These are antiques and represent the oldest terminals. They are like remotely controlled typewriters but are large and noisy. Made by the Teletype Corp., the first models were made in the 1920's and predate the computer by over 30 years. Early models used electro-mechanical relays and rotating distributors instead of electronics. Their Baudot code was only 5-bits per character as compared to 7-bit ASCII. See the book "Small Computer Systems Handbook" by Sol Libes, Hayden Books, 1978: pp. 138-141 ("Teletypes").

22.4 VT (DEC)

Digital Equipment Corporation made the famous VT series of terminals including the commonly emulated VT100. In 1995 they sold their terminal business to SunRiver which is now named Boundless Technologies. More information is available at Shuford's Website Information on current products is available from the Boundless website. See Terminal Info on the Internet.

VT220: Some have a BNC connector for video output (not for input). Sometimes people erroneously think this is for an ethernet connection.

VT520: Supports full DTR/DSR flow control.

22.5 Wyse

Wyse has some FAQ's for terminal numbers under 100 (such as WY60). See http://www.wyse.com/service/faq/wysetter.htm For the specs on more recent terminals see See http://www.wyse.com/terminal/.

Wyse 99-GT

Here is the setup Menus of the Wyse99GT (late 1980's). Note that TERM means "termination" (character) and not "terminal".

```
  WYSE 99-GT Terminal Set-Up as used at the University of CA, Irvine
                 by David Lawyer, April 1990

                          F1 DISP:
COLUMNS=80                LINES=24                  CELL SIZE=10 X 13
STATUS LINE=STANDARD      BACKGROUND=DARK           SCROLL SPEED=JUMP
SCREEN SAVER=OFF          CURSOR=BLINK BLOCK        DISPLAY CURSOR=ON
ATTRIBUTE=CHAR            END OF LINE WRAP=ON       AUTO SCROLL=ON
-------------------------------------------------------------------------
                          F2  GENERAL:
PERSONALITY=VT 100        ENHANCE=ON                FONT LOAD=OFF
COMM MODE=FULL DUPLEX     RCVD CR=CR                SEND ACK=ON
```

```
RESTORE TABS=ON          ANSWERBACK MODE=OFF     ANSWERBACK CONCEAL=OFF
WIDTH CHANGE CLEAR=OFF   MONITOR=OFF             TEST=OFF
-----------------------------------------------------------------------------
                         F3 KEYBRD:
KEYCLICK=OFF             KEYLOCK=CAPS            KEY REPEAT=ON
RETURN=CR                ENTER=CR                FUNCT KEY=HOLD
XMT LIMIT=NONE           FKEY XMT LIMIT=NONE     BREAK=170MS
LANGUAGE=US              MARGIN BELL=OFF         PRINTER RCV=OFF
-----------------------------------------------------------------------------
                         F4 COMM:
DATA/PRINTER=AUX/MODEM     MDM RCV BAUD RATE=9600  MDM XMT BAUD RATE=9600
MDM DATA/STOP BITS=8/1     MDM RCV HNDSHAKE=NONE   MDM XMT HNDSHAKE=NONE
MDM PARITY=NONE            AUX BAUD RATE=9600      AUX DATA/STOP BITS=8/1
AUX RCV HNDSHAKE=NONE      AUX XMT HNDSHAKE=NONE   AUX PARITY=NONE
(There is a main port (Modem=MDM) and an Auxiliary Port (AUX)
-----------------------------------------------------------------------------
                         F5 MISC 1:
WARNING BELL=ON          FKEY LOCK=OFF           FEATURE LOCK=ON
KEYPAD=NUMERIC           DEL=DEL/CAN             XFER TERM=EOS
CURSOR KEYS=NORMAL       MARGIN CTRL=0           DEL FOR LOW Y=ON
GIN TERM=CR              CHAR MODE=MULTINATIONAL
-----------------------------------------------------------------------------
                         F6 MISC 2:
LOCAL=OFF                SEND=ALL                PRINT=NATIONAL
PORT=EIA DATA            SEND AREA=SCREEN        PRINT AREA=SCREEN
DISCONNECT=60 MSEC       SEND TERM=NONE          PRINT TERM=NONE
PRINT MODE=NORMAL        VT100 ID=VT100          POUND=US
-----------------------------------------------------------------------------
F7 TABS: You should see several "T" characters spaced 8 dots apart.
    If you don't, hit backspace.
F8 F/KEYS: Normally you will see no definitions for the Function Keys
    here (unless someone has set them up and saved them).  This means that
    they will normally generate their default settings (not displayed here).
    <ctrl><F5> shows the "user defined definition" of the F5 key, etc.
F9 A/BACK: Normally not defined: ANSWERBACK =
F10 EXIT: Selecting "DEFAULT ALL" will make the factory default settings
    the default.
```

HINTS on use of WY-99GT User's Guide: Note that much that is missing from this Guide may be found in the WY-99GT Programmer's Guide. The VT100 emulation (personality) is known as ANSI and uses ANSI key codes per p. A-10+ even though the keyboard may be ASCII. A sub-heading on p. A-13 "ASCII Keyboard" also pertains to VT100 because it has an "ANSI KEY ..." super-heading a few pages previously. But not all ASCII keyboard headings pertain to VT100 since they may fall under a non-ANSI personality super-heading which may found be a few pages previously. Appendix H is the "ANSI Command Guide" except for the VT52 (ANSI) personality which is found in Appendix G.

Wyse 150

When exiting set-up using F12, hitting space changes "no" to "yes" to save the set-up. The sentence to the left of this no/yes is about "vertical alignment" and has nothing to do with this no/yes for saving the set-up (confusing menu design).

END OF Text-Terminal-HOWTO

The Linux Thai HOWTO

Poonlap Veeratanabutr, poon-v@fedu.uec.ac.jp

v0.4, 4 August 1998

This document describes how to use Thai language with Linux. This will cover setting Thai fonts, Thai keyboard and some Thai applications.

1. Introduction

2. Thai Input and Output

- 2.1 Linux console
- 2.2 X Window system

3. Applications with Thai language

- 3.1 Some X applications and Thai language
- 3.2 Printing Thai document
- 3.3 X Application Resources
- 3.4 Thai Extension for Linux (TE)

4. References and FTP sites

- 4.1 Other documents of relevance
- 4.2 Thai related stuffs
- 4.3 FTP and Web sites

5. Acknowledgments and Copyright

1. Introduction

It's about one year that I didn't update this document. There were a lot of movement in Thai computing and using Linux in Thailand. For example, Linux boxes are used as server in many schools in Thailand.

The purpose of this document is to show how to set your Linux to use Thai language. I use Linux RedHat 5.0 as I wrote this document, so directories which I mention in this document may be different from other distribution.

First I would like to talk about Thai standard character set. Thai standard character set is TIS-620. There are also other Thai standard character sets such as ISO-IR-166, CP874, etc. Please see http://www.inet.co.th/cyberclub/trin/thairef/ for further information about Thai standard character set. TIS-620 is 8-bit character set. It has the same range as ISO-8859-1, so we can use apllications that support ISO-8859-1 character set also. Although we can use Thai language with apllications that support ISO-8859-1 character set, but it does not mean those applications support Thai language.

Thai characters are different from English characters. There is a variation of position, normal position, character can be on other character, character can be under other character. There is no space between words. These are some problems in developing Thai supported application.

You can find the lastest version of Thai-HOWTO document from http://www.fedu.uec.ac.jp/ZzzThai/Linux. Your comment is welcome.

2. Thai Input and Output

2.1 Linux console

Thai characters do not display properly in Linux console. If you mainly use X window, you may pass this section.

Thai fonts

You can obtain Linux Thai console fonts which created by Mr. Phaisarn Techajaruwong from ZzzThai ftp site

For example, there is a font name "phaisarn.psf". Put it in `/usr/lib/kbd/consolefonts/` directory. Then, you can load the new font from Linux console by command

```
%setfont phaisarn.psf
```

Keyboard layout

You can set keyboard behavior as you like by using `loadkeys` command. Usually, you use `loadkeys` to load the file located in `/usr/lib/kbd/keytables`. You can create a US/Thai keyboard-map file and save it in this directory. Here is a sample.

```
keycode   0 =
keycode   1 = Escape           Escape
        alt     keycode   1 = Meta_Escape
keycode   2 = +one              exclam          +0x0e5          plus
        alt     keycode   2 = Meta_one
        alt shift keycode 2 = Meta_exclam
keycode   3 = +two              at              +slash          0x0f1
        control keycode   3 = nul
        control shift keycode 3 = nul
        alt     keycode   3 = Meta_two
        alt shift keycode 3 = Meta_at
keycode   4 = +three            numbersign      +underscore     0x0f2
        control keycode   4 = Escape
        alt     keycode   4 = Meta_three
        alt shift keycode 4 = Meta_numbersign
keycode   5 = +four             dollar          +0x0c0          0x0f3
        control keycode   5 = Control_backslash
        alt     keycode   5 = Meta_four
        alt shift keycode 5 = Meta_dollar
keycode   6 = +five             percent         +0x0b6          0x0f4
        control keycode   6 = Control_bracketright
        alt     keycode   6 = Meta_five
        alt shift keycode 6 = Meta_percent
keycode   7 = +six              asciicircum     +0x0d8          0x0d9
        control keycode   7 = Control_asciicircum
        alt     keycode   7 = Meta_six
        alt shift keycode 7 = Meta_asciicircum
keycode   8 = +seven            ampersand       +0x0d6          0x0df
        control keycode   8 = Control_underscore
        alt     keycode   8 = Meta_seven
keycode   9 = +eight            asterisk        +0x0a4          0x0f5
        control keycode   9 = Delete
```

TOC

```
        alt     keycode   9 = Meta_eight
keycode  10 = +nine             parenleft       +0x0b5          0x0d6
        alt     keycode  10 = Meta_nine
keycode  11 = +zero             parenright      +0x0a8          0x0f7
        alt     keycode  11 = Meta_zero
keycode  12 = +minus            underscore      +0x0a2          0x0f8
        control keycode  12 = Control_underscore
        control shift keycode 12 = Control_underscore
        alt     keycode  12 = Meta_minus
keycode  13 = +equal            plus            +0x0aa          0x0f9
        alt     keycode  13 = Meta_equal
keycode  14 = Delete           Delete           Delete          Delete
        alt     keycode  14 = Meta_Delete
keycode  15 = Tab              Tab              Tab             Tab
        alt     keycode  15 = Meta_Tab
keycode  16 = +q        Q       +0x0e6          0x0f0
keycode  17 = +w        W       +0x0e4          quotedbl
keycode  18 = +e        E       +0x0d3          0x0ae
keycode  19 = +r        R       +0x0be          0x0b1
keycode  20 = +t        T       +0x0d0          0x0b8
keycode  21 = +y        Y       +0x0d1          0x0ed
keycode  22 = +u        U       +0x0d5          0x0ea
keycode  23 = +i        I       +0x0c3          0x0b3
keycode  24 = +o        O       +0x0b9          0x0cf
keycode  25 = +p        P       +0x0c2          0x0ad
keycode  26 = +bracketleft      braceleft       +0x0ba          0x0b0
        control keycode  26 = Escape
        alt     keycode  26 = Meta_bracketleft
        alt shift keycode 26 = Meta_braceleft
keycode  27 = +bracketright     braceright      +0x0c5          comma
        control keycode  27 = Control_bracketright
        alt     keycode  27 = Meta_bracketright
        alt shift keycode 27 = Meta_braceright
keycode  28 = Return            Return          Return          Return
        alt     keycode  28 = 0x080d
keycode  29 = Control           Control         Control         Control
keycode  30 = +a        A       +0x0bf          0x0c4
keycode  31 = +s        S       +0x0cb          0x0a6
keycode  32 = +d        D       +0x0a1          0x0af
keycode  33 = +f        F       +0x0b4          0x0e2
keycode  34 = +g        G       +0x0e0          0x0ac
keycode  35 = +h        H       +0x0e9          0x0e7
keycode  36 = +j        J       +0x0e8          0x0eb
keycode  37 = +k        K       +0x0d2          0x0c9
keycode  38 = +l        L       +0x0ca          0x0c8
keycode  39 = +semicolon        colon           +0x0c7          0x0ab
        alt     keycode  39 = Meta_semicolon
keycode  40 = +apostrophe       quotedbl        +0x0a7          period
        control keycode  40 = Control_g
        alt     keycode  40 = Meta_apostrophe
keycode  41 = +grave            asciitilde      +minus          percent
        control keycode  41 = nul
        alt     keycode  41 = Meta_grave
keycode  42 = Shift            Shift            Shift           Shift
keycode  43 = +backslash        bar             +0x0a3          0x0a5
        control keycode  43 = Control_backslash
```

TOC

```
        alt     keycode  43 = Meta_backslash
        alt shift keycode 43 = Meta_bar
keycode  44 = +z        Z       +0x0bc          parenleft
keycode  45 = +x        X       +0x0bb          parenright
keycode  46 = +c        C       +0x0e1          0x0a9
keycode  47 = +v        V       +0x0cd          0x0ce
keycode  48 = +b        B       +0x0d4          0x0da
keycode  49 = +n        N       +0x0d7          0x0ec
keycode  50 = +m        M       +0x0b7          question
keycode  51 = +comma    less    +0x0c1          0x0b2
        alt     keycode  51 = Meta_comma
        alt shift keycode 51 = Meta_less
keycode  52 = +period           greater         +0x0e3          0x0cc
        alt     keycode  52 = Meta_period
        alt shift keycode 52 = Meta_greater
keycode  53 = +slash            question        +0x0bd          0x0c6
        control keycode  53 = Delete
        alt     keycode  53 = Meta_slash
keycode  54 = Shift             Shift           Shift           Shift
keycode  55 = KP_Multiply
keycode  56 = Alt               Alt             Alt             Alt
keycode  57 = space             space           space           space
        control keycode  57 = nul
        alt     keycode  57 = Meta_space
keycode  58 = Caps_Lock         Caps_Lock       Caps_Lock       Caps_Lock
keycode  59 = F1                F11             Console_13
        control keycode  59 = F1
        alt     keycode  59 = Console_1
        control alt     keycode  59 = Console_1
keycode  60 = F2                F12             Console_14
        control keycode  60 = F2
        alt     keycode  60 = Console_2
        control alt     keycode  60 = Console_2
keycode  61 = F3                F13             Console_15
        control keycode  61 = F3
        alt     keycode  61 = Console_3
        control alt     keycode  61 = Console_3
keycode  62 = F4                F14             Console_16
        control keycode  62 = F4
        alt     keycode  62 = Console_4
        control alt     keycode  62 = Console_4
keycode  63 = F5                F15             Console_17
        control keycode  63 = F5
        alt     keycode  63 = Console_5
        control alt     keycode  63 = Console_5
keycode  64 = F6                F16             Console_18
        control keycode  64 = F6
        alt     keycode  64 = Console_6
        control alt     keycode  64 = Console_6
keycode  65 = F7                F17             Console_19
        control keycode  65 = F7
        alt     keycode  65 = Console_7
        control alt     keycode  65 = Console_7
keycode  66 = F8                F18             Console_20
        control keycode  66 = F8
        alt     keycode  66 = Console_8
```

TOC

```
        control alt     keycode  66 = Console_8
keycode  67 = F9                F19              Console_21
        control keycode  67 = F9
        alt     keycode  67 = Console_9
        control alt     keycode  67 = Console_9
keycode  68 = F10               F20              Console_22
        control keycode  68 = F10
        alt     keycode  68 = Console_10
        control alt     keycode  68 = Console_10
keycode  69 = Num_Lock
keycode  70 = Scroll_Lock       Show_Memory      Show_Registers
        control keycode  70 = Show_State
        alt     keycode  70 = Scroll_Lock
keycode  71 = KP_7
        alt     keycode  71 = Ascii_7
keycode  72 = KP_8
        alt     keycode  72 = Ascii_8
keycode  73 = KP_9
        alt     keycode  73 = Ascii_9
keycode  74 = KP_Subtract
keycode  75 = KP_4
        alt     keycode  75 = Ascii_4
keycode  76 = KP_5
        alt     keycode  76 = Ascii_5
keycode  77 = KP_6
        alt     keycode  77 = Ascii_6
keycode  78 = KP_Add
keycode  79 = KP_1
        alt     keycode  79 = Ascii_1
keycode  80 = KP_2
        alt     keycode  80 = Ascii_2
keycode  81 = KP_3
        alt     keycode  81 = Ascii_3
keycode  82 = KP_0
        alt     keycode  82 = Ascii_0
keycode  83 = KP_Period
        altgr   control keycode  83 = Boot
        control alt     keycode  83 = Boot
keycode  84 = Last_Console
keycode  85 =
keycode  86 = less              greater          bar
        alt     keycode  86 = Meta_less
keycode  87 = F11               F11              Console_23
        control keycode  87 = F11
        alt     keycode  87 = Console_11
        control alt     keycode  87 = Console_11
keycode  88 = F12               F12              Console_24
        control keycode  88 = F12
        alt     keycode  88 = Console_12
        control alt     keycode  88 = Console_12
keycode  89 =
keycode  90 =
keycode  91 =
keycode  92 =
keycode  93 =
keycode  94 =
```

```
keycode  95 =
keycode  96 = KP_Enter
keycode  97 = Control
keycode  98 = KP_Divide
keycode  99 = Control_backslash
        control keycode  99 = Control_backslash
        alt     keycode  99 = Control_backslash
keycode 100 = AltGr_Lock
keycode 101 = Break
keycode 102 = Find
keycode 103 = Up
keycode 104 = Prior
        shift   keycode 104 = Scroll_Backward
keycode 105 = Left
        alt     keycode 105 = Decr_Console
keycode 106 = Right
        alt     keycode 106 = Incr_Console
keycode 107 = Select
keycode 108 = Down
keycode 109 = Next
        shift   keycode 109 = Scroll_Forward
keycode 110 = Insert
keycode 111 = Remove
        altgr   control keycode 111 = Boot
        control alt     keycode 111 = Boot
keycode 112 =
keycode 113 =
keycode 114 =
keycode 115 =
keycode 116 =
keycode 117 =
keycode 118 =
keycode 119 =
keycode 120 =
keycode 121 =
keycode 122 =
keycode 123 =
keycode 124 =
keycode 125 =
keycode 126 =
keycode 127 =
string F1 = "\033[[A"
string F2 = "\033[[B"
string F3 = "\033[[C"
string F4 = "\033[[D"
string F5 = "\033[[E"
string F6 = "\033[17~"
string F7 = "\033[18~"
string F8 = "\033[19~"
string F9 = "\033[20~"
string F10 = "\033[21~"
string F11 = "\033[23~"
string F12 = "\033[24~"
string F13 = "\033[25~"
string F14 = "\033[26~"
string F15 = "\033[28~"
```

```
string F16 = "\033[29~"
string F17 = "\033[31~"
string F18 = "\033[32~"
string F19 = "\033[33~"
string F20 = "\033[34~"
string Find = "\033[1~"
string Insert = "\033[2~"
string Remove = "\033[3~"
string Select = "\033[4~"
string Prior = "\033[5~"
string Next = "\033[6~"
string F21 = ""
string F22 = ""
string F23 = ""
string F24 = ""
string F25 = ""
string F26 = ""
```

Suppose you save this file as `thai.map`. From Linux console, use command `loadkeys` to load `thai.map`.

```
%loadkeys thai.map
```

You can switch to Thai keyboard by pressing the right Alt key. If you want to switch the keyboard back, press the right Alt key again.

2.2 X Window system

Thai fonts

You can obtain Thai fonts in bdf format or pcf format from internet. You can also use scalable fonts such as Type1 or TrueType fonts. But I will not describe about these.

Installing Thai fonts

You must log in as root. Let's put Thai fonts in `/usr/X11R6/lib/X11/fonts/misc/`, this is a default font path. Change directory to `/usr/X11R6/lib/X11/fonts/misc/` and run command

```
%mkfontdir
%xset fp rehash
```

If you put Thai fonts in different directory, you must use `xset` to add the new font path. Please see man-page for further information. You can check the new fonts by running command `xlsfonts` and see whether there are Thai fonts or not. If you can not see any Thai fonts from this command, you may need to restart X window.

Thai keyboard layoutThere are two ways to map Thai keyboard on X window, using X KeyboardExtension (XKB) and using xmodmap. Please select how you map Thai keyboard. Irecommend using XKB.

XKB and Thai keyboard layout.

Beginning with XFree86 3.1.2D, you can use the new X11R6.1 XKEYBOARD extension to manage the keyboard layout. This is very helpful.

During X server configuration with `xf86config` you will be asked about XKB, if you want to to set Thai keyboard layout for your system, say yes. There are a list of pre-configured keymaps. Choose Standard 101-key, Thai encoding.

`XF86Setup` is the graphical X server configuration utility for XFree86 X server. It is easier than traditional `xf86config`. You

can select a keyboard layout easily with this tool.

Ther are many choices of keyboard switch key to select. The default is LeftAlt+RightShift switch to Thai and LeftAlt+LeftShift switch to US. You can type Thai characters in any applications which support ISO-8859-1 character set, but don't forget to use Thai fonts with those applications too.

I found that pre-configured keymaps that came with XFree86-3.2 is not correct. You may not be able to type THO THUNG which located at " 5 key " . To fix this problem, you should add the line

```
        key <AE05> {     [], [   paragraph,      ocircumflex     ]       };
```

in the file `/usr/X11R6/lib/X11/xkb/symbols/th` as the example.

```
    ..................
        key <AE03> {        [], [       minus,      ograve          ]       };
        key <AE04> {        [], [      Agrave,      oacute          ]       };
        key <AE05> {     [], [   paragraph,      ocircumflex     ]       };
        key <AE06> {        [], [    Ooblique,      Ugrave          ]       };
        key <AE07> {        [], [  Odiaeresis,      ssharp          ]       };
    ..................
```

You can not type SORUSI also. Please change the line from

```
        key <AC08> {        [], [      Ograve,      eacute          ]       };
```

to

```
        key <AC08> {        [], [      Ograve,      Eacute          ]       };
```

Note that eacute is equal to MAITHO and Eacute is equal to SORUSI.

There are also XKB extension utilities such as `setxkbmap`, `xkbcomp`, etc. Please see man-page for more information. I recommend to use `xkbvled`. The leds will be on when you are using Thai keyboard so you can know your keyboard's status.

The following is part of `XF86Config` file about keyboard section. If you want to configure the keyboard by hand, change the content of `/usr/X11R6/lib/X11/XF86Config` as an example below. This configuration uses the default toggle key.

```
    Section "Keyboard"
       Protocol        "Standard"
       AutoRepeat      500 5
       LeftAlt         Meta
       RightAlt        Meta
       ScrollLock      Compose
       RightCtl        Control
    #  XkbDisable
       XkbKeycodes     "xfree86"
       XkbTypes        "default"
       XkbCompat       "default"
       XkbSymbols      "us(pc101)"
       XkbGeometry     "pc"
       XkbRules        "xfree86"
       XkbModel        "pc101"
       XkbLayout       "th"
    EndSection
```

If you use XKB extension, Thai keyboard mapping with `xmodmap` may not work. See `XF86Config` man-page for mor information.

Thai keyboard layout with xmodmap

You can use the utility `xmodmap` to map Thai keyboard. Normally `xmodmap` is used to load a keyboard configured file. For most Linux distributions, when you start X window with `startx`, X server will find `.Xmodmap` in `/usr/X11R6/lib/X11/xinit/` first. If `.Xmodmap` does not exist, X server will find `.Xmodmap` in your home directory. Please see the content of `/usr/X11R6/lib/X11/xinit/xinitrc`.

The following is the sample of `.Xmodmap` for Thai Kedmanee keyboard layout.

```
!
! Linux/XFree86 Thai Kedmanee layout (based on US keyboard)
! Use ScrollLock to switch to Thai keyboard.
! This file will work with XFree86 only.
!

keycode 0x09 =  Escape
keycode 0x43 =  F1
keycode 0x44 =  F2
keycode 0x45 =  F3
keycode 0x46 =  F4
keycode 0x47 =  F5
keycode 0x48 =  F6
keycode 0x49 =  F7
keycode 0x4A =  F8
keycode 0x4B =  F9
keycode 0x4C =  F10
keycode 0x5F =  F11
keycode 0x60 =  F12
keycode 0x6F =  Print
keycode 0x4E =  Mode_switch     XF86ModeLock
keycode 0x6E =  Pause
keycode 0x31 =  grave           asciitilde      minus           percent
keycode 0x0A =  1               exclam          0x0e5           plus
keycode 0x0B =  2               at              slash           0x0f1
keycode 0x0C =  3               numbersign      underscore      0x0f2
keycode 0x0D =  4               dollar          0x0c0           0x0f3
keycode 0x0E =  5               percent         0x0b6           0x0f4
keycode 0x0F =  6               asciicircum     0x0d8           0x0d9
keycode 0x10 =  7               ampersand       0x0d6           0x0df
keycode 0x11 =  8               asterisk        0x0a4           0x0f5
keycode 0x12 =  9               parenleft       0x0b5           0x0f6
keycode 0x13 =  0               parenright      0x0a8           0x0f7
keycode 0x14 =  minus           underscore      0x0a2           0x0f8
keycode 0x15 =  equal           plus            0x0aa           0x0f9
keycode 0x33 =  backslash       bar             0x0a3           0x0a5
keycode 0x16 =  BackSpace
keycode 0x6A =  Insert
keycode 0x61 =  Home
keycode 0x63 =  Prior
keycode 0x4D =  Num_Lock
keycode 0x70 =  KP_Divide
keycode 0x3F =  KP_Multiply
keycode 0x52 =  KP_Subtract
keycode 0x17 =  Tab
keycode 0x18 =  q               Q               0x0e6           0x0f0
keycode 0x19 =  w               W               0x0e4           quotedbl
```

TOC

```
keycode 0x1A =  e               E               0x0d3           0x0ae
keycode 0x1B =  r               R               0x0be           0x0b1
keycode 0x1C =  t               T               0x0d0           0x0b8
keycode 0x1D =  y               Y               0x0d1           0x0ed
keycode 0x1E =  u               U               0x0d5           0x0ea
keycode 0x1F =  i               I               0x0c3           0x0b3
keycode 0x20 =  o               O               0x0b9           0x0cf
keycode 0x21 =  p               P               0x0c2           0x0ad
keycode 0x22 =  bracketleft     braceleft       0x0ba           0x0b0
keycode 0x23 =  bracketright    braceright      0x0c5           comma
keycode 0x24 =  Return
keycode 0x6B =  Delete
keycode 0x67 =  End
keycode 0x69 =  Next
keycode 0x4F =  KP_7
keycode 0x50 =  KP_8
keycode 0x51 =  KP_9
keycode 0x56 =  KP_Add
keycode 0x42 =  Caps_Lock
keycode 0x26 =  a               A               0x0bf           0x0c4
keycode 0x27 =  s               S               0x0cb           0c0a6
keycode 0x28 =  d               D               0x0a1           0x0af
keycode 0x29 =  f               F               0x0b4           0x0e2
keycode 0x2A =  g               G               0x0e0           0x0ac
keycode 0x2B =  h               H               0x0e9           0x0e7
keycode 0x2C =  j               J               0x0e8           0x0eb
keycode 0x2D =  k               K               0x0d2           0x0c9
keycode 0x2E =  l               L               0x0ca           0x0c8
keycode 0x2F =  semicolon       colon           0x0c7           0x0ab
keycode 0x30 =  apostrophe      quotedbl        0x0a7           period
keycode 0x53 =  KP_4
keycode 0x54 =  KP_5
keycode 0x55 =  KP_6
keycode 0x32 =  Shift_L
keycode 0x34 =  z               Z               0x0bc           parenleft
keycode 0x35 =  x               X               0x0bb           parenright
keycode 0x36 =  c               C               0x0e1           0x0a9
keycode 0x37 =  v               V               0x0cd           0x0ce
keycode 0x38 =  b               B               0x0d4           0x0da
keycode 0x39 =  n               N               0x0d7           0x0ec
keycode 0x3A =  m               M               0x0b7           question
keycode 0x3B =  comma           less            0x0c1           0x0b2
keycode 0x3C =  period          greater         0x0e3           0x0cc
keycode 0x3D =  slash           question        0x0bd           0x0c6
keycode 0x3E =  Shift_R
keycode 0x62 =  Up
keycode 0x57 =  KP_1
keycode 0x58 =  KP_2
keycode 0x59 =  KP_3
keycode 0x6C =  KP_Enter
keycode 0x25 =  Control_L
keycode 0x40 =  Alt_L           Meta_L
keycode 0x41 =  space
keycode 0x71 =  Alt_R           Meta_R
keycode 0x6D =  Control_R
keycode 0x64 =  Left
```

```
keycode 0x68 =  Down
keycode 0x66 =  Right
keycode 0x5A =  KP_0
keycode 0x5B =  KP_Decimal

clear Shift
clear Lock
clear Control
clear Mod1
clear Mod2
clear Mod3
clear Mod4
clear Mod5

add    Shift   = Shift_L Shift_R
add    Lock    = Caps_Lock
add    Control = Control_L Control_R
add    Mod1    = Alt_L Alt_R
add    Mod2    = Mode_switch
```

Just put `.Xmodmap` in your home directory will be OK. When you start X window, X server will load this file.

You can also load `.Xmodmap` from command line.

```
%xmodmap ~/.Xmodmap
```

In above `.Xmodmap` file, US/Thai switch key is assigned to keycode 0x4E (78), Scroll Lock key, with the statement

---

```
keycode 0x4E =  Mode_switch     XF86ModeLock
```

`XF86ModeLock` is the special keysym for XFree86 X server. If you don't add this keysym, you have to hold the scroll lock key while you are typing Thai characters. Note that if you use commercial X server, some keycodes are different. You may have to map keyboard by yourself. See man-pages of `X` and `xev` for further information.

Note: If you are using XFree86 version 3.1.2D or later, you need to add the line `XkbDisable` in keyboard section of `XF86Config` file. You may config the keyboard section like the following sample.

```
Section "Keyboard"
   Protocol        "Standard"
   AutoRepeat      500 5
   LeftAlt         Meta
   RightAlt        ModeShift
   ScrollLock      ModeLock
   RightCtl        Compose
   XkbDisable
EndSection
```

3. Applications with Thai language

This is the tricky part. Most applications support ISO-8859-1 character set. For example, emacs can display ISO-8859-1 character. If we set emacs to display ISO-8859-1 and use Thai font, you can edit Thai document with emacs. But this is not a good policy. You should avoid using this trick as possible. What we need is Thai locale or Thai supported applications to manage these things.

To make X window application displays Thai font, you should run the application with `-fn` option. For example,

```
%xedit -fn thai8x16
```

Note that `thai8x16` is just a one of Thai font names. You can see all available fonts by command `xlfonts`. If you don't want to fill `-fn` option every time you run application. You should set Thai font in your `~/.Xdefaults` or `~/.Xresources` like this

```
XTerm*font:     thai8x16
```

3.1 Some X applications and Thai language

txterm

`txterm` is Thai version of `xterm`. There are several programs running under `xterm` such as shell, `pine`, `vi`, `less`, etc. We can type Thai characters without any problems with `txterm`. `Txterm` also provides its own Thai input method by pressing " F1 " key. `Txterm` will use fonts `thai9x13` as default Thai font. You can change this by add `-fn` option.

You can get `txterm` from Thaigate or ZzzThai.

`bash shell:`

Normally, shell accepts only ASCII character set. To type Thai characters in shell command line, you should set environment `LC_CTYPE` to `iso_8859_1`.

I don't set `LC_CTYPE` environment variable to iso_8859_1 because this environment variable will effect other applications too. With bash shell, you can specify which environment variable to be passed to the application. For example, I can make a fake Thai X terminal with this syntax.

```
LC_CTYPE=iso_8859_1 xterm -fn thai8x16
```

This xterm display Thai characters well, but not good for typing Thai characters. I strongly recommend you to use `txterm`.

`ls:`

If you name a filename in Thai. Issue the command as

```
ls -N
```

You may set `alias` in `~/.bashrc` or `~/.cshrc`, so you can type `ls` without option. If you don't use

`ls` with `-N` option, you may see Thai filename as ?????.

Emacs, Mule

Mule stands for " Multilingual Enhancement to GNU Emacs " . It has the same functions as emacs and supports many languages. `Mule` provides its own input method, so you don't need any configuration for typing Thai. You needs only Thai fonts for `mule` which you can get from, ZzzThai or Etl site. These Thai fonts are fixed width fonts.

You need some configuration for `mule`. Puts the following lines in your `.emacs`.

```
;;
;; Thai System, add in .emacs
;;
(set-file-coding-system-for-read '*tis620*)
(set-default-file-coding-system '*tis620*)
(set-display-coding-system '*tis620*)
(set-keyboard-coding-system '*tis620*)
(setq-deafault quail-current-package (assoc "thai" quail-package-alist))
```

Add the following lines in `.Xdefaults`.

```
!
! Emacs, Mule - Font menu
!
Emacs*FontSetList: thai14, thai16, thai24
Emacs*FontSet-thai14:\
        -etl-fixed-medium-r-normal--14-140-72-72-m-70-tis620.2529-1
Emacs*FontSet-thai16:\
        -etl-fixed-medium-r-normal--16-160-72-72-m-80-tis620.2529-1
Emacs*FontSet-thai24:\
        -etl-fixed-medium-r-normal--24-500-72-72-m-120-tis620.2529-1
```

When you hold shift key and press left mouse's button, you can select Thai fonts to display in `mule` window. To type Thai characters, press " Ctrl +] " . To type English, press " Ctrl +] " again.

You can get `mule` from ElectroTechnical Laboratory(ETL)

vi

Vi should be run on `txterm`.

pine

In the past, we could not send 8-bit characters through E-mail. Now, although mail transfer agent can handle 8-bit characters but some old mail transfer agent can not. We can send Thai e-mail by using e-mail application that supports MIME (Multipurpose Internet Mail Extensions) E-mail applications that support MIME are `pine`, `elm`, Netscape mail, etc.

Put the following definition in your `~/.pinerc` file:

```
character-set=ISO-8859-1
```

This can also be set via the `Setup` option in `pine` window. You can find it under `Config`. You can read Thai news from pine, too.

`Pine` should be run in `txterm`.

Netscape

If you have Thai fonts in your system. Just set Thai font from preference. Thai fonts will appear in User defined area. See http://www.fedu.uec.ac.jp/ZzzThai/unix for setting Thai language on Netscape.

Some movement about Thai Mozilla project at http://members.xoom.com/inThai/mozthai.html.

Ss, Simple thai word Separator

`ss` is a dictionary based Thai word separation program similar to `cttex`. It can be used to insert a configurable string between Thai words. It can also show words that cannot be found in the dictionary. More words can be added to the dictionary. Developed by Mr.Teera Kitticharеonpot.

We can use this program to insert `< WBR >` tag between Thai words in html file. Browser will display Thai homepage better than normal html document.

Xzthai, X keyboard configurator + simple editor

`Xzthai`, this is the Tcl/Tk application for mapping Thai keyboard on any keyboard with graphical user interface. Also provides simple editor and keyboard layout figure. It actually uses `xmodmap` program in background to map Thai keyboard. This may be useful for commercial X server and X server on UNIX.

3.2 Printing Thai document

`Thai2ps` is used to convert plain text file to postscript file.You can use ghostscript(`gs`) to print your Thai document. For better quality document, you have to use (La)TeX.

Latex and Thai language

Dr. Manop Wongsaisuwan first tried to use Thai language with `latex`. He wrote some perl script as filter for latex source code that contains Thai language. Then pass the result to `latex`. Mr. Vuthichai Ampornaramvech used this concept and wrote a program in C language, `cttex`, to handle this. It runs faster and makes Thai word segmentation based on dictionary. `Cttex` also fixes the position of Thai characters in word, so SARA and WANNAYUK will be placed in the beautiful position.

You can find Thai latex filter from http://thaigate.nacsis.ac.jp/files/ttex.html.

Latex's configuration for Thai language

You must have `latex` installed in your computer. First, download Thai postscript (Type1) fonts, tfm fonts and Thai style file. These fonts are needed by `Latex`. This is the list of files you should download.

```
tfm fonts:
    dbtt.tfm    dbttb.tfm   dbttbi.tfm  dbtti.tfm
postscript fonts:
    dbtt.pfa    dbttb.pfa   dbttbi.pfa  dbtti.pfa
style files:
    thai.sty sakka.sty
Thai Latex filter:
    cttex
Sample Latex file:
    ttex.ttex test.ttex
```

There is `latex`'s directory at `/usr/lib/texmf/texmf/` (RedHat 5.0). I will call `/usr/lib/texmf/texmf/` as " $texroot " . We will concentrate at `$texroot/texmf/` directory. In `$texroot/texmf/` directory, there are many files about tex's configuration. You have to edit files in `dvips, fonts, tex` subdirectories.

Add the following lines to `$texroot/texmf/dvips/misc/psfonts.map`

```
dbtt  DBThaiText <dbtt.pfa
dbttb DBThaiTextBold <dbttb.pfa
dbttbi DBThaiTextBoldItalic <dbttbi.pfa
dbtti DBThaiTextItalic <dbtti.pfa
```

Make a new directories and copy files to the appropriate directories.

```
%mkdir /usr/lib/texmf/texmf/fonts/tfm/public/thai
%mkdir /usr/lib/texmf/texmf/fonts/type1/public/thai
%mkdir /usr/lib/texmf/texmf/tex/generic/thai
%cp *.tfm /usr/lib/texmf/texmf/fonts/tfm/public/thai
%cp *.pfa /usr/lib/texmf/texmf/fonts/type1/public/thai
%cp *.sty /usr/lib/texmf/texmf/tex/generic/thai
```

Run `texhash` or `MakeTeXls-R`(in some system) to update `Tex` database.

```
%/usr/bin/texhash
texhash: updating /usr/lib/texmf/texmf/ls-R ...
texhash: Done.
```

Use Thai LaTeX filter

We can use `cttex` as filter like this,

```
%cttex < ttex.ttex > ttex.tex
C-TTeX $Revivsion: 1.15 $
Usage : cttex [cutcode] < infile > outfile
Usage : cutcode=0 forces operation in HTML mode.
Build-in dictionary size: 9945 words
 343
Done
%latex ttex.tex
```

```
...
%xdvi ttex.dvi
```

You can convert dvi file to postscript file by,

```
%dvips -o ttex.ps ttex.dvi
```

Finally, you can print ttex.ps by using `gs` or `lpr`. You must configure printer before printing. See man-pages of `printcap`, `gs`, `lpr` for more information.

Editing LaTex source file

For new `latex` user, `lyx` is helpful. But I recommend to use `mule` to edit Thai `latex` source file because `mule` supports Thai language and it is a powerful editor. You may take a look a Thai Latex tutorial.

3.3 X Application Resources

Because Xt based applications allow user to configure the applications by resources. We can make the menu or label to be Thai language.

For example, if you want `xman` to display Thai labels. You may add these lines in your `.Xdefaults`

```
......
!!  Xman section
Xman*Font:                          thai8x16
Xman*helpButton.Label:              ªèÇÂ
Xman*quitButton.Label:              ÍÍ¡
Xman*manpageButton.Label:           ¤ÙèÁ×Í¡ÒÃãªé
......
```

You can use the same idea to set window manager to be more Thai environment too.

3.4 Thai Extension for Linux (TE)

Thai Extension for Linux is a installation package comes with applicaions and Thai fonts. You don't have to configure Linux system and applications by yourself. Let TE do configuration task for you. After installation, you can use Thai language suddenly. Get TE from ftp://fedu.uec.ac.jp/pub/thai/UEC/ZzzThai/Software/Linux/

---

4. References and FTP sites

4.1 Other documents of relevance

The HOWTOs ought to be available from all mirrors of `sunsite.unc.edu`.

The Linux Danish/International HOWTO by Niels Kristian Bech Jensen

The Linux Cyrillic HOWTO by Alexander L. Belikoff

The Keystroke mini-HOWTO by Zenon Fortuna.

The Locales mini-HOWTO by Peeter Joot. (This one is mainly for developers.)

The ISO-8859-1 FAQ and Programming for Internationalization FAQ (plus much more) by Michael Gschwind is available from his homepage.

4.2 Thai related stuffs

" NACSIS R& D Thai Project Page " , http://thaigate.nacsis.ac.jp

- Information about Thai computing.
- Discussion groups in Thai language, such as thai-l (Thai Mailing list), Thai news, etc.
- Thai references and Thai softwares.
- Thai Latex filter.

" ZzzThai Project " , http://www.fedu.uec.ac.jp

- Most softwares and Thai fonts introduced here can download from ZzzThai.
- Describe how to use Thai with 3 main computer platforms, UNIX like, Windows and Mac.
- Linux information at http://www.fedu.uec.ac.jp/ZzzThai/Linux, TE, Thai LaTeX tutorial, etc.
- By The group of students at The University of Electro-Communications, Tokyo.

" Vuthichai's Page " , http://www.ctrl.titech.ac.jp:80/~vuthi/

- Information about Thai computing by Mr. Vuthichai Ampornaramveth.

" An annotated reference to the Thai implementations " , http://www.inet.co.th/cyberclub/trin/thairef/

- Information about Thai character standard.
- By Mr.Trin Tantsetthi.

" X window and Thai language " , http://members.xoom.com.cwg.x11th/

- By Mr.Rawat S.Pirom

TOC

" SchoolNet Internet Sever " , http://www.school.net.th/linux-sis/

- Using Linux in School, Thailand.
- By NECTEC (National Electronics and Computer Technology Center).

" Thai Open Source Development " , http://members.xoom.com/inThai

- Mozilla Thai enabling.
- Open source Thai softwares and Libraries.
- By Mr.Samphan Raruenrom

" Linux Thai Project " , http://www.geocities.com/SiliconValley/8302

- Information about Linux in Thai language.
- By Kaiwal Development Team.

" ThaiLinux unofficial Webboard " , http://lulu.mptc.eng.cmu.ac.th/HyperNews/get/ThaiLinux.html

- Questions and answers about Linux in Thai language.
- By Mr.Pruet Boonma

" Thai Linux installation project " , http://www.geocities.com/Tokyo/Bay/4521/

- Installation guide in Thai language

4.3 FTP and Web sites

Most softwares and Thai fonts which introdued in this howto.

- ftp://ftp.fedu.uec.ac.jp/pub/thai/UEC/ZzzThai/Software/Linux
- http://thaigate.nacsis.ac.jp/files/index.html
- http://www.nectec.or.th/pub/software/i18n/thai

`Mule`

- ftp://etlport.etl.go.jp/pub/mule

`Ss`

- http://members.xoom.com/theera/ss/

SunSite and mirrors. `doc/howto` has the above mentioned HOWTOs. `utils/nls` and subdirectories contain files related to National Language Support. Developers should take a look at `locale-tutorial-0.8.txt.gz`, `locale-pack-0.8.tar.gz` and `cat-pack.tar.gz`.

The GNU archives has the `recode` package for character table conversion, the `ABOUT-NLS` file and the `gettext` package for locale support of some GNU applications and (of course) the latest versions of GNU emacs.

---

5. Acknowledgments and Copyright

Some parts of this HOWTO comes from The Linux Danish/International HOWTO by Thomas Petersen, `petersen@post1.tele.dk` (the original author) and Niels Kristian Bech Jensen, `nkbj@image.dk`.

Thank you to Phaisarn Techajaruwong for building Thai fonts and valuable discussion.

Thank you to Thai students at The University of Electro-Communications and Mr. Vuthichai Ampornaramveth for every help.

This HOWTO is copyrighted by Poonlap Veeratanabutr, `poon-v@fedu.uec.ac.jp`. It is distributed as other Linux HOWTOs under the terms described below.

Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the authors would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at `linux-howto@sunsite.unc.edu` via email.

The Linux Tips HOWTO

Paul Anderson, paul@geeky1.ebtech.net

v3.6, June 1998

This HOWTO contains those hard to find hints and tweekings that make Linux a bit nicer.

1. Introduction

2. Short Tips

- 2.1 Handy Syslog Trick *Paul Anderson, Tips-HOWTO maintainer*
- 2.2 Script to view those compressed HOWTOs. *Didier Juges,* dj@destin.nfds.net
- 2.3 Is there enough free space??? *Hans Zoebelein,* zocki@goldfish.cube.net
- 2.4 Util to clean up your logfiles. *Paul Anderson, Tips-HOWTO Maintainer>*
- 2.5 Handy Script to Clean Up Corefiles. *Otto Hammersmith,*
- 2.6 Moving directories between filesystems. *Alan Cox,* A.Cox@swansea.ac.uk
- 2.7 Finding out which directories are the largest. *Mick Ghazey,*
- 2.8 The Linux Gazette
- 2.9 Pointer to patch for GNU Make 3.70 to change VPATH behavior.
- 2.10 How do I stop my system from fscking on each reboot? *Dale Lutz,* dal@wimsey.com
- 2.11 How to avoid fscks caused by "device busy" at reboot time. *Jon Tombs,* jon@gtex02.us.es
- 2.12 How to find the biggest files on your hard-drive.
- 2.13 How to print pages with a margin for hole punching. *Mike Dickey,* mdickey@thorplus.lib.purdue.edu
- 2.14 A way to search through trees of files for a particular regular expression.
- 2.15 A script for cleaning up after programs that create autosave and backup files.
- 2.16 How to find out what process is eating the most memory. *Simon Amor,*
- 2.17 Rigging vi for C programming, *Paul Anderson,*Tips-HOWTO Maintainer

- 2.18 Using ctags to ease programming.
- 2.19 Why does sendmail hang for 5 minutes on startup with RedHat? *Paul Anderson*,
- 2.20 How do I configure RedHat for using color-ls? *Paul Anderson*, paul@geeky1.ebtech.net
- 2.21 How do I find which library in /usr/lib holds a certain function? *Pawel Veselow*,
- 2.22 I compiled a small test program in C, but when I run it, I get no output!

3. Detailed Tips

- 3.1 Sharing swap partitions between Linux and Windows. *Tony Acero*, ace3@midway.uchicago.edu
- 3.2 Desperate Undelete. *Michael Hamilton*, michael@actrix.gen.nz
- 3.3 How to use the immutable flag. *Jim Dennis*, jadestar@rahul.net
- 3.4 A suggestion for where to put new stuff.
- 3.5 Converting all files in a directory to lowercase. *Justin Dossey*, dossey@ou.edu
- 3.6 How To Upgrade Sendmail
- 3.7 Some tips for new sysadmins.
- 3.8 How to configure xdm's chooser for host selection. *Arrigo Triulzi*, a.triulzi@ic.ac.uk

1. Introduction

Welcome to the **Linux Tips HOWTO**, a list of neato tricks and optimizations that make Linux more fun. All I have in here right now are tips off of the top of my head, and tips from the old Tips-HOWTO (Why take out decent tips, right?). So send all your favorite hints and tips to me so I can put them in the next Tips-HOWTO.

Paul Anderson *Maintainer--Linux TIPS HOWTO*

`panderso@ebtech.net`

2. Short Tips

2.1 Handy Syslog Trick *Paul Anderson, Tips-HOWTO maintainer*

Edit your /etc/syslog.conf, and put in the following line:

```
# Dump everything on tty8
*.*                                     /dev/tty8
```

One caveat: *REMEMBER TO USE TABS!* syslog doesn't like spaces...

2.2 Script to view those compressed HOWTOs. *Didier Juges,* `dj@destin.nfds.net`

From a newbie to another, here is a short script that eases looking for and viewing howto documents. My howto's are in /usr/doc/faq/howto/ and are gzipped. The file names are XXX-HOWTO.gz, XXX being the subject. I created the following script called "howto" in the /usr/local/sbin directory:

```
#!/bin/sh
if [ "$1" = "" ]; then
    ls /usr/doc/faq/howto | less
else
    gunzip -c /usr/doc/faq/howto/$1-HOWTO.gz | less
fi
```

When called without argument, it displays a directory of the available howto's. Then when entered with the first part of the file name (before the hyphen) as an argument, it unzips (keeping the original intact) then displays the document.

For instance, to view the Serial-HOWTO.gz document, enter:

$ howto Serial

2.3 Is there enough free space??? *Hans Zoebelein,* `zocki@goldfish.cube.net`

Here comes a short script which will check from time to time that there is enough free space available on anything which shows up in mount (disks, cdrom, floppy...)

If space runs out, a message is printed every X seconds to the screen and 1 mail message per filled device is fired up.

```
#!/bin/sh

#
# $Id: check_hdspace,v 1.18 1996/12/11 22:33:29 root Exp root $
#

#
# Since I got mysterious error messages during compile when
# tmp files filled up my disks, I wrote this to get a warning
# before disks are full.
```

TOC

```
#
# If this stuff saved your servers from exploding,
# send praising email to zocki@goldfish.cube.net.
# If your site burns down because of this, sorry but I
# warned you: no comps.
# If you really know how to handle sed, please forgive me :)
#

#
# Shoot and forget: Put 'check_hdspace &' in rc.local.
# Checks for free space on devices every $SLEEPTIME sec.
# You even might check your floppies or tape drives. :)
# If free space is below $MINFREE (kb), it will echo a warning
# and send one mail for each triggering device to $MAIL_TO_ME.
# If there is more free space than trigger limit again,
# mail action is also armed again.
#

# TODO: Different $MINFREE for each device.
# Free /*tmp dirs securely from old junk stuff if no more free space.


DEVICES='/dev/sda2 /dev/sda8 /dev/sda9'         # device; your put disks here
MINFREE=20480                                   # kb; below this do warning
SLEEPTIME=10                                    # sec; sleep between checks
MAIL_TO_ME='root@localhost'                     # fool; to whom mail warning


# ------- no changes needed below this line (hopefully :) -------

MINMB=0
ISFREE=0
MAILED=""
let MINMB=$MINFREE/1024         # yep, we are strict :)

while [ 1 ]; do
        DF="`/bin/df`"
                for DEVICE in $DEVICES ; do
                ISFREE=`echo $DF | sed s#.\*$DEVICE" "\*[0-9]\*""\*[0-9]\*" "\*## |
sed s#" ".\*##`

                if [ $ISFREE -le $MINFREE ] ; then
                        let ISMB=$ISFREE/1024
                        echo  "WARNING: $DEVICE only $ISMB mb free." >&2
                        #echo "more stuff here" >&2
                        echo -e "\a\a\a\a"

                        if [ -z  "`echo $MAILED | grep -w $DEVICE`" ] ; then
                                echo "WARNING: $DEVICE only $ISMB mb free.
(Trigger is set to $MINMB mb)" \
                                | mail -s "WARNING: $DEVICE only $ISMB mb free!"
$MAIL_TO_ME
                                MAILEDH="$MAILED $DEVICE"
                                MAILED=$MAILEDH
                                # put further action here like cleaning
                                # up */tmp dirs...
                        fi
                        elif [ -n  "`echo $MAILED | grep -w $DEVICE`" ] ; then
```

```
                                        # Remove mailed marker if enough disk space
                                        # again. So we are ready for new mailing action.
                                        MAILEDH="`echo $MAILED  | sed s#$DEVICE##`"
                                        MAILED=$MAILEDH
                                fi

                        done
                        sleep $SLEEPTIME

done
```

2.4 Util to clean up your logfiles. *Paul Anderson, Tips-HOWTO Maintainer>*

If you're like me, you have a list with 430 subscribers, plus 100+ messages per day coming in over UUCP. Well, what's a hacker to do with these huge logs? Install chklogs, that's what. Chklogs is written by Emilio Grimaldo, `grimaldo@panama.iaehv.nl`, and the current version 1.8 available from ftp.iaehv.nl:/pub/users/grimaldo/chklogs-1.8.tar.gz. It's pretty self explanatory to install(you will, of course, check out the info in the doc subdirectory). Once you've got it installed, add a crontab entry like this:

```
# Run chklogs at 9:00PM daily.
00 21 * * *       /usr/local/sbin/chklogs -m
```

While you're at it, mention to the author how nice a peice of software this is:)

2.5 Handy Script to Clean Up Corefiles. *Otto Hammersmith,* `ohammers@cu-online.com`

Create a file called rmcores(the author calls it handle-cores) with the following in it:

```
#!/bin/sh
USAGE="$0 <directory> <message-file>"

if [ $# != 2 ] ; then
        echo $USAGE
        exit
fi

 echo Deleting...
find $1 -name core -atime 7 -print -type f -exec rm {} \;

echo e-mailing
for name in `find $1 -name core -exec ls -l {} \; | cut -c16-24`
do
        echo $name
        cat $2 | mail $name
done
```

And have a cron job run it every so often.

TOC

2.6 Moving directories between filesystems. *Alan Cox,* `A.Cox@swansea.ac.uk`

Quick way to move an entire tree of files from one disk to another

```
(cd /source/directory && tar cf - . ) | (cd /dest/directory && tar xvfp -)
```

[Change from cd /source/directory; tar....etc. to prevent possibility of trashing directory in case of disaster. Thanks to Jim Dennis, jim@starshine.org, for letting me know. -Maint.]

2.7 Finding out which directories are the largest. *Mick Ghazey,* `mick@lowdown.com`

Ever wondered which directories are the biggest on your computer? Here's how to find out.

```
du -S | sort -n
```

2.8 The Linux Gazette

Kudos go to John Fisk, creator of the Linux Gazette. This is an excellent e-zine plus, it's **FREE!!!** Now what more could you ask? Check it out at:

```
http://www.linuxgazette.com
```

BTW, It turns out that (1) LG is now out on a monthly basis, and (2) John Fisk no longer maintains it, the fellows at SSC do.

2.9 Pointer to patch for GNU Make 3.70 to change VPATH behavior. *Ted Stern,* `stern@amath.washington.edu`

I don't know if many people have this problem, but there is a "feature" of GNU make version 3.70 that I don't like. It is that VPATH acts funny if you give it an absolute pathname. There is an extremely solid patch that fixes this, which you can get from Paul D. Smith `<psmith@wellfleet.com>`. He also posts the documentation and patch after every revision of GNU make on the newsgroup "gnu.utils.bug" Generally, I apply this patch and recompile gmake on every system I have access to.

2.10 How do I stop my system from fscking on each reboot? *Dale Lutz,* `dal@wimsey.com`

Q: How do I stop e2fsck from checking my disk every time I boot up.

A: When you rebuild the kernel, the filesystem is marked as 'dirty' and so your disk will be checked with each boot. The fix is to run:

rdev -R /zImage 1

This fixes the kernel so that it is no longer convinced that the filesystem is dirty.

Note: If using lilo, then add `read-only` *to your linux setup in your lilo config file (Usually /etc/lilo.conf)*

2.11 How to avoid fscks caused by "device busy" at reboot time. *Jon Tombs,* `jon@gtex02.us.es`

If you often get device busy errors on shutdown that leave the filesystem in need of an fsck upon reboot, here is a simple fix:

To `/etc/rc.d/init.d/halt` or `/etc/rc.d/rc.0`, add the line

```
mount -o remount,ro /mount.dir
```

for all your mounted filesystems except /, before the call to umount -a. This means if, for some reason, shutdown fails to kill all processes and umount the disks they will still be clean on reboot. Saves a lot of time at reboot for me.

2.12 How to find the biggest files on your hard-drive.

Simon Amor, `simon@foobar.co.uk`

```
ls -l | sort +4n
```

Or, for those of you really scrunched for space this takes awhile but works great:

```
cd /
ls -lR | sort +4n
```

2.13 How to print pages with a margin for hole punching. *Mike Dickey*, `mdickey@thorplus.lib.purdue.edu`

```
        #!/bin/sh
        # /usr/local/bin/print
        # a simple formatted printout, to enable someone to
        # 3-hole punch the output and put it in a binder

        cat $1 | pr -t -o 5 -w 85 | lpr
```

2.14 A way to search through trees of files for a particular regular expression. *Raul Deluth Miller*, `rockwell@nova.umd.edu`

I call this script 'forall'. Use it like this:

```
forall /usr/include grep -i ioctl
forall /usr/man grep ioctl
```

Here's forall:

```
#!/bin/sh
if [ 1 = `expr 2 \> $#` ]
then
        echo Usage: $0 dir cmd [optargs]
        exit 1
fi
dir=$1
shift
find $dir -type f -print | xargs "$@"
```

2.15 A script for cleaning up after programs that create autosave and backup files. *Barry Tolnas*, `tolnas@nestor.engr.utk.edu`

Here is a simple two-liner which recursively descends a directory hierarchy removing emacs auto-save (#) and backup (~) files, .o files, and TeX .log files. It also compresses .tex files and README files. I call it 'squeeze' on my system.

```
#!/bin/sh
#SQUEEZE removes unnecessary files and compresses .tex and README files
#By Barry tolnas, tolnas@sun1.engr.utk.edu
#
echo squeezing $PWD
find  $PWD \( -name \*~ -or -name \*.o -or -name \*.log -or -name \*\#\) -exec
rm -f {} \;
find $PWD \( -name \*.tex -or -name \*README\* -or -name \*readme\* \) -exec gzip -9
{} \;
```

2.16 How to find out what process is eating the most memory. *Simon Amor*, `simon@foobar.co.uk`

```
ps -aux | sort +4n
```

-OR-

```
ps -aux | sort +5n
```

2.17 Rigging vi for C programming, *Paul Anderson*,`Tips-HOWTO Maintainer`

I do a lot of C programming in my spare time, and I've taken the time to rig vi to be C friendly. Here's my .exrc:

```
set autoindent
set shiftwidth=4
set backspace=2
set ruler
```

What does this do? autoindent causes vi to automatically indent each line following the first one indented, shiftwidth sets the distance of ^T to 4 spaces, backspace sets the backspace mode, and ruler makes it display the line number. Remember, to go to a specific line number, say 20, use:

```
vi +20 myfile.c
```

2.18 Using ctags to ease programming.

Most hackers already have ctags on their computers, but don't use it. It can be very handy for editing specific functions. Suppose you have a function, in one of many source files in a directory for a program you're writing, and you want to edit this function for updates. We'll call this function foo(). You don't where it is in the source file, either. This is where ctags comes in handy. When run, ctags produces a file named tags in the current dir, which is a listing of all the functions, which files they're in and where they are in

said files. The tags file looks like this:

```
ActiveIconManager       iconmgr.c       /^void ActiveIconManager(active)$/
AddDefaultBindings      add_window.c    /^AddDefaultBindings ()$/
AddEndResize    resize.c        /^AddEndResize(tmp_win)$/
AddFuncButton   menus.c /^Bool AddFuncButton (num, cont, mods, func, menu, item)$/
AddFuncKey      menus.c /^Bool AddFuncKey (name, cont, mods, func, menu, win_name,
action)$/
AddIconManager  iconmgr.c       /^WList *AddIconManager(tmp_win)$/
AddIconRegion   icons.c /^AddIconRegion(geom, grav1, grav2, stepx, stepy)$/
AddStartResize  resize.c        /^AddStartResize(tmp_win, x, y, w, h)$/
AddToClientsList        workmgr.c       /^void AddToClientsList (workspace, client)$/
AddToList       list.c  /^AddToList(list_head, name, ptr)$/
```

To edit, say AddEndResize() in vim, run:

```
vim -t AddEndResize
```

This will bring the appropriate file up in the editor, with the cursor located at the beginning of the function.

2.19 Why does sendmail hang for 5 minutes on startup with RedHat? *Paul Anderson,* `paul@geeky1.ebtech.net`

This is a fairly common problem, almost to the point of being a FAQ. I don't know if RedHat has fixed this bug in their distribution, but you can repair it yourself. If you look in your /etc/hosts file, you will find it looks something like:

```
127.0.0.1               localhost       yourbox
```

When sendmail starts, it does a lookup on your hostname(in this example, yourbox). It then finds that the IP for yourbox is 127.0.0.1, sendmail doesn't like this, so it does the lookup again. It continues with this for a while until it eventually gives up and exits. Fixing the problem is extremely easy, edit your /etc/hosts file and change it to something like this:

```
127.0.0.1               localhost
10.56.142.1             yourbox
```

2.20 How do I configure RedHat for using color-ls? *Paul Anderson,* `paul@geeky1.ebtech.net`

RedHat's distribution comes with color-ls, however why they don't configure it for colour use by default is beyond me. Here's to fix it.

First, type eval `DIRCOLORS`

Next, alias ls='ls --color=auto'

And put the 'alias.....' in your /etc/bashrc

2.21 How do I find which library in /usr/lib holds a certain function? *Pawel Veselow,* `vps@unicorn.niimm.spb.su`

What if you're compiling and you've missed a library that needed linking in? All gcc reports are function names... Here's a simple command that'll find what you're looking for:

```
for i in *; do echo $i:;nm $i|grep tgetnum 2>/dev/null;done
```

Where tgetnum is the name of the function you're looking for.

2.22 I compiled a small test program in C, but when I run it, I get no output!

You probably compiled the program into a binary named test, didn't you? Linux has a program called test, which tests if a certain condition is true, it never produces any output on the screen. Instead of just typing test, try: ./test

3. Detailed Tips

3.1 Sharing swap partitions between Linux and Windows. *Tony Acero,* `ace3@midway.uchicago.edu`

1. Format the partition as a dos partition, and create the Windows swap file on it, but don't run windows yet. (You want to keep the swap file completely empty for now, so that it compresses well).
2. Boot linux and save the partition into a file. For example if the partition was /dev/hda8:

   ```
   dd if=/dev/hda8 of=/etc/dosswap
   ```
3. Compress the dosswap file; since it is virtually all 0's it will compress very well

   ```
   gzip -9 /etc/dosswap
   ```
4. Add the following to the /etc/rc file to prepare and install the swap space under Linux: *XXXXX is the number of blocks in the swap partition*

   ```
   mkswap /dev/hda8 XXXXX
   swapon -av
   ```

   Make sure you add an entry for the swap partition in your /etc/fstab file
5. If your init/reboot package supports /etc/brc or /sbin/brc add the following to /etc/brc, else do this by hand when you want to boot to dos|os/2 and you want to convert the swap partition back to the dos/windows version:

```
swapoff -av
zcat /etc/dosswap.gz | dd of=/dev/hda8 bs=1k count=100
```

Note that this only writes the first 100 blocks back to the partition. I've found empirically that this is sufficient

>> What are the pros and cons of doing this?

Pros: you save a substantial amount of disk space.

Cons: if step 5 is not automatic, you have to remember to do it by hand, and it slows the reboot process by a nanosecond :-)

3.2 Desperate Undelete. *Michael Hamilton,* `michael@actrix.gen.nz`

Here's a trick I've had to use a few times.

Desperate person's text file undelete.

If you accidentally remove a text file, for example, some email, or the results of a late night programming session, all may not be lost. If the file ever made it to disk, ie it was around for more than 30 seconds, its contents may still be in the disk partition.

You can use the grep command to search the raw disk partition for the contents of file.

For example, recently, I accidentally deleted a piece of email. So I immediately ceased any activity that could modify that partition: in this case I just refrained from saving any files or doing any compiles etc. On other occasions, I've actually gone to the trouble of bring the system down to single user mode, and unmounted the filesystem.

TOC

I then used the egrep command on the disk partition: in my case the email message was in /usr/local/home/michael/, so the output from df, I could see this was in /dev/hdb5

```
sputnik3:~ % df
  Filesystem          1024-blocks  Used Available Capacity Mounted on
  /dev/hda3               18621    9759     7901     55%   /
  /dev/hdb3              308852  258443    34458     88%   /usr
  /dev/hdb5              466896  407062    35720     92%   /usr/local

  sputnik3:~ % su
  Password:
  [michael@sputnik3 michael]# egrep -50 'ftp.+COL' /dev/hdb5 > /tmp/x
```

Now I'm ultra careful when fooling around with disk partitions, so I paused to make sure I understood the command syntax BEFORE pressing return. In this case the email contained the word 'ftp' followed by some text followed by the word 'COL'. The message was about 20 lines long, so I used -50 to get all the lines around the phrase. In the past I've used -3000 to make sure I got all the lines of some source code. I directed the output from the egrep to a different disk partition - this prevented it from over writing the message I was looking for.

I then used strings to help me inspect the output

```
strings /tmp/x | less
```

Sure enough the email was in there.

This method can't be relied on, all, or some, of the disk space may have already been re-used.

This trick is probably only useful on single user systems. On multi-users systems with high disk activity, the space you free'ed up may have already been reused. And most of use can't just rip the box out from under our users when ever we need to recover a file.

On my home system this trick has come in handy on about three occasions in the past few years - usually when I accidentally trash some of the days work. If what I'm working survives to a point where I feel I made significant progress, it get's backed up onto floppy, so I haven't needed this trick very often.

3.3 How to use the immutable flag. *Jim Dennis,* `jadestar@rahul.net`

Use the Immutable Flag

Right after you install and configure your system go through the /bin, /sbin/, /usr/bin, /usr/sbin and /usr/lib (and a few of the other usual suspects and make liberal use of the 'chattr +i command'. Also add that to the the kernel files in root. Now 'mkdir /etc/.dist/' copy everything from /etc/ on down (I do this in two steps using /tmp/etcdist.tar to avoid recursion) into that directory. (Optionally you can just create /etc/.dist.tar.gz) -- and mark that as immutable.

The reason for all of this is to limit the damage that you can do when logged in as root. You won't overwrite files with a stray redirection operator, and you won't make the system unusable with a stray space in an 'rm -fr' command (you might still do alot of damage to your data -- but your libs and bins will be safer.

This also makes a variety of security and denial of service exploits either impossible or more difficult (since many of them rely on overwriting a file through the actions of some SUID program that *isn't providing an arbitrary shell command*).

The only inconvenience of this is when building and doing your 'make install' on various sorts of system binaries. On the other hand it also prevents the 'make install' from over-writing the files. When you forget to read the Makefile and chattr -i the files that are to be overwritten (and the directories to which you want to add files) -- the make fails, you just use the chattr

TOC

command and rerun it. You can also take that opportunity to move your old bin's, libs, or whatever into a .old/ directory, rename or tar them or whatever.

3.4 A suggestion for where to put new stuff. *Jim Dennis*, `jadestar@rahul.net`

All new stuff starts under /usr/local! or /usr/local/`hostname`

If your distribution is one that leaves /usr/local empty then just create your /usr/local/src, /usr/local/bin etc and use that. If your distribution puts things in the /usr/local tree than you may want to 'mkdir /usr/local/`hostname`' and give the 'wheel' group +w to it (I also make it SUID and SGID to insure that each member of the wheel group can only mess with their own files thereunder, and that all files created will belong to the 'wheel' group.

Now discipline yourself to *ALWAYS! ALWAYS! ALWAYS!* put new packages under /usr/local/src/.from/$WHEREVER_I_GOT_IT/ (for the .tar or whatever files) and build them under /usr/local/src (or .../$HOSTNAME/src). Make sure that it installs under the local hierarchy. If it *absolutely must* be installed back in /bin or /usr/bin or somewhere else -- put a symlink from the local heirarchy to each element that when anywhere else.

The reason for this -- even though it's more work -- is that it helps isolate what has to be backed up and restored or reinstalled in the event of a full re-install from the distribution medio (usually CD these days). By using a /usr/local/.from directory you also keep an informal log of where your sources are coming from -- which helps when you're looking for new updates -- and may be critical when monitoring the security announcement lists.

One of my systems at home (the one I'm calling from) was put together before I adopted these policies for myself. I still don't "know" all the ways that it differs from the stock "as installed" system. This is despite the fact that I've done very little with my home system's configuration and I'm the *only* person who ever uses it.

By contrast the systems I've set up at work (when I was thrust into the role of system administrator there) have all been configured this way -- have been administered by many contractors and other MIS people, and have had a large number of upgrades and package installations. Nonetheless I have a very good idea which precise elements were put in *after* the initial installation and configuration.

3.5 Converting all files in a directory to lowercase. *Justin Dossey*, `dossey@ou.edu`

I noticed a few overly difficult or unnecessary procedures recommended in the 2c tips section of Issue 12. Since there is more than one, I'm sending it to you:

```
#!/bin/sh
        # lowerit
        # convert all file names in the current directory to lower case
        # only operates on plain files--does not change the name of directories
        # will ask for verification before overwriting an existing file
        for x in `ls`
          do
          if [ ! -f $x ]; then
            continue
            fi
          lc=`echo $x  | tr '[A-Z]' '[a-z]'`
          if [ $lc != $x ]; then
            mv -i $x $lc
          fi
          done
```

TOC

Wow. That's a long script. I wouldn't write a script to do that; instead, I would use this command:

```
for i in * ; do [ -f $i ] && mv -i $i `echo $i | tr '[A-Z]' '[a-z]'`;
done;
```

on the command line.

The contributor says he wrote the script how he did for understandability (see below).

On the next tip, this one about adding and removing users, Geoff is doing fine until that last step. Reboot? Boy, I hope he doesn't reboot every time he removes a user. All you have to do is the first two steps. What sort of processes would that user have going, anyway? An irc bot? Killing the processes with a simple

```
kill -9 `ps -aux |grep ^<username> |tr -s " " |cut -d " " -f2`
```

Example, username is foo

```
kill -9 `ps -aux |grep ^foo |tr -s " " |cut -d " " -f2`
```

That taken care of, let us move to the forgotten root password.

The solution given in the Gazette is the most universal one, but not the easiest one. With both LILO and loadlin, one may provide the boot parameter "single" to boot directly into the default shell with no login or password prompt. From there, one may change or remove any passwords before typing "init 3" to start multiuser mode. Number of reboots: 1 The other way Number of reboots: 2

Justin Dossey

3.6 How To Upgrade Sendmail *Paul Anderson,* `paul@geeky1.ebtech.net`

We're starting from raw, clean source. First, obtain the sendmail source code. I've d/led version 8.9.0, which is, as you will notice, bleeding edge. I grabbed it from ftp.sendmail.org:/pub/sendmail/sendmail.8.9.0.tar.gz

It's about 1Meg, and considering I'm running 8.7.6, I think it's worth the effort. If this works, you'll undoubtedly hear about it, elsewise I can't get the new HOWTO versions out without e-mail:)

Now, once you've got the source d/led, unpack it. It'll create a dir called `sendmail-8.9.0` in the current directory. Change into that directory, read the README and RELEASE_NOTES files(and be amazed at the updates they've done). Now, cd in src. This is where most of your work will be done.

A quick note: Sendmail is a small, powerful and well-written program. The sendmail binary itself compiled in less than 5 minutes on my 5x86 133 with 32Megs RAM! The entire compile and install(sans config) took under 15 minutes!

I don't normally run BIND on my system, so I found the lines:

```
# ifndef NAMED_BIND
#  define NAMED_BIND    1       /* use Berkeley Internet Domain Server */
# endif
```

---

and changed the 1 to a 0, ala:

```
# ifndef NAMED_BIND
#  define NAMED_BIND    0       /* use Berkeley Internet Domain Server */
```

```
# endif
```

On Debian 1.3.1, db.h is by default installed in /usr/include/db, instead of /usr/include, where sendmail hopes to find it. Change to the src, mailstats, makemap, praliases, rmail and smrsh directories and execute the following command:

```
./Build -I/usr/include/db
```

Once you've done that, cd .. and type make install. There! Sendmail version 8.9.0 should now be installed! This is, of course, assuming you already have your original configuration. For everything to work smoothly on my system, since I host free mailing lists for people using majordomo, I had to add the following to the beginning of my /etc/sendmail.cf:

```
O DontBlameSendmail=forwardfileinunsafedirpath, forwardfileinunsafedirpathsafe
```

Sendmail 8.9.0 is rather pedantic about directory and file permissions these days, and will complain about dirs and files in aliases or .forward files that are group or world writeable. While it's not a good idea to disable this pedantry, I am only running with a single person at the console and I felt it was okay to allow this minor security hole. YMMV.

3.7 Some tips for new sysadmins. *Jim Dennis,* `jadestar@rahul.net`

Create and maintain a /README.`hostname` and/or a /etc/README.`hostname` *[Or possibly /usr/local/etc/README.`hostname` -Maint.]*

Absolutely, from *day one* of administering a system take notes in an online log file. You might make "vi /README.$(hostname)" a line in root's /bash_logout. Another way to do this is to write an su or a sudo script that does something like:

```
function exit \
        { unset exit; exit; \
          cat ~/tmp/session.$(date +%y%m%d) \
          >> /README.$(hostname) && \
          vi /README.$(hostname)
          }
script -a ~/tmp/session.$(date +%y%m%d)
/bin/su.org -
```

(use the typescript command to create a session log and create a function to automate appending and updating the log).

I'll admit that I haven't implemented this automation of policy -- I've just relied on self-discipline so far. However I have been toying with the idea (even to the point of prototyping the scripts and shell functions as you see them). One thing that holds me back on this is the 'script' command itself. I think I'll have to grab the sources and add a couple of command line parameters (to pause/stop the script recording from the command line) before I commit to using this).

My last suggestion (for this round):

Root's path should consist of 'PATH= /bin'

That's it. Nothing else on root's path. Everything root does is provided by a symlink from /bin or by an alias or shell function, or is a script or binary in /bin, or is typed out with an explicit path.

This makes anyone running as root aware (sometimes painfully so) of how he or she is trusting binaries. The wise admin of a multi-user host will periodically look through his or here /bin and /.*history files to look for patterns and loopholes.

The really motivated admin will spot sequences that can be automated, places where sanity checks can be inserted, and tasks

for which "root" privileges should be temporarily eschewed (launching editors, MTA's and other large interactive programs with elaborate scripting features that *might* be embedded in transparent or data files -- like the infamous vi ./.exrc and emacs ./.emacs and the even more insidous $EXINIT and the embedded header/footer macros). Naturally those sorts of commands can be run with something like:

```
cp $data $some_users_home/tmp
su -c $origcommand $whatever_switches
cp $some_users_home/tmp $data
```

(...where the specifics depend on the command).

Mostly these last sorts of precautions are overboard for the home or "single" user workstation -- but they are very good policy the admin of a multi-user -- particular a publicly exposed system (like the one's at netcom).

3.8 How to configure xdm's chooser for host selection. *Arrigo Triulzi,* `a.triulzi@ic.ac.uk`

1. Edit the file that launches xdm most likely /etc/rc/rc.6 or /etc/rc.local) so that it contains the following lines in the xdm startup section.

```
/usr/bin/X11/xdm
exec /usr/bin/X11/X -indirect hostname
```

2. Edit /usr/lib/X11/xdm/Xservers and comment out the line which starts the server on the local machine (i.e. starting 0:)
3. Reboot the machine and you're home and away.

I add this because when I was, desperately, trying to set it up for my own subnet over here it took me about a week to suss out all the problems.

Caveat: with old SLS (1.1.1) for some reason you can leave a -nodaemon after the xdm line -- this does **NOT** work for later releases.

UMSDOS HOW-TO

Jacques Gelinas, jacques@solucorp.qc.ca

v1.1, 13 November 1995

Umsdos is a linux file system. It provide an alternative to the EXT2 file-system. Its main goal is to achieve easier coexistence with Ms-DOS data by sharing the same partition. This document explain first how to use Umsdos in different configuration, and later explain its operation and try to provide some information letting you decide if it is a good choice for you (see UMSDOS-WHY-TO at the end).

1. UMSDOS: Where is it ?

- 1.1 History
- 1.2 Availability
- 1.3 Distribution supporting it
- 1.4 Home site
- 1.5 Technical documentation
- 1.6 Who wrote it

2. Umsdos as your root partition

- 2.1 The pseudo-root concept.
- 2.2 Things to know about the pseudo-root

3. Different topics about the operation of *Umsdos*

- 3.1 Mount option
- 3.2 How to set defaults for the root
- 3.3 To swap or not to swap

TOC

4. How to boot a Umsdos system

- 4.1 Loadlin
- 4.2 From a floppy
- 4.3 LILO
- 4.4 How to defragment a *Umsdos* partition
- 4.5 Advance tricks

5. Basic principle

- 5.1 Introduction
- 5.2 *Umsdos* can replace the *Ms-DOS* file-system.
- 5.3 Directory promotion
- 5.4 How to promote: `/sbin/umssync`
- 5.5 Using `/sbin/umssync` at boot time
- 5.6 How to UN-promote
- 5.7 What about files created during a *DOS* session ?

6. Installation/UN-installation and some tricks

- 6.1 The pseudo-root `/mnt/linux`
- 6.2 Preparing `/mnt/linux`
- 6.3 Making sure `/mnt/linux` is correctly setup
- 6.4 Oops releasing pseudo root ...
- 6.5 How to UN-install a *Umsdos* system
- 6.6 Moving a *Umsdos* system to another *DOS* drive
- 6.7 About installing 50 *Umsdos* systems.

7. Setting a *Linux* section in a *DOS* partition

8. UMSDOS-WHY-TO

- 8.1 The goal of *Umsdos*
- 8.2 Who needs it

- 8.3 Performance issue

1. UMSDOS: Where is it ?

1.1 History

The *Umsdos* project was started in 1992 and made available to the net in January 1994 as a patch. It was included in the standard kernel distribution in July, starting with kernel 1.1.36.

Umsdos was early adopted in the *Slackware* distribution even before it was officially included in the official kernel.

Umsdos was improved starting at kernel 1.1.60. Its performance has been dramatically enhanced, especially for writing. Since 1.1.70 (around this), it is stable again.

A major bug was solve in *Linux* 1.2.2. This bug was causing some grief to users since the beginning (some file were silently renamed, giving the sad impression that they were deleted). Beware that *Slackware* 2.2 is still shipping release 1.2.1 of the kernel, so has this bug.

1.2 Availability

It is available as a patch for kernel 1.0.x. It is built-in for kernel 1.2. It can be compiled in or load as a module. Beware that for now, if you intend to load *umsdos* as a module, you must also use the Ms-DOS fs as a module. This come from a limitation in the module system (some symbols are only export when the drivers is installed as a module).

1.3 Distribution supporting it

So far, I think only *Slackware* does support it. I am surely wrong, so please send me info to correct this.

1.4 Home site

The home site for *Umsdos* is sunsite.unc.edu. Look in the directory `/pub/Linux/system/Filesystems/umsdos`.

1.5 Technical documentation

There is quite a lot of documentation about the internal of *Umsdos*. It is available both in `HTML` and text format at the same location as the utilities.

As far as I know, the `HTML` version is not available online on any web site. You must down-load it and "UN-tar" it and read it locally.

1.6 Who wrote it

Jacques Gelinas jacques@solucorp.qc.ca

2. Umsdos as your root partition

2.1 The pseudo-root concept.

With *Umsdos*, *Linux* can be installed in a standard DOS partition. *Linux* is then installed as a second (or third) OS in the partition. To avoid name collision (there is maybe a bin or tmp directory in the drive C: already), *Umsdos* use a smart trick: The pseudo-root.

All *Linux* files are installed in a DOS subdirectory called `linux`, generally `C: LINUX`. The normal *Linux/Unix* directory structure goes there. So you get

- `C:\LINUX\BIN`
- `C:\LINUX\ETC`
- `C:\LINUX\LIB`
- `C:\LINUX\ROOT`
- `C:\LINUX\SBIN`
- `C:\LINUX\TMP`
- `C:\LINUX\USR`
- `C:\LINUX\VAR`

When the *Umsdos* boot, it probes for the directory `linux` and then `/linux/etc`. If it exist, it activates the pseudo-root mode.

Mostly, the pseudo-root mode switch the root of the partition to `C:\\LINUX` giving the conventional *Unix* directory layout

- `/bin`
- `/etc`
- `/lib`
-

- `/root`
- `/sbin`
- `/tmp`
- `/usr`
- `/var`

To this list, it adds a new one called `DOS`. This one is a virtual directory.

2.2 Things to know about the pseudo-root

- This mode can only be triggered at boot time. There is no way to activate this by a mount command.
- This mechanism is purely a different view of a normal *Umsdos* file-system. This means that a partition normally used as a root partition can be normally mounted. There won't be any pseudo-root effect. For example, if you boot linux with a maintenance floppy and mount your normal root partition in `/mnt`, you will find all your linux directory in `/mnt/linux/bin`, `/mnt/linux/etc` and so on.

---

3. Different topics about the operation of *Umsdos*

3.1 Mount option

You can use the same mount option as for the Ms-DOS file system. The option conv= is questionable on a *Umsdos* system. I suggest to avoid it. Mostly the option you may want to look at are

- uid=
- gid=
- umask=

Just remember that *Umsdos* manage non promoted directory the same way as the *Ms-DOS* file system. The option above will apply globally to all non promoted directory. *uid* setup the default owner, *gid* setup the default group and *umask* setup the default permissions.

3.2 How to set defaults for the root

umssetup was created to provide at run time default ownership for the root partition. For other *Umsdos* partition, mount option may be used or umssetup. Storing mount option in /etc/fstab is the prefered way for non root partition. Here is an example. Put this in /etc/rc.d/rc.S.

```
/sbin/umssetup -u jack -g group -m 0755 /
```

3.3 To swap or not to swap

Using a swap file is generally slower than a swap partition. It is however much more flexible. You can setup a swap file in a *Umsdos* partition the same way you do it for any other *Linux* file systems. For example, to setup a 8 megabytes swap file in the root directory:

```
dd if=/dev/zero bs=1024k count=8 of=/swap
mkswap /swap 8192
sync
swapon /swap
```

Once done, you can put the following line in /etc/fstab

```
/swap   swap    swap    default
```

And the swap file will be activated at each boot (There is generally a "swapon -a" in `/etc/rc.d/rc.S`).

4. How to boot a Umsdos system

4.1 Loadlin

The package lodlin15.tgz available from sunsite.unc.edu in `/pub/Linux/system/Bootutils`. This utility is particularly suited to boot a *Umsdos* system. Generally all you need to do is

```
Boot DOS
C:>loadlinx zimage root=D:
```

where zimage is a normal kernel image (compressed) simply copied somewhere in the DOS drive. `D:` is the DOS drive where you have installed *Linux*.

4.2 From a floppy

Booting a *Umsdos* system from a floppy is not different from booting a *Ext2* system. You need a kernel zImage file properly initialize to locate your root *Umsdos* partition. This is generally achieved using the command `rdev`. The following sequence will initialize a zImage and put it on a floppy.

```
rdev zImage /dev/hda1
rdev -R zImage 0
dd if=zImage bs=8192 of=/dev/fd0
```

If this looks confusing, just format a boot-able DOS floppy and put the following component on it.

- loadlin.exe
- loadlinx.exe
- zimage

and setup the autoexec.bat like this

```
loadlinx zimage rw root=C:
```

4.3 LILO

LILO, the official *Linux* boot loader can also be used to boot a *Umsdos* system. I have no experience with it though. Since 1.1.60, it should work. Please email if you know something.

TOC

4.4 How to defragment a *Umsdos* partition

It can be done using any popular DOS tool. There is nothing particular about file produced by *Umsdos*. And *Umsdos* do not expect anything particular (directory layout, directory entry sequence, etc...) from the file system under it.

As far as I know, there is no *Linux* tool to achieve this.

4.5 Advance tricks

Umsdos rely on the `--linux-.---` which rely on the *DOS* directory. Some users may want to experiment a bit. The utility `udosctl` part of the umsdos_progs package (containing `umssync` and `umssetup`) allows basic directory operation (listing, deletion) independently on the `--linux-.---` and the *DOS* directory.

---

5. Basic principle

5.1 Introduction

Umsdos map *Linux* files directly to *Ms-DOS* files. This is a one for one translation. File content is not manipulated at all. *Umsdos* only works on names. For special files (links and devices for example), it introduces special management.

For each directory, there is a file named `--linux-.---`.

5.2 *Umsdos* can replace the *Ms-DOS* file-system.

Umsdos can be thought as a general purpose superset of the *Ms-DOS* file system of linux. In fact this capability or flexibility yields much confusion about *Umsdos*. Here is why. Try to mount a newly formatted *DOS* floppy like this.

```
mount -t umsdos /dev/fd0 /mnt
```

And do this,

```
ls / >/mnt/LONGFILENAME
ls -l /mnt
```

You will get the following result

```
-rwxr-xr-x   1 root     root          302 Apr 14 23:25 longfile
```

So far, it seems that the *Umsdos* file system does not do much more (in fact nothing at all) than the normal *Ms-DOS* file system of *Linux*.

???

5.3 Directory promotion

Pretty unimpressive so far. Here is the trick. Unless promoted a *DOS* directory will be managed the same way with *Umsdos* than the *Ms-DOS* file-system will. *Umsdos* use a special file in each subdirectory to achieve the translation between the extended capabilities (long name, ownership, etc...) of *Umsdos* and the limitation of the *DOS* file-system. This file is invisible to *Umsdos* users, but visible when you boot *DOS*. To avoid cluttering the *DOS* partition with those file (`--linux-.---`) uselessly, the file is now optional. If absent, *Umsdos* behave like *Ms-DOS*.

When a directory is promoted, any subsequent operation will be done with the full semantic normally available to *Unix* and *Linux* users. And all subdirectory created afterward will be silently promoted.

This feature allows you to logically organize your *DOS* partition into *DOS* stuff and *Linux* stuff. It is important to understand that those `--linux-.---` file do take some place (generally 2k per directory). *DOS* generally use large cluster (as big as 16k for a 500meg partition), so avoiding putting `--linux-.---` everywhere can save your day.

5.4 How to promote: `/sbin/umssync`

A directory can be promoted any time using `/sbin/umssync`. It can be used at any time. Promoting a directory do the following operation

- Create a `--linux-.---`.
- Establish a one to one relation between the `--linux-.---` and the current content of the directory.

`/sbin/umssync` maintain an existing `--linux-.---` file. It does not create it from scratch all the time. It simply add missing entries in it (Files created during a *DOS* session). It will also removed files which do not exist anymore in the *DOS* directory from the --*linux*-.---. `umssync` gets its name from that. It put --*linux*-.--- in sync with the underlying *DOS* directory.

5.5 Using `/sbin/umssync` at boot time

It is a good idea to place a call to `/sbin/umssync` at the end of your /etc/rc.d/rc.S if it's not there. The following command is adequate for most system:

```
/sbin/umssync -r99 -c -i+ /
```

The `-c` option prevent `umssync` from promoting directories. It will only update existing `--linux-.---`.

This command is useful if you access *Linux* directory during a *DOS* session. *Linux* has no efficient way to tell that a directory has been modified by *DOS* so *Umsdos* can't do a *umssync* operation as needed.

5.6 How to UN-promote

Remove the `--linux-.---` file using *DOS*. You will be sorry.

5.7 What about files created during a *DOS* session ?

Unless you use `umssync` on a directory where files have been added or removed by *DOS*, you will notice some problems:

- It won't crash the system nor it won't cause major problems, only annoyance :-)
- Files created by *DOS*.
  - They will be invisible in *Linux*.
  - When trying to create a file with the same name, you will get an error message stating that the file already exist.

    - This creates more confusion that real problem. It does not harm the file system.
- Files deleted by *DOS* won't cause problem. *Umsdos* will notice the absence at the first access. A message will be output (and generally written into `/var/adm/syslog`).

---

6. Installation/UN-installation and some tricks

The installation of a *Umsdos* is not much different from the installation of an ordinary (*Ext2 based*) *Linux* system.

There are two main differences.

6.1 The pseudo-root `/mnt/linux`

The normal steps for an installation are

1. Setting a partition with fdisk and formatting it.
2. Mounting it as /mnt relative to our installation root disk.
3. Copy all packages into `/mnt`.

With *Umsdos*, the step 1 is not required (wasn't it the goal of *Umsdos* not to reformat ?).

It is possible to install a *Umsdos* system just by copying all packages into `/mnt`. This will certainly work. But it will create a bunch of subdirectories into your *DOS* root directory (C:) and you won't like it. This is the reason all *Umsdos* installation use the pseudo-root. And this is the major difference between a normal *Ext2* installation and a *Umsdos* one: All files are copied into `/mnt/linux`.

6.2 Preparing `/mnt/linux`

`/mnt/linux` is not an ordinary directory. It has to be promoted so it will correctly handle *Linux* long file name and special files (links, device). The step required to setup `/mnt/linux` are:

1. `mkdir /mnt/linux`
2. `umssync /mnt/linux`

That's it!

6.3 Making sure `/mnt/linux` is correctly setup

Even if the setup of `/mnt/linux` is pretty simple, there are many installation package out there who get it wrong. How can ?

The biggest installation problem come from an incompatible `umssync` program. *Umsdos* has been update in linux 1.1.88 (Can't remember exactly) and a flaw was uncovered in `umssync`. To avoid confusion in the *Linux* community, it was decided to raise the compatibility level required for all *Umsdos* tools. Old version of the tools were simply rejected.

It sounds like many distribution did not update their `umssync` utility on the installation disk.

TOC

There are still many distribution like this out there. The net result is that the directory `/mnt/linux` not promoted at all and will truncate all long file name and will reject all special file.

It is possible to do a test very early during the installation to find out if something went wrong. Thanks to the pseudo console mechanism of *Linux*, you can do that without leaving the installation program. Do the following steps:

1. Press `Alt-F2` (`Alt` key at the same time as the `F2` key).
2. login as root.
3. `cd /mnt/linux` If this fail, you are trying this too early. A good time to do this is at the end of the packages selection.
4. `>TOTO`
5. `ls -l` You should see an empty file `TOTO` in uppercase. If you see it in lowercase, something went wrong. Try to do the `umssync` step again. `umssync` can be use over and over without problem. `umssync .` If there is no error message, try the `TOTO` test again. If `TOTO` appears fine, then all is OK. Something is strange in this installation, but you just save it. Continue
6. Press `Alt-F1` to get back to the installation screen.

If the test fail, the best fix is to get a newer installation root disk. You can generally fix this root disk by installing a newer version of `umssync`. This is not difficult but required a working *Linux* system. You simply have to mount the root disk floppy and replace the offending `umssync` with a new one.

6.4 Oops releasing pseudo root ...

Most *Umsdos* installation which fail, do this by printing this strange message. This is not a bug in *Umsdos* although the message looks strange. Here are the known causes.

- The most common one

  The *Slackware* installation try to setup a swap file very early during the installation. To do so, it asks you to select a partition (dos drive), then mount it and set the swap file.

  When installing a *Slackware* system, you must setup the target partition prior to install. This normally mounts the *DOS* partition on `/mnt`, creates the `/mnt/linux` directory and applies `umssync` on it.

  This is where most problems come from. Most user just forget the "setup target partition" step and go directly to the rest of the installation. Since `/mnt` is already mounted, this mistake goes unnotice. This means that `/mnt/linux` was not created properly (Not promoted). All special files and links and long names can't be created properly.
- Invalid umssync utility

  `/mnt/linux` was improperly setup-ed. Generally caused by an improper `umssync` utility on the installation root disk.
- Old bug in umsdos

  There was a bug in *Umsdos* prior to *Linux 1.2.2*. The pseudo-root mode would not activate

TOC

properly if the file `/etc/init` was missing. `init` is now located in `/sbin`. You can fix it getting a newer kernel. This is recommended because another bug was uncover and fixed in 1.

If you can't upgrade, do this

1. Boot from you installation disk.
2. Login as root.
3. `mount -t umsdos /dev/hdXX /mnt` where `/dev/hdXX` is your *DOS* partition.
4. `cd /mnt/linux/etc`
5. `ln -s ../sbin/init init`
6. `cd /`
7. `Ctrl-Alt-Del`
8. Boot your *Umsdos* normally.

Unfortunatly, the first two (Installation problems) produce a completly unusable installation. Uninstall it (See next section) and install again.

6.5 How to UN-install a *Umsdos* system

One neat thing about *Umsdos* and its pseudo-root mechanism, is that you can UN-install it without pain. You just boot *DOS* and recursively delete the `linux` directory. That's all. *Umsdos* requires no special drivers in the config.sys, nor it creates anything special outside of the `linux` directory.

6.6 Moving a *Umsdos* system to another *DOS* drive

This can be done from *Linux* or from *DOS*. You just have to copy recursively the `linux` directory from one drive to the other. After that you will have to adjust you boot mechanism (generally loadlin command) and the `/etc/fstab` file.

Umsdos can live on any *DOS* drive. There is no need to install it on the `C:` drive, nor it is important to have it on the first hard drive. It does not matter at all.

In fact, one may decide to have several *Umsdos* installations on different drive just to do experiments.

6.7 About installing 50 *Umsdos* systems.

How about installing a bunch of *Linux* systems in no time ?

Umsdos systems are living in a *DOS* world. You can take advantage of this if you wish to install *Linux* easily.

You can install and configure a *Umsdos* system at your site. When you are satisfied with the configuration and the different packages you have selected, you can boot *DOS* and copy the complete `linux` directory to your *DOS* file server. Then you go to other *DOS* station and simply copy the files on the network drive to the local drive. That's it. Only adjust the boot script (*Loadlinx*) and go.

With minimal adjustment (Host name, IP number), anyone will be able to install a *Linux* system in a matter of minute.

Interest readers may note that installing *Linux* systems by copying running system also works for any other *Linux* systems, including *Ext2* based one.

One beauty of *Linux* is that there is no hidden files which have to be install by magic installation program.

---

7. Setting a *Linux* section in a *DOS* partition

Umsdos has some use even for *Ext2* (Native *Linux* file-system) users. One common scenario is this:

- *Linux* being your *OS* of choice, the *Linux* partition start to fill and fill and fill.
- Your *DOS* partition is collecting dust, being half empty.
- You are suddenly out of space in the *Ext2* partition.
- You are still not sure you want to get rid of *DOS*.

Umsdos may save the day here. You can setup a *Linux* directory in the *DOS* partition and use it without restriction for *Linux* usage. For example, say you want to setup a new directory named `"extra"` in your `C:` drive. And you want this directory to behave as a normal *Linux* directory. Do this (assuming that C: is /dev/hda1).

```
mkdir /c
/sbin/mount -t umsdos /dev/hda1 /c
mkdir /c/extra
umssync /c/extra
```

You must be root to do this.

By setting up `/etc/fstab` like this, you will always have access to the `/c/extra` directory.

8. UMSDOS-WHY-TO

Explaining how to operate or install a *Umsdos* system is not enough. Most people are seeking some advises about using *Umsdos* or not.

8.1 The goal of *Umsdos*

The goal of *Umsdos* was to ease the installation of *Linux*. An other goal was to ease its UN-installation. The idea here was to promote the spreading of *Linux*. Installing a new OS on a system is always troublesome. *OS/2* for one will happily pollute your `C:` root with a bunch of new directories. If you are clever like me, it will also erase your config.sys and autoexec.bat files :-(

The pseudo-root feature of *Umsdos* avoid this unwanted invasion. *Linux* can be UN-install without side effect.

8.2 Who needs it

If you have a small hard drive, *Umsdos* will allow you to share disk space between *DOS* and *Linux*. A disk below 300 megs is in my opinion a small disk. This opinion is based on the size of the different package available today. One popular word processor may eat as much as 70 megabytes if you select all features.

If you have a larger drive, you may consider having a dedicated *Linux* partition running the *Ext2* file-system. *Ext2* use a smaller cluster size that *DOS* (1k in fact) so installing many small files will eat less space than in a *Umsdos* partition.

8.3 Performance issue

The following point apply to *Umsdos* compared with *Ext2*.

- Directory management is faster on *Ext2*. This come from the overhead of the double directory structure of *Umsdos*.
- File access (reading and writing) is probably faster on *Umsdos* than *Ext2*. This come from the simplicity of the *FAT* file-system used by *DOS*. Beware that this simplicity come with a cost:
  - A maximum of around 65,000 files or clusters per partitions. This also means that a 500 megabytes partition will use cluster 16k large. In other word, a file containing a single byte will use 16k of disk storage.
  - Everything is controlled by the `FAT` located at the beginning of the hard drive. The *DOS* file-system is probably more fragile because of this.
  - No provision to avoid fragmentation of files. A *Umsdos* system will generally be used as a

single user workstation. In this case, this does not matter much. As a multi-user engine, will get spread-ed all around the drive, lowering file access performance.

- Symbolic links are stored in normal file. If you intend to have a lot of them, you will find that *Umsdos* use quite a lot of disk space compared to *Ext2*.

The UPS Howto

Harvey J. Stein, hjstein@bfr.co.il, Berger Financial Research, Ltd.

v2.42, 18 November 1997

This document will help you connect an uninterruptable power supply to a Linux box... if you're lucky... Copyright (c) 1994, 1995, 1996, 1997 by Harvey J. Stein. You may use this document as you see fit, as long as it remains intact. In particular, this notice (along with the contributions below) must remain untouched.

1. Introduction

- 1.1 Contributors
- 1.2 Important disclaimer
- 1.3 Other documents

2. Important note on obsolete information

3. Smart and dumb UPSs.

4. Software

5. Do it yourself guide

- 5.1 What you need to do (summary)
- 5.2 How it's supposed to work
- 5.3 How to set things up
- 5.4 User Enhancements

TOC

6. Hardware notes

- 6.1 How to make a cable
- 6.2 Reverse-engineering cables and hacking powerd.c
- 6.3 Serial port pin assignments
- 6.4 Ioctl to RS232 correspondence

7. What to do when you're really stuck

8. Info on selected UPSs

- 8.1 General Experiences.
- 8.2 Advice 1200 A
- 8.3 Trust Energy Protector 400/650
- 8.4 Trust UPS 400-A
- 8.5 Sustainer S-40a
- 8.6 Systel
- 8.7 Deltec Power, Fiskars Power Systems and Exide.
- 8.8 Beaver model UB500 UPS
- 8.9 Sendom
- 8.10 Best
- 8.11 GPS1000 from ACCODATA
- 8.12 TrippLite BC750LAN (Standby UPS)
- 8.13 APC

9. How to shutdown other machines on the same UPS

---

1. Introduction

This HOWTO covers connecting a UPS to a computer running Linux. The idea is to connect the two in such a way that Linux can shutdown cleanly when the power goes out, and before the UPS's battery gives out.

This includes pointing out the existence of software packages which aid in establishing such communications, and detailing exactly how such communications are carried out. The latter often is unnecessary if you can find a software package that's already been configured for your UPS. Otherwise, you'll have to read on.

To a large extent this document is even more redundant than when I originally wrote it three years ago. All the basic information has always been contained in the `powerd` man page that comes with the `SysVinit` package. Whereas three years ago one could commonly find Linux distributions which didn't even include this man page, I don't believe this is the case any longer.

Furthermore, when I first wrote this Howto, there was no software other than `powerd.c` for Linux/UPS communications and control. Today there are quite afew UPS control packages available in Sunsite's UPS directory.

None the less, I'm continuing to maintain the UPS Howto. Why bother? Well,

- An additional general overview might help to understand how to connect a Linux system to a UPS, even if it's just the same information written differently.
- The HOWTO is serving as a repository for UPS specific data - there are many UPSs that haven't yet been incorporated into the general packages.
- The HOWTO contains additional details that aren't available in other documents.
- Some of the UPS software packages available in Sunsite's UPS directory seem to be quite sparsely documented. You might need to read this before you can understand how to use them.
- This thing seems to have a life of it's own now. It's clear when a Howto should be born. It's less clear when it should be put to sleep.

1.1 Contributors

I am forever indebted to those from whom I've received help, suggestions, and UPS specific data. The list includes:

- Hennus Bergman (hennus@sky.owl.nl)
- Charli (mefistos@impsat1.com.ar)
- Ciro Cattuto (Ciro Cattuto)
- Nick Christenson (npc@minotaur.jpl.nasa.gov)

TOC

- Lam Dang (angit@netcom.com)
- Markus Eiden (Markus@eiden.de)
- Dan Fandrich (dan@fch.wimsey.bc.ca)
- Ben Galliart (bgallia@orion.it.luc.edu)
- Danny ter Haar (dth@cistron.nl)
- Christian G. Holtje (docwhat@uiuc.edu)
- Raymond A. Ingles (inglesra@frc.com)
- Peter Kammer (pkammer@ics.uci.edu)
- Marek Michalkiewicz (ind43@sun1000.ci.pwr.wroc.pl)
- Jim Ockers (ockers@umr.edu)
- Evgeny Stambulchik (fnevgeny@plasma-gate.weizmann.ac.il)
- Clive A. Stubbings (cas@vjet.demon.co.uk)
- Miquel van Smoorenburg (miquels@cistron.nl)
- Slavik Terletsky (ts@polynet.lviv.ua)
- Tom Webster (webster@kaiwan.com)

Note that email addresses appearing below as excerpts from email messages can be out of date. The above is probably out of date too, but some of it's more recent than what's below.

Also, many apologies to anyone whom I've failed to note in this list. Please email me and I'll add you.

1.2 Important disclaimer

I really can't guarantee that any of this will work for you. Connecting a UPS to a computer can be a tricky business. One or the other or both might burn out, blow up, catch fire, or start World War Three. Furthermore, I only have direct experience with the Advice 1200 A UPS, and a 5kva Best Ferrups, and I didn't have to make a cable. So, BE CAREFUL. GATHER ALL INFORMATION YOU CAN ON YOUR UPS. THINK FIRST. DON'T IMPLICITLY TRUST ANYTHING YOU READ HERE OR ANYWHERE ELSE.

On the other hand, I managed to get everything working with my UPSs, without much information from the manufacturer, and without blowing anything up, so it is possible.

1.3 Other documents

This document does not cover the general features and capabilities of UPSs. For that type of information, you might turn to The UPS FAQ. It can also be found at ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/comp/answers/UPS-faq. It is maintained by Nick Christenson (npc@minotaur.jpl.nasa.gov), but seems to have last been updated in 1995. In email to him, he'd like that you put `UPS` or `UPS FAQ` or something along these lines in the Subject line of the message.

There're also more and more UPS manufactures sprouting up on the net. Some of them actually supply useful information on their web sites. A convenient list of UPS manufacturers' web sites is available at The UPS Directory. Said site also has a UPS FAQ.

---

2. Important note on obsolete information

I've just discovered that some of the documentation below is obsolete. In particular, the `init` daemon that comes with the latest sysvinit package is more sophisticated than I've portrayed it to be. Although it seems that the current version is backward compatible with what's written here, it looks like it has some undocumented features which are **very important** for UPS support.

The control mechanism outlined below only allows `powerd` to give `init` one of two messages, namely `powerfail` or `powerok`. `init` runs one command when it receives `powerfail`, and another when it receives `powerok`. This leads to complicated `powerd` logic for dealing with low battery signals and other sorts of special situations.

Newer versions of `init` (as of version 2.58, it seems) are more sophisticated. These versions can be signaled to run one of `three` scripts. Thus, `init` can have a `powerfail` script for announcing a power outage, a `powerfailnow` script for doing an immediate shutdown, and a `powerok` script for halting any pending shutdowns. This is much cleaner than the gyrations one would have to go through with the mechanisms detailed below.

Although most of the discussion here assumes the old `init` communication method, I just added two new sections where the authors uses the new communcation method. These are sections Trust Energy Protector 400/650 and APC Smart-UPS 700. The former is especially detailed. Both include a `powerd.c` which signals `init` to do an immediate shutdown when a low battery signal is received, as well as the relevant `/etc/inittab` lines to make this work. Other than this, all I can tell you is to look at the source code for `init`.

Also, for all I know, many of the software packages listed below also use this newer communication method.

3. Smart and dumb UPSs.

UPSs fall into two categories, which I'll call ``smart'' and ``dumb''. The difference between the two is the amount of information one can get from the UPS and the amount of control one can exert over the UPS.

Dumb UPS

- Connects to the computer via serial port.
- Uses modem control lines to communicate with the computer.
- Can signal whether or not the power is out.
- Typically can signal whether or not the battery is low.
- The computer can usually signal the UPS to turn itself off.

Smart UPS

- Connects to the computer via serial port.
- Communicates with the computer via normal data transfer through the serial port.
- Typically has some sort of command language that the computer can use to get various pieces of information from the UPS, to set various operating parameters for the UPS, and to control the UPS (such as turning it off).

Usually smart UPSs can be operated in dumb mode. This is useful because as far as I know, the company which manufactures the most popular smart UPS (namely APC) will only disclose the communication protocol for their UPSs to people who sign a non-disclosure agreement.

As far as I know, the only smart UPS available which is easy to communicate with under Linux are those made by Best. Furthermore, BEST fully documents the smart mode (and the dumb mode) of their UPSs. BEST also supplies source code for programs which can communicate with their UPSs.

All the packages listed in section Software will communicate with a UPS in dumb mode. This is all you really need. The ones specifically for the APC UPSs make various claims as to being usable in smart mode, but I don't know exactly what they permit. A full implementation would give you a pop-up window with all sorts of fun gauges displaying various statistics for the UPS, such as load, internal temperature, fault history, input voltage, output voltage, etc. It seems like the `smupsd-0.9-1.i386.rpm` package (section Software) approaches this. I'm not sure about the others.

The rest of this document is pretty much confined to configuring your system to work with a dumb UPS. The general idea is about the same with a smart UPS, but the details of how `powerd` would need to work and what kind of cable you need are different for a smart UPS.

4. Software

Basically, all you need is a working `powerd` binary, usually found in `/sbin/powerd`. This is usually part of the `SysVinit` package. As far as I know, all current Linux distributions include a recent version of `SysVinit`. Very old versions didn't include `powerd`.

The only problem you might have is that your cable might not match how `powerd` is set up, in which case you'll have to either rewire your cable, or pick up a copy of `powerd.c` and modify it to work with your cable. Or, for that matter, you can always pick up one of the following packages, most of which allow you to configure them to match your cable.

As mentioned, an alternative to using the `powerd` that comes with the `SysVinit` package would be to use one of the UPS packages now available. There are many packages currently available to aid in setting up computer/ups communications. None of this was available when I first wrote this Howto, which is why I had to write it. In fact, there's a good chance that you might be able to use one of these software packages, and avoid this Howto entirely!

As of 15 March 1997 or so, Sunsite's UPS directory had quite a few packages available. Other sites seem to have UPS control packages available too. Here's what I've found to date (all but two from sunsite):

Enhanced_APC_BackUPS.tar.gz

> A package for controlling APC Smart UPSs. Seems to basically follow the BUPS Howto (included here), but also seems to have some low battery warning support.

Enhanced_APC_UPSD-v1.4.tar.gz

> The `.lsm` file says that it's formerly the above package, but it actually includes the above package as a `.tar.gz` file inside of this `tar.gz` file! The documentation is spotty. It seems to support APC UPSs in both smart mode and dumb mode, but I can't be sure.

apcd-0.5.tar.gz

> Another package for controlling APC Smart UPSs. Seems to include some sort of master/slave support (i.e. - one machine signals others to shut down when the power goes out). Seems to use the UPS in smart mode, as opposed to via modem signal line toggling.

smupsd-0.9-1.i386.rpm

smupsd-0.9-1.src.rpm

> The author (David E. Myers, dem@netsco.com) writes:
>
> smupsd monitors an APC Smart-UPS[TM] under Red Hat[TM] Linux. Should power fail, smupsd will power down the system and the UPS in an orderly fashion.
>
> smupsd has the following features:
>
> - Shuts down the system and the UPS based on either remaining UPS battery charge or

elapsed time since power failure.

- UPS parameters can be monitored live from any host with the graphical monitor program upsmon, written in JavaTM.
- UPS parameters can be logged to a file for analysis and reporting.
- When additional systems share the same UPS, instances of smupsd running on these systems can read UPS parameters from the one running on the system serially connected to the UPS (master/slave).
- Network access from remote hosts can be controlled via the `/etc/hosts.allow` file.

genpower-1.0.1.tgz

A general UPS handling package. Includes configurations for many UPSs - two TrippLite configurations, and three APC configurations. Includes good documentation. A best buy.

powerd-2.0.tar.gz

A replacement for the `powerd` that comes with the `SysVinit` package. As opposed to comments included in the documentation it doesn't seem to have been merged into the `SysVinit` package as of version 2.62. Its advantages are that it can act as a server for other `powerd`s running on other machines (for when you have a network of machines hanging off a single UPS), and it can be configured by config file - the source code doesn't have to be edited and recompiled.

upsd-1.0.tgz

Another replacement for `powerd`. Seems to be quite comparable in features to powerd-2.0.tar.gz.

checkups.tar

This package is for controlling Best UPSs. It's direct from Best's web site. Includes binaries for lots of `unix` flavors, but more importantly, it includes source code, so you can try it out under Linux, and if it doesn't work, you can try to fix it. The source code includes both ``basic checkups'' which controls the UPS in dumb mode, and ``advanced checkups'' which is a little more sophisticated - it will signal a shutdown when the UPS says it has X minutes of power remaining instead of just shutting down X minutes after the power goes out. The advanced checkups program also will shut down when the UPS registers various alarms such as High Ambient Temperature, Near Low Battery, Low AC Out, or User Test Alarm.

bestups-0.9.tar.gz

A package that might very well be on sunsite by the time you read this. It's a pair of communications module which works with Best Ferrups UPSs. It operates the UPS in smart mode. It inter-operates well with powerd-2.0 - useful if you have a big Best Ferrups UPS keeping up all the machines on a network.

NOTE - This package has yet to be uploaded to Sunsite. I keep begging the author to finish and upload it, but he has yet to find the time.

LanSafe III

Deltec Electronics (and Exide) sell a software package called LanSafe III. They have a Linux version. It comes with their UPSs. They also say that it works with other UPSs (on the dumb

level).

apcupsd-2.8.tar.gz

The author (Andre Hedrick, hedrick@astro.dyer.vanderbilt.edu) writes:

apcupsd-2.1.tar.gz replaces Enhanced-APC-UPSD.tar.gz

It is a very complete package for APC UPSs. There is support for the entire range of UPSs in their product line. I have now added smart mode signaling to the package and support with APC's own cables or a custom cable if you don't have an APC cable that is supported to date.

smartups-1.1.tgz

From the LSM:

A powerd and an X11 graphing utility which shows you the voltages, frequencies, load percentage and battery level in realtime. The protocol that the "Safeware" software uses, and "Tripplite" UPSs are supported. Source + ELF binaries.

ups.tar.gz

From the LSM:

Program to interact with battery backups (Powerbox UPS).

usvd-2.0.0.tgz

From the LSM:

usvd is a daemon that monitors the state of an uninterrupted power supply and reacts upon state changes (line fail, line back, battery low situations). You can write your own scripts that are called in these cases. It does *not* require SYSVINIT.

Note that I've only glanced at these packages. I haven't used them. We were just about to start using bestups-0.9.tar.gz in conjunction with powerd-2.0.tar.gz, but we never quite got around to it.

---

5. Do it yourself guide

This discussion is specifically tailored for dumb UPS control. However, most of the process is about the same for dumb UPSs and smart UPSs. The biggest difference is in the details of how the UPS monitoring daemon (typically `powerd`) communicates with the UPS.

Before doing anything, I suggest the following algorithm:

- Skim this document.
- Download and investigate all packages which seem specifically tailored to your UPS.
- Download and investigate the more generic packages. Note that some of the more generic packages are actually more powerful, better documented, and easier to use than their more specific counterparts.
- If you still can't get things working, or if points are still unclear, read this document more carefully, and hack away...

5.1 What you need to do (summary)

- Plug the computer into the UPS.
- Connect the computer's serial port to the UPS with a special cable.
- Run `powerd` (or some sort of equivalent) on the computer.
- Setup your `init` to do something reasonable on powerfail and powerok events (like start a `shutdown` and kill any currently running `shutdown`s, respectively, for example).

5.2 How it's supposed to work

UPS's job

When the power goes out, the UPS continues to power the computer and signals that the power went out by throwing a relay or turning on an opticoupler on it's control port.

Cable's job

The cable is designed so that when the UPS throws said relay, this causes a particular serial port control line (typically `DCD`) to go high.

Powerd's job

The `powerd` daemon monitors the serial port. Keeps raised/lowered whatever serial port control lines the UPS needs to have raised/lowered (typically, `DTR` must be kept high and whatever line shuts off the UPS must be kept low). When `powerd` sees the UPS control line go high, it writes `FAIL` to `/etc/powerstatus` and sends the `init` process a `SIGPWR` signal. (Older versions of `powerd` and `initd` wrote to `/etc/powerfail`.) When the control line goes low again, it writes `OK` to `/etc/powerstatus` and sends `init` a `SIGPWR` signal.

Init's job (aside from everything else it does)

When it receives a `SIGPWR`, it looks at `/etc/powerstatus`. If it contains `FAIL` it runs the `powerfail` entry from `/etc/inittab`. If it contains `OK` it runs the `powerokwait` entry from `inittab`.

TOC

5.3 How to set things up

The following presupposes that you have a cable that works properly with `powerd`. If you're not sure that your cable works (or how it works), see section Reverse-engineering cables and hacking powerd.c for information on dealing with poorly described cables and reconfiguring `powerd.c`. Sections Serial port pin assignments and Ioctl to RS232 correspondence will also be useful.

If you need to make a cable, see section How to make a cable for the overall details, and the subsection of section Info on selected UPSs that refers to your UPS. The latter might also include information on manufacturer supplied cables. You may want to at least skim all of section Info on selected UPSs because each section has a few additional generally helpful details.

- Edit `/etc/inittab`. Put in something like this:

  ```
  # What to do when power fails (Halt system & drain battery :):
  pf::powerfail:/etc/powerfailscript +5

  # If power is back before shutdown, cancel the running shutdown.
  pg:0123456:powerokwait:/etc/powerokscript
  ```

- Write scripts `/etc/powerfailscript` and `/etc/powerokscript` to shutdown in 5 minutes (or whatever's appropriate) and kill any existing `shutdown`, respectively. Depending on the version of `shutdown` that you're using, this will be either so trivial that you'll dispense with the scripts, or be a 1 line `bash` script, something along the lines of:

  ```
  kill `ps -aux | grep "shutdown" | grep -v grep | awk '{print $2}'`
  ```

  and you'll keep the scripts. (In case it doesn't come out right, the first single quote on the above line is a backquote, the second and third are single quotes, and the last is also a backquote.)
- Tell `init` to re-process the `inittab` file with the command:

  ```
  telinit q
  ```

- Edit rc.local so that `powerd` gets run upon startup. The syntax is:

  ```
  powerd <line>
  ```

  Replace `<line>` with the serial port that the UPS is connected, such as `/dev/cua1`.
- Connect computer's serial port to UPS's serial port. DO NOT PLUG THE COMPUTER INTO UPS YET.
- Plug a light into the UPS.
- Turn on the UPS and the light.

TOC

- Run `powerd`.
- Test the setup:
  - Yank the UPS's plug.
    - Check that the light stays on.
    - Check that `/etc/powerfailscript` runs.
    - Check that `shutdown` is running.
  - Plug the UPS back in.
    - Check that the light stays on.
    - Check that `/etc/powerokscript` runs.
    - Check that `/etc/powerfailscript` is not running.
    - Check that `shutdown` is no longer running.
  - Yank the UPS's plug again. Leave it out and make sure that the computer shuts down properly in the proper amount of time.
  - **The Dangerous Part.** After everything seems to be proper, power down the computer and plug it into the UPS. Run a script that sync's the hard disk every second or so. Simultaneously run a second script that keeps doing a find over your entire hard disk. The first is to make this a little safer and the second is to help draw lots of power. Now, pull the plug on the UPS, check again that `shutdown` is running and wait. Make sure that the computer shuts down cleanly before the battery on the UPS gives out. This is dangerous because if the power goes out before the computer shuts down, you can end up with a corrupt file system, and maybe even lose all your files. You'll probably want to do a full backup before this test, and set the shutdown time extremely short to begin with.

Congratulations! You now have a Linux computer that's protected by a UPS and will shutdown cleanly when the power goes out!

5.4 User Enhancements

- Hack `powerd.c` to monitor the line indicating that the batteries are low. When the batteries get low, do an **immediate** shutdown.
- Modify the shutdown procedure so that if it's shutting down in a `powerfail` situation, then it turns off the UPS after doing everything necessary.

---

6. Hardware notes

6.1 How to make a cable

This section is just from messages I've seen on the net. I haven't done it so I can't write from experience. If anyone has, please write this section for me :). See also the message about the GPS1000 contained in section GPS1000 from ACCODATA, not to mention all the UPS specific data in section Info on selected UPSs.

```
>From miquels@caution.cistron.nl.mugnet.org Wed Jul 21 14:26:33 1993
Newsgroups: comp.os.linux
Subject: Re: UPS interface for Linux?
From: miquels@caution.cistron.nl.mugnet.org (Miquel van Smoorenburg)
Date: Sat, 17 Jul 93 18:03:37
Distribution: world
Organization: Cistron Electronics.

In article <1993Jul15.184450.5193@excaliber.uucp>
joel@rac1.wam.umd.edu (Joel M. Hoffman) writes:
>I'm in the process of buying a UPS (Uninteruptable Power Supply), and
>notice that some of them have interfaces for LAN's to signal the LAN
>when the power fails.
>
>Is there such an interface for Linux?
>
>Thanks.
>
>-Joel
>(joel@wam.umd.edu)
>

When I worked on the last versioon of SysVinit (Now version 2.4),
I temporarily had a UPS on my computer, so I added support for it.
You might have seen that in the latest <signal.h> header files there
is a #define SIGPWR 30 now :-). Anyway, I did not have such a special
interface but the output of most UPS's is just a relais that makes or breaks
on power interrupt. I thought up a simple way to connect this to the
DCD line of the serial port. In the SysVinit package there is a daemon
called 'powerd' that keeps an eye on that serial line and sends SIGPWR
to init when the status changes, so that init can do something (such as
bringing the system down within 5 minutes). How to connect the UPS to
the serial line is described in the source "powerd.c", but I will
draw it here for explanation:

                        +-------------------------o  DTR
                        |
                      +---+
```

```
                    |   | resistor
                    |   | 10 kilo-Ohm
                    |   |
                    +---+                                To serial port.
                      |
       +-----o-------+------------------------o  DCD
       |             |
       o  UPS        |
     \    relais     |
      \              |
       |             |
       +-----o-------+------------------------o  GND
```

Nice drawing eh?

Hope this helps.
SysVinit can be found on sunsite (and tsx-11 probably) as
SysVinit2.4.tar.z

Mike.

--

Miquel van Smoorenburg, <miquels@cistron.nl.mugnet.org>
Ibmio.com: cannot open CONFIG.SYS: file handle broke off.

>From danny@caution.cistron.nl.mugnet.org Wed Jul 21 14:27:04 1993
Newsgroups: comp.os.linux
Subject: Re: UPS interface for Linux?
From: danny@caution.cistron.nl.mugnet.org (Danny ter Haar)
Date: Mon, 19 Jul 93 11:02:14
Distribution: world
Organization: Cistron Electronics.

In article <9307174330@caution.cistron.nl.mugnet.org>
miquels@caution.cistron.nl.mugnet.org (Miquel van Smoorenburg) writes:
>How to connect the UPS to the serial line is described in the source
>"powerd.c", but I will draw it here for explanation:

The drawing wasn't really clear, please use this one in stead !

```
>
>                     +------------------------o  DTR
>                     |
>                   +---+
>                   |   | resistor
>                   |   | 10 kilo-Ohm
>                   |   |
>                   +---+                                To serial port.
>                     |
>       +-----o-------+------------------------o  DCD
```

```
>       |
>       o  UPS
>     \    relais
>      \
>       |
>       +-----o--------------------------------o  GND
>

The DTR is kept high, when the UPS's power input is gone it
will close the relais . The computer is monitoring
the DCD input port to go LOW . When this happens it will start a
shutdown sequence...


_____
Danny

--
<=====================================================================>
Danny ter Haar  <dannyth@hacktic.nl> or <danny@cistron.nl.mugnet.org>
Robins law #103: 'a couple of lightyears can't part good friends'
```

6.2 Reverse-engineering cables and hacking powerd.c

Try to get documentation for the cables that your UPS seller supplies. In particular find out:

- What lines need to be kept high.
- What line(s) turn off the UPS.
- What lines the UPS toggles to indicate that:
  - Power is out.
  - Battery is low.

You then need to either hack `powerd.c` appropriately, or use one of the above configurable packages (see the packages `genpower-1.0.1.tgz`, `powerd-2.0.tar.gz`, or `upsd-1.0.tgz` described in section Software). If you use one of the packages, follow the instructions there. If you want to hack `powerd.c`, keep reading.

If you have trouble getting the above information, or just want to check it (a *good* idea) the following program might help. It's a hacked version of `powerd.c`. It allows you to set the necessary port flags from the command line and then monitors the port, displaying the control lines every second. I used it as ``upscheck /dev/cua1 2" (for example) to set the 2nd bit (`DTR`) and to clear the other bits. The number base 2 indicates which bits to set, so for example to set bits 1, 2 and 3, (and clear the others) use 7. See the code for details.

Here's the (untested) upscheck.c program. It's untested because I edited the version I originally used to make it clearer, and can't test the new version at the moment.

```
/*
 * upscheck     Check how UPS & computer communicate.
 *
 * Usage:       upscheck <device> <bits to set>
 *              For example, upscheck /dev/cua4 4 to set bit 3 &
 *              monitor /dev/cua4.
 *
```

```
 * Author:      Harvey J. Stein <hjstein@math.huji.ac.il>
 *              (but really just a minor modification of Miquel van
 *              Smoorenburg's <miquels@drinkel.nl.mugnet.org> powerd.c
 *
 * Version:     1.0 19940802
 *
 */
#include <sys/types.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <errno.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <signal.h>

/* Main program. */
int main(int argc, char **argv)
{
  int fd;

/*  These TIOCM_* parameters are defined in <linux/termios.h>, which  */
/*  is indirectly included here.                                      */
  int dtr_bit = TIOCM_DTR;
  int rts_bit = TIOCM_RTS;
  int set_bits;
  int flags;
  int status, oldstat = -1;
  int count = 0;
  int pc;

  if (argc < 2) {
        fprintf(stderr, "Usage: upscheck <device> <bits-to-set>\n");
        exit(1);
  }

  /* Open monitor device. */
  if ((fd = open(argv[1], O_RDWR | O_NDELAY)) < 0) {
    fprintf(stderr, "upscheck: %s: %s\n", argv[1], sys_errlist[errno]);
    exit(1);}

  /* Get the bits to set from the command line. */
  sscanf(argv[2], "%d", &set_bits);

  while (1) {
    /* Set the command line specified bits (& only the command line */
    /* specified bits).                                             */
    ioctl(fd, TIOCMSET, &set_bits);
    fprintf(stderr, "Setting %o.\n", set_bits);

    sleep(1);
```

```
      /* Get the current line bits */
      ioctl(fd, TIOCMGET, &flags);
      fprintf(stderr, "Flags are %o.\n", flags);

  /*  Fiddle here by changing TIOCM_CTS to some other TIOCM until    */
  /*  this program detects that the power goes out when you yank     */
  /*  the plug on the UPS.  Then you'll know how to modify powerd.c. */
      if (flags & TIOCM_CTS)
        {
          pc = 0 ;
          fprintf(stderr, "power is up.\n");
        }
      else
        {
          pc = pc + 1 ;
          fprintf(stderr, "power is down.\n");
        }
      }

    close(fd);
  }
```

6.3 Serial port pin assignments

The previous section presupposes knowledge of the correspondence between terminal signals and serial port pins. Here's a reference for that correspondence, taken from David Tal's ``Frequently Used Cables and Connectors'' document. I'm including a diagram illustrating the connectors, and a table listing the correspondence between pin numbers and terminal line signals.

If you need a general reference for cable wiring, connectors, etc, then David Tal's would be a good one, but I can't seem to locate this document on the net any more. But I've found a good replacement. It's The Hardware Book.

Other useful sites:

- Yost Serial Device Wiring Standard which contains interesting information on how to use RJ-45 jacks and eight wire cables for all serial port connections.
- Stokely Consulting for general Unix info, and in particular their Unix Serial Port Resources.
- Unix Workstation System Administration Education Certification which contains RS-232: Connectors and Cabling

Incidentally, it seems that the Linuxdoc-sgml package still doesn't format tables very well in the `html` output. If you want to be able to read the following table, you're probably going to have to look at either the `DVI` version or the plain text version of this document.

DB-25 DB-9 Name EIA CCITT DTE-DCE Description
Pin # Pin #

---

1 FG AA 101 --- Frame Ground/Chassis GND
2 3 TD BA 103 ---> Transmitted Data, TxD
3 2 RD BB 104 <--- Received Data, RxD

TOC

4 7 RTS CA 105 ---> Request To Send
5 8 CTS CB 106 <--- Clear To Send
6 6 DSR CC 107 <--- Data Set Ready
7 5 SG AB 102 ---- Signal Ground, GND
8 1 DCD CF 109 <--- Data Carrier Detect
9 -- -- - - Positive DC test voltage
10 -- -- - - Negative DC test voltage
11 QM -- - <--- Equalizer mode
12 SDCD SCF 122 <--- Secondary Data Carrier Detect
13 SCTS SCB 121 <--- Secondary Clear To Send
14 STD SBA 118 ---> Secondary Transmitted Data
15 TC DB 114 <--- Transmitter (signal) Clock
16 SRD SBB 119 <--- Secondary Receiver Clock
17 RC DD 115 ---> Receiver (signal) Clock
18 DCR -- - <--- Divided Clock Receiver
19 SRTS SCA 120 ---> Secondary Request To Send
20 4 DTR CD 108.2 ---> Data Terminal Ready
21 SQ CG 110 <--- Signal Quality Detect
22 9 RI CE 125 <--- Ring Indicator
23 -- CH 111 ---> Data rate selector
24 -- CI 112 <--- Data rate selector
25 TC DA 113 <--- Transmitted Clock

Pin Assignment for the Serial Port (RS-232C), 25-pin and 9-pin

```
  1                          13         1         5
_________________________________      _______________
\  . . . . . . . . . . . . .  /       \  . . . . .  /    RS232-connectors
 \  . . . . . . . . . . . . .  /       \  . . . .  /     seen from outside
  ---------------------------           -----------      of computer.
  14                        25           6       9
```

```
DTE : Data Terminal Equipment (i.e. computer)
DCE : Data Communications Equipment (i.e. modem)
RxD : Data received; 1 is transmitted "low", 0 as "high"
TxD : Data sent; 1 is transmitted "low", 0 as "high"
DTR : DTE announces that it is powered up and ready to communicate
DSR : DCE announces that it is ready to communicate; low=modem hangup
RTS : DTE asks DCE for permission to send data
CTS : DCE agrees on RTS
RI  : DCE signals the DTE that an establishment of a connection is attempted
DCD : DCE announces that a connection is established
```

6.4 Ioctl to RS232 correspondence

TOC

Since you also might need to modify `powerd.c` to raise and lower the correct lines, you might also need the numeric values of different terminal signals. The can be found in `/usr/include/linux/termios.h`, but are reproduced here for reference. Since they could change, you're best off confirming these values against said file.

```
/* modem lines */
#define TIOCM_LE        0x001
#define TIOCM_DTR       0x002
#define TIOCM_RTS       0x004
#define TIOCM_ST        0x008
#define TIOCM_SR        0x010
#define TIOCM_CTS       0x020
#define TIOCM_CAR       0x040
#define TIOCM_RNG       0x080
#define TIOCM_DSR       0x100
#define TIOCM_CD        TIOCM_CAR
#define TIOCM_RI        TIOCM_RNG
```

Note that the 3rd column is in Hex.

---

7. What to do when you're really stuck

Here's a novel solution to UPS control for when the UPS and the computer just aren't on speaking terms. I must say that every time I read this, I'm struck by how clever a solution it is.

```
From: " Raymond A. Ingles" <inglesra@frc.com>
To: hjstein@math.huji.ac.il
Subject: UPS HOWTO tip
Date: Mon, 24 Feb 1997 11:48:32 -0500 (EST)


 I don't know if others would find this useful, but I thought I might
pass this along for possible inclusion in the HOWTO. Thanks for
maintaining a HOWTO that I found very useful!

-----------------

 My fiancee bought me a UPS as a present, a Tripp-Lite 400, I believe. It
was very welcome and seems to operate as expected, but unfortunately
doesn't have a serial interface to let the computer know the line power
has failed. It's apparently intended for home or office use where the
computer will not be left unattended.

 This, of course, was unacceptable and I began working on a line monitor,
planning on opening up the case and figuring out how to add the hardware
that the manufacturer had left out. Then I realized that there was a
quicker and simpler and cheaper (if somewhat less functional) way.

 I had an old 2400 baud modem that I wasn't using, and hooked it up to an
unused serial port on my computer. I then plugged the modem into a surge
supressor plugged into the wall power. I set up powerd with the options
as follows:

-----
serialline    /dev/ttyS1
monitor     DCD
failwhen    low
-----

 Now, when the wall power fails (or, since that hasn't happened lately,
when I pull the surge supressor from the wall to test this setup) the modem
fails but the UPS starts supplying power to the computer. When powerd
notices the modem has dropped DCD, it triggers the powerfail sequence.

 Obviously, this has some limitations. You can't tell from the modem when
the battery is low and so on. You can only tell that the wall power has
failed. Still, it's certainly cheap and I hate to see functioning
computer equipment lie unused. These days you should be able to get a
```

2400 baud modem for very nearly free.

I'd still suggest getting a real UPS with full communication capability. But if you're stuck with a less-functional one, this may at least make it useful.

Sincerely,

Ray Ingles (810) 377-7735 inglesra@frc.com

"Anybody who has ever seen a photograph showing the kind of damage that a trout traveling that fast can inflict on the human skull knows that such photographs are very valuable. I paid $20 for mine." - Dave Barry

8. Info on selected UPSs

This section contains UPS specific information. What I'd like is to have the UPS control port information (what each pin does and needs to have done), information on the manufacturer supplied cable (what it connects where), and a hacked version of powerd.c which works with the UPS. What I currently have is fairly complete descriptions of setting up each UPS. I'd try to distill out the relevant information, but since I can't test each UPS, it's hard to decide exactly what's relevant. Furthermore, each UPS seems to have some additional quirks that are nicely described by the authors of each section. So for now I'm leaving everything in. Makes for a hefty Howto.

Please send me your experiences for inclusion here.

8.1 General Experiences.

I've been saving peoples comments, but haven't gotten permission yet to include them here. Here's a general summary of what I've heard from people.

APC: Won't release info on their smart mode without your signature on a non-disclosure agreement. Thus, people are forced to run their smart UPSs in the dumb mode as outlined above. Various people have had varying amounts of success reverse engineering

Best: Helpful and friendly. Supply source code and documentation both for dumb modes and smart modes.

TrippLite: One person reported that TrippLite won't release info either.

Upsonic: One person reported that Upsonic has discussed technical details over the phone, answered questions via fax and are generally helpful.

8.2 Advice 1200 A

UPS from Advice Electronics, Tel Aviv Israel (they stick their own name on the things).

I don't recommend them. Our experiences with them have been very bad. We've twice had a 17" monitor fry when the power failed. We've had computers spontaneously reboot when the power failed.

None the less, for completeness, here's he UPS Control Port's pin specifications.

- 2 - Power Fail.
- 5 - Battery Low.
- 6 - Shut Down UPS.
- 4 - Common ground for pin 2, 5, 6.

They also gave me the following picture which didn't help me, but may help you if you want to build a cable yourself:

```
2 ----------+
            |
            \
             \|
              |--------------
             /|
           \/      <--- The "\/" here indicates the type of
           |            this transister.  I forget what
           |            denotes what, but this one points
```

```
                 +-----+         away from the center line.
                /  /  /


        5 ----------+
                    |
                    \
                     \|
                      |--------------
                     /|
                   \/
                   |
                   |
                 +-----+
                /  /  /


                      +-------------
                      |
                      /
             10K    |/
        6 --\/\/\/--|
                    |\
                      \/
                      |
                      |
                   +-----+
                  /  /  /


        4 ----------+
                    |
                    |
                 +-----+
                /  /  /
```

Cable supplied

They first gave me a cable that was part of a DOS UPS control package called RUPS. I used this for testing. When I was satisfied, they gave me a cable they use for Netware servers connected to UPSs. It functioned identically. Here are the details:

- DTR - Powers cable (make powerd.c keep it high).
- CTS - Power out (stays high and goes low when power goes out).
- DSR - Battery low (stays high. Goes low when battery does).
- RTS - Turns off UPS (keep it low. Set it high to turn off UPS).

(The powerd.c that comes with SysVinit set or left RTS high, causing the UPS to shut off immediately when powerd was started up!)

8.3 Trust Energy Protector 400/650

This section is good for more than just the Trust Energy Protector. It illustrates how to work with the new features of `init`.

How to use a Trust Energy Protector 400/650 under Linux

by Ciro Cattuto

Version 1.0 - 31 March 1997

The computer to UPS connection

The Trust Energy Protector 400/650 is equipped with a remote signal port. Using a properly designed cable, it is possible to connect the UPS port to the serial port of a computer, thus making it aware of power failure events.

The UPS signal port

These are the pin assignments for the DB-9 signal port of the Trust Energy Protector 400/650, as described in the user's manual:

pin 2

The relay will close when input power fails.

pin 4

Common for pins 2 and 5.

pin 5

The relay will close when the battery inside the Trust Energy Protector 400/650 has less than 1.5 minutes of backup time left.

pin 6

The user may send a high level signal (+5V - +12V) for over 1ms to turn off the Trust Energy Protector 400/650. However this option can only be activated when the input power fails.

pin 7

Common for pin 6.

The Cable

This is the cable I used to connect the UPS to the serial port of my computer:

```
computer side (DB-25)               UPS side (DB-9)
=====================================================

 6 DSR --+             [R] = 10 kilo-Ohm resistor
         |
20 DTR --+----+
         |    |
        [R]  [R]                          +--- 7
         |    |                           |
 8 DCD --+----|-----------        ---------|--- 2
              |                           |
 7 GND -------|-----------        ---------+--- 4
              |             ....
 5 CTS -------+-----------        ------------- 5

 2 TX  -------------------        ------------- 6

=====================================================
```

In the case of a DB-9 serial port, the pins 6,20,8,7,5,2 are mapped to pins 6,4,1,5,8,3.

How the cable works

The computer raises DTR and checks whether DSR is high, to ensure that the cable is connected to the computer. While the power is good, DCD and CTS are both high (because of the pull-up resistors).

When the power fails, the relay between pins 2 and 4 of the UPS port closes, and DCD becomes low, signalling the failure condition.

Similarly, when the UPS batteries are getting low, the relay between pins 5 and 4 closes, thus lowering CTS.

During a power failure the computer is able to turn off the UPS by raising TX for over 1ms. This can be easily accomplished sending a 0xFF byte to the serial port, at a low baud rate.

The powerd daemon

To make use of the information available at the serial port we need to run a program which monitors the port, decodes the signals and sends the appropriate messages to the operating system, i.e. to the init process. The init process can execute scripts and programs designed to handle (gracefully!) the power failure event.

Compiling powerd

In Appendix A you'll find the source code of powerd, the daemon I use to monitor the Trust Energy Protector 400/650. To compile it you will need the source code of the sysvinit package (I used the code from sysvinit-2.60). Just overwrite the original powerd.c and compile it.

How powerd works

As soon as powerd starts it opens the serial device connected to the UPS and forces DTR high. It then forks a daemon and exits, leaving the daemon running. The powerd daemon can be in one of three states:

State 0 - POWER IS GOOD

In this state powerd reads the serial port every T0_SLEEP seconds (see the #define lines at the beginning of the code). If DCD drops, powerd switches to state 1. If CTS drops powerd switches to state 2 (this shouldn't happen without DCD dropping before, but I decided to stay on the safe side).

State 1 - POWER FAILURE

A power failure was detected. DCD is low and powerd reads the UPS port every T1_SLEEP seconds. If DCD becomes high, it switches to state 0. If CTS drops, it switches to state 2.

State 2 - POWER CRITICAL

UPS batteries are low. The powerd daemon will remain in this state.

Each time powerd changes state, it notifies the init process, so that the appropriate action can be taken. These events are logged using the system logging facility.

If DSR is low there must be something wrong with the cable. Powerd keeps monitoring the DSR line, and every two minutes sends a warning message to the system logging facility.

Running powerd

The powerd daemon should be launched from the system initialization scripts, during system startup. I added the following lines to my /etc/rc.d/rc.local script:

```
# Add support for the UPS
echo "Starting powerd daemon..."
rm -f /etc/turnUPSoff
```

```
stty -crtscts speed 75 < /dev/cua3 > /dev/null
if [ -x /usr/sbin/powerd ]; then
        /usr/sbin/powerd /dev/cua3
fi
```

First we remove (if present) the file `/etc/turnUPSoff`. This file is used by the system shutdown script (`/etc/rc.d/rc.0`, in my case) to decide whether we want to turn the UPS off. See later in this document for more information.

Then we disable hardware flow control on the serial device connected to the UPS, and set its baud rate to 75. Now we're confident that the TX signal will stay high for a time long enough to turn the UPS off, if we send a character to the serial port (again, see later).

Finally we launch the `powerd` daemon, specifying the serial port to monitor. Notice that we're not going to read characters from the serial device, so don't worry if you have interrupt conflicts - they'll do no harm.

The inittab file and the shutdown scripts

The `powerd` process is now running, and it will send signals to `init` whenever a power failure occurs. Now we have to configure the system so that it can react in a useful way when those signals are received.

Modifying inittab

Add the following lines near the beginning of your /etc/inittab file:

```
# What to do when power fails (delayed shutdown).
pf::powerfail:/etc/powerfail_script

# If power is back before shutdown, cancel the running shutdown.
pg::powerokwait:/etc/powerokay_script

# If UPS batteries are getting low, do an immediate shutdown.
pc::powerfailnow:/etc/powerfailnow_script
```

The scripts

The scripts `powerfail_script`, `powerokay_script` and `powerfailnow_script` are executed when `init` receives the corresponding signal. They have the responsibility of shutting down the system in a clean way or cancelling a running shutdown in case power comes back. These are the scripts I'm currently using:

`/etc/powerfail_script`:

```
#!/bin/sh
/bin/sync
/usr/bin/sleep 10m
kill -9 `ps auxw | grep "shutdown" | grep -v grep | awk '{print $2}'`
> /etc/turnUPSoff
/sbin/shutdown -t30 -h +3 "POWER FAILURE"
```

My Trust Energy Protector 400 powers only the computer, so I have quite a long backup time. Since power failures only last for some minutes in my zone, the system responds to a blackout in the following way: it waits for 10 minutes (usually the power comes back before) and then halts the system, allowing the users to close their applications and leave the machine. Before issuing the shutdown command, I make sure that there are no running shutdowns. I also create the file `/etc/turnUPSoff`, so that the system will turn off the UPS.

`/etc/powerokay_script`:

TOC

```
#!/bin/sh
kill `ps auxw | grep "powerfail_script" | grep -v grep | awk '{print $2}'`
kill -9 `ps auxw | grep "shutdown" | grep -v grep | awk '{print $2}'`
rm -f /etc/turnUPSoff
```

If power comes back, we kills the running powerfail_script and any running shutdown. We also remove `/etc/turnUPSoff`.

`/etc/powerfailnow_script`:

```
#!/bin/sh
kill -9 `ps auxw | grep "shutdown" | grep -v grep | awk '{print $2}'`
> /etc/turnUPSoff
/sbin/shutdown -h now "UPS batteries low. IMMEDIATE SHUTDOWN."
```

If batteries are getting low, we make sure that there are no running shutdowns, create the `/etc/turnUPSoff` file and then shutdown the system immediately.

The system shutdown script

When system shutdown is complete, we can turn off the UPS raising the `TX` signal of the serial port for over 1ms. The serial device is already properly configured (see the `stty` command in the `rc.local` script). If the file `/etc/turnUPSoff` is present, we send the byte `0xff` (all '1' bits) to the serial port.

To do this, add the following lines near the bottom of your system shutdown script (`/etc/rc.d/rc.0`, in my case). The proper place depends on the way your system is configured, but it should be okay to insert the lines before the echo command which prints the "System is halted" message.

```
# Is this a powerfail situation?
if [ -f /etc/turnUPSoff ]; then
  echo "Turning off UPS. Bye."
  sleep 5
  echo -e "\377" > /dev/cua3
  exit 1
fi
```

General remarks

This document contains things I learned while trying to configure *my* Linux system to use the Trust Energy Protector 400. Some informations (the path of the system inizialization scripts, for example) may be specific to my system, and you probably will need some customization. However, I hope this document will be a useful trace for those trying to use a Trust Energy Protector 400/650 under Linux. If you experience difficulties, look for general information in the rest of this UPS-Howto. Good luck!

Feedback

I would greatly appreciate receiving feedback about this document, so that I can polish it and correct possible mistakes (I know the English is not very good, but I'm Italian after all!). Direct any comments/suggestions/critics to the following e-mail address:

ciro@stud.unipg.it

If you have problems using Trust Energy Protector 400/650 under Linux, feel free to contact me. I'll try to help you.

Legal Issues

I have no relation at all with Trust Networking Products.

The information contained in this document comes "as is". Use it at your own risk. I can't be held responsible for any damage or loss of data resulting from the use of the code and information given here.

Ciro Cattuto

Appendix A - Source code for the powerd daemon

powerd.c:

```
/*
 * powerd       Catch power failure signals from
 *              a Trust Energy Protector 400/650
 *              and notify init
 *
 * Usage:       powerd /dev/cua3 (or any other serial device)
 *
 * Author:      Ciro Cattuto <ciro@stud.unipg.it>
 *
 * Version 1.0 - 31 March 1997
 *
 * This code is heavily based on the original powerd.c code
 * by Miquel van Smoorenburg <miquels@drinkel.ow.org>.
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License
 * as published by the Free Software Foundation; either version
 * 2 of the License, or (at your option) any later version.
 *
 */

/* state 0 - power is good */
#define T0_SLEEP        10      /* interval between port reads, in seconds */
#define T0_DCD          3       /* number of seconds DCD has to be high
                                   to cause an action                      */
#define T0_CTS          3       /* number of seconds CTS has to be high
                                   to cause an action                      */
/* state 1 - power is failing */
#define T1_SLEEP        2       /* interval between ports reads            */
#define T1_DCD          3       /* same as T0_DCD                          */
#define T1_CTS          3       /* same as T0_CTS                          */

#define DSR_SLEEP       2
#define DSR_TRIES       60

/* Use the new way of communicating with init. */
#define NEWINIT

#include <sys/types.h>
#include <sys/stat.h>
```

TOC

```
#include <sys/ioctl.h>
#include <fcntl.h>
#include <errno.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <signal.h>
#include <syslog.h>
#include <string.h>
#include "paths.h"
#ifdef NEWINIT
#include "initreq.h"
#endif

#ifndef SIGPWR
#  define SIGPWR SIGUSR1
#endif

#ifdef NEWINIT
void alrm_handler()
{
}
#endif

/* Tell init that the power has gone (1), is back (0),
   or the UPS batteries are low (2). */
void powerfail(int event)
{
  int fd;
#ifdef NEWINIT
  struct init_request req;

  /* Fill out the request struct. */
  memset(&req, 0, sizeof(req));
  req.magic = INIT_MAGIC;
  switch (event)
        {
        case 0:
                req.cmd = INIT_CMD_POWEROK;
                break;
        case 1:
                req.cmd = INIT_CMD_POWERFAIL;
                break;
        case 2:
        default:
                req.cmd = INIT_CMD_POWERFAILNOW;
        }

  /* Open the fifo (with timeout) */
  signal(SIGALRM, alrm_handler);
  alarm(3);
  if ((fd = open(INIT_FIFO, O_WRONLY)) >= 0
                && write(fd, &req, sizeof(req)) == sizeof(req)) {
        close(fd);
```

TOC

```
            return;
      }
      /* Fall through to the old method.. */
   #endif

      /* Create an info file for init. */
      unlink(PWRSTAT);
      if ((fd = open(PWRSTAT, O_CREAT|O_WRONLY, 0644)) >= 0) {
      switch (event)
            {
            case 0:
                    write(fd, "OK\n", 3);
                    break;

            case 1:
                    write(fd, "FAIL\n", 5);
                    break;

            case 2:
            default:
                    write(fd, "LOW\n", 4);
                    break;
            }
      close(fd);
      }

      kill(1, SIGPWR);
   }

   /* Main program. */
   int main(int argc, char *argv[])
   {
      int fd;
      int dtr_bit = TIOCM_DTR;
      int flags;
      int DCD, CTS;
      int status = -1;
      int DCD_count = 0, CTS_count = 0;
      int tries;

      if (argc < 2) {
            fprintf(stderr, "Usage: powerd <device>\n");
            exit(1);
      }

      /* Start syslog. */
      openlog("powerd", LOG_CONS|LOG_PERROR, LOG_DAEMON);

      /* Open monitor device. */
      if ((fd = open(argv[1], O_RDWR | O_NDELAY)) < 0) {
            syslog(LOG_ERR, "%s: %s", argv[1], sys_errlist[errno]);
            closelog();
            exit(1);
      }
```

TOC

```
/* Line is opened, so DTR is high. Force it anyway to be sure. */
ioctl(fd, TIOCMBIS, &dtr_bit);

/* Daemonize. */
switch(fork()) {
      case 0: /* Child */
              closelog();
              setsid();
              break;
      case -1: /* Error */
              syslog(LOG_ERR, "can't fork.");
              closelog();
              exit(1);
      default: /* Parent */
              closelog();
              exit(0);
}

/* Restart syslog. */
openlog("powerd", LOG_CONS, LOG_DAEMON);

/* Now sample the DCD line. */
while(1) {
      /* Get the status. */
      ioctl(fd, TIOCMGET, &flags);

      /* Check the connection: DSR should be high. */
      tries = 0;
      while((flags & TIOCM_DSR) == 0) {
              /* Keep on trying, and warn every two minutes. */
              if ((tries % DSR_TRIES) == 0)
                  syslog(LOG_ALERT, "UPS connection error");
              sleep(DSR_SLEEP);
              tries++;
              ioctl(fd, TIOCMGET, &flags);
      }
      if (tries > 0)
              syslog(LOG_ALERT, "UPS connection OK");

      /* Calculate present status. */
      DCD = flags & TIOCM_CAR;
      CTS = flags & TIOCM_CTS;

      if (status == -1)
              {
              status = (DCD != 0) ? 0 : 1;
              if (DCD == 0)
                      {
                      syslog(LOG_ALERT, "Power Failure. UPS active.");
                      powerfail(1);
                      }
              }
```

TOC

```
        switch (status)
                {
                case 0:
                        if ((DCD != 0) && (CTS != 0))
                                {
                                DCD_count = 0;
                                CTS_count = 0;
                                sleep(T0_SLEEP);
                                continue;
                                }
                        if (DCD == 0)
                                DCD_count++;
                        if (CTS == 0)
                                CTS_count++;
                        if ((DCD_count < T0_DCD) && (CTS_count < T0_CTS))
                                {
                                sleep(1);
                                continue;
                                }
                        if (CTS_count == T0_CTS)
                                {
                                status = 2;
                                syslog(LOG_ALERT, "UPS batteries low!");
                                break;
                                }
                        status = 1;
                        DCD_count = 0;
                        syslog(LOG_ALERT, "Power Failure. UPS active.");
                        break;

                case 1:
                        if ((DCD == 0) && (CTS != 0))
                                {
                                DCD_count = 0;
                                CTS_count = 0;
                                sleep(T1_SLEEP);
                                continue;
                                }
                        if (DCD != 0)
                                DCD_count++;
                        if (CTS == 0)
                                CTS_count++;
                        if ((DCD_count < T1_DCD) && (CTS_count < T1_CTS))
                                {
                                sleep(1);
                                continue;
                                }
                        if (CTS_count == T1_CTS)
                                {
                                status = 2;
                                syslog(LOG_ALERT, "UPS batteries low!");
                                break;
                                }
                        status = 0;
```

```
                        DCD_count = 0;
                        CTS_count = 0;
                        syslog(LOG_ALERT, "Power okay.");
                        break;

                case 2:
                        sleep(1);
                        continue;

                default:
                        break;
                }

        powerfail(status);
  }
  /* Never happens */
  return(0);
}
```

8.4 Trust UPS 400-A

I received a submission about the Trust UPS 400-A. I don't know if it's the same as the Trust Energy Protector 400, so I'm including the submission.

```
From: "Marcel Ammerlaan" <marcel@ch.twi.tudelft.nl>
To: hjstein@math.huji.ac.il
Subject: UPS addition
Date: Wed, 16 Jul 1997 01:17:11 +100

Hello Harvey,

I've got an addition to your UPS Howto. I've got a
"Trust UPS 400-A" which isn't listed. This product doesn't seem
to be manufactured anymore by it's producer (www.trust.box.nl).
But that doesn't mean it's not available anymore, I've got mine
really cheap just a month ago. Also this company just relabels
products so maybe there are others that have got the same UPS.

I have included a picture of the UPS in case anybody got such a beast
under another label.

The cable was easily constructed based on the original powerd cable
and the documentation from trust.
It clearly describes which pins of the D-shell connector of the UPS
carry which signal.

It extends the original design with 2 extra functions:
1) Battery low indication
2) Power down UPS

The cable I created looks like (see the other attachement).

This cable has been tested with powergend by Tom Webster and did work
completely (although your milage may vary).
```

```
Type:                "pleur"
Cable Power:         {TIOCM_DTR,0}
Inverter Kill:       {TIOCM_RTS,1}
Inverter Kill Time:  5
Power Check:         {TIOCM_CTS,0}
Battery Check:       {TIOCM_CAR,0}
Cable Check:         {TIOCM_RI,0}

Although (just as the powerd cable) the cable check function isn't
used because the UPS doesn't seem to support it.

Well that's about it I guess. If you need more information about the
UPS the cable or the software feel free to contact me.

And remember, everything described here works for me but I don't
guarantee it will for you.

Marcel Ammerlaan
CEO Pleursoft (explains the cablename doesn't it :-)
The Netherlands

<RSA implemented in 3 lines of perl deleted by the editor ;)>

Marcel Ammerlaan   | <m.j.ammerlaan@twi.tudelft.nl>
Paardenmarkt 78    | Just another nerd on the loose
2611 PD Delft      |
The Netherlands    |
```

8.5 Sustainer S-40a

Information on the Sustainer S-40a.

```
From: fnevgeny@plasma-gate.weizmann.ac.il (Evgeny Stambulchik)
To: hjstein@math.huji.ac.il, hjstein@math.huji.ac.il, hjstein@math.huji.ac.il,
        hjstein@math.huji.ac.il
Subject: UPS-HowTo add-ons
Date: Sun, 10 Sep 1995 13:09:50 +0300 (IST)

Hi Harvey,

This is an addition to your UPS-HowTo. I'm using Sustainer S-40a UPS for a few
months with unipower package (now it's called genpower) and home-made cable
constructed as follows (I've sent all this stuff to Tom Webster, author of the
package, too, and it should appear in the next version):

         UPS SIDE                                    LINUX SIDE


             2 POWER FAIL                                1(8)
   +-----------o-------------------------+----------------o  DCD
   |                                     |
   o                                     |
    /                                    |
```

TOC

```
   /                                        |
   |            4 COMMON                    |              5(7)
   +-----------o------+-------------------|----------------o  GND
   |                  |                    |
   \                  |                    |
    \                 |                    |
   o                  |                    |
   |            5 BATT | LOW                |              8(5)
   +-----------o------|-------------------|--------+-------o  CTS
                      |                    |        |
                      |                 +-+-+    +-+-+
                      |                 |   |    |   |
                      |    3 resistors  |   |    |   |
                      |                 |   |    |   |
                      |    by 10 kOhm   |   |    |   |
                      |                 +-+-+    +-+-+
                      |                   |        |     4(20)
                      |                  +--------+-------o  DTR
                      |
|              6 SHUT | DOWN               +-------+     7(4)
+-+       +----o------|-------------------+       +-------o  RTS
  \       |           |                    +-------+
   \|    -+-          |
    | <- \ /          |
   /|    -+-          |
  /       |    7      |
  |       +----o------+
--+--
 ---
  -
```

NOTE!!!: Shutdown pins in the tech info supplied with UPS (4 and 6) are given incorrectly! The valid ones are 6 and 7, as shown above.
Note2: Pin numbers on the PC side in the brackets are for 25-pin connector, outside - for 9-pin one.

Here's the unipowerd.h file I used:

```
/*********************************************************************/
/* File Name            : unipowerd.h                                */
/* Program Name         : unipowerd                  Version: 1.0.0  */
/* Author               : Tom Webster <webster@kaiwan.com>           */
/* Created              : 1994/04/20                                 */
/* Last Modified By     : Tom Webster                Date: 1995/04/09 */
/* Last Modified By     : Evgeny Stambulchik (for Sustainer UPS)     */
/*                                                                   */
/* Compiler (created)   : GCC 2.5.8                                  */
/* Compiler (env)       : Linux 1.0.9                                */
/* ANSI C Compatable    : No                                         */
/* POSIX Compatable     : Yes?                                       */
/*                                                                   */
```

TOC

```
/* Purpose              : Header file for unipowerd.                    */
/*                      : Contains the configuration information for    */
/*                      : unipowerd.  Edit this file as indicated       */
/*                      : below to activate features and to customize   */
/*                      : unipowerd for your UPS.                       */
/*                                                                      */
/* Copyright            : GNU Copyleft                                  */
/************************************************************************/

/* The following are the RS232 control lines      */
/*                                                */
/*                                            D D */
/*                                            T C */
/* Macro           English                    E E */
/* ---------------------------------------------- */
/* TIOCM_DTR       DTR - Data Terminal Ready  --> */
/* TIOCM_RTS       RTS - Ready to send        --> */
/* TIOCM_CTS       CTS - Clear To Send        <-- */
/* TIOCM_CAR       DCD - Data Carrier Detect  <-- */
/* TIOCM_RNG       RI  - Ring Indicator       <-- */
/* TIOCM_DSR       DSR - Data Signal Ready    <-- */

#define HIGH            (1)
#define LOW             0
#define PWRSTAT         "/etc/powerstatus"
#define UPSSTAT         "/etc/upsstatus"

/* CABLEPOWER is the line which provides power to */
/* the cable for normal monitoring activities.    */
#define CABLEPOWER      TIOCM_DTR

#define POWERBIT        TIOCM_CAR
#define POWEROK         HIGH

/* Define CABLECHECK as 1 to check for low battery */
/* Define CABLECHECK as 0 value to skip            */
#define CABLECHECK      0
#define CABLEBIT        TIOCM_RNG
#define CABLEOK         HIGH

/* Define BATTCHECK as 1 to check for low battery  */
/* Define BATTCHECK as 0 value to skip.            */
#define BATTCHECK       1
#define BATTBIT         TIOCM_CTS
#define BATTOK          HIGH

/* Define INVERTERKILL as 1 to hndle killing the inverter */
/* Define INVERTERKILL as 0 value to skip.                */
/* INVERTERBIT is the line which will kill the inverter   */
/*    while the UPS is in powerfail mode.                 */
/* INVERTERTIME is the time in seconds to hold the line   */
/*    defiined by INVERTERBIT high to kill the inverter.  */
#define INVERTERKILL    1
#define INVERTERBIT     TIOCM_RTS
```

```
#define INVERTERTIME    5

/*********************************************************************/
/* End of File unipowerd.h                                           */
/*********************************************************************/
```

```
I'm aware that current name of the package is genpower. I haven't try it yet as
see no reason to switch to the new version meantime; the former seems to work
very stable. Nevertheless, here is the add-on for genpower-1.0.1's genpowerd.h
file (hopefully, I "translated" unipowerd.h correctly):
Add-on for genpower-1.0.1's genpowerd.h file:

/* Evgeny's Sustainer S-40A */
 {"sustainer",   {TIOCM_DTR,0}, {TIOCM_RTS,1},  5, {TIOCM_CAR,0}, {TIOCM_CTS,0},
{0,0}}


Evgeny
```

8.6 Systel

Another Israeli company. I never ended up purchasing a UPS from them, but they were very good about getting me detailed documentation on their communication port. It should be easy enough to control their UPS. Their phone number is 972-8-409-019 (972-8-407-216 for fax).

8.7 Deltec Power, Fiskars Power Systems and Exide.

Fiskars is a Finnish holding company. They used to own Deltec Power. In March of 1996 Fiskars sold Deltec Power to Exide Electronics Group. At that time, Deltec Power was one of the world's largest makers of UPSs.

Under Fiskars, Deltec used to make the PowerServers 10, 20, 30, and 40. The Deltec Power home page mentions other UPSs.

Exide now bundles UPS control software with their UPSs that works under Linux. They also sell the software separately. They say that their software works with other UPSs too.

I'd like to hear from people using their software.

Here's the advertisement they emailed me:

Exide Electronics announces LanSafe III UPS Power Management Software for Linux.

LanSafe III is a UPS Power Management application. It provides automatic orderly shutdown functionality incase of an extended power failure that should outlast the UPS battery run time.

LanSafe III enables broadcast messages and e-mail to be sent according to user defined power condition changes. The shutdown procedure can also be customized.

LanSafe III works together with the vast majority of all Exide Electronics UPS models. It goes even one step further by supporting basic shutdown functionality also with other manufacturers UPSs.

LanSafe III for Linux runs on Intel based Linux systems. Both character based and X11/Motif based user interfaces are provided.

LanSafe III supports all the major OS platforms: Linux, IBM AIX, HP UX, Digital UNIX, SCO UNIX, Solaris, SunOS, AT&T UNIX, all Windows platforms, OS/2, Novell and Macintosh among others.

LanSafe III is bundled with the following Exide Electronics UPSs: OneUPS Plus, NetUPS, PowerWare Prestige, PowerWare

Profile, PowerWare Plus 5xx.

It also ships with FPS Power Systems UPSs: PowerRite Plus, PowerRite Max, PowerWorks A30, PowerWorks A40, Series 9000 and Series 10000.

It is also possible to purchase a separate software license to use with a previous UPS model or an other manufactures UPS. Regular licenses are S$149, with site licenses also available.

For details please visit our Web sites at www.exide.com, www.fiskarsUPS.com and www.deltecpower.com.

Incidentally, when I tried to connect to www.fiskarsUPS.com, it prompted me for a username and password.

8.8 Beaver model UB500 UPS

dan@fch.wimsey.bc.ca (Dan Fandrich) writes:

I seem to have gotten my old Beaver model UB500 UPS working with genpower. The interface uses RS-232 compatible voltage levels, so installing it is a snap. There is a DE-9 female connector on the back which plugs directly into a 9-pin PC serial port using a plain 9-pin video monitor extension cable.

The DIP switches allow quite versatile pinouts. To emulate genpower's apc1-nt type of UPS, they must be set as follows:

1 on (CTS = power fail)
2 off (CTS = low battery)
3 off (DSR = power fail)
4 off (DSR = low battery)
5 off (CD = power fail)
6 on (CD = low battery)
7 off (RI = power fail)
8 off (RI = low battery)
9 on (DTR = inverter off)
10 off (RTS = inverter off)

DIP
switch
SW601
for
Beaver
model
UB500
UPS.

The switches form groups of adjacent pairs for each output pin. They are mutually exclusive--don't try to turn on both switch 5 and 6 simultaneously, for example, or you'll be shorting the low battery and power fail signals.

That's all there is to it. Feel free to add this do your documentation.

8.9 Sendom

Documentation on using the Sendom UPS.

```
From: charli <mefistos@impsat1.com.ar>
To: hjstein@math.huji.ac.il
Subject: ups howto contribution
Date: Wed, 13 Nov 1996 19:07:41 -0200
```

```
hjstein@math.huji.ac.il


sir:

i connected a sendom ups with the help of your UPS-howto and man powerd
and discovered something useful. perhaps this thing extends to some
other ups.
im using slackware 3.0 distribution. i has the soft configuration in
/etc/inittab already done. its only needed to add the /rc.local powerd
/cuaX


i used the man powerd diagram:
        9pin    25pin

DTR     4       20      ---------
                         |      >
DSR     6       6       --      < 10k
                                 >
DCD     1       8       -------------------------
                                                relais
GND     5       7       -------------------------


the fact is that the sendom ups dont use relais but some electronic
solid state device, and it works one way BUT NO THE OTHER. so if you
make the cable and doesnt work, first try inverting the cable in the
ups "relais"

i hope this can be useful, if you want to include this somewhere, feel
free to correct my english. please aknowledge this mail even with an
empty
mail so i know it arrived
end
```

8.10 Best

Information on Best UPSs is available on at the Best Power website. Their website includes the `checkups.tar` (section Software) package for communicating with Best UPSs, both in smart mode and in dumb mode, and it includes source code, so you can compile it under Linux.

Best Fortress - Using Best's software

Linux Best Power UPS Mini-HOWTO by Michael Stutz (stutz@dsl.org, and http://dsl.org/m/) v1.0, 14 Aug 97

0.0 Disclaimer

Copyright 1997 by Michael Stutz; this information is free; it may be redistributed and/or modified under the terms of the GNU General Public License, either Version 2 of the License, or (at your preference) any later version, and as long as this sentence remains; this information comes WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE; see the GNU General Public License for more details.

1.0 Introduction

Best Power are the makers of high quality UPS products, with their Fortress line in particular being well-suited for typical Linux users. Although their products are not currently priced as low as some (such as APC), Best Power provide source code for their UPS software and have been very respondent to queries from Linux users. Furthermore, their hardware seems to be highly regarded, making Best Power a winning choice for Linux users.

This document describes the installation of a Best Power Fortress UPS (model used was a 1996-model 660a with accompanying Best Power CD-ROM) to a Linux box.

2.0 Installation

2.1 Hardware

Install the hardware as indicated in the instructions. The Best Power ``Fortress'' series comes with an RS-232 cable that should attach to a spare serial port on the back of the computer.

2.2 Software

This is where it differs from the manual, which does not currently have Linux-specific instructions. The accompanying Fortress CD-ROM, however, does come with source code for the UPS software, so getting it up and running on a Linux system is a trivial task.

To do this, follow these steps, and use the manual as a reference to get an overall feel for how the software works. I took the liberty of making a few changes in this HOWTO from the way the Fortress software is set up on other UNIX systems that I feel are better suited for a Linux system. For example, I eliminated the need for an `/etc/best` directory and put the executables in `/usr/local/sbin`, which I feel is a more appropriate place.

- First, create the "upsdown" script that is executed during a power outage. This one will halt the system:

```
cat > /etc/upsdown << EOF
#!/bin/sh
shutdown -h now < /dev/console &
EOF
```

- Now, make directories for the documentation and the source code:

```
mkdir /usr/doc/best
mkdir /usr/local/src/best
```

- Mount the CD-ROM, and untar the unix/checkups.tar file into the /tmp directory or somewhere similar:

```
cd /tmp
tar /cdrom/unix/checkups.tar
```

- Change into the etc/best/advanced directory that should have been extracted from the checkups tarball.
- Copy the documentation and UPS script files to their proper place in the sytem:

```
cp README /usr/doc/best
cp manual.txt /usr/doc/best
cp bestsend /etc
cp source/*.c /usr/local/src/best
```

- Clean up the /tmp mess and compile the software:

```
cd /usr/local/src/best
rm -R /tmp/etc
gcc -o checkups checkups.c
gcc -o mftalk mftalk.c
mv checkups /usr/local/sbin
mv mftalk /usr/local/sbin
```

TOC

- Test the UPS. Replace ttySx with the serial port of your choice. If you have a good connection, you should see a r characters print across the screen:

```
mftalk /dev/ttySx
```

- Make the checkups program run at startup for testing. This can be done in several different ways (described in the manual). The way I did it is by adding this line to /etc/inittab:

```
ups:234:once:/usr/local/sbin/checkups -c500 /dev/ttyS1
```

- Test it. Do this by taking out power to UPS by pulling out the fuse connected to the UPS, and wait a couple of minutes. It print a warning messages and then halt the system after a few mintues.
- If that works, take out the "-c500" from the line in your inittab (which basically means shut down the system right away instead of when the UPS power runs out), and you're good to go!

3.0 Conclusions

I welcome suggestions for improving this document or the techniques described herein. As of this writing, Best Power seemed interested in including this or other information in their documentation to help Linux users with their product, so this is definitely a company to support. Let them know how you feel at sales@bestpower.com and support@bestpower.com.

Best Fortress LI-950

Some comments on the Best Fortress.

```
From lnz@dandelion.com Wed May 31 19:53:09 1995
Newsgroups: comp.os.linux.hardware
Subject: Re: UPS for use with Linux?
From: Leonard N. Zubkoff <lnz@dandelion.com>
Date: 25 May 1995 16:27:55 -0700
Organization: Dandelion Digital
NNTP-Posting-Host: dandelion.com
NNTP-Posting-User: root
In-reply-to: nautix@community.net's message of 23 May 1995 09:41:40 -0700

In article <3pt384$sic@odin.community.net> nautix@community.net writes:

  Ditto what Craig says.  APC was very uncooperative, but I have only
  good things to say about Best.  I use their Fortress LI 660 model;
  660 VA, lots of status features on the front, etc.  The CheckUPS
  software costs extra and needs some hacking to fit into my
  FSSTND-ish file system (the directories and file names are hard-coded
  to fit into SunOS 4.1.x).  I'd be happy to send you my diffs, if
  you want them.  (I love it when a vendor ships the source as
  a normal business practice!)

  The CheckUPS software is limited to doing automagic shutdowns, though.
  The UPS can give lots of status information; CheckUPS only asks for
  ``If the power has failed, how much battery time remains?''

  Best follows up on their customer satisfaction cards, too.
  I indicated that I was dissappointed that CheckUPS didn't do more
  status reporting (like input voltage, output voltage, percent load,
  etc.), which is available from the UPS.  I asked for a the
  spec on the interface lingo; they said ``sure'' and had it to me in
```

TOC

```
    2 days, free.  A full-featured UPS status checker is on my back burner.
    Does anyone see a demand for such a utility?

  Let me add yet another recommendation for Best Power.  I just purchased a
  Fortress LI-950, though I declined on the CheckUPS software.  Unlike some
  other brands, a simple three wire cable is all that's needed to connect the
  Fortress to a serial port -- no need for pull-up circuitry in the cabling.
  A few minutes hacking and I had program to act as both a shutdown monitor
  daemon, and to kill the inverter output when the system is shutdown while
  on battery power.

  I may eventually want to use the smarter serial communication mode rather
  than the simple contact mode, so I asked Best technical support for the
  documentation, and it arrived today, a week after I called them.  Once I
  peruse the documentation I'll decide if a smarter interface is really
  worthwhile, especially since at some point I'll need to shut down two
  networked machines sharing the UPS.

                  Leonard
```

Best Ferrups

In addition to the doumentation and softare on Best's web site, you could also use the `bestups-0.9.tar.gz` (section Software) package. We've just started testing it with our 5kva FERRUPS.

The basic idea is that there are two modules. One which fields information requests on a network port, relays those requests to the UPS, and returns the results. The second module talks to the first, interprets the results, and responds with either OK or FAIL.

This is sufficient to allow the `powerd-2.0.tar.gz` package (section Software) to do the rest of the work.

The details can be gotten from the `bestups-0.9.tar.gz` package (section Software).

Incidentally, our 5kva Ferrups has performed flawlessly in keeping our 10 computers and 30 screens humming.

8.11 GPS1000 from ACCODATA

```
>From hennus@sky.nl.mugnet.org Thu Mar 10 15:10:22 1994
Newsgroups: comp.os.linux.help
Subject: Re: auto-shutdown with UPS
From: hennus@sky.nl.mugnet.org (Hennus Bergman)
Date: Tue, 1 Mar 1994 22:17:45 GMT
Distribution: world
Organization: The Organization For Removal Of On-Screen Logos

In article <CRAFFERT.94Feb28125452@nostril.lehman.com>,
Colin Owen Rafferty <craffert@nostril.lehman.com> wrote:
>I am about to buy an Uninterruptable Power Supply for my machine, and
>I would like to get one that has the "auto-shutdown" feature.
>
I just got one of those real cheap :-)
It's a GPS1000 by ACCODATA. Anybody know how good the output
signal of these things is? [Don't have a scope myself :-(]
```

```
>I assume that these each have some kind of serial connection that
>tells the system information about it.
>
I took it apart to find out how it worked. There were three optocouplers
(two output, one input) connected to a 9 pin connector at the back.
One turns on when the power fails, and goes off again when the power
returns. While the power is off, you can use the `input' to shut the
battery off. [It releases the power-relay.] The third one is some kind
of feedback to tell that it did accepted the `shut-down command'.
I think the interface for my UPS was designed to be connected to TTL-level
signals, but with some resistors it could be connected to serial port.
It's wired in such a way that using a RS-232 port you cannot use both
output optocouplers; but the shutdown feedback is not necessary anyway,
just use the important one. ;-)
[Note that it is possible to blow the transistor part in optocouplers
with RS-232 levels if you wire it the wrong way round ;-)]

I was hoping I would be able to connect it to my unused game port,
but that doesn't have an output, does it?
I'll probably end up getting an extra printer port for this.

Not all UPS' use optocouplers, some use simple relays, which are
less critical to connect, but of course not as `nice'.

>Has anyone written a package that watches the UPS and does a shutdown
>(or something) when the power is off?
SysVinit-2.4 (and probably 2.5 as well) has a `powerd' daemon that
continually watches a serial port for presence of the CD (Carrier
Detect) line and signals init when it drops. Init then activates
shutdown with a time delay. If the power returns within a few minutes
the shutdown is cancelled. Very Nice.
The only problem I had with it is that it doesn't actually tell the
UPS to turn off when the shutdown is complete. It just sits there with
a root prompt. I'll probably write a small program to shut it down
>from /etc/brc. RSN.

>    Colin Rafferty, Lehman Brothers <craffert@lehman.com>

Hennus Bergman
```

8.12 TrippLite BC750LAN (Standby UPS)

Tom Webster (webster@kaiwan.com, the author of the genpower package) sent me information on the TrippLite BC750LAN. If you have one of these, your probably best off using his package.

But, for completeness, here's his cable wiring diagram (done by trial and error, and without documentation):

```
  UPS                 System
 DB-25                DB-25
   1 <--------------> 1        Ground

   2 <--------------> 4        Power Fail
   8 <--------------> 8        Sensing Circuit
```

```
 3 <-------------->  2       Inverter Shutdown
20 <--------------> 22       Circuit
```

8.13 APC

If the above plethora of APC packages doesn't get you running, maybe the following sections will help.

APC Back-UPS

There seems to be some controversy as to the accuracy of the information here on APC Back-UPSs. So, please be careful. I'm prefacing this section with one message of caution I received. It might not make a lot of sense before the rest of this section is read, but this way, at least you're more likely to see it. And again, since I don't have any APC UPS units, I can't verify the accuracy of either of these messages.

A message of caution

```
From ind43@sun1000.ci.pwr.wroc.pl Sun Oct  9 11:00:42 1994
Newsgroups: comp.os.linux.admin
Subject: BUPS-HOWTO warning
From: ind43@sun1000.ci.pwr.wroc.pl (Marek Michalkiewicz)
Date: 6 Oct 1994 18:38:15 GMT
Organization: Technical Univeristy of Wroclaw
NNTP-Posting-Host: ci3ux.ci.pwr.wroc.pl
X-Newsreader: TIN [version 1.2 PL2]

If you want to connect the APC Back-UPS to your Linux box, this might
be of interest to you.

There is a good BUPS-HOWTO which describes how to do this. But it has
one "bug".

The RTS serial port signal is used to shut down the UPS. The UPS will
shut down only if it operates from its battery. The manual says that
the shutdown signal must be high for at least 0.5s. But few milliseconds
is enough, at least for my APC Back-UPS 600.

Using RTS to shut down the UPS can be dangerous, because the RTS goes
high when the serial device is opened. The backupsd program then turns
RTS off, but it is on (high) for a moment. This kills the power when
backupsd is first started and there is a power failure at this time.
This can happen for example when the UPS is shut down, unattended,
and the power comes back for a while.

Either start backupsd before mounting any filesystems for read-write,
or (better) use TX (pin 3) instead of RTS (pin 7) to shut down the
UPS (pin numbers are for 9-pin plug). Use ioctl(fd, TCSBRKP, 10);
to make TX high for one second, for example. Using TX should be safe.
Maybe I will post the diffs if time permits...

-- Marek Michalkiewicz
ind43@ci3ux.ci.pwr.wroc.pl
```

BUPS-HOWTO

TOC

Luminated Software Group Presents

HOWTO use Back-UPS (by APC) (to keep your linux box from frying)

Version: 1.01 BETA

Document by: Christian G. Holtje <docwhat@uiuc.edu> Cabling info and help: Ben Galliart <bgallia@orion.it.luc.edu>

This document, under one condition, is placed in Public Domain. The one condition is that credit is given where credit is due. Modify this as much as you want, just give some credit to us who worked.

********** Warning! I, nor any of us who have written or helped with this document, make and guarantees or claims for this text/source/hints. If anything is damaged, we take NO RESPONSIBILITY! This works to the BEST OF OUR KNOWLEDGE, but we may have made mistakes. So be careful! **********

Al right, you just bought (or are going to buy) a Back-UPS from APC. (Other brands might be able to use this info, with little or no modification, but we don't know) You've looked at the price of the Power-Chute software/cabling, and just are not sure it's worth the price. Well, I made my own cable, and my own software and am using it to automatically shut off the power to my linux box when a power failure hits. Guess what? You can too!

*** The Cable ***

This was the hardest part to figure out (I know little about hardware, so Ben did the most work for this). To build one, you need to buy from your local radio shack (or other part supplier) this stuff:

```
1 9-Position Male D-Subminature Connector (solder-type)
        [Radio Shack cat. no. 276-1537c]
1 9-Position Female D-Subminature Connector (solder-type)
        [Radio Shack cat. no. 276-1538c]
2 casings for the above plugs (usually sold separately)
Some stranded wire (wire made of strands, not solid wire)
```

You also need, but may be able to borrow:

```
1 soldering iron
solder
```

Okay...this is how you connect it up!

These diagrams are looking into the REVERSE SIDE (the side where you solder the wire onto the plugs) The letters G, R, and B represent the colors of the wires I used, and help to distinguish one line from the next. (NOTE: I'm use standard rs-232 (as near as we can tell) numbering. The APC book uses different numbers. Ignore them! Use ours...I already changed the numbers for you!)

```
---------------------     Male Side! (This goes into the UPS)
 \  B   R  *  *  * /
   \  *  *  *  G  /
     ------------


---------------------     Female Side! (This goes into your COM port)
 \  R   *  *  *  G /
   \  *  B  *  *  /
     ------------
```

TOC

For those who like the numbers better:

```
        Male            Female
-------------------------------------
        1               7               Black
        2               1               Red
        9               5               Green
```

---------Aside: What the rs-232 pins are for!---------- Since we had to dig this info up anyway:

>From the REAR (the soldering side) the pins are numbered so:

```
    ---------------------
     \  1   2  3  4  5 /
       \  6  7  8  9  /
        ------------
```

The pins mean:

```
        Number  Name                    Abbr. (Sometimes written with D prefix)
        1       Carrier Detect          CD
        2       Receive Data            RD
        3       Transmit Data           TD(?)
        4       Data Terminal Ready     DTR
        5       Signal Ground           Gnd
        6       Data Set Ready          DSR
        7       Request to Send         RTS(?)
        8       Clear to Send           CS
        9       Ring Indicator          RI
```

What we did is connect the UPS's RS-232 Line Fail Output to the CD, the UPS's chassis to Gnd, and the UPS's RS-232 Shut Down Input to RTS. Easy now that we told you, no?

I have no idea if the software below will work, if you purchase the cable from APC. It might, and it might not.

*** The Software ***

Okay, I use the SysVInit package by Miquel van Smoorenburg for Linux. (see end for file locations, credits, email addresses, etc.) I don't know what would have to be changed to use someone elses init, but I know this code (following) will work with Miquel's stuff. Just so I give credit where credit's due. I looked at Miquel's code to figure out how ioctl()'s worked. If I didn't have that example, I'd have been in trouble. I also used the powerfail() routine (verbatim, I think), since it must interact with his init, I thought that he should know best. The .c file is at the end of this document, and just needs to be clipped off. To clip the file, edit away and extra '.sigs' and junk. This document should end on the line /* End of File */.....cut the rest.

This program can either be run as a daemon to check the status of the UPS and report it to init, or it can be run to send the kill-power command to the UPS. The power will only be killed if there is a power problem, and the UPS is running off the battery. Once the power is restored, it turns back on.

To run as a daemon, just type: backupsd /dev/backups

/dev/backups is a link to /dev/cua0 at the moment (COM 1, for you DOSers). The niceness of the link is that I can just re-link the device if I change to com 2 or 3.

Then, if the power dies init will run the commands for the powerwait. An example (This is from my `/etc/inittab`):

```
#Here are the actions for powerfailure.
pf::powerwait:/etc/rc.d/rc.power start
po::powerokwait:/etc/rc.d/rc.power stop
```

TOC

The powerwait will run, if the power goes down, and powerokwait will run if the power comes back up.

Here is my entire rc.power:

```
#! /bin/sh
#
# rc.power      This file is executed by init when there is a powerfailure.
#
# Version:      @(#)/etc/rc.d/rc.power   1.50    1994-08-10
#
# Author:       Christian Holtje, <docwhat@uiuc.edu>
#

  # Set the path.
  PATH=/sbin:/etc:/bin:/usr/bin:/sbin/dangerous

  # Find out how we were called.
  case "$1" in
        start)
                echo "Warning there is Power problems."  | wall
                # Save current Run Level
                ps | gawk '{ if (($5 == "init") && ($1 == "1")) print $6 }' \
                         | cut -f2 -d[ | cut -f1 -d] \
                         > /tmp/run.level.power
                /sbin/shutdown -h +1m
                ;;
        stop)
                echo "Power is back up.  Attempting to halt shutdown." | wall
                shutdown -c
                ;;
        *)
                echo "Usage:  $0 [start|stop]"
                exit 1
                ;;
  esac
```

Pretty nifty, no? Actually, there is a problem here...I haven't had time to figure it out...If there is a 'sh' wizard out there....

There is one little detail left, that is having the UPS turn off the power if it was halted with the power out. This is accomplished by adding this line into the end of your halt script:

```
/sbin/backupsd /dev/backups killpower
```

This will only kill the power if there is no power being supplied to your UPS.

*** Testing the stuff ***

This is just a short section saying this:

BE CAREFUL!

I recommend backing up your linux partitions, syncing several times before testing and just being careful in general. Of course, I'm just recommending this. I wasn't careful at all, and had to clean my partition several times testing my config. But it works. :)

*** Where to Get It ***

TOC

Miquel van Smoorenburg's SysVInit can be gotten at:

`sunsite.unc.edu:/pub/Linux/system/Daemons/SysVinit-2.50.tgz`

and a fix for some bash shells is right next-door as:

`sunsite.unc.edu:/pub/Linux/system/Daemons/SysVinit-2.50.patch1`

As to getting this HOWTO, you can email me. docwhat@uiuc.edu with the subject saying 'request' and the keyword 'backups' in body of the letter. (I may automate this, and other stuff)

*** Credit Where Credit's Due Dept. ***

Thanks to Miquel van Smoorenburg <miquels@drinkel.nl.mugnet.org> for his wonderful SysVInit package and his powerd.c which helped me very much.

Christian Holtje <docwhat@uiuc.edu> Documentation backupsd.c (what wasn't Miquel's) rc.power

Ben Galliart <bgallia@orion.it.luc.edu> The cable Information for the RS-232 standard Lousy Jokes (none quoted here)

```
/*  backupsd.c -- Simple Daemon to catch power failure signals from a
 *                Back-UPS (from APC).
 *
 *  Parts of the code are from Miquel van Smoorenburg's powerd.c
 *  Other parts are original from Christian Holtje <docwhat@uiuc.edu>
 *  I believe that it is okay to say that this is Public Domain, just
 *  give credit, where credit is due.
 *
 *  Disclaimer:  We make NO claims to this software, and take no
 *               resposibility for it's use/misuse.
 */

#include <sys/types.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <errno.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <signal.h>

/* This is the file needed by SysVInit */
#define PWRSTAT         "/etc/powerstatus"

void powerfail(int fail);

/* Main program. */
int main(int argc, char **argv)
{
  int fd;
  int killpwr_bit = TIOCM_RTS;
  int flags;
  int status, oldstat = -1;
  int count = 0;

  if (argc < 2) {
        fprintf(stderr, "Usage: %s <device> [killpower]\n", argv[0]);
```

TOC

```
            exit(1);
    }

    /* Open the the device */
    if ((fd = open(argv[1], O_RDWR | O_NDELAY)) < 0) {
            fprintf(stderr, "%s: %s: %s\n", argv[0], argv[1], sys_errlist[errno]);
            exit(1);
    }

    if ( argc >= 3  && (strcmp(argv[2], "killpower")==0) )
        {
            /* Let's kill the power! */
            fprintf(stderr, "%s: Attempting to kill the power!\n",argv[0] );
            ioctl(fd, TIOCMBIS, &killpwr_bit);
            /* Hmmm..... If you have a power outtage, you won't make it! */
            exit(0);
        }
    else
        /* Since we don't want to kill the power, clear the RTS. (killpwr_bit) */
        ioctl(fd, TIOCMBIC, &killpwr_bit);

  /* Become a daemon. */
    switch(fork()) {
    case 0: /* I am the child. */
                  setsid();
                  break;
    case -1: /* Failed to become daemon. */
                  fprintf(stderr, "%s: can't fork.\n", argv[0]);
                  exit(1);
    default: /* I am the parent. */
                  exit(0);
    }


    /* Now sample the DCD line. */
    while(1) {
        ioctl(fd, TIOCMGET, &flags);
        status = (flags & TIOCM_CD);
        /* Did DCD jumps to high? Then the power has failed. */
        if (oldstat == 0 && status != 0) {
            count++;
            if (count > 3) powerfail(0);
            else { sleep(1); continue; }
        }
        /* Did DCD go down again? Then the power is back. */
        if (oldstat > 0 && status == 0) {
            count++;
            if (count > 3) powerfail(1);
            else { sleep(1); continue; }
        }
        /* Reset count, remember status and sleep 2 seconds. */
        count = 0;
        oldstat = status;
        sleep(2);
```

```
  }
  /* Error! (shouldn't happen) */
  return(1);
}


/* Tell init the power has either gone or is back. */
void powerfail(ok)
int ok;
{
  int fd;

  /* Create an info file needed by init to shutdown/cancel shutdown */
  unlink(PWRSTAT);
  if ((fd = open(PWRSTAT, O_CREAT|O_WRONLY, 0644)) >= 0) {
        if (ok)
                write(fd, "OK\n", 3);
        else
                write(fd, "FAIL\n", 5);
        close(fd);
  }
  kill(1, SIGPWR);
}

/* End of File */
```

More notes

```
From ockers@carnot02.maem.umr.edu Mon Jan 16 15:27:29 1995
Newsgroups: comp.os.linux.hardware
Subject: Back-UPS, backupsd, and low battery signal
From: ockers@carnot02.maem.umr.edu (Jim Ockers)
Date: 12 Jan 1995 04:22:44 GMT
Reply-To: ockers@umr.edu
Organization: the all-male wasteland of Rolla, MO
NNTP-Posting-Host: carnot02.maem.umr.edu
X-Newsreader: TIN [version 1.2 PL2]

Hello all,

I use the backupsd on my linux system and I like it a lot.  I also
run Windows NT when I have to and it has a UPS daemon too.  The pinouts
required by Windows NT are different from the ones you specify in the
program but that is easily changed since I have the source for your
program..

Anyways I was browsing through the Windows NT knowledge base (KB) and
found something interesting.  If you look in the documentation for your
Back-UPS under "computer interface port" you will see that this UPS will
send a Low Battery signal at least two minutes before the battery fails.

At least the manual for my Back-UPS 400 says that...
```

TOC

However they also speak some Electrical Engineering gibberish ("Outputs ... are actually open collector outputs which must be pulled up to a common referenced supply no greater than +40 Vdc. The transistors are capable of a maximum non-inductive load of 25mAdc.)

Well that means nothing to me, but what I discovered in the NT KB was that it is possible to use the low battery signal in the same manner that the other signals are used. The output from pin 5 on the UPS should go to the pin on which you are reading the LowBatt signal into the computer. When that line goes high, the battery is running out of charge. When the situation is normal, that line will be low. (Hi/Lo in a standard RS-232 signal, just like the other lines.)

What they don't tell you in the APC manual, and they should, is that you need to buy a 10 KOhm resistor (50 cents at Radio Shack) and connect pins 5 and 8 on the UPS side using the resistor. Pin 8 provides the "common referenced supply no greater than 40vdc". Here's how you would make the cable (the 1st three lines are the same as the HOWTO):

```
       PC side                          UPS side
pin      7 <------------------------------> 1                 ShutDownUPS
         1 <------------------------------> 2                 LineFail
         5 <------------------------------> 4 (same as 9 )    GND
         ? your choice  <-----------------> 5                 LowBatt
                                            |
                                             > 10
                                            <  KOhm
                                            |
                                            8
```

So then when the LowBatt line is HIGH then the computer has 2 minutes to shut down before the battery runs out.

This is not mentioned in the Back-UPS HOWTO nor is it addressed in the backupsd source. However I would think that it would be a Good Thing to have in there; especially since a power failure would not require a shutdown unless the UPS batteries were low. In most cases it would mean that the backupsd could send a warning to everyone if the LineFails, and give everyone a one (or two) minute warning when the batteries start running down.

As far as I know this applies to all the APC Back-UPS and Smart-UPS products. These instructions were for a Smart-UPS 900,1250, and 2000 according to the NT KB. However they have been tested with a Back-UPS 400 running Windows NT and everything works properly...

I'd sure like to have a backupsd that handled the LowBatt situation too. Does anyone feel like modifying the backupsd.c source so that it will do this too? (I can't program in C yet...)

P.S. The APC manual says to use only pin 4 as the common and even though in the diagram it says that pin 9 is connected to pin 4 you might want to be sure and use pin 4 . This differs from the instructions in the HOWTO.

P.P.S. I mailed this to the Back-UPS HOWTO authors.

--
Jim (ockers@umr.edu) Ask me about Linux!
http://www.umr.edu/~ockers/ - home page

From: Peter Kammer <pkammer@liege.ICS.UCI.EDU>
To: "Harvey J. Stein" <hjstein@math.huji.ac.il>
cc: "Christian G. Holtje" <docwhat@uiuc.edu>
Subject: UPS-Howto--minor correction
Date: Mon, 07 Oct 1996 12:00:16 -0700

Mr. Stein,

Let me first thank you for putting together and maintaining the Linux UPS-HowTo document. I recently attached a APC Back-UPS 400 to a a Linux box and the document turned out to be very helpful.

I would like to suggest a correction to the the text diagrams which accompany the description in section 11.5.2. The diagrams are presented as being the rear of the plug. This in mind, the diagram of the male is backwards:

```
---------------------     Male Side! (This goes into the UPS)
 \  B   R  *  *  * /
   \  *  *  *  G  /
     ------------
```

From the rear, the pins on the male connector are numbered right-to-left. The correct diagram should be:

```
---------------------     Male Side! (This goes into the UPS)
 \  *   *  *  R  B /
   \  G  *  *  *  /
     ------------
```

Similarly, the numbered diagram should be labeled as for the rear of the female plug.

```
---------------------
 \  1   2  3  4  5 /
   \  6  7  8  9  /
     ------------
```

The rear of the male is numbered the reverse:

```
---------------------
 \  5   4  3  2  1 /
   \  9  8  7  6  /
     ------------
```

```
This caused us some confusion until we realized our mistake.  With four
different configurations to be aware of (front, rear) x (male, female) it
is easy to get confused.  Even now, reference in hand, I keep reexamining
my diagrams.

It might also be helpful to add a reference to the APC technical document
for the Back-UPS line which is available on-line at:

   http://www.apcc.com/english/techs/techref4/224e.htm

Once we corrected our wiring, setting up the software was relatively simple
thanks to your documentation.  We used the alternative (using TD to kill
the UPS power rather than RTS) wiring scheme and ran into few problems.
Your efforts in maintaining this information are much appreciated.

------------
Peter Kammer                        Dept. of Information and Computer Science
pkammer@ics.uci.edu                 University of California
http://www.ics.uci.edu/~pkammer/    Irvine, CA 92697-3425
```

APC Back-UPS Pro 650

```
From: Troy Muller <tmuller@agora.rdrop.com>
Sender: tmuller@napalm.it.wsu.edu
To: abel@netvision.net.il
Subject: APC Back-UPS Pro 650
Date: Sun, 06 Apr 1997 12:50:40 -0700

Dear Mr. Stein,

I have a Back-UPS Pro 650 from APC and finally got it working with a
standard APC cable.

I used cable number 940-0023A and Enhanced_APC_BackUPS software.  My
only grudge is the software broadcasts every 2 seconds, but hacking the
dowall.c code to sleep 10 sec before broadcasting seems to limit it to
every 10 seconds with a 2-3 message queued to be printed (ie. much more
acceptable).
```

APC Smart-UPS

Many people have APC Smart UPSs. There seem to be packages for using them in smart modes (see the afore mentioned packages `Enhanced_APC_UPSD-v1.4.tar.gz`, `apcd-0.5.tar.gz`, and `smupsd-0.7-1.i386.rpm` described in section Software). I don't know how the support in each package is. It seems that APC **still** refuses to release the protocol for the ``smart'' mode without a non-disclosure agreement, so everyone's left reverse engineering it.

The general consensus is to buy from a brand which does release the information, such as Best.

Another possibility is to run the SCO Unix version of APC's Powerchute UPS control software under Linux via the iBCS compatibility package. I'm told by Clive A. Stubbings (cas@vjet.demon.co.uk) that this works nicely after some install script adjustments. He says that the only problem is "the GUI stuff seems to have difficulty controlling non-local UPSs across the net".

If you have an APC Smart-UPS, and you can't get any of the above software to work in smart mode, you can still use it in

dumb mode. The following sections detail how to do that. In particular, I've received messages from people regarding th Model 600, the Model 700, and the model 1400. You'll probably have to hack `powerd.c` as outlined in section Reverse-engineering cables and hacking powerd.c.

TOC

APC Smart-UPS, Model 600

```
From dangit@netcom.com Mon Aug 22 10:16:23 1994
Newsgroups: comp.os.linux.misc
Subject: UPS Monitoring Cable For APC
From: dangit@netcom.com (Lam Dang)
Date: Fri, 19 Aug 1994 11:56:28 GMT
Organization: NETCOM On-line Communication Services (408 261-4700 guest)
X-Newsreader: TIN [version 1.2 PL1]

[Didn't make it the first time.]

A few netters have asked about UPS monitoring cables.  This is what I
found when I made one for my APC Smart-UPS, Model 600.  A disclaimer is in
order.  This is just an experimenter's report; use it at your own risks.
Please read the User's Manual first, especially Section 6.4, Computer
Interface Port.

The cable is to run between a 9-pin female connector on the UPS and a
25-pin male connector on the PC.  Since I cut off one end of a 9-pin
cable and replaced it with a 25-pin connector, I had to be VERY
CAREFUL ABOUT PIN NUMBERS.  The 25-pin hood is big enough to contain a
voltage regulator and two resistors.  I got all the materials (listed
below) from Radio Shack for less than 10 bucks.  As required by Windows NT
Advanced Server 3.5 (Beta 2), the "interface" between the UPS connector
and the PC connector is as follows:

        UPS (9-pin)              PC (25-pin)

        1 (Shutdown)             20 (DTR)
        3 (Line Fail)             5 (CTS)
        4 (Common)                7 (GND)
        5 (Low Battery)           8 (DCD)
        9 (Chassis Ground)        1 (Chassis Ground)

This is pretty straightforward.  You can use UPS pin 6 instead of 3
(they're the inverse of each other).  The complication is in pulling up
UPS open collector pins 3 (or 6) and 5.

This APC model provides an unregulated output of 24 Vdc at UPS pin 8. The
output voltage is available all the time (at least until some time after
Low Battery has been signalled).  The supply is limited to 40 mA.  To
pull up, UPS pin 8 is input to a +5 Vdc voltage regulator.  The output of
the regulator goes into two 4.7K resistors.  The other end of one
resistor connects both UPS pin 3 (Line Fail) and PC pin 5 (CTS).  That
of the other resistor connects both UPS pin 5 (Low Battery) and PC pin 8
(DCD).  The two resistors draw about 2 mA when closed.

Test your cable without connecting it to the PC.  When the UPS is on
```

TOC

```
line, pins 5 (CTS) and 8 (DCD) at the PC end of the cable should be very
close to 5 Vdc, and applying a high to pin 20 (DTR) for 5 seconds should
have no effect.  Now pull the power plug to put the UPS on battery.  Pin
5 (CTS) should go down to zero Vdc, pin 8 (DCD) should stay the same at 5
Vdc, and applying a high to pin 20 (DTR), e.g., by shorting pins 8 and 20,
should shut down the UPS after about 15 seconds.

Keep the UPS on battery until Low Battery is lighted on its front panel.
Now pin 8 (DCD) should go down to zero Vdc too.  Wait until the UPS
battery is recharged.  Then connect your cable to the PC, disable the UPS
option switches by turning all of them ON, and run your favorite UPS
monitoring software.

For those who want to run it with Windows NT Advanced Server, the UPS
interface voltages are NEGATIVE for both power failure (using UPS pin 3)
and low battery conditions, and POSITIVE for remote shutdown.  Serial
line parameters such as baud rate don't matter.

I haven't tested my cable with Linux powerd.  When you do, please let us
know.  I run NT as often as Linux on the same PC.  I must conform to NT's
UPS scheme.  Perhaps somebody can modify powerd to work with it and post
the source code here.

List of materials:

        1 shielded D-sub connector hood (Radio Shack 276-1510)
        1 25-pin female D-sub crimp-type connector (276-1430)
        1 7805 +5Vdc voltage regulator (276-1770)
        2 4.7K resistors
        1 component perfboard (276-148)
        1 cable with at least one 9-pin male connector.

You'll need a multimeter, a soldering iron, and a couple of hours.

Hope this helps.

Regards,

--
Lam Dang
dangit@netcom.com
```

APC Smart-UPS 700

Here're some details for running an APC Smart-UPS 700 in dumb mode.

It has a clever usage of a transistor in the cable so that the UPS will turn off when the computer is turned off. And it includes a `powerd.c` which also does a fast low battery shutdown.

Also, note that Markus' is also using `init`'s new capabilities. So we have here another illustration of how to use the new `init` to your advantage.

```
From: Markus Eiden <Markus@eiden.de>
Sender: eiden@eiden.de
To: "Harvey J. Stein" <abel@netvision.net.il>
```

TOC

Subject: Re: APC Smart-UPS
Date: Sun, 30 Mar 1997 16:21:05 +0200

I'm using an APC Smart-UPS 700 for my Linux box, running 2.0.21 on an ASUS-Board.

To use some features of the UPS you need four things:

1) You have to build a RS232-cable with a small interface.
2) You need the powerd-source from the sysvinit-package (I use version 2.65 from Miquel van Smoorenburg). Then you have to patch his powerd.
3) You have to change your /etc/inittab
4) You need a script to run some commands if the power is down or battery is low.

Some features:

When the power goes down, a script will start and a syslog-entry is done.

If the battery is low, an other script will start (which shutdown your computer of course) and a syslog-entry is done.

If you shutdown your computer and the power is down, the UPS will be shut down too.

1)Let's start with the cable:

If you have a look at the back side of you UPS you will see a female connector like this:

```
             8             1: Shutdown the UPS when the power is down and
                               pin 1 is high.
     X   X   X   X         3: Goes low on "Linefail"
   X   X   X   X   X       4: GND
                           5: Goes low on  "Low battery"
   1       3   4   5       8: +24V
```

On the other hand at the back side of your PC there exist a male connector like this:

```
         8       6         1: DCD
```

```
      X   X   X   X          4: DTR
    X   X   X   X   X        5: GND
    5   4           1        6: DSR
                             8: CTS
```

You have to build the following interface between these connectors:

```
 PC                                                          UPS


                                          #------------------  (8)
                                          |
                                         470 Ohm
                                          |
          #-----#-----#-----#-----#-----#----- ca. 9-12V
          |     |     |     |     |     |
          47    3.3   3.3   3.3   1     470
          kOhm  kOhm  kOhm  kOhm  kOhm  Ohm
          |     |     |     |     |     |
 (8) ------------------------#     |     |
          |     |     |           |     |
 (6) ------------#-------------------------------------------- (5)
          |           |           |     |
 (1) ------------------#------------------------------------- (3)
          |                       |     |
          |                      C#------------------------- (1)
          |                      -|     |
          |                    B/       |
 (4) ------#-----12kOhm---------|        |
                                \>E     |
                                |       |
 (5)-----------------------------#-------#------------------ (4)
```

-I use a "BC140" - transistor, but nearly any simple NPN-transistor should work ;-)

-The transistor works as an "inverter". If you shutdown your PC, AND the power ist down, then pin 4(PC) goes low and 1(UPS) goes high. This shuts down the UPS for saving batteries power.

2) The powerd-Source:

```
=======================

I patched the powerd-source very little (so it is really the source
of Miquel).

(a) Give an "alert" to the syslogd if 8(PC, CTS) is down (Cause
     the cable is not connected)

(b) DCD droped to zero => power has failed => call powerfail(0)
    => Give INIT_CMD_POWERFAIL to the init-process

(c) DCD comes up again => power is back => call powerfail(1)
    => Give INIT_CMD_POWEROK to the init-process

(d) DSR and DCD are dropped to zero => power has failed and
    battery is low => call powerfail(2) => Give
    INIT_CMD_POWERFAILNOW to the init-process

 Thats it.


------------------------------>8---- Schnipp ----------------------------


/*
 * powerd       Monitor the DCD line of a serial port connected to
 *              an UPS. If the power goes down, notify init.
 *              If the power comes up again, notify init again.
 *              As long as the power is OK, the DCD line should be
 *              "HIGH". When the power fails, DCD should go "LOW".
 *              Powerd keeps DTR high so that you can connect
 *              DCD and DTR with a resistor of 10 Kilo Ohm and let the
 *              UPS or some relais pull the DCD line to ground.
 *              You also need to connect DTR and DSR together. This
 *              way, powerd can check now and then if DSR is high
 *              so it knows the UPS is connected!!
 *
 * Usage:       powerd /dev/cua4 (or any other serial device).
 *
 * Author:      Miquel van Smoorenburg, <miquels@drinkel.cistron.nl>.
 *              Some minor changes by Markus Eiden, <Markus@Eiden.de>
 *              for the APC-Smart-UPS-powerd.
 *
 * Version:     1.31,  29-Feb-1996.
 *
 *              This program was originally written for my employer,
 *                      ** Cistron Electronics **
 *              who has given kind permission to release this program
 *              for general puppose.
 *
 *              Copyright 1991-1996 Cistron Electronics.
 *
 *              This program is free software; you can redistribute it and/or
 *              modify it under the terms of the GNU General Public License
```

```
 *              as published by the Free Software Foundation; either version
 *              2 of the License, or (at your option) any later version.
 *
 *              Some minor changes for the APC-powerd by Markus Eiden
 *              Markus@Eiden.de
 */

/* Use the new way of communicating with init. */
#define NEWINIT

#include <sys/types.h>
#include <sys/stat.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <errno.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <signal.h>
#include <syslog.h>
#include <string.h>
#include "paths.h"
#ifdef NEWINIT
#include "initreq.h"
#endif

#ifndef SIGPWR
#  define SIGPWR SIGUSR1
#endif

#ifdef NEWINIT
void alrm_handler()
{
}
#endif

/* Tell init the power has either gone or is back. */
void powerfail(ok)
int ok;
{
  int fd;
#ifdef NEWINIT
  struct init_request req;

  /* Fill out the request struct. */
  memset(&req, 0, sizeof(req));
  req.magic = INIT_MAGIC;


  /* INIT_CMD_* are definied in initreq.h                    *
   * Have a look at  init.c and /etc/inittab                 *
   *                                                         *
   * ok=0 -> INIT_CMD_POWERFAIL      -> powerwait            *
   * ok=1 -> INIT_CMD_POWEROK        -> powerokwait          *
```

TOC

```
     * ok=2 -> INIT_CMD_POWERFAILNOW   -> powerfailnow        */


    switch (ok) {
      case 0 : req.cmd = INIT_CMD_POWERFAIL;
               /* Linefail -> warning */
               break;
      case 1 : req.cmd = INIT_CMD_POWEROK;
               /* Power is back -> cancel warning */
               break;
      case 2 : req.cmd = INIT_CMD_POWERFAILNOW;
               /* Linefail and LowBatt -> reboot */
               break;
               }

    /* Open the fifo (with timeout) */
    signal(SIGALRM, alrm_handler);
    alarm(3);
    if ((fd = open(INIT_FIFO, O_WRONLY)) >= 0
                  && write(fd, &req, sizeof(req)) == sizeof(req)) {
          close(fd);
          return;
    }
    /* Fall through to the old method.. */
  #endif

    /* Create an info file for init. */
    unlink(PWRSTAT);
    if ((fd = open(PWRSTAT, O_CREAT|O_WRONLY, 0644)) >= 0) {
          if (ok)
                  write(fd, "OK\n", 3);
          else
                  write(fd, "FAIL\n", 5);
          close(fd);
    }
    kill(1, SIGPWR);
  }

  /* Main program. */
  int main(int argc, char **argv)
  {
    int fd;
    int dtr_bit = TIOCM_DTR;
    int flags;
    int status, oldstat = -1;
    int count = 0;
    int tries = 0;
    int powerfailed = 0;
    int rebootnow   = 0;

    if (argc < 2) {
          fprintf(stderr, "Usage: powerd <device>\n");
          exit(1);
    }
```

TOC

```
 /* Start syslog. */
 openlog("powerd", LOG_CONS|LOG_PERROR, LOG_DAEMON);

 /* Open monitor device. */
 if ((fd = open(argv[1], O_RDWR | O_NDELAY)) < 0) {
       syslog(LOG_ERR, "%s: %s", argv[1], sys_errlist[errno]);
       closelog();
       exit(1);
 }

 /* Line is opened, so DTR is high. Force it anyway to be sure. */

 /* USE: low Batt -> Reboot -> DTR goes low                      *
  * transistor is open -> shutdown-pin  of the UPS goes          *
  * high -> UPS goes down after about 20s. If there is a         *
  * linefail and the computer is off, the                        *
  * UPS goes down. If the power is back, the                     *
  * UPS goes on, the computer boots, and powerd                  *
  * is startet.                                                  */

/* Verwendung: Die UPS meldet low Batt -> Reboot -> DTR geht   *
  * auf Low -> Transistor oeffnet -> Shutdown der UPS geht auf  *
  * High -> UPS schaltet sich nach circa 20s aus. Bei jedem     *
  * Linefail und ausgeschaltetem Computer, schaltet sich die    *
  * UPS aus. Kommt der Strom zurueck, dann schaltet sich die    *
  * UPS selbstaendig ein, der Computer bootet, und der powerd   *
  * wird gestartet.                                              */

 ioctl(fd, TIOCMBIS, &dtr_bit);




 /* Daemonize. */
 switch(fork()) {
       case 0: /* Child */
               closelog();
               setsid();
               break;
       case -1: /* Error */
               syslog(LOG_ERR, "can't fork.");
               closelog();
               exit(1);
       default: /* Parent */
               closelog();
               exit(0);
 }

 /* Restart syslog. */
 openlog("powerd", LOG_CONS, LOG_DAEMON);

  syslog(LOG_INFO, "APCpowerd started...");
```

TOC

```
    /* Now sample the DCD line. */
    while(1) {
          /* Get the status. */
          ioctl(fd, TIOCMGET, &flags);

          /* Check the connection: CTS should be high. */
          tries = 0;
          /* TIOCM_*- have a look at  .../ams/termios.h */
          while((flags & TIOCM_CTS) == 0) {
                  /* Keep on trying, and warn every two minutes. */
                  if ((tries % 60) == 0)
                      syslog(LOG_ALERT, "UPS connection error");
                  sleep(2);
                  tries++;
                  ioctl(fd, TIOCMGET, &flags);
          }
          if (tries > 0)
                  syslog(LOG_ALERT, "UPS connection OK");

          /* Calculate present status. */
          status = (flags & TIOCM_CAR);

          /* Did DCD drop to zero? Then the power has failed. */
          if (oldstat != 0 && status == 0) {
                  count++;
                  if (count > 3) {
                          powerfailed = 1;
                          powerfail(0);
                          }
                  else {
                          sleep(1);
                          continue;
                  }
          }
          /* Did DCD come up again? Then the power is back. */
          if (oldstat == 0 && status > 0) {
                  count++;
                  if (count > 3) {
                          powerfailed = 0;

                          /* eigentlich unnoetig: */
                          rebootnow = 0;

                          powerfail(1);
                          }
                  else {
                          sleep(1);
                          continue;
                  }
          }

          /* Low battery and Linefail ? */
          if (rebootnow==0)
          if (powerfailed==1)
```

```
        if ((flags & TIOCM_DSR) == 0)
        {
           rebootnow=1;
           powerfail(2);

        }

        /* Reset count, remember status and sleep 2 seconds. */
        count = 0;
        oldstat = status;
        sleep(2);
  }
  /* Never happens */
  return(0);
}
```

---------- schnap ---------- 8<----------

3) Change your inittab:

Init gets the INIT_CMDs and will start a script:

```
pf::powerwait:/sbin/init.d/powerfail    start
pn::powerfailnow:/sbin/init.d/powerfail now
po::powerokwait:/sbin/init.d/powerfail  stop
```

(Which means for example: if the power has failed (powerwait) start the script /sbin/init.d/powerfail with the parameter "start".)

4) The powerfail-Script

---------- 8< ------- schnipp ----------

```
#! /bin/sh
# Copyright (c) 1997 Markus Eiden, Markus@Eiden.de
#

case "$1" in
    start)
        echo "THE POWER IS DOWN!" | wall
        logger "Powerfail"
        ;;
    now)
        echo "BATTERY IS LOW! Shutdown in 1 minute" | wall
        logger "Battery is low, shutdown in 1 minute"
        sync
```

TOC

```
        /sbin/shutdown -r -t 5 +1
        ;;
    stop)
        echo "THE POWER IS BACK!!" | wall
        logger "Power is back"

        /sbin/shutdown -c >/dev/null 2>/dev/null

        ;;
  *)
        echo "Usage: $0 {start|now|stop}"
        exit 1
        ;;
esac

exit 0

--------------------- >8 ----------- schnapp -------------
```

Well, that should be easy ;-)

You are ready now, but be careful: It works for me, but I really can't guarantee that any of this will work for you.

Some advice at the end: If /sbin/init.d/powerfail shuts down your PC then DTR goes down, so the shutdown pin (UPS) goes high. >From that time it takes about 20 or 30 seconds for the UPS to shut down. It is your job to prevent your Linux-box from booting within these 20 seconds (in particular to mount the filesystem). On my system it was no problem. There are four easy ways to prevent the PC from the fast booting:

1) The BIOS should do some routines (Like searching the number of tracks of your floppydisk if you have one)

2) If you have LILO installed, tell him to wait.

3) You do nothing (like I did)

4) You buy some more memory so that counting the memory takes 30 seconds. That should be about 1024 Megabytes ;-).

Markus Eiden

Markus@Eiden.de

```
--
-----------------------------------------------------------------------
StR Dipl.-Ing. Markus Eiden \\://                        Markus@eiden.de
Am alten Sportplatz 3       (o -)        http://www.rp.schule.de/eiden/
D-67599 Gundheim      ---ooO-(_)-Ooo---             NIC-HDL: ME256-RIPE
```

APC Smart-UPS 1400

Another day, another APC. This is for the Smart-UPS 1400, in dumb mode.

```
From: "Slavik Terletsky" <ts@polynet.lviv.ua>
To: hjstein@math.huji.ac.il
Subject: my contribution to UPS HOWTO
Date: Mon, 27 Jan 1997 21:10:16 +0000

Hello
I just hacked ups daemon, if you want, you may enclose it
in your UPS HOWTO document (i used some info from).
Please replay.
--

UPS daemon for FreeBSD (2.1.5 - tested).
Interacts with APC Smart-UPS 1400.

Connection scheme:

UPS (pin, signal name)          PC (pin, signal name)
----------------------          ---------------------
1 Shutdown              >----------->   4 Data Terminal Ready
2 Line Failed           >----------->   8 Clear To Send
4 Common                >----------->   5 Ground
5 Battery Low           >--------+-->   1 Data Carrier Detector
8 Battery (+24V)        >--|10K|-+


UPSD DESCRIPTION

usage: upsd <device> [wait [script]]

device  - device name upsd interacts thru (e.g. /dev/cuaa1)
wait    - time (secs) to wait before running script, (default value 0 sec).
script  - system shutdown script (default /etc/rc.shutdown).

Actions:
upsd logs all the changes of UPS status (power {up,down}, battery {low,ok}).
When "power down" and "battery low" upsd activates UPS SHUTDOWN signal,
waits for a <wait> seconds, and then runs system shutdown script - <script>.

Script sample:

#!/bin/sh
# Script is executed when system is going down.

PATH=/sbin:/bin:/usr/sbin:/usr/bin

echo "System is going DOWN right NOW" | wall

reboot
```

Upsd source:

```
/* UPS daemon
 * Copyright 1997 Slavik Terletsky. All rights reserved.
 * Author: Slavik Terletsky <ts@polynet.lviv.ua>
 * System: FreeBSD
 */
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <syslog.h>
#include <unistd.h>
#include <varargs.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/uio.h>
#include <sys/types.h>
#include <sys/ioctl.h>
#include <sys/ttycom.h>

int status;
int wait = 0;
FILE *fd;
char *scr = "/etc/rc.shutdown";
char *idf = "/var/run/upsd.pid";

void upsterm();
void upsdown(int);

int main(int argc, char *argv[]) {
 int pd;
 int zero = 0;
 char d5, d6, d7;
 char low = 0;
 char pow = 1;

 /* check arguments */
 switch(argc) {
 case  4:
 scr = argv[3];
 case  3:
 wait = atoi(argv[2]);
 case  2:
 break;
 default:
 fprintf(stderr, "usage: %s <device> [wait [script]]\n", argv[0]);
 exit(1);
 }

 /* check if script exists */
 if(!(fd = fopen(scr, "r"))) {
 fprintf(stderr, "fopen: %s: %s\n", scr, sys_errlist[errno]);
 exit(1);
 }
 fclose(fd);
```

TOC

```
/* check if upsd is already running */
if(fd = fopen(idf, "r")) {
fprintf(stderr, "fopen: %s: File already exists\n", idf);
exit(1);
}

/* become a daemon */
switch(fork()) {
case -1:       /* error */
fprintf(stderr, "fork: %s\n", sys_errlist[errno]);
exit(1);
case  0:       /* child */
break;
default:       /* parent */
exit(0);
}

/* save the pid */
if(!(fd = fopen(idf, "w"))) {
fprintf(stderr, "fopen: %s: %s\n", idf, sys_errlist[errno]);
exit(1);
}
fprintf(fd, "%d\n", (int)getpid());
fclose(fd);

/* open monitor device */
if((pd = open(argv[1], O_RDWR | O_NDELAY)) < 0) {
fprintf(stderr, "open: %s: %s\n", argv[1], sys_errlist[errno]);
exit(1);
}

/* daemon is alive */
openlog("upsd", LOG_PID, LOG_DAEMON);
syslog(LOG_INFO, "daemon started");

/* signal reaction */
(void)signal(SIGTERM, upsterm);

/* monitor device */
while(1) {
/* clear bits */
if(ioctl(pd, TIOCMSET, &zero) < 0) {
 fprintf(stderr, "ioctl: %s\n", sys_errlist[errno]);
 exit(1);
}

/* get device status */
if(ioctl(pd, TIOCMGET, &status) < 0) {
 fprintf(stderr, "ioctl: %s\n", sys_errlist[errno]);
 exit(1);
}

/* determin status */
```

```
    d5 = status & 0x20;
    d6 = status & 0x40;
    d7 = status & 0x80;

    /* power up */
    if(!(d7 + d5)) {
     if(!pow) {
      syslog(LOG_CRIT, "power up");
      pow = 1;
     }
    /* power down */
    } else {
     if(pow) {
      syslog(LOG_CRIT, "power down");
      pow = 0;
     }
    }

    /* battery low */
    if(!d6 && !low) {
     syslog(LOG_ALERT, "battery low");
     low = 1;

     /* down ups */
     if(!pow) {
      upsdown(pd);
     }
    }

    /* battery ok */
    if(d6 && low) {
     syslog(LOG_CRIT, "battery ok");
     low = 0;
    }

    sleep(1);
    }

    /* not reached */
    return 0;

   }

   void upsterm() {
    /* log termination message */
    syslog(LOG_INFO, "daemon terminated");

    /* remove pid file */
    unlink(idf);

    exit(0);
   }

   void upsdown(int pd) {
```

```
    /* log shutdown message */
    syslog(LOG_ALERT, "system is going down");

    /* remove pid file */
    unlink(idf);

    /* save our filesystem */
    system("/bin/sync");
    system("/bin/sync");
    system("/bin/sync");

    /* shutdown ups */
    status = TIOCM_DTR;
    if(ioctl(pd, TIOCMSET, &status) < 0) {
    fprintf(stderr, "ioctl: %s\n", sys_errlist[errno]);
    exit(1);
    }

    /* wait and then run script */
    sleep(wait);
    system(scr);
   }
   # Slavik Terletsky      # University "Lvivska Poytechnika" #
   # Network Administrator # mailto:ts@polynet.lviv.ua        #
```

9. How to shutdown other machines on the same UPS

Some people (myself included) have several computers running Linux connected to one UPS. One computer monitors the UPS and needs to get the other computers to shut down when the power goes out.

We assume the computers can communicate over a network. Call the computer that monitors the UPS the master and the other computers the slaves.

In the old days this required some fancy programming.

These days, the best thing to do is to pick up either the `powerd-2.0.tar.gz` package or the `upsd-1.0.tgz` package (see section Software), and follow the instructions. Both are able to run on the slaves in a mode where they connect over the network to a `powerd` or `upsd` process running on the master to query the status of the UPS. Some of the APC specific packages seem to have this ability too.

Note, though, that if your network is insecure, you might want to add a little security to this, lest someone spoof the slave `powerd` processes into thinking that the power is out.

Another possibility is to go for SNMP (Simple Network Management Protocol). Detailing the use of SNMP is beyond the scope of this document, not to mention currently being beyond me.

The Linux UUCP HOWTO

Guylhem Aznar <guylhem at oeil.qc.ca>

v2.0, 6 February 1998

*This document describes the setup, care & feeding of UUCP under Linux. You need to read this if you plan to connect to remote sites via UUCP via a modem, via a direct-connection, or via Internet. You probably do *not* need to read this document if don't talk UUCP or if you don't know what it means.*

1. Introduction, copyright & standard disclaimer

- 1.1 Email & spamming
- 1.2 Goals
- 1.3 New versions
- 1.4 Feedback
- 1.5 Copyright
- 1.6 Limited warranty

2. Other sources of information

- 2.1 USENET
- 2.2 Mailing Lists
- 2.3 Other documents from LDP
- 2.4 Books

3. Requirements

- 3.1 Hardware
- 3.2 Software

4. Setting up the config files

- 4.1 Installing config. files
- 4.2 "Poll" file
- 4.3 "call" file
- 4.4 "config" file
- 4.5 "crontab" file
- 4.6 "dial" file
- 4.7 "passwd" file
- 4.8 "port" file
- 4.9 "sys" file
- 4.10 Now let's test all this
- 4.11 Additional informations

5. It doesn't work - now what?

6. Frequently Asked Questions about Linux UUCP

- 6.1 Why is all the info here for UUCPs configured in "Taylor" rather than
- 6.2 Why do I get "timeout" on connections when I upgraded to uucp-1.04?
- 6.3 Why doesn't HDB anonymous uucp seem to work?
- 6.4 What does "no matching ports found" mean?
- 6.5 What are known good config files for HDB mode?
- 6.6 Getting uucico to call alternate numbers

7. Acknowledgements

---

1. Introduction, copyright & standard disclaimer

1.1 Email & spamming

First, convert all "at" in Emails addresses given in this document into "@".

It's simple for humans, but not for bots searching the web to spam ; therefore it's enough to protect generous contributors from being spammed.

1.2 Goals

The intent of this document is to answer some of the questions & comments that appear to meet the definition of "frequently asked questions" about UUCP software under Linux genrally & the version in the Linux Debian and RedHat distributions in particular.

1.3 New versions

New versions of this document will be periodically posted to comp.os.linux.announce, comp.answers & news.answers. They will also be added to the various anonymous ftp sites who archive such information including sunsite FTP.

In addition, you should be generally able to find this document on the Linux WorldWideWeb home page at the LDP page.

1.4 Feedback

I am interested in any feedback (please e-mail), positive or negative, regarding the content of this document. Definitely contact me if you find errors or obvious omissions.

I read, but do not necessarily respond to, all e-mail I receive. Requests for enhancements will be considered & acted upon based on that day's combination of available time, merit of the request & daily blood pressure :-)

Flames will quietly go to /dev/null so don't bother.

Feedback concerning the actual format of the document should go to the HOWTO coordinator: Greg Hankins (`gregh at sunsite.unc.edu`).

TOC

1.5 Copyright

The UUCP-HOWTO is copyrighted (c)1997 Guylhem Aznar. Distributed under LDP copyright license.
If you have questions, please contact Greg Hankins, the Linux HOWTO coordinator, at `gregh at sunsite.unc.edu`.

1.6 Limited warranty

Of course, I disavow any potential liability for the contents of this document. Use of the concepts, examples, &/ or other content of this document is entirely at your own risk.

---

2. Other sources of information

2.1 USENET

There is nothing "special" about configuring & running UUCP under Linux (any more). Accordingly, you almost certainly do *NOT* want to be posting generic UUCP-related questions to the comp.os.linux.* newsgroups.

Don't post in comp.os.linux hierarchy unless it's really linux specific, for example: "What's wrong with Debian 1.2 uucp?" or "RedHat 5.0 uucp crashes when I run it" ...

Let me repeat that.

There is virtually no reason to post anything uucp-related in the comp.os.linux hierarchy any more. There are existing newsgroups in the comp.mail.* hierarchy to handle *ALL* your questions.

IF YOU POST TO COMP.OS.LINUX. FOR NON-LINUX-SPECIFIC QUESTIONS, YOU ARE LOOKING IN THE WRONG PLACE FOR HELP. THE UUCP EXPERTS HANG OUT IN THE PLACES INDICATED ABOVE AND GENERALLY DO NOT RUN LINUX.*

POSTING TO THE LINUX HIERARCHY FOR NON-LINUX-SPECIFIC QUESTIONS WASTES YOUR TIME AND EVERYONE ELSE'S AND IT FREQUENTLY DELAYS YOU FROM GETTING THE ANSWER TO YOUR QUESTION.

The GOOD PLACE is `comp.mail.uucp` since you can get answers for most of your UUCP questions.

2.2 Mailing Lists

There is a Taylor UUCP mailing list.

To join (or get off) the list, send mail to

```
taylor-uucp-request@gnu.ai.mit.edu
```

This request goes to a person, not to a program, so please make sure that you include the address at which you want to receive mail in the text of the message.

To send a message to the list, send it to

```
taylor-uucp@gnu.ai.mit.edu
```

TOC

2.3 Other documents from LDP

There is plenty of exceptional material provided in the other Linux HOWTO documents & from the Linux DOC project.

In particular, you might want to take a look at the following:

- on your own computer in /usr/doc/uucp & /usr/info/uucp* :-)
- the Linux Networking Administrators' Guide
- the Serial Communications HOWTO
- the Ethernet HOWTO
- the News HOWTO
- the Mail HOWTO

2.4 Books

HDB & V2 versions of UUCP are documented in about every vendor's documentation as well as in almost all *nix communications books.

Taylor config. files are currently only documented in the info files provided with the sources (& in your distribution hopefully). The following is a non-inclusive set of books that will help.

- `"Managing UUCP & USENET"` from O'Reilly & Associates is in my opinion the best book out there for figuring out the programs & protocols involved in being a USENET site.
- `"Unix Communications"` from The Waite Group contains a nice description of all the pieces (& more) & how they fit together.
- `"Practical Unix Security"` from O'Reilly & Associates has a nice discussion of how to secure UUCP.
- `"The Internet Complete Reference"` from Osborne is a fine reference book that explains the various services available on Internet & is a great source for information on news, mail & various other Internet resources.
- "`The Linux Networking Administrators' Guide`" from Olaf Kirch of the Linux DOC Project is available on the net & is also published by (at least) O'Reilly & SSC. It makes a fine *one-stop shopping* to learn about everything you ever imagined you'd need to know about Unix networking.

---

3. Requirements

3.1 Hardware

There are no specific hardware requirements for UUCP under Linux. Basically any Hayes-compatible modem works painlessly with UUCP.

In most cases, you'll want the fastest modem you can afford, i.e. 56000 bps actually. In general, you want to have a 16550 UART on your serial board or built into your modem to handle speeds of above 9600 baud.

If you don't know what that last sentence means, please consult the *comp.dcom.modems* group or the various fine modem & serial communications FAQs & periodic postings on USENET.

3.2 Software

UUCP for linux is available everywhere, for example on sunsite.unc.edu. But before trying to get any version, try to install & make your current uucp work ; there're many little differences between each linux distribution, therefore it's easier for you to configure/install your distribution's UUCP package rather than editing sources for some options, setting the right paths & permissions, installing, etc.

But if you prefer sources ...

1) Unpack

To extract a gzip'd tar archive, I do the following:

```
gunzip -c filename.tar.z | tar xvf -
```

A "modern" tar can just do a:

```
tar -zxvf filename.tgz
```

2) Run "configure"

Type `"sh configure"`.

The configure script will compile a number of test programs to see what is available on your system & will calculate many things.

The configure script will create `conf.h` from conf.h.in & `Makefile` from Makefile.in. It will also create config.status, which is a shell script which actually creates the files.

3) Decide where to install

TOC

Rather than editing the Makefile.in file in the sources you can get the same effect by:

```
"configure --prefix=/usr/lib"
```

4) Edit "policy.h" for your local system

- - set the type of lockfiles you want (HAVE_HDB_LOCKFILES)
- - set the type of config files you want built in (HAVE_TAYLOR_CONFIG, HAVE_V2_CONFIG, HAVE_HDB_CONFIG)
- - set the type of spool directory structure you want (SPOOLDIR_HDB)
- - set the type of logging you want (HAVE_HDB_LOGGING)
- - set the default search path for commands (I added /usr/local/bin to mine)

5) Then compile & install the software

- Type `"make"` to compile.
- Type `"make install"` to install.

---

4. Setting up the config files

I recommend you start by installing the attached known-good config. files included in the document.

4.1 Installing config. files

Put these file in their "standard" location: `/etc/uucp` on recent linux distributions or `/usr/lib/uucp` on older ones.

Then make sure that the `permissions` of the files are

```
(guylhem@barberouge:uucp)$ ls -l
total 11
-rw-r--r--   1 uucp     uucp          501 Jan 23 11:33 Poll
-rw-r-----   1 uucp     uucp          589 Jan 23 11:34 call
-rw-r-----   1 uucp     uucp         1184 Jan 23 12:06 config
-rw-r-----   1 uucp     uucp          476 Jan 23 12:31 crontab
-rw-r-----   1 uucp     uucp         1256 Jan 23 11:47 dial
-rw-r-----   1 uucp     uucp          486 Jan 23 11:48 passwd
-rw-r-----   1 uucp     uucp          810 Jan 23 11:55 port
-rw-r--r--   1 uucp     uucp         1690 Jan 23 12:04 sys
(guylhem@barberouge:uucp)$
```

To change file owner, as root, type:

```
(root@barberouge:uucp)$ chown uucp.uucp *
```

Then you must change file permissions; as root once again, type:

```
(root@barberouge:uucp)# chmod 640 *
(root@barberouge:uucp)# chmod +r Poll sys
```

4.2 "Poll" file

This file is used to set polling timetables for any system.

```
schedule polux 01
poll     polux 01
```

My machine calls polux at 01:00, that's all!

You can add more line if you must call many other machines, but don't forget to put the 2 lines (schedule & poll) for each.

4.3 "call" file

It contains your login/password for each system you poll:

```
polux uudan password
```

My machine uses "uudan" login & "password" password :-) when it polls "polux".

As for Poll, adapt this to your situation.

4.4 "config" file

```
nodename barberouge                     # The UUCP name of this system

spool /var/spool/uucp                   # The UUCP spool directory
pubdir /var/spool/uucppublic            # The UUCP public directory
logfile /var/log/uucp/log               # The UUCP log file
statfile /var/log/uucp/stats            # The UUCP statistics file
debugfile /var/log/uucp/debug           # The UUCP debugging file

#sysfile        /etc/uucp/sys           # Default "sys"
#portfile       /etc/uucp/port          # Default "port"
#dialfile       /etc/uucp/dial          # Default "dial"
#dialcodefile   /etc/uucp/dialcode      # Default "dialcode"
#callfile       /etc/uucp/call          # Default "call"
#passwdfile     /etc/uucp/passwd        # Default "passwd"

# No commands may be executed by unknowns (empty list of permitted commands)
# Upload is authorized in /var/spool/uucp
unknown commands
unknown pubdir /var/spool/uucp
unknown remote-send ~ !~/upload
unknown remote-receive ~/upload
```

Here just replace "barberouge" by your system name ; run "hostname" if you can't remind it.

4.5 "crontab" file

```
# Every day just before morning generate reports.
#
0 7 * * *     /usr/lib/uucp/uudemon.day root
#
# Every hour start the uudemon.hr. To actually poll a remote system,
# enter its name in /etc/uucp/Poll. You are encouraged to change the "8".
#
8 * * * *      /usr/lib/uucp/uudemon.hr
```

Just run "crontab -u uucp /etc/uucp/crontab" to add it to the others crontabs.

4.6 "dial" file

```
# 1) expect nothing (i.e., continue with step 2)
# 2) send "ATZ", then a carriage return, then sleep for 1 to 2 seconds.
# The \c means to not send a final carriage return.
# 3) wait until the modem echoes "OK", then do the the same for "ATX4" & "OK"
# 4) send "ATDT", then the telephone number (after translating any dialcodes).
# 5) wait until the modem echoes "CONNECT"
# 6) if we get "BUSY", "NO CARRIER" ... during the chat script we abort dialing
# 7) when the call is over, we make sure we hangup the modem
```

```
dialer hayes
chat "" ATZ\r\d\c OK\r \dATX4\r\d\c OK\r ATDT\D CONNECT
chat-fail RING
chat-fail NO\sCARRIER
chat-fail ERROR
chat-fail NO\sDIALTONE
chat-fail BUSY
chat-fail NO\sANSWER
chat-fail VOICE
complete \d\d+++\d\dATH\r\c
abort \d\d+++\d\dATH\r\c

# You can also add other dialers: inetd, nullmodem ...
#dialer nullmodem
#complete \d\dexit\r\c
#abort \d\dexit\r\c
```

Syntax is complicated ... you'd rather not touch anything here but "ATZ" & "ATX4" which are my modem init string.

4.7 "passwd" file

```
#uuguest                guestpassword
```

If you allow uucp dialin, just add system/passwords in this file.

It's *that* simple.

But it's recommended for security reasons to make sure each have a separate account & home directory so you can track things.

4.8 "port" file

```
# Description for the modem entry
# Debianers, make SURE this device is root:dialout, mode 0660 (crw-rw---)

port ACU
type modem
device /dev/ttyS0
dialer hayes
speed 57600
# hardflow n


# Description for the TCP port - pretty trivial. DON'T DELETE.
# Change service number if non standard, cf /etc/services

port TCP
type tcp
#service 540


# Description for the nullmodem entry
# (ttyS1 means COM2)

port nullmodem
```

```
type direct
device /dev/ttyS1
dialer nullmodem
speed 115200
```

You shouldn't change anything here ... except your modem port.

On recent distributions with mgetty, it's `/dev/ttySN` while on older distributions it's `/dev/cuaN`, where N is you serial port:

N starts at 0 & ttyS(N) means COM(N+1), for example, my null-modem is on ttyS1 (COM2) while my modem is on ttyS0 (COM1).

Most of recent modems support hardware flow control, if your doesn't, just uncomment the line "# hardflow n".

4.9 "sys" file

```
# First some defaults. These are for ALL other entries (unless overridden).
#
protocol gvG
protocol-parameter G packet-size 1024
# protocol-parameter G window 7
protocol-parameter G short-packets

#
# Our remote uucp connection.
#
system polux
call-login *
call-password *
local-send /
local-receive /var/spool/uucppublic
remote-send /
remote-receive /var/spool/uucppublic
time any
phone 0111111110
port ACU
chat "" \r\c ogin:-BREAK-ogin:-BREAK- \L word: \P
#chat "" \d\d\r\c ogin: \d\L word: \P

# This is an alternate - it means that if a connection using the above
# "system polux" fails it falls through to this entry.
# Only useful if your service provider has more then one phone number for UUCP.
#
#alternate polux-2
#alias polux-2
#phone 0222222220

# Here's another alternate - we poll the system over TCP/IP.
# This is useful if we have a PPP connection to our provider.
# The first two entries will fail because the modem is busy & we will poll
# over TCP/IP.
#
#alternate polux-tcp
#alias polux-tcp
#time any
#address uucp.polux
```

```
#port TCP
#protocol t

#
# Last example - a system that we poll over TCP/IP.
#
#system horizon
#call-login *
#call-password *
#time any
#chat "" \d\d\r\c ogin: \d\L word: \P
#address uucp.horizon.nl
#port TCP
#protocol t
```

Here, you must change "polux" by the name of the system you poll & "0111111110" by its phone number.

"polux-2", "polux-tcp" & "horizon" are just examples of user specific needs ; if the system you poll has more than one line, define it as "itsname-2" ; if you call it by PPP sometimes, just define "itsname-tcp" also.

It's useful when chat script fail (busy ...)

4.10 Now let's test all this

First run:

```
(root@barberouge:uucp)# su uucp
(uucp@barberouge:uucp)# /usr/lib/uucp/uuchk
Local node name barberouge
Spool directory /var/spool/uucp
Public directory /var/spool/uucppublic
Lock directory /var/lock
Log file /var/log/uucp/log
Statistics file /var/log/uucp/stats
Debug file /var/log/uucp/debug
Global debugging level
uucico -l will strip login names and passwords
uucico will strip UUCP protocol commands
Start uuxqt once per uucico invocation

System: polux
 When called using any login name
 Call out using port ACU
 The possible ports are:
  Port name ACU
   Port type modem
   Device /dev/ttyS0
   Speed 57600
   Carrier available
   Hardware flow control available
   Dialer hayes
    Chat script "" ATZ\r\d\c OK\r \dATX4\r\d\c OK\r ATDT\D CONNECT
    Chat script timeout 60
    Chat failure strings RING NO\sCARRIER ERROR NO\sDIALTONE BUSY NO\sANSWER VOICE
    Chat script incoming bytes stripped to seven bits
    Wait for dialtone ,
```

```
      Pause while dialing ,
      Carrier available
      Wait 60 seconds for carrier
      When complete chat script "" \d\d+++\d\dATH\r\c
      When complete chat script timeout 60
      When complete chat script incoming bytes stripped to seven bits
      When aborting chat script "" \d\d+++\d\dATH\r\c
      When aborting chat script timeout 60
      When aborting chat script incoming bytes stripped to seven bits
   Phone number 0111111110
   Chat script "" \r\c ogin:-BREAK-ogin:-BREAK- \L word: \P
   Chat script timeout 10
   Chat script incoming bytes stripped to seven bits
   Login name uudan
   Password your_password_here
   At any time may call if any work
   May retry the call up to 26 times
   May make local requests when calling
   May make local requests when called
   May send by local request: /
   May send by remote request: /
   May accept by local request: /var/spool/uucppublic
   May receive by remote request: /var/spool/uucppublic
   May execute /usr/bin/uucp /usr/bin/rmail /usr/bin/rnews
   Execution path /bin /usr/bin /usr/local/bin /usr/sbin
   Will leave 50000 bytes available
   Public directory is /var/spool/uucppublic
   Will use protocols gvG
   For protocol G will use the following parameters
    packet-size 1024
    short-packets
```

to check if all the informations you've set are correct.

Warning: your mileage may vary ; different distributions use different paths, regardless Linux FSSTD!

If everything is correct, run:

```
/usr/sbin/uucico -r 1 -x 9 -s remote_system_name
```

The `-x 9` will have maximum debugging information written to the `/var/log/uucp/debug` file for help in initial setup.

I normally run `-x 4` here since that level logs details that help me with login problems. Obviously, this text contains clear informations from your systems file (account/password) so protect it against world-read.

4.11 Additional informations

>From *Pierre.Beyssac at emeraude.syseca.fr*

Taylor has more logging levels. Use -x all to get the highest level possible.

Also, do a `"tail -f /var/log/uucp/debug"` while debugging to watch things happen on the fly.

5. It doesn't work - now what?

In general, you can refer to the documentation mentioned above if things don't work.

You can also refer to your more experienced UUCP neighbors for help. Usually, it's something like a typo anyway.

6. Frequently Asked Questions about Linux UUCP

6.1 Why is all the info here for UUCPs configured in "Taylor" rather than "HDB" mode?

(religious mode on - I know some people are just as religious about "ease of use" as I am about "being standard". That's why they make source code you can build your own from :-))

Because IMHO it's the de-facto standard UUCP implementation at this time. There are thousands of sites with experienced admins & there are many places you can get incredibly good information concerning the HDB setup.

Moreover, if you know what HDB is, you shouldn't be reading this HOWTO :-)

Use the `uuconv` utility in /usr/sbin to convert config files from one mode to another.

6.2 Why do I get "timeout" on connections when I upgraded to uucp-1.04?

- from *Ed Carp - erc at apple.com*

  If you use a "Direct" device in the Devices file, there's now a 10 second timeout compiled in. Make the name of the Device anything other than "Direct". If you tweak the example /etc/uucp files provided, you won't have problems with this one.
- from *Greg Naber - greg at squally.halcyon.com*

  If you get chat script timeouts, you can tweak the sources by editing at line 323 in uuconf/syssub.c & changing the default timeouts from 10 seconds to something larger.
- from *Ed Rodda - ed at orca.wimsey.bc.ca*

  If you get chat script timeouts, typically connecting to other Taylor sites, a pause after login can fix this.

  ```
  feed Any ACU,ag 38400 5551212 ogin: \c\d "" yourname word: passwd
  ```
- from *Dr. Eberhard W. Lisse - el at lisse.NA*

  Some kernels experience modems hanging up after a couple of seconds. The following patch sent by Ian Taylor might help.

  ```
  *** conn.c.orig Mon Feb 22 20:25:24 1993
  --- conn.c      Mon Feb 22 20:33:10 1993
  ***************
  *** 204,209 ****
  --- 204,212 ----

       /* Make sure any signal reporting has been done before we set
         fLog_sighup back to TRUE.  */
  +   /* SMR: it seems to me if we don't care about SIGHUPS, we should clear
  +      the flag before we return  */
  +   afSignal[INDEXSIG_SIGHUP] = FALSE;
  ```

```
            ulog (LOG_ERROR, (const char *) NULL);
            fLog_sighup = TRUE;
```

TOC

6.3 Why doesn't HDB anonymous uucp seem to work?

Taylor in HDB mode seems to be sensitive to white space & blank lines. To be safe, make sure that there are no blank lines or trailing spaces in the Permissions file.

Lastly, make sure that you have a file called `remote.unknown` in /usr/lib/uucp or /etc/uucp & that it's **NOT** executable.

See the O'Reilly & Associates book `"Managing UUCP & USENET"` for details regarding this file.

6.4 What does "no matching ports found" mean?

In all probability, you are attempting to use a dialer that doesn't exist, or the dialer you've specified in the port files doesn't match up with any valid devices in the dial file.

6.5 What are known good config files for HDB mode?

The following are "known-good" config files for Taylor 1.05 under Linux in HoneyDanBer mode. They work on kernels of 0.99-8 or later. All files should be in /usr/lib/uucp or /etc/uucp unless you've tweaked the sources to put the uucp library elsewhere.

If you *HAVE* put things in non-standard places, be aware that things like sendmail might get very confused. You need to ensure that all communications-related programs agree on your idea of "standard" paths.

```
#------------- Devices -------------
# make sure the device (cua1 here) matches your system
# cua N = COM N+1
#
# here "scout" is the Digicom Scout Plus 19.2 modem I use
# tbfast etc. is for a Telebit Trailblazer Plus modem's various speeds
#
ACU cua1 - 19200 scout
ACU cua1 - 9600 tbfast
ACU cua1 - 1200 tbslow
ACU cua1 - 2400 tbmed

#------------- dialers --------------
# note the setting of the Trailblazer registers "on the fly"
# "scout" is a Digicom Scout Plus (Hayes-like) modem I use here
#
scout   =W-,    "" ATM0DT\T CONNECT
tbfast  =W-,    "" A\pA\pA\pT OK ATS50=255DT\T CONNECT\sFAST
tbslow  =W-,    "" A\pA\pA\pT OK ATS50=2DT\T CONNECT\s1200
tbmed   =W-,    "" A\pA\pA\pT OK ATS50=3DT\T CONNECT\s2400

#-------------- Systems -------------
# this is a very generic entry that will work for most systems
#
# the Any;1 means that you can call once per minute with using -f (force)
```

```
# the ACU,g means force "g" protocol rather than Taylor's default "i"
#
fredsys Any;1 ACU,g 19200 scout5555555 "" \r ogin:--ogin: uanon word: uanon

#-------------------------------- Permissions -------------------------

# Taylor UUCP in HDB mode appears to be sensitive to blank lines.
# Make sure all Permissions lines are real or commented out.
#
# this is a anonymous uucp entry
#
LOGNAME=nuucp MACHINE=OTHER \
READ=/var/spool/uucp/nuucp \
WRITE=/var/spool/uucp/nuucp \
SENDFILES=yes REQUEST=yes \
COMMANDS=/bin/rmail
# # this is a normal setup for a remote system that talks to us
# note the absolute path to rnews since this site puts things
# in locations that aren't "standard"
#
LOGNAME=fredsys MACHINE=fredsys \
READ=/var/spool/uucp/fredsys:/var/spool/uucp/uucppublic:/files \
WRITE=/var/spool/uucp/fredsys:/var/spool/uucppublic \
SENDFILES=yes REQUEST=yes \
COMMANDS=/bin/rmail:/usr/bin/rnews
#-----------------------------------------------------------------------
```

6.6 Getting uucico to call alternate numbers

The new v1.05 has an added '-z' switch to uucico that will try alternate numbers for a remote system.

You can else use Taylor mode & "systemyouarecalling-2" (see "sys" in config. files section for more details)

---

7. Acknowledgements

The following people have helped in the assembly of the information & experience that helped make this document possible:

Ed Carp, Steve Robbins, Ian Taylor, Greg Naber, Matt Welsh, Pierre Beyssac & especially many thanks to Vince Skahan for his huge contribution.

If I forgot anybody, my apologies: just email me.

The Unix and Internet Fundamentals HOWTO

by Eric S. Raymond

v1.1, 3 December 1998

This document describes the working basics of PC-class computers, Unix-like operating systems, and the Internet in non-technical language.

1. Introduction

- 1.1 Purpose of this document
- 1.2 Related resources
- 1.3 New versions of this document
- 1.4 Feedback and corrections

2. Basic anatomy of your computer

3. What happens when you switch on a computer?

4. What happens when you run programs from the shell?

5. How do input devices and interrupts work?

6. How does my computer do several things at once?

TOC

7. How does my computer keep processes from stepping on each other?

8. How does my computer store things on disk?

- 8.1 Low-level disk and file system structure
- 8.2 File names and directories
- 8.3 Mount points
- 8.4 How a file gets looked up
- 8.5 How things can go wrong

9. How do computer languages work?

- 9.1 Compiled languages
- 9.2 Interpreted languages
- 9.3 P-code languages

10. How does the Internet work?

- 10.1 Names and locations
- 10.2 Packets and routers
- 10.3 TCP and IP
- 10.4 HTTP, an application protocol

---

1. Introduction

1.1 Purpose of this document

This document is intended to help Linux and Internet users who are learning by doing. While this is a great way to acquire skills, sometimes it leaves peculiar gaps in one's knowledge of the basics -- gaps which can make it hard to think creatively or troubleshoot effectively, from lack of a clear mental model of what is really going on.

I'll try to describe in clear, simple language how it all works. The presentation will be tuned for people using Unix or Linux on PC-class hardware. Nevertheless I'll usually refer simply to `Unix' here, as most of what I will describe is constant across platforms and across Unix variants.

I'm going to assume you're using an Intel PC. The details differ slightly if you're running an Alpha or PowerPC or some other Unix box, but the basic concepts are the same.

I won't repeat things, so you'll have to pay attention, but that also means you'll learn from every word you read. It's a good idea to just skim when you first read this; you should come back and reread it a few times after you've digested what you have learned.

This is an evolving document. I intend to keep adding sections in response to user feedback, so you should come back and review it periodically.

1.2 Related resources

If you're reading this in order to learn how to hack, you should also read the How To Become A Hacker FAQ. It has links to some other useful resources.

1.3 New versions of this document

New versions of the Unix and Internet Fundamentals HOWTO will be periodically posted to comp.os.linux.help and news:comp.os.linux.announce and news.answers. They will also be uploaded to various Linux WWW and FTP sites, including the LDP home page.

You can view the latest version of this on the World Wide Web via the URL http://sunsite.unc.edu/LDP/HOWTO/Fundamentals-HOWTO.html.

1.4 Feedback and corrections

If you have questions or comments about this document, please feel free to mail Eric S. Raymond, at esr@thyrsus.com. I welcome any suggestions or criticisms. I especially welcome hyperlinks to more detailed explanations of individual concepts. If you find a mistake with this document, please let me know so I can correct it in the next version. Thanks.

---

2. Basic anatomy of your computer

Your computer has a processor chip inside it that does the actual computing. It has internal memory (what DOS/Windows people call ``RAM'' and Unix people often call ``core''). The processor and memory live on the *motherboard* which is the heart of your computer.

Your computer has a screen and keyboard. It has hard drives and floppy disks. The screen and your disks have *controller cards* that plug into the motherboard and help the computer drive these outboard devices. (Your keyboard is too simple to need a separate card; the controller is built into the keyboard chassis itself.)

We'll go into some of the details of how these devices work later. For now, here are a few basic things to keep in mind about how they work together:

All the inboard parts of your computer are connected by a *bus*. Physically, the bus is what you plug your controller cards into (the video card, the disk controller, a sound card if you have one). The bus is the data highway between your processor, your screen, your disk, and everything else.

The processor, which makes everything else go, can't actually see any of the other pieces directly; it has to talk to them over the bus. The only other subsystem it has really fast, immediate access to is memory (the core). In order for programs to run, then, they have to be *in core*.

When your computer reads a program or data off the disk, what actually happens is that the processor uses the bus to send a disk read request to your disk controller. Some time later the disk controller uses the bus to signal the computer that it has read the data and put it in a certain location in core. The processor can then use the bus to look at that memory.

Your keyboard and screen also communicate with the processor via the bus, but in simpler ways. We'll discuss those later on. For now, you know enough to understand what happens when you turn on your computer.

3. What happens when you switch on a computer?

A computer without a program running is just an inert hunk of electronics. The first thing a computer has to do when it is turned on is start up a special program called an *operating system*. The operating system's job is to help other computer programs to work by handling the messy details of controlling the computer's hardware.

The process of bringing up the operating system is called *booting* (originally this was *bootstrapping* and alluded to the difficulty of pulling yourself up ``by your bootstraps''). Your computer knows how to boot because instructions for booting are built into one of its chips, the BIOS (or Basic Input/Output System) chip.

The BIOS chip tells it to look in a fixed place on the lowest-numbered hard disk (the *boot disk*) for a special program called a *boot loader* (under Linux the boot loader is called LILO). The boot loader is pulled into core and started. The boot loader's job is to start the real operating system.

The loader does this by looking for a *kernel*, loading it into core, and starting it. When you boot Linux and see "LILO" on the screen followed by a bunch of dots, it is loading the kernel. (Each dot means it has loaded another *disk block* of kernel code.)

(You may wonder why the BIOS doesn't load the kernel directly -- why the two-step process with the boot loader? Well, the BIOS isn't very smart. In fact it's very stupid, and Linux doesn't use it at all after boot time. It was originally written for primitive 8-bit PCs with tiny disks, and literally can't access enough of the disk to load the kernel directly. The boot loader step also lets you start one of several operating systems off different places on your disk, in the unlikely event that Unix isn't good enough for you.)

Once the kernel starts, it has to look around, find the rest of the hardware, and get ready to run programs. It does this by poking not at ordinary memory locations but rather at *I/O ports* -- special bus addresses that are likely to have device controller cards listening at them for commands. The kernel doesn't poke at random; it has a lot of built-in knowledge about what it's likely to find where, and how controllers will respond if they're present. This process is called *autoprobing*.

Most of the messages you see at boot time are the kernel autoprobing your hardware through the I/O ports, figuring out what it has available to it and adapting itself to your machine. The Linux kernel is extremely good at this, better than most other Unixes and *much* better than DOS or Windows. In fact, many Linux old-timers think the cleverness of Linux's boot-time probes (which made it relatively easy to install) was a major reason it broke out of the pack of free-Unix experiments to attract a critical mass of users.

But getting the kernel fully loaded and running isn't the end of the boot process; it's just the first stage (sometimes called *run level 1*).

The kernel's next step is to check to make sure your disks are OK. Disk file systems are fragile things; if they've been damaged by a hardware failure or a sudden power outage, there are good reasons to take

recovery steps before your Unix is all the way up. We'll go into some of this later on when we talk about how file systems can go wrong.

The kernel's next step is to start several *daemons*. A daemon is a program like a print spooler, a mail listener or a WWW server that lurks in the background, waiting for things to do. These special programs often have to coordinate several requests that could conflict. They are daemons because it's often easier to write one program that runs constantly and knows about all requests than it would be to try to make sure that a flock of copies (each processing one request and all running at the same time) don't step on each other. The particular collection of daemons your system starts may vary, but will almost always include a print spooler (a gatekeeper daemon for your printer).

Once all daemons are started, we're at *run level 2*. The next step is to prepare for users. The kernel starts a copy of a program called `getty` to watch your console (and maybe more copies to watch dial-in serial ports). This program is what issues the `login` prompt to your console. We're now at *run level 3* and ready for you to log in and run programs.

When you log in (give a name and password) you identify yourself to `getty` and the computer. It then runs a program called (naturally enough) `login`, which does some housekeeping things and then starts up a command interpreter, the *shell*. (Yes, `getty` and `login` could be one program. They're separate for historical reasons not worth going into here.)

In the next section, we'll talk about what happens when you run programs from the shell.

---

4. What happens when you run programs from the shell?

The normal shell gives you the '$' prompt that you see after logging in (unless you've customized it to something else). We won't talk about shell syntax and the easy things you can see on the screen here; instead we'll take a look behind the scenes at what's happening from the computer's point of view.

After boot time and before you run a program, you can think of your computer of containing a zoo of processes that are all waiting for something to do. They're all waiting on *events*. An event can be you pressing a key or moving a mouse. Or, if your machine is hooked to a network, an event can be a data packet coming in over that network.

The kernel is one of these processes. It's s special one, because it controls when the other *user processes* can run, and it is normally the only process with direct access to the machine's hardware. In fact, user processes have to make requests to the kernel when they want to get keyboard input, write to your screen, read from or write to disk, or do just about anything other than crunching bits in memory. These requests are known as *system calls*.

Normally all I/O goes through the kernel so it can schedule the operations and prevent processes from stepping on each other. A few special user processes are allowed to slide around the kernel, usually by being given direct access to I/O ports. X servers (the programs that handle other programs' requests to do screen graphics on most Unix boxes) are the most common example of this. But we haven't gotten to an X server yet; you're looking at a shell prompt on a character console.

The shell is just a user process, and not a particularly special one. It waits on your keystrokes, listening (through the kernel) to the keyboard I/O port. As the kernel sees them, it echos them to your screen then passes them to the shell. When the kernel sees an `Enter' it passes your line of text to the shell. The shell tries to interpret those keystrokes as commands.

Let's say you type `ls' and Enter to invoke the Unix directory lister. The shell applies its built-in rules to figure out that you want to run the executable command in the file `/bin/ls'. It makes a system call asking the kernel to start /bin/ls as a new *child* process and give it access to the screen and keyboard through the kernel. Then the shell goes to sleep, waiting for ls to finish.

When /bin/ls is done, it tells the kernel it's finished by issuing an *exit* system call. The kernel then wakes up the shell and tells it it can continue running. The shell issues another prompt and waits for another line of input.

Other things may be going on while your `ls' is executing, however (we'll have to suppose that you're listing a very long directory). You might switch to another virtual console, log in there, and start a game of Quake, for example. Or, suppose you're hooked up to the Internet. Your machine might be sending or receiving mail while /bin/ls runs.

5. How do input devices and interrupts work?

Your keyboard is a very simple input device; simple because it generates small amounts of data very slowly (by a computer's standards). When you press or release a key, that event is signalled up the keyboard cable to raise a *hardware interrupt*.

It's the operating system's job to watch for such interrupts. For each possible kind of interrupt, there will be an *interrupt handler*, a part of the operating system that stashes away any data associated with them (like your keypress/keyrelease value) until it can be processed.

What the interrupt handler for your keyboard actually does is post the key value into a system area near the bottom of core. There, it will be available for inspection when the operating system passes control to whichever program is currently supposed to be reading from the keyboard.

More complex input devices like disk or network cards work in a similar way. Above, we referred to a disk controller using the bus to signal that a disk request has been fulfilled. What actually happens is that the disk raises an interrupt. The disk interrupt handler then copies the retrieved data into memory, for later use by the program that made the request.

Every kind of interrupts has an associated *priority level*. Lower-priority interrupts (like keyboard events) have to wait on higher-priority interrupts (like clock ticks or disk events). Unix is designed to give high priority to the kinds of events that need to be processed rapidly in order to keep the machine's response smooth.

In your OS's boot-time messages, you may see references to *IRQ* numbers. You may be aware that one of the common ways to misconfigure hardware is to have two different devices try to use the same IRQ, without understanding exactly why.

Here's the answer. IRQ is short for "Interrupt Request". The operating system needs to know at startup time which numbered interrupts each hardware device will use, so it can associate the proper handlers with each one. If two different devices try use the same IRQ, interrupts will sometimes get dispatched to the wrong handler. This will usually at least lock up the device, and can sometimes confuse the OS badly enough that it will flake out or crash.

6. How does my computer do several things at once?

It doesn't, actually. Computers can only do one task (or *process*) at a time. But a computer can change tasks very rapidly, and fool slow human beings into thinking it's doing several things at once. This is called *timesharing*.

One of the kernel's jobs is to manage timesharing. It has a part called the *scheduler* which keeps information inside itself about all the other (non-kernel) processes in your zoo. Every 1/60th of a second, a timer goes off in the kernel, generating a clock interrupt. The scheduler stops whatever process is currently running, suspends it in place, and hands control to another process.

1/60th of a second may not sound like a lot of time. But on today's microprocessors it's enough to run tens of thousands of machine instructions, which can do a great deal of work. So even if you have many proceses, each one can accomplish quite a bit in each of its timeslices.

In practice, a program may not get its entire timeslice. If an interrupt comes in from an I/O device, the kernel effectively stops the current task, runs the interrupt handler, and then returns to the current task. A storm of high-priority interrupts can squeeze out normal processing; this misbehavior is called *thrashing* and is fortunately very hard to induce under modern Unixes.

In fact, the speed of programs is only very seldom limited by the amount of machine time they can get (there are a few exceptions to this rule, such as sound or 3-D graphics generation). Much more often, delays are caused when the program has to wait on data from a disk drive or network connection.

An operating system that can routinely support many simultaneous processes is called "multitasking". The Unix family of operating systems was designed from the ground up for multitasking and is very good at it -- much more effective than Windows or the Mac OS, which have had multitasking bolted into it as an afterthought and do it rather poorly. Efficient, reliable multitasking is a large part of what makes Linux superior for networking, communications, and Web service.

7. How does my computer keep processes from stepping on each other?

The kernel's scheduler takes care of dividing processes in time. Your operating system also has to divide them in space, so that processes don't step on each others' working memory. The things your operating system does to solve this problem are called *memory management*.

Each process in your zoo needs its own area of core memory, as a place to run its code from and keep variables and results in. You can think of this set as consisting of a read-only *code segment* (containing the process's instructions) and a writeable *data segment* (containing all the process's variable storage). The data segment is truly unique to each process, but if two processes are running the same code Unix automatically arranges for them to share a single code segment as an efficiency measure.

Efficiency is important, because core memory is expensive. Sometimes you don't have enough to hold the entirety of all the programs the machine is running, especially if you are using a large program like an X server. To get around this, Unix uses a strategy called *virtual memory*. It doesn't try to hold all the code and data for a process in core. Instead, it keeps around only a relatively small *working set*; the rest of the process's state is left in a special *swap space* area on your hard disk.

As the process runs, Unix tries to anticipate how the working set will change and have only the pieces that are needed in core. Doing this effectively is both complicated and tricky, so I won't try and describe it all here -- but it depends on the fact that code and data references tend to happen in clusters, with each new one likely to refer to somewhere close to an old one. So if Unix keeps around the code or data most frequently (or most recently) used, you will usually succeed in saving time.

Note that in the past, that "Sometimes" two paragraphs ago was "Almost always," -- the size of core was typically small relative to the size of running programs, so swapping was frequent. Memory is far less expensive nowadays and even low-end machines have quite a lot of it. On modern single-user machines with 64MB of core and up, it's possible to run X and a typical mix of jobs without ever swapping.

Even in this happy situation, the part of the operating system called the *memory manager* still has important work to do. It has to make sure that programs can only alter their own data segments -- that is, prevent erroneous or malicious code in one program from garbaging the data in another. To do this, it keeps a table of data and code segments. The table is updated whenever a process either requests more memory or releases memory (the latter usually when it exits).

This table is used to pass commands to a specialized part of the underlying hardware called an *MMU* or *memory management unit*. Modern processor chips have MMUs built right onto them. The MMU has the special ability to put fences around areas of memory, so an out-of-bound reference will be refused and cause a special interrupt to be raised.

If you ever see a Unix message that says "Segmentation fault", "core dumped" or something similar, this is exactly what has happened; an attempt by the running program to access memory outside its segment has raised a fatal interrupt. This indicates a bug in the program code; the *core dump* it leaves behind is

diagnostic information intended to help a programmer track it down.

8. How does my computer store things on disk?

When you look at a hard disk under Unix, you see a tree of named directories and files. Normally you won't need to look any deeper than that, but it does become useful to know what's going on underneath if you have a disk crash and need to try to salvage files. Unfortunately, there's no good way to describe disk organization from the file level downwards, so I'll have to describe it from the hardware up.

8.1 Low-level disk and file system structure

The surface area of your disk, where it stores data, is divided up something like a dartboard -- into circular tracks which are then pie-sliced into sectors. Because tracks near the outer edge have more area than those close to the spindle at the center of the disk, the outer tracks have more sector slices in them than the inner ones. Each sector (or *disk block*) has the same size, which under modern Unixes is generally 1 binary K (1024 8-bit words). Each disk block has a unique address or *disk block number*.

Unix divides the disk into *disk partitions*. Each partition is a continuous span of blocks that's used separately from any other partition, either as a file system or as swap space. The lowest-numbered partition is often treated specially, as a *boot partition* where you can put a kernel to be booted.

Each partition is either *swap space* (used to implement virtual memory or a *file system* used to hold files. Swap-space partitions are just treated as a linear sequence of blocks. File systems, on the other hand, need a way to map file names to sequences of disk blocks. Because files grow, shrink, and change over time, a file's data blocks will not be a linear sequence but may be scattered all over its partition (from wherever the operating system can find a free block when it needs one).

8.2 File names and directories

Within each file system, the mapping from names to blocks is handled through a structure called an *i-node*. There's a pool of these things near the ``bottom'' (lowest-numbered blocks) of each file system (the very lowest ones are used for housekeeping and labeling purposes we won't describe here). Each i-node describes one file. File data blocks live above the inodes.

Every i-node contains a list of the disk block numbers in the file it describes. (Actually this is a half-truth, only correct for small files, but the rest of the details aren't important here.) Note that the i-node does *not* contain the name of the file.

Names of files live in *directory structures*. A directory structure just maps names to i-node numbers. This is why, in Unix, a file can have multiple true names (or *hard links*); they're just multiple directory entries that happen to point to the same inode.

TOC

8.3 Mount points

In the simplest case, your entire Unix file system lives in just one disk partition. While you'll see this arrangement on some small personal Unix systems, it's unusual. More typical is for it to be spread across several disk partitions, possibly on different physical disks. So, for example, your system may one small partition where the kernel lives, a slightly larger one where OS utilities live, and a much bigger one where user home directories live.

The only partition you'll have access to immediately after system boot is your *root partition*, which is (almost always) the one you booted from. It holds the root directory of the file system, the top node from which everything else hangs.

The other partitions in the system have to be attached to this root in order for your entire, multiple-partition file system to be accessible. About midway through the boot process, your Unix will make these non-root partitions accessible. It will *mount* each one onto a directory on the root partition.

For example, if you have a Unix directory called `/usr', it is probably a mount point to a partition that contains many programs installed with your Unix but not required during initial boot.

8.4 How a file gets looked up

Now we can look at the file system from the top down. When you open a file (such as, say, /home/esr/WWW/ldp/fundamentals.sgml) here is what happens:

Your kernel starts at the root of your Unix file system (in the root partition). It looks for a directory there called `home'. Usually `home' is a mount point to a large user partition elsewhere, so it will go there. In the top-level directory structure of that user partition, it will look for a entry called `esr' and extract an inode number. It will go to that i-node, notice it is a directory structure, and look up `WWW'. Extracting *that* i-node, it will go to the corresponding subdirectory and look up `ldp'. That will take it to yet another directory inode. Opening that one, it will find an i-node number for `fundamentals.sgml'. That inode is not a directory, but instead holds the list of disk blocks associated with the file.

8.5 How things can go wrong

Earlier we hinted that file systems can be fragile things. Now we know that to get to file you have to hopscotch through what may be an arbitrarily long chain of directory and i-node references. Now suppose your hard disk develops a bad spot?

If you're lucky, it will only trash some file data. If you're unlucky, it could corrupt a directory structure or i-node number and leave an entire subtree of your system hanging in limbo -- or, worse, result in a corrupted structure that points multiple ways at the same disk block or inode. Such corruption can be spread by normal file operations, trashing data that was bot in the original bad spot.

Fortunately, this kind of contingency has become quite uncommon as disk hardware has become more reliable. Still, it means that your Unix will want to integrity-check the file system periodically to make sure nothing is amiss. Modern Unixes do a fast integrity check on each partition at boot time, just before

mounting it. Every few reboots they'll do a much more thorough check that takes a few minutes long.

If all of this sounds like Unix is terribly complex and failure-prone, it may be reassuring to know that these boot-time checks typically catch and correct normal problems *before* they become really disasterous. Other operating systems don't have these facilities, which speeds up booting a bit but can leave you much more seriously screwed when attempting to recover by hand (and that's assuming you have a copy of Norton Utilities or whatever in the first place...).

---

9. How do computer languages work?

We've already discussed how programs are run. Every program ultimately has to execute as a stream of bytes that are instructions in your computer's *machine language*. But human beings don't deal with machine language very well; doing so has become a rare, black art even among hackers.

Almost all Unix code except a small amount of direct hardware-interface support in the kernel itself is nowadays written in a *high-level language*. (The `high-level' in this term is a historical relic meant to distinguish these from `low-level' *assembler languages*, which are basically thin wrappers around machine code.)

There are several different kinds of high-level languages. In order to talk about these, you'll find it useful to bear in mind that the *source code* of a program (the human-created, editable version) has to go through some kind of translation into machine code that the machine can actually run.

9.1 Compiled languages

The most conventional kind of language is a *compiled language*. Compiled languages get translated into runnable files of binary machine code by a special program called (logically enough) a *compiler*. Once the binary has been generated, you can run it directly without looking at the source code again. (Most software is delivered as compiled binaries made from code you don't see.)

Compiled languages tend to give excellent performance and have the most complete access to the OS, but also to be difficult to program in.

C, the language in which Unix itself is written, is by far the most important of these (with its variant C++). FORTRAN is another compiled language still used among engineers and scientists but years older and much more primitive. In the Unix world no other compiled languages are in mainstream use. Outide it, COBOL is very widely used for financial and business software.

There used to be many other compiler languages, but most of them have either gone extinct or are strictly research tools. If you are a new Unix developer using a compiled language, it is overwhelmingly likely to be C or C++.

9.2 Interpreted languages

An *interpreted language* depends on an interpreter program that reads the source code and translates it on the fly into computations and system calls. The source has to be re-interpreted (and the interpreter present) each time the code is executed.

Interpreted languages tend to be slower than compiled languages, and often have limited access to the underlying operating system and hardware. On the other hand, they tend to be easier to program and more forgiving of coding errors than compiled languages.

TOC

Many Unix utilities, including the shell and bc(1) and sed(1) and awk(1), are effectively small interp languages. BASICs are usually interpreted. So is Tcl. Historically, the most important interpretive language has been LISP (a major improvement over most of its successors). Today Perl is very widely used and steadily growing more popular.

9.3 P-code languages

Since 1990 a kind of hybrid language that uses both compilation and interpretation has become increasingly important. P-code languages are like compiled languages in that the source is translated to a compact binary form which is what you actually execute, but that form is not machine code. Instead it's *pseudocode* (or *p-code*), which is usually a lot simpler but more powerful than a real machine language. When you run the program, you interpret the p-code.

P-code can can run nearly as fast as a compiled binary (p-code interpreters can be made quite simple, small and speedy). But p-code languages can keep the flexibility and power of a good interpreter.

Important p-code languages include Python and Java.

---

10. How does the Internet work?

To help you understand how the Internet works, we'll look at the things that happen when you do a typical Internet operation -- pointing a browser at the front page of this document at its home on the Web at the Linux Documentation Project. This document is

```
http://sunsite.unc.edu/LDP/HOWTO/Fundamentals.html
```

which means it lives in the file LDP/HOWTO/Fundamentals.html under the World Wide Web export directory of the host sunsite.unc.edu.

10.1 Names and locations

The first thing your browser has to do is to establish a network connection to the machine where the document lives. To do that, it first has to find the network location of the *host* sunsite.unc.edu (`host' is short for `host machine' or `network host'; sunsite.unc.edu is a typical *hostname*). The corresponding location is actually a number called an *IP address* (we'll explain the `IP' part of this term later).

To do this, your browser queries a program called a *name server*. The name server may live on your machine, but it's more likely to run on a service machine that yours talks to. When you sign up with an ISP, part of your setup procedure will almost certainly involve telling your Internet software the IP address of a nameserver on the ISP's network.

The name servers on different machines talk to each other, exchanging and keeping up to date all the information needed to resolve hostnames (map them to IP addresses). Your nameserver may query three or four different sites across the network in the process of resolving sunsite.unc.edu, but this usually happens very quickly (as in less than a second).

The nameserver will tell your browser that Sunsite's IP address is 152.2.22.81; knowing this, your machine will be able to exchange bits with sunsite directly.

10.2 Packets and routers

What the browser wants to do is send a command to the Web server on Sunsite that looks like this:

```
GET /LDP/HOWTO/Fundamentals.html HTTP/1.0
```

Here's how that happens. The command is made into a *packet*, a block of bits like a telegram that is wrapped with three important things; the *source address* (the IP address of your machine), the *destination address* (152.2.22.81), and a *service number* or *port number* (80, in this case) that indicates that it's a World Wide Web request.

Your machine then ships the packet down the wire (modem connection to your ISP, or local network) until it gets to a specialized machine called a *router*. The router has a map of the Internet in its memory, not always a complete one, but one that completely describes your network neighborhood and knows how to get to the routers for other neighborhoods on the Internet.

Your packet may pass through several routers on the way to its destination. Routers are smart. They watch how long it takes for other routers to acknowledge having received a packet. They use that information to direct traffic over fast links. They use it to notice when another routers (or a cable) have dropped off the network, and compensate if possible by finding another route.

There's an urban legend that the Internet was designed to survive nuclear war. This is not true, but the Internet's design is extremely good at getting reliable performance out of flaky hardware in am uncertain world.. This is directly due to the fact that its intelligence is distributed through thousands of routers rather than a few massive switches (like the phone network). This means that failures tend to be well localized and the network can route around them.

Once your packet gets to its destination machine, that machine uses the service number to feed the packet to the web server. The web server can tell where to reply to by looking at the command packet's source IP address. When the web server returns this document, it will be broken up into a number of packets. The size of the packets will vary according to the transmission media in the network and the type of service.

10.3 TCP and IP

To understand how multiple-packet transmissions are handled, you need to know that the Internet actually uses two protocols, stacked one on top of the other.

The lower level, *IP* (Internet Protocol), knows how to get individual packets from a source address to a destination address (this is why these are called IP addresses). However, IP is not reliable; if a packet gets lost or dropped, the source and destination machines may never know it. In network jargon, IP is a *connectionless* protocol; the sender just fires a packet at the receiver and doesn't expect an acknowledgement.

IP is fast and cheap, though. Sometimes fast, cheap and unreliable is OK. When you play networked Doom or Quake, each bullet is represented by an IP packet. If a few of those get lost, that's OK.

The upper level, *TCP* (Transmission Control Protocol), gives you reliability. When two machines negotiate a TCP connection (which they do using IP), the receiver knows to send acknowledgements of the packets it sees back to the sender. If the sender doesn't see an acknowledgement for a packet within some timeout period, it resends that packet. Furthermore, the sender gives each TCP packet has a sequence number, which the receiver can use you reassemble packets in case they show up out of order. (This can happen if network links go up or down during a connection.)

TCP/IP packets also contain a checksum to enable detection of data corrupted by bad links. So, from the point of view of anyone using TCP/IP and nameservers, it looks like a reliable way to pass streams of bytes between hostname/service-number pairs. People who write network protocols almost never have to think about all the packetizing, packet reassembly, error checking, checksumming, and retransmission that goes on below that level.

10.4 HTTP, an application protocol

TOC

Now let's get back to our example. Web browsers and servers speak an *application protocol* that runs on top of TCP/IP, using it simply as a way to pass strings of bytes back and forth. This protocol is called *HTTP* (Hyper-Text Transfer Protocol) and we've already seen one command in it -- the GET shown above.

When the GET command goes to sunsite.unc.edu's webserver with service number 80, it will dispatched to a *server daemon* listening on port 80. Most Internet services are implemented by server daemons that do nothing but wait on ports, watching for and executing incoming commands.

If the design of the Internet has one overall rule, it's that all the parts should be as simple and human-accessible as possible. HTTP, and its relatives (like the Simple Mail Transfer Protocol, *SMTP*, that is used to move electronic mail between hosts) tend to use simple printable-text commands that end with a carriage-return/line feed.

This is marginally inefficient; in some circumstances you could get more speed by using a tightly-coded binary protocol. But experience has shown that the benefits of having commands be easy for human beings to describe and understand outweigh any marginal gain in efficiency that you might get at the cost of making things tricky and opaque.

Therefore, what the server daemon ships back to you via TCP/IP is also text. The beginning of the response will look something like this (a few headers have been suppressed):

```
HTTP/1.1 200 OK
Date: Sat, 10 Oct 1998 18:43:35 GMT
Server: Apache/1.2.6 Red Hat
Last-Modified: Thu, 27 Aug 1998 17:55:15 GMT
Content-Length: 2982
Content-Type: text/html
```

These headers will be followed by a blank line and the text of the web page (after which the connection is dropped). Your browser just displays that page. The headers tell it how (in particular, the Content-Type header tells it the returned data is really HTML).

Linux User Group HOWTO

Kendall Grant Clark

v.1.6.2, 24 April 1998

The Linux User Group HOWTO is a guide to founding, maintaining, and growing a Linux User Group.

1. Introduction

- 1.1 Purpose
- 1.2 Other sources of information

2. What is a Linux User Group?

- 2.1 What is Linux?
- 2.2 How is Linux unique?
- 2.3 What is a user group?
- 2.4 Summary

3. What LUGs are there?

- 3.1 Lists of LUGs
- 3.2 Solidarity versus convenience

4. What does a LUG do?

- 4.1 Linux advocacy
- 4.2 Linux education
- 4.3 Linux support
- 4.4 Linux socializing

TOC

5. Local LUG activities

- 5.1 Meetings
- 5.2 Online resources

6. Practical suggestions

- 6.1 LUG support organizations
- 6.2 Founding a LUG
- 6.3 Maintaining and growing a LUG

7. Legal and political issues

- 7.1 Legal issues
- 7.2 United States
- 7.3 Canada
- 7.4 Political issues

8. About this document

- 8.1 Terms of use
- 8.2 New versions
- 8.3 Please contribute to this HOWTO
- 8.4 Document history
- 8.5 Acknowledgements

---

1. Introduction

1.1 Purpose

The Linux User Group HOWTO is intended to serve as a guide to founding, maintaining, and growing a Linux User Group.

Linux is a freely-distributable implementation of Unix for personal computers, servers and workstations. It was developed on the i386 and now supports i486, Pentium, Pentium Pro, and Pentium II processors, as well as x86-clones from AMD, Cyrix, and others. It also supports many SPARC, DEC Alpha, PowerPC/PowerMac, Motorola 68x0 Mac/Amiga machines.

1.2 Other sources of information

If you want to learn more about Linux, the Linux Documentation Project is a good place to start.

For general information about computer user groups, please see the Association of PC Users Groups.

2. What is a Linux User Group?

2.1 What is Linux?

In order to appreciate and understand fully the significant role of LUGs in the Linux Movement, it is important to understand what makes Linux unique among computer operating systems.

Linux as an operating system is very efficient and very powerful. But, Linux as an ***idea*** about how software ought to be developed is even more powerful. Linux is a **free** operating system: it is licensed under the GNU Public License. The source code is freely available to anyone who wants it and always will be. It is developed by a unstructured group of programmers from around the world, under the technical direction of Linus Torvalds and other key developers. Linux is a world-wide movement without any central structure, bureaucracy, or entity to control, coordinate, or otherwise direct its affairs. While this situation is a powerful part of the appeal and technical quality of Linux as an computer operating system, it can make for inefficient allocation of human resources, ineffective and even detrimental advocacy, public relations, user education and training.

2.2 How is Linux unique?

This loose structure is not likely to change with regard to Linux as a software project. And it's a good thing, too. Linux works precisely because people are free to come and go as they please: **free programmers are happy programmers are effective programmers**.

But this loose structure can make the average Linux user's life a little complicated--especially if that user isn't a programmer by profession or by vocation. Who does she call for support, training, or education? How does she know the kinds of uses for which Linux is well-suited?

In large part local LUGs provide the answers to these kinds of question. This is why LUGs are a crucial part of the Linux Movement. Because there is no ``regional office'' of the Linux Corporation in your town or village or metropolis, the local LUG takes on many of the same roles that a regional office does for a large multi-national corporation..

Linux is unique because it does not have, nor is it burdened by, a central structure or bureaucracy to allocate its resources, train its users, or provide support for its products. These jobs get done in a variety of ways: the Internet, consultants, VARs, support companies, colleges and universities. But, increasingly, in many places around the globe, they get done by a local LUG.

2.3 What is a user group?

Computer user groups, at least in the United States, are not a new phenomenon; in fact, they played an important role in the history of the personal computer. The personal computer arose in large part to

satisfy the demand of electronics, Ham Radio, and other hobbyist user groups, as well as trade shows, swap meets, for affordable, personal access to computing resources. Of course eventually giants like IBM discovered that the PC was a good and profitable thing, but the impetus for the PC came from the people, by the people, and for the people.

In the United States, user groups have changed, and many for the worse, with the times. The financial woes of the largest user group ever, the Boston Computer Society have been well-reported; but all over the U.S. most of the big PC user groups have seen a decline in real membership. American user groups in their heyday concentrated on the production of newsletters, the maintenance of shareware and diskette libraries, meetings, social events, and, sometimes, even Bulletin Board Systems. With the advent of the Internet, however, many of the services that user groups once provided were transferred to things like CompuServe, AOL, and the Web.

The rise of Linux, however, coincided with and was intensified by general public's ``discovery'' of the Internet. As the Internet grew more popular, so did Linux: the Internet brought new users, developers, and vendors to the Linux Movement.

So just when traditional PC user groups were declining because of the Internet's popularity, this popularity propelled Linux forward, creating new demand for new user groups dedicated exclusively to Linux. To give just one indication of the ways in which a LUG is different than a traditional user group, I call the reader's attention to a curious fact: traditional user groups have had to maintain a fairly tight control over the kinds of software that its users copy and trade at its meetings. While illegal copying of commercial software certainly occurred at these meetings, it was officially discouraged and for good reason.

At a LUG meeting, however, this entire mindset simply does not apply. Far from being the kind of thing that a LUG ought to discourage, the free copying of Linux itself ought to be one of the primary activities of a LUG. In fact there is anecdotal evidence that traditional user groups sometimes have a difficult time adapting to the fact that Linux can be freely copied as many times as one needs or wants.

2.4 Summary

In order for the Linux Movement to continue to flourish, the proliferation and success of local LUGs, along with other factors, is an absolute requirement. Because of the unique status of Linux, the local LUG must provide some of the same functions that a ``regional office'' provides for large computer corporations like IBM, Microsoft, or Sun. LUGs can and must train, support, and educate Linux users, coordinate Linux consultants, advocate Linux as a computing solution, and even serve as a liason to local media outlets like newspapers and television.

---

3. What LUGs are there?

Since this document is meant as a guide not only to maintaining and growing LUGs but also to founding them, it would be well before we go much further to determine what LUGs there are.

3.1 Lists of LUGs

There are several lists of LUGs available on the Web. If you want to found a local LUG, one of the first things to do is to determine where the nearest LUG is. *Your best bet may be to join a LUG that is already established in your area rather than founding a new one.*

As of the mid-1997, there are LUGs in all 50 states, the District of Columbia, and 26 other countries, including India, Russia, and most of Western and Eastern Europe.

Note: the biggest untapped computing market on the planet, China, does not yet appear to have a LUG, and India, the second most populous country on the planet, has only a few.

- Finding Groups of Linux Users Everywhere
- LUG List Project
- LUG Registry

It appears that the GLUE list is more comprehensive for American LUGs, while the LUG List Project offers more comprehensive international coverage.

3.2 Solidarity versus convenience

While the lists of LUGs on the Web are well-maintained, it is likely that they do not list every LUG. In addition to consulting these lists, I suggest, if you are considering founding a LUG, that you post a short message asking about the existence of a local LUG to comp.os.linux.announce, comp.os.linux.misc, or an appropriate regional Usenet hierarchy. If there isn't a LUG already in your area, then posting mesages to these groups will alert potential members of your plans.

If you plan to found a local LUG, you should carefully balance convenience against solidarity. In other words, if there is a LUG in your metropolitan area, but on the other side of the city, it may be better to start a new group for the sake of convenience. But it may be better to join the pre-existing group for the sake of unity and solidarity. ***Greater numbers almost always means greater power, influence, and efficiency***. While it might be nice to have two groups of 100 members each, there are certain advantages to one group of 200 members. Of course if you live in a small town or village, any group is better than no group at all.

The point is that starting a LUG is an arduous undertaking, and one that ought to be entered into with all the relevant facts, and with some appreciation of the effect on other groups.

4. What does a LUG do?

The goals of local LUGs are as varied as the locales in which they operate. There is no master plan for LUGs, nor is this document meant to supply one. Remember: Linux is free from bureaucracy and centralized control and so are local LUGs.

It is possible, however, to identify a core set of goals for a local LUG:

- advocacy
- education
- support
- socializing

Each local LUG will combine these and other goals in a unique way in order to satisfy the unique needs of its membership.

4.1 Linux advocacy

The urge to advocate the use of Linux is as natural to computer users as is eating or sleeping. When you find something that works and works well, the natural urge is to tell as many people about it as you can. The role of LUGs in Linux advocacy cannot be overestimated, especially since the wide-scale commercial acceptance of Linux which it so richly deserves has not yet been achieved. While it is certainly beneficial to the Linux Movement each and every time a computer journalist writes a positive review of Linux, it is also beneficial every time satisfied Linux users tell their friends, colleagues, employees or employers about Linux.

There is effective advocacy and there is ineffective carping: as Linux users, we must be constantly vigilant to advocate Linux in such a way as to reflect positively on both the product, its creators and developers, and our fellow users. The Linux Advocacy mini-HOWTO, available at the Linux Documentation Project, gives some helpful suggestions in this regard. Suffice it to say that advocacy is an important aspect of the mission of a local LUG.

There may come a time when Linux advocacy is pretty much beside the point because Linux has more or less won the day, when the phrase ``No one ever got fired for using Linux'' becomes a reality. Until that time, however, the local LUG plays an indispensable role in promoting the use of Linux. It does so because its advocacy is free, well-intentioned, and backed up by organizational commitment. If a person comes to know about Linux through the efforts of a local LUG, then that person, as a new Linux user, is already ahead of the game: *she is already aware of the existence of an organization that will help her install, configure, and even maintain Linux on whatever computers she is willing to dedicate to it.*

New Linux users who are already in contact with a local LUG are ahead of those whose interest in Linux has been piqued by a computer journalist, but who have no one to whom to turn to aid them in their quest to install, run, and learn Linux.

TOC

It is, therefore, important for local LUGs to advocate Linux because their advocacy is effective, well-supported, and free.

4.2 Linux education

Not only is it the business of a local LUG to advocate the use of Linux, it may also turn its efforts to training its members, as well as the computing public in its area, to use Linux and associated components. In my own estimation, the goal of user education is the single most important goal a LUG may undertake. Of course, as I have already pointed out, LUGs are perfectly free to organize themselves and their activities around any of these, or other, goals. I believe, however, that LUGs can have the greatest impact on the Linux Movement by educating and training Linux users.

Local LUGs may choose to undertake the goal of education simply because there is no other local entity from which a Linux user may receive technically-oriented education. While it is certainly the case that universities, colleges, and junior colleges are increassingly turning to Linux as a way to educate their students, both efficiently and cheaply, about Unix-like operating systems, some Linux users are either unable or unwilling to register for courses in order to learn Linux. For these users the local LUG is a valuable resource for enhancement or creation of advanced computer skills: Unix-like system administration, system programming, support and creation of Internet and Intranet technologies, etc.

In an ironic twist, many local LUGs are even sharing the burden of worker training with large corporations. Every worker at Acme Corp that expands her computer skills by participating in a local LUG is one less worker Acme Corp has to train or pay to train. Even though using and administering a Linux PC at home isn't the same as administering a corporate data warehouse, call center, or similar high-availability facility, it is light years more complex, more rewarding, and more educational than using and administering a Windows 95 PC at home. As Linux itself advances toward things like journalling filesystems, high-availability, real-time capacity, and other high-end Unix features, the already blurry line between Linux and the ``real'' Unixes will get even more indistinct.

Not only is such education a form of worker training, but it will also serve, as information technology becomes an increasingly vital part of the global economy, as a kind of community service. In most metropolitan areas in the United States, for example, it is possible for a local LUG to take Linux into local schools, small businesses, community and social organizations, and other non-corporate environments. This accomplishes the task of Linux advocacy and also helps train the general public about Linux as a Unix-like operating system. As more and more of these kinds of organizations seek to establish an Internet presence or provide dial-in access to their workers, students, and constituents, the opportunities arise for local LUGs to participate in the life of their community by educating it about a free and freely-available operating system. This kind of community service allows the average Linux user to emulate the kind of generosity that has characterized Linux, and the free software community, from the very beginning. Most Linux users can't program like Linus Torvalds, but we can all all give our time and abilities to other Linux users, the Linux community, and the broader community in which work and live.

Linux is a natural fit for these kinds of organization because deploying it doesn't commit them to expensive license, upgrade, or maintenance fees. Because Linux is also technically elegant and economical, it runs very well on the the kinds of disposable hardware that corporations typically cast off

and that non-profit organizations are only too happy to use. As more and more people discover every day, that old 486 collecting dust in the closet can do **real work** if someone will install Linux on it.

In addition, Linux education has a cumulative effect on the other goals of a local LUG, in particular the goal of Linux support discussed below. Better Linux education means better Linux support. The more people that a LUG can count on to reach its support goals, the easier support becomes and, therefore, the more of it can be done. The more new and inexperienced users a local LUG can support and eventually educate about Linux, the larger and more effective the LUG can become. In other words, if a LUG focuses solely on Linux support to the neglect of Linux education, the natural barriers to organizational growth will be more restrictive. If only two or three percent of the members of a LUG take upon themselves the task of supporting the others, the growth of the LUG will be stifled. One thing you can count on: ***if new and inexperienced users don't get the help with Linux they need from a local LUG, they won't participate in that LUG for very long***. If a larger percentage of members support the others, the LUG will be able to grow much larger. Linux education is the key to this dynamic: education turns new Linux users into experienced ones.

Free education about free Linux also highlights the degree to which Linux is part and parcel of the free software Community. So it seems appropriate that local LUGs focus not solely on Linux education but also education about all of the various software systems and technologies that run under Linux. These include, for instance, the GNU suite of programs and utilities, the Apache Web server, the XFree86 implementation of X Windows, TeX, LaTeX, etc. Fortunately the list of free software that runs under Linux is a long and diverse one.

Finally, Linux is a self-documenting operating environment; in other words, if we don't write the documentation, nobody is going to do it for us. Toward that end, make sure that LUG members are well aware of the Linux Documentation Project, which can be found at mirrors worldwide. Consider providing an LDP mirror for the local Linux community and for LUG members. Also make sure to publicize---through `comp.os.linux.announce`, the LDP, and other pertinent sources of Linux information---any relevant documentation that is developed by the LUG: technical presentations, tutorials, local FAQs, etc. There is a lot of Linux documentation produced in LUGs that doesn't benefit the worldwide Linux community because no one outside the LUG knows about it. Don't let the LUGs efforts in this regard go to waste: it is highly probable that if someone at one LUG had a question or problem with something, then people at other LUGs around the world will have the same questions and problems.

4.3 Linux support

Of course for the desperate **newbie** the primary role of a local LUG is Linux support. But it is a mistake to suppose that Linux support only means *technical* support for new Linux users. It can and should mean much more.

Local LUGs have the opportunity to support:

- users
- consultants
- businesses, non-profit organizations, and schools

- the Linux Movement

Users

The most frequent complaint from new Linux users, once they have gotten Linux installed, is the steep learning curve which is not at all unique to Linux but is, rather, a characteristic of all modern Unixes. With the steepness of the learning curve, however, comes the power and flexibility of a complex operating system. A local LUG is often the only resource that a new Linux user has available to help flatten out the learning curve.

But even if a new Linux user doesn't know it yet, she needs more than just technical support: Linux and the free software worlds are both rapidly moving targets. The local LUGs form an invaluable conduit of information about Linux and other free software products. Not only does Linux lack a central bureaucracy, but it also for the most part lacks the kind of journalistic infrastructure from which users of other computer systems benefit. The Linux Movement does have resources like Linux Journal and Linux Gazette, but many new Linux users are unaware of these resources. In addition, as *monthly* publications they are often already out of date about bugfixes, security problems, patches, new kernels, etc. This is where the local LUG as a source and conduit of timely information is so vital to new and experienced Linux users alike.

For example, until a new Linux user knows that the newest kernels are available from ftp.kernel.org or that the **Linux Documentation Project** usually has newer versions of Linux HOWTOs than a CD-based Linux distribution, it is up to the local LUG, as the primary support entity, to be a conduit of timely and useful information.

In fact it may be just a bit misleading to focus on the support role that local LUGs provide to new users: intermediate and advanced users also benefit from the proliferation of timely and useful tips, facts, and secrets about Linux. Because of the complexity of Linux, even advanced users often learn new tricks or techniques simply by becoming involved in a local LUG. Sometimes they learn about software packages they didn't know existed, sometimes they just remember that arcane `vi` command sequence they've not used since college.

Consultants

It is, I think, rather obvious to claim that local LUGs ought to be in the business of supporting new Linux users. After all, if they're not supposed to be doing that, what are they to do? It may not be as obvious that local LUGs can play an important role in supporting local Linux consultants. Whether they do Linux consulting full-time or only part-time, consultants can be an important part of a local LUG. How can the LUG support them?

The answer to that question is just the answer to another question: what is it that Linux consultants want and need? ***They need someone for whom to consult.*** A local LUG provides the best way for those who *offer* Linux consulting to find those who *need* Linux consulting. The local LUG can informally broker connections between consulting suppliers and consulting consumers simply by getting all, or as many as possible, of the people interested in Linux in a local area together and talking with one another. How LUGs do that will occupy us below. What is important here is to point out that LUGs can and should play this role as well. The Linux Consultants HOWTO is an important document in this regard, but it is

TOC

surely the case that only a fraction of the full-time and part-time Linux consultants worldwide are registered in the Consultants HOWTO.

The relationship is mutually beneficial. Consultants aid LUGs by providing experienced leadership, both technically and organizationally, while LUGs aid consultants by putting them in contact with the kinds of people who need their services. New and inexperienced users gain benefit from both LUGs and consultants since their routine or simple requests for support are handled by LUGs *gratis*, and their complex needs and problems---the kind that obviously require the services of a paid consultant---can be handled by the consultants whom the local LUG helps them contact.

The line between support requests that need a consultant and those that do not is sometimes indistinct; but in most cases the difference is clear. While a local LUG doesn't want to gain the reputation for pawning new users off unnecessarily on consultants--as this is simply rude and very anti-Linux behavior--there is no reason for LUGs not to help broker contacts between the users who need consulting services and the professionals who offer them.

Please see Martin Michlmayr's Linux Consultants HOWTO for an international list of Linux consultants.

Businesses, non-profit organizations, and schools

LUGs also have the opportunity to support local businesses and organizations. This support has two aspects. First, LUGs can support businesses and organizations that want to use Linux as a part of their computing and IT efforts. Second, LUGs can support local businesses and organizations that develop for Linux, cater to Linux users, support or install Linux, etc.

The kinds of support that LUGs can provide to local businesses that want to use Linux as a part of their computing operations isn't really all that different from the kinds of support LUGs give to individuals who want to run Linux at home. For example, compiling the Linux kernel doesn't really vary from home to business. Supporting businesses using Linux, however, may mean that a LUG needs to concentrate on commercial software that runs on Linux, rather than concentrating solely on free software. If Linux is going to continue to maintain its momentum as a viable computing alternative, then it's going to take software vendors who are willing to write for and port to Linux as a commercially-viable platform. If local LUGs can play a role in helping business users evaluate commercial Linux solutions, then more software vendors will be encouraged to consider Linux in their development and planning.

This leads us directly to the second kind of support that a local LUG can give to local businesses. Local LUGs can serve as a clearing house for the kind of information that is available in very few other places. For example:

- Which local ISP is Linux-friendly?
- Are there any local hardware vendors that build Linux PCs?
- Does anyone sell Linux CDs locally?

Maintaining and making this kind of information public not only helps the members of a local LUG, but it also helps Linux-friendly local businesses as well, and it encourages them to continue to be Linux-friendly. It may even, in some cases, help contribute to a competitive atmosphere in which other businesses are encouraged to become Linux-friendly too.

TOC

Free software development

Finally, LUGs may also support the Linux Movement by soliciting and organizing charitable giving. Chris Browne has thought about this issue as much as anyone I know, and he contributes the following.

Chris Browne on free software philanthropy

A further involvement can be to encourage sponsorship of various Linux-related organizations in a financial way. With the multiple millions of Linux users, it would be entirely plausible for grateful users to individually contribute a little. Given millions of users, and the not unreasonable sum of a hundred dollars of ``gratefulness'' per Linux user ($100 being roughly the sum *not* spent this year upgrading a Microsoft OS), that could add up to *hundreds of millions* of dollars towards development of improved tools and applications for Linux.

A users group can encourage members to contribute to various ``development projects.'' If it has some form of ``charitable tax exemption'' status, that can encourage members to contribute directly to the group, getting tax deductions as appropriate, with contributions flowing on to other organizations.

It is appropriate, in any case, to encourage LUG members to direct contributions to organizations with projects and goals that they individually wish to support.

This section lists possible candidates. None are explicitly being recommended here, but the list can represent useful ``food for thought.'' Many are registered as charities in the United States, thus making U.S. contributions tax deductible.

Here are organizations with activities particularly directed towards development of software that works with Linux:

- Linux International Project Sponsorship Fund
- Debian/Software In the Public Interest
- Free Software Foundation
- The XFree86 Project

Contributions to these organizations has the direct effect of supporting the creation of freely redistributable software usable with Linux. Dollar for dollar, such contributions almost certainly have greater effect on the Linux community as a whole than any other specific kind of spending.

There are also organizations that are less directly associated with Linux that may nonetheless be worthy of assistance, such as:

- League for Programming Freedom

  This is not a Linux-specific organization; they are involved in general advocacy activities that touch on people involved with software development. Involvement in this organization represents something closer to involvement in a ``political lobby'' group.

  There is somewhat of a ``USA bias;'' there are nonetheless international implications, and the international community as often follows the American lead in computing-related matters as

vice-versa.

- The LaTeX3 Project Fund

  The TeX Users Group (TUG) is working on the ``next generation'' version of the LaTeX publishing system, known as LaTeX3. Linux is one of the platforms on which TeX and LaTeX are best supported.

  Donations for the project can be sent to:

  ```
  TeX Users Group
  P.O. Box 1239
  Three Rivers, CA 93271-1239
  USA
  ```

  or, for those in Europe,

  ```
  UK TUG
  1 Eymore Close
  Selly Oaks
  Burmingham B29 4LB
  UK
  ```

- Project Gutenberg

  Their purpose is to make freely available in electronic form the texts of out-of-copyright books. This isn't directly a ``Linux thing,'' but it seems fairly worthy, and they actively encourage platform independence, which means that their ``products'' are quite usable with Linux.

Linux Movement

I have referred throughout this HOWTO to something I call the **Linux Movement**. There really is no better way to describe the international Linux phenomenon than to call it a movement: it isn't a bureaucracy, but it is organized; it isn't a corporation, but it is important to businesses all over the world. The best way for a local LUG to support the international Linux movement is to work to insure that the local Linux community is robust, vibrant, and growing. Linux is *developed* internationally, which is easy enough to see by reading /usr/src/linux/MAINTAINERS. But Linux is also *used* internationally. And this ever-expanding user base is the key to Linux's continued success. And that is where the local LUG plays an incalculably important role.

The strength of the Linux Movement internationally is the simple fact that Linux offers unprecedented computing power and sophistication for its cost and for its freedom. The keys are value and independence from proprietary control. Every time a new person, group, business, or organization has the opportunity to be exposed to Linux's inherent value the Linux Movement grows in strength and numbers. Local LUGs can make that happen.

4.4 Linux socializing

TOC

The last goal of a local LUG that I will mention here is socializing. In some ways this is the most difficult goal to discuss because it is not clear how many or to what degree LUGs engage in it. While it would be strange to have a local LUG that didn't engage in the other goals, there very well may be local LUGs somewhere in the world for which socialization isn't an important consideration.

It seems, however, that whenever two or three Linux users get together fun, highjinks, and, often, beer are sure to follow. Linus Tovalds has always had one enduring goal for Linux: to have more fun. For hackers, kernel developers, and Linux users, there's nothing quite like downloading a new kernel, recompiling an old one, twittering with a window manager, or hacking some code. It is the sheer fun of Linux that keeps many LUGs together, and it is this kind of fun that leads many LUGs naturally to socializing.

By ``socializing'' here I mean primarily sharing experiences, forming friendships, and mutually-shared admiration and respect. There is another meaning, however, one that social scientists call *socialization*. In any movement, institution, or human community, there is the need for some process or pattern of events in and by which, to put it in Linux terms, newbies are turned into hackers. In other words, socialization turns you from ``one of them'' to ``one of us''.

For armed forces in the U.S. and in most countries, this process is called boot camp or basic training. This is the process whereby civilians are transformed into soldiers. The Linux movement has analogous requirements. It is important that new Linux users come to learn what it means to be a Linux user, what is expected of them as a member of an international community, the special vocabulary of the Linux movement, its unique requirements and opportunities. This may be as simple as how Linux users in a partcicular locale pronounce ``Linux''. It may be as profound as the ways in which Linux users should advocate, and the ways in which they should, more importantly, *refrain* from advocating Linux.

Linux socialization, unlike `real world' socialization, can occur on mailing lists and Usenet, although the efficacy of the latter is constantly challenged precisely by poorly socialized users. In my view, socialization and socializing are both done best in the company of real, flesh-and-blood fellow human beings, and not by incorporeal voices on a mailing list or Usenet group.

---

5. Local LUG activities

In the previous section I focused exclusively on what LUGs do and what they ought to be doing. In this section the focus shifts to practical strategies for accomplishing these goals.

There are, despite the endless permutations of form, two basic things that local LUGs do: first, they meet together in physical space; second, they communicate with each other in cyberspace. Everything or nearly everything that LUGs do can be seen in terms of meetings and online resources.

5.1 Meetings

As I said above, physical meetings are synonymous with LUGs and with most computer user groups. LUGs have these kinds of meetings:

- social
- technical presentations
- informal discussion groups
- user group business
- Linux installation
- configuration and bug-squashing

What do LUGs do at these meetings?

- Install Linux for newbies and strangers
- Teach members about Linux
- Compare Linux to other operating systems
- Teach members about the software that runs on Linux
- Discuss the ways in which Linux can be advocated
- Discuss the importance of the Free Software Movement
- Discuss the business of the user group
- Eat, drink, and be merry

5.2 Online resources

The commercial rise of the Internet coincided roughly with the rise of Linux, and the latter in large part owes something to the former. The Internet has always been an important asset for Linux development. It is no different for LUGs. Most LUGs have web pages if not whole Web sites. In fact, I am not sure how else to find a local LUG but to check the Web.

It makes sense, then, for a local LUG to make use of whatever Internet technologies they can

appropriate: Web sites, mailing lists, gopher, FTP, e-mail, WAIS, finger, news, etc. As the world of commerce is discovering, the Internet can be an effective way to advertise, inform, educate, and even sell. The other reason that LUGs make extensive use of Internet technologies is that it is the very essence of Linux to *provide* a stable and rich platform for the deployment of these technologies. So not only do LUGs benefit from, say, the establishment of a Web site because it advertizes their existence and helps organize their members, but in deploying these technologies, the members of the LUG are provided an opportunity to learn about this technology and see Linux at work.

Some LUGs that use the Internet effectively:

- Atlanta Linux Enthusiasts
- North Texas Linux Users Group
- Boston Linux and Unix
- Colorado Linux Users and Enthusiasts
- BLUG - BHZ Linux Users Group (Brazil)
- Ottawa Carleton Linux Users Group
- Provence Linux Users Group
- Duesseldorf Linux Users Group
- Linux User Group Austria
- Israeli Linux Users Group
- Tokyo Linux Users Group
- Linux in Mexico
- Netherlands Linux Users Group (NLLGG)
- St. Petersburg Linux User Group
- Linux User Group of Singapore
- Victoria Linux User Group
- Essex Linux User Group
- Turkish Linux User Group
- Linux User Group of Rochester
- Korean Linux Users Group

Please let me know if your LUG uses the Internet in an important or interesting way; I'd like this list to include your group.

---

6. Practical suggestions

Finally, I want to make some very practical, even mundane, suggestions for anyone wanting to found, maintain, or grow a LUG.

6.1 LUG support organizations

There are several organizations that offer assistance to local LUGs.

GLUE

Groups of Linux Users Everywhere is a user group coordination and support program started by SSC, the same people who publish *Linux Journal*. The GLUE program is an inexpensive way for a local LUG to provide some benefits to its membership.

Linux Systems Labs

LSL offers their Tri-Linux Disk set (Three Linux distributions on four CDs: Red Hat, Slackware, and Debian) to LUGs for resale at a considerable discount.

Linux Mall User Group Program

Sponsored by WorkGroup Solutions, the Linux Mall User Group Program offers a range of benefits for participating User Groups. LUGs are also free to participate in Linux Mall's Referral Program as well.

Cleveland Linux User's Group

Owns the Internet domain, `lug.net`. They will provide your LUG an Internet domain name at `lug.net`: your-LUG-name-or-city`lug.net`. More information may be found at LUG.NET or by e-mailing Jeff Garvas.

Red Hat Software's User Group Program

Assists LUGs to develop and grow. More information may be found at Red Hat Web site

6.2 Founding a LUG

- Determine the nearest pre-existing LUG
- Announce your intentions on `comp.os.linux.announce` and on an appropriate regional hierarchy
- Announce your intention wherever computer users are in your area: bookstores, swap meets, cybercafes, colleges and universities, corporations, Internet service providers, etc.
- Find Linux-friendly businesses or institutions in your area that may be willing to help you form the

LUG

- Form a mailing list or some means of communication between the people who express an interest in forming a LUG
- Ask key people specifically for help in spreading the word about your intention to form a LUG
- Solicit space on a Web server to put a few HTML pages together about the group
- Begin looking for a meeting place
- Schedule an initial meeting
- Discuss at the initial meeting the goals for the LUG

6.3 Maintaining and growing a LUG

- Make the barriers to LUG membership as low as possible
- Make the LUG's Web site a priority: keep all information current, make it easy to find details about meetings (who, what, and where), and make contact information and feedback mechanisms prominent
- Install Linux for anyone who wants it
- Post flyers, messages, or handbills wherever computer users are in your area
- Secure dedicated leadership
- Follow Linus's *benevolent dictator* model of leadership
- Take the big decisions to the members for a vote
- Start a mailing list devoted to technical support and ask the ``gurus'' to participate on it
- Schedule a mixture of advanced and basic, formal and informal, presentations
- Support the software development efforts of your members
- Find way to raise money without dues: for instance, selling Linux merchandise to your members and to others
- Consider securing formal legal standing for the group, such as incorporation or tax-exempt status
- Find out if your meeting place is restricting growth of the LUG
- Meet in conjunction with swap meets, computer shows, or other community events where computer users---i.e., potential Linux converts---are likely to gather
- Elect formal leadership for the LUG as soon as is practical: some helpful officers might include President, Treasurer, Secretary, Meeting Host (general announcements, speaker introductions, opening and closing remarks, etc.), Publicity Coordinator (handles Usenet and e-mail postings, local publicity), and Program Coordinator (organizes and schedules speakers at LUG meetings)
- Provide ways for members and others to give feedback about the direction, goals, and strategies of the LUG
- Support Linux and Free Software development efforts by donating Web space, a mailing list, or FTP site
- Establish an FTP site for relevant software
- Archive everything the LUG does for the Web site

TOC

- Solicit ``door prizes'' from Linux vendors, VARs, etc. to give away at meetings
- Give credit where credit is due
- Join SSC's GLUE (Groups of Linux Users Everywhere) but be aware they charge a membership fee
- Submit your LUG's information to all of the Lists of LUGs
- Publicize your meetings on appropriate Usenet groups and in local computer publications and newspapers
- Compose promotional materials, like Postscript files, for instance, that members can use to help publicize the LUG at workplaces, bookstores, computer stores, etc.
- Make sure you know what LUG members want the LUG to do
- Release press releases to local media outlets about any unusual LUG events like an Installation Fest, Net Day, etc.
- Use LUG resources and members to help local non-profit organizations and schools with their Information Technology needs
- Advocate the use of Linux zealously but responsibly
- Play to the strengths of LUG members
- Maintain good relations with Linux vendors, VARs, developers, etc.
- Identify and contact Linux consultants in your area
- Network with the leaders of other LUGs in your area, state, region, or country to share experiences, tricks, and resources
- Keep LUG members advised on the state of Linux software---new kernels, bugs, fixes, patches, security advisories---and the state of the Linux world at large---new ports, trademark and licensing issues, where Linus is living and working, etc.
- Notify the Linux Documentation Project---and other pertinent sources of Linux information---about the documentation that the LUG produces: technical presentations, tutorials, local HOWTOs, etc.

---

7. Legal and political issues

7.1 Legal issues

7.2 United States

There is a strong case to be made for formal organization of local LUGs. I will not make that case here. If, however, you are interested in formally organizing your local LUG, then this section will introduce you to some of the relevant issues.

Note: this section should not be construed as competent legal counsel. These issues require the expertise of competent legal counsel; you should, before acting on any of the statements made in this section, consult an attorney.

There are at least two different legal statuses that a local LUG in the United States may attain:

1. incorporation as a non-profit entity
2. tax-exemption

Although the relevant statutes differ from state to state, most states allow user groups to incorporate as non-profit entitites. The benefits of incorporation for a local LUG may include limitations of liability of LUG members and volunteers, as well as limitation or even exemption from state corporate franchise taxes.

While you should consult competent legal counsel before incorporating your LUG as a non-profit entity, you can probably reduce your legal fees if you are acquainted with the relevant issues before consulting with an attorney. I recommend the *Non-Lawyers Non-Profit Corporation Kit* (ISBN 0-937434-35-3).

As for the second status, tax-exemption, this is not a legal status so much as a judgment by the Internal Revenue Service. It is important for you to know that incorporation as a non-profit entity **does not** insure that the IRS will rule that your LUG is to be tax-exempt. It is possible to have a non-profit corporation that is **not** also tax-exempt.

The IRS has a relatively simple document that explains the criteria and process for tax-exemption. It is **Publication 557:** *Tax-Exempt Status for Your Organization*. It is available as an Adobe Acrobat file from the IRS's Web site. I strongly recommend that you read this document **before** filing for incorporation as a non-profit entity. While becoming a non-profit corporation cannot insure that your LUG will be declared tax-exempt by the IRS, there are ways to incorporate that will **prevent** the IRS from declaring your LUG to be tax-exempt. *Tax-Exempt Status for Your Organization* clearly sets out the necessary conditions for your LUG to be declared tax-exempt.

Finally, there are resources available on the Internet for non-profit and tax-exempt organizations. Some of the material is probably relevant to your local LUG.

7.3 Canada

Thanks to Chris Browne for the following comments about the Canadian situation.

The Canadian tax environment strongly parallels the US environment, in that the ``charitable organization'' status confers similar tax advantages for donors over mere ``not for profit'' status, while requiring that similar sorts of added paperwork be filed by the ``charity'' with the tax authorities in order to attain and maintain certified charity status.

7.4 Political issues

Chris Browne has the following to say about the kinds of inter-LUG political dynamics that often crop up.

People have different feelings about free software.

Linux users are a diverse bunch. As soon as you try to put a lot of them together, there are *some* problem issues that can come up.There are those that are nearly political radicals that believe that all software, always, should be ``free.'' Because Caldera charges quite a lot of money for their distribution, and doesn't give all profits over to *(pick favorite advocacy organization)*, they must be ``evil.'' Ditto for Red Hat or S.u.S.E. Keep in mind that all three of these companies have made and continue to make significant contributions to free software.

Others may figure that they can find some way to highly exploit the ``freeness'' of the Linux platform for their fun and profit. Be aware that many users of the BSD UNIX variants consider that *their* licenses that *do* permit companies to build ``privatized'' custom versions of their OSes are preferable to the ``enforced permanent freeness'' of the GPL as applied to Linux. Do not presume that all people promoting this sort of view are necessarily greedy leeches.

If these people are put together in one place, disagreements can occur.

Leaders should be clear on the following facts:

- There are a lot of opinions about the GPL and how it is supposed to work. It is easy to misunderstand both the GPL and alternative licensing schemes.
- Linux benefits from contributions from many places, and can support some freeloaders, particularly if this encourages more people to get involved, thus pulling in further contributors.
- Many significant contributions have been made to Linux by commercial enterprises. Examining the sources to the Linux kernel, and notable subsystems such as XFree86 and GCC show a surprising number of commercial contributors.
- Commercial does not always imply ``better,'' but it also does not always imply ``horrible.''

The main principle can be extended well beyond this; computer ``holy wars'' have long been waged over the virtues of one system over another, whether that be (in modern day) between Linux, other UNIX variants, and Microsoft OSes, or between the ``IBM PC'' and the various Motorola 68000-based systems, or between the many 8 bit systems of the 1970s. Or of KDE versus GNOME.

TOC

A wise LUG leader will seek to smooth over such differences, rather than inciting them. LUG leaders *must* have thick skins.

There *will* be disagreements at some point as diverse views collide with one another, and leaders must be able to cope with this, resolving disagreements rather than contributing to the problem.

Nonprofit organizations and money don't mix terribly well.

It is important to be quite careful in dealing with finances in a nonprofit organization of any sort. In businesses, where profitable flows of monies are the goal, people are not typically too worried about ``nagging details'' such as possible misspending of immaterial sums of money.

The same cannot be said about nonprofit organizations. Some people are involved for reasons of principle, and can easily give minor problems inordinate attention. And the potential for wide participation at business meetings correspondingly expands the potential for inordinate attention to be drawn to things.

As a result, it is probably preferable for there to *not* be a membership fee for a LUG, as that provides a specific thing for which people can reasonably demand accountability. Fees that are not collected cannot, by virtue of the fact that they don't exist, be misused.

If there *is* a lot of money and/or other such resources floating around, it is important for the user group to be accountable to its members for it.

In a vital, growing group, there should be more than one person involved. In troubled nonprofit organizations, financial information is often tightly held by someone who will not willingly relinquish control of funds. Ideally, there should be *some* rotation of duties in a LUG including that of control of the finances.

Regular useful financial reports should be made available to those that wish them. A LUG that maintains an official ``charitable status'' for tax purposes will have to file at least annual financial reports with the local tax authorities, which would represent a minimum financial disclosure for the purposes of the members.

With the growth of Linux-based financial software, it should be readily possible to create reports on a regular basis. With the growth of the Internet, it should even be possible to publish these on the World Wide Web.

---

8. About this document

8.1 Terms of use

Copyright (c) 1997 by Kendall Grant Clark. This document may be distributed under the terms set forth in the LDP license at http://sunsite.unc.edu/LDP/COPYRIGHT.html.

8.2 New versions

New versions of the Linux User Group HOWTO will be periodically uploaded to various Linux WWW and FTP sites, principally my homepage and the Linux Documentation Project

8.3 Please contribute to this HOWTO

I welcome questions about and feedback on this document. Please send them to me at kclark@cmpu.net. *I am especially interested in hearing from leaders of LUGs from around the world*. I would like to include real-life examples of the things described here. I would also like to include a section on LUGs outside the United States, since this HOWTO as it stands now is rather US-centric. Please let me know if your group does things that should be mentioned in this HOWTO.

8.4 Document history

- 1.0 released on 13 July 1997
- 1.1: expanded online resources section
- 1.3: added LUG Support Organizations and expanded the Legal and Organizational Issues section
- 1.3.1: general editing for clarity and conciseness
- 1.4: general editing, added new LUG resources
- 1.4.1: general editing for clarity
- 1.5: added some resources, some discussion of LUG documentation, also general editing
- 1.5.1: changed Web location for this document and author's email address.
- 1.5.2: new copyright and license
- 1.5.3: miscellaneous edits and minor re-organizations
- 1.6: added Chris Browne's material: Linux philanthropic donations and LUG political considerations
- 1.6.1: very minor additions
- 1.6.2: minor corrections

8.5 Acknowledgements

I want to thank all the great people I've met and worked with during the time I've served as President of the North Texas Linux Users Group. They helped inspire me to use Linux full-time. The best thing about Linux really is the people you meet.

I especially want to thank Chris Browne for describing the situation with non-profit and charitable groups in Canada, his thoughts on financial donations as a way to participate in Linux and the free software movement, and his ideas about the kinds of political issues that may arise within LUGs.

In addition, the following people have made helpful comments and suggestions:

- Hugo van der Kooij
- Greg Hankins
- Charles Lindahl
- Rick Moen
- Jeff Garvas
- James Hertzler

Linux VAR HOWTO

Mr. Poet poet@linuxports.com

v2.84, 22 August 1999

This document contains a listing of service companies which do not manufacture hardware or create packaged software, but add value to existing products. It is maintained by Mr. Poet poet@linuxports.com.

1. About this document

2. Copyright Information

3. VAR ID: Australia, Cybersource

4. VAR ID: Australia, EMUSYS Unix Consulting

5. VAR ID: Australia, Sydnet Group Pty Ltd

6. VAR ID: Austria, QUANT-X Service & Consulting GmbH

7. VAR ID: Brazil, Millennium Consultoria e Informática Ltda.

8. VAR ID: Canada, Affinity Systems Inc.

9. VAR ID: Canada, Centurion Services

10. VAR ID: Canada, Mediabase

11. VAR ID: Canada, Net Direct Inc.

12. VAR ID: Canada, Softouch Computer Solutions

13. VAR ID: Canada, VL

14. VAR ID: Finland, SOT Finnish Software Engineering Ltd.

15. VAR ID: France, Paralline

16. VAR ID: Germany, IGEL GmbH

17. VAR ID: Germany, IGEL

18. VAR ID: Germany, dynamis EDV-Consulting

19. VAR ID: Hong Kong, Genesis Systems Int'l Ltd.

20. VAR ID: India, G.T.Enterprises

21. VAR ID: Indonesia, PT Cakram DataLingga Duaribu (CDL2000)

22. VAR ID: Italy, Conecta Telematica srl

23. VAR ID: Italy, Nauta Srl - Internet per l'impresa

24. VAR ID: Italy, Studio C

25. VAR ID: Japan, SoftWare Research Co.

26. VAR ID: Malaysia, Linux Resources Sdn. Bhd.

27. VAR ID: Malaysia, MAGNIFIX

28. VAR ID: Malaysia, Malaysian Linux Competency Center Sdn. Bhd.

29. VAR ID: Netherlands, CMG Trade, Transport & Industry

30. VAR ID: Netherlands, X OS Experts in Open Systems BV

31. VAR ID: Poland, Emit Sp z o.o.

32. VAR ID: Romania, EVERCOMM Serv SRL

33. VAR ID: Spain, Innovacion y Futuro s.l.

34. VAR ID: Spain, LANGRE INGENIEROS S.L.

35. VAR ID: Sweden, South Pole AB

36. VAR ID: Taiwan, Whinstone Info. Corp.

37. VAR ID: Turkey, Gelecek Bilisim ve Iletisim A.S

38. VAR ID: US, AZ, Sunset Systems

39. VAR ID: US, CA, ASL Workstations, Inc.

40. VAR ID: US, CA, Linux Laptops Ltd.

41. VAR ID: US, CA, PSSC Labs

42. VAR ID: US, CO, Apache Digital Corporation

43. VAR ID: US, CT, GNUTEC

44. VAR ID: US, FL, eBuilders Network

45. VAR ID: US, IL, Advanced Horizons, Inc

46. VAR ID: US, IL, QLITech Linux COmputers

47. VAR ID: US, IL, QLITech Linux Computers

48. VAR ID: US, IL, WellThot Inc.

49. VAR ID: US, KY, IndyBox Systems, Inc.

50. VAR ID: US, KY, Neville Technologies

TOC

51. VAR ID: US, MA, Network Concierge Inc.

52. VAR ID: US, MD, AmNet Computers

53. VAR ID: US, MN, Cranston Software

54. VAR ID: US, MN, Donnal Group Inc

55. VAR ID: US, MO, Atipa Linux Solutions

56. VAR ID: US, MO, Omnitec Corporatin

57. VAR ID: US, NC, ApeX Systems Integration Corp.

58. VAR ID: US, NC, ComputerNet Inc.

59. VAR ID: US, NC, Indelible Blue

60. VAR ID: US, NE, BrainStorm Technologies

61. VAR ID: US, NE, BrainStorm Technologies

62. VAR ID: US, NH, DCG Computers, Inc.

63. VAR ID: US, NON United States, Microway

64. VAR ID: US, NON United States, Telenet System Solutions, Inc.

TOC

65. VAR ID: US, NY, Logicept Corporation

66. VAR ID: US, OH, 800Linux

67. VAR ID: US, OH, LANshark Systems, Inc.

68. VAR ID: US, OH, The Computer Underground

69. VAR ID: US, OH, Unique Systems, Inc.

70. VAR ID: US, OR, 3D Technology, Inc.

71. VAR ID: US, PR, OG Consulting

72. VAR ID: US, TX, Computer Innovations International

73. VAR ID: US, TX, Heights Open Systems

74. VAR ID: US, TX, Parachute

75. VAR ID: US, VA, NetSource of Virginia, Inc.

76. VAR ID: US, WA, Cheek Consulting

77. VAR ID: US, WA, Lodgepole Technology

78. VAR ID: United Arab Emirates, JM Infotech Solutions STPI

TOC

79. VAR ID: United Kingdom, DGC-NMS

80. VAR ID: United Kingdom, Definite Linux Systems Ltd

81. VAR ID: United Kingdom, Digital Networks

82. VAR ID: United Kingdom, iCode Systems Ltd

---

1. About this document

This is the Linux VAR HOWTO. It contains a listing of service companies which do not manufacture hardware or create packaged software, but add value to existing products. This added value can take many forms, such as installation of Linux on computer hardware, configuration, or custom applications. However, at the moment the Linux VAR HOWTO only contains companies selling preconfigured hardware for the purpose of running Linux. Companies offering other services are invited to contact the maintainer poet@linuxports.com at poets@linuxports.com for a listing.

If you contact any companies listed in this document, please mention the Linux VAR HOWTO.

If you need to know more about the Linux Documentation Project or about Linux HOWTO's, feel free to contact the supervisor Tim Bynum <linux-howto@sunsite.unc.edu>.

Tim Bynum will post the listing to several national and international newsgroups on a monthly basis. In addition, the Linux VAR HOWTO can be found on the World Wide Web at http://www.linuxports.com. New versions of the Linux VAR HOWTO are always placed at this site first, so please be sure to check if the copy you are reading is still up to date!

Companies selling hardware for the purpose of running Linux are invited to fill out the form located at . Companies offering other services should so a more appropriate template can be worked out.

2. Copyright Information

This HOWTO s Copyright © 1999 Mr. Poet and LinuxPorts

A verbatim copy may be reproduced or distributed in any medium physical or electronic without permission of the author. Translations are similarly permitted without express permission if it includes a notice on who translated it.

Short quotes may be used without prior consent by the author. Derivative work and partial distributions of the Linux VAR HOWTO must be accompanied with either a verbatim copy of this file.

Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

We further want that *all* information provided in the HOWTOs is disseminated. If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@metalab.unc.edu.

3. VAR ID: Australia, Cybersource

```
Level 8
140 Queen Street
Melbourne, NON United States 3000

Australia
```

Platform: x86 AXP

Distribution: RedHat

Company Phone Number:

+61 3 9642 5997

Company Fax:

+61 3 9642 5997

Contact Email Address:

info@cyber.com.au

Company URL:

http://www.cyber.com.au

Primary Contact:

Con Zymaris

4. VAR ID: Australia, EMUSYS Unix Consulting

```
20 Ivanhoe Rd
Croydon
Sydney, NON United States 2132

Australia
```

Platform: x86 AXP PPC

Distribution: Caldera Debian RedHat Suse LinuxPPC

Company Phone Number:

+61 2 9798 7604

Company Fax:

+61 2 9798 2854

Contact Email Address:

sales@emusys.com.au

Company URL:

http://www.emusys.com.au

Primary Contact:

Anthony Rumble

5. VAR ID: Australia, Sydnet Group Pty Ltd

PO Box 4664

North Rocks, NON United States NSW 2151

Australia

Platform: x86 AXP Sparc M68k

Distribution: Caldera Debian RedHat LinuxRouter UltraPenguin

Company Phone Number:

+61 2 9873 6400

Company Fax:

+61 2 9873 6411

Contact Email Address:

sales@syd.net.au

Company URL:

http://www.syd.net.au

Primary Contact:

Mark Lipscombe

6. VAR ID: Austria, QUANT-X Service & Consulting GmbH


```
Hintnausdorf 14/1

Sankt Veit a.d.Glan, NON United States

Austria
```


Platform: AXP

Distribution: RedHat

Company Phone Number:

+43 4212 90555-0

Company Fax:

+43 4212 90555-20

Contact Email Address:

office@quant-x.com

Company URL:

http://www.quant-x.com

Primary Contact:

Wolfgang Dreyer

7. VAR ID: Brazil, Millennium Consultoria e Informática Ltda.

R. Francisco Parolin, 758

Curitiba, NON United States 802220-360

Brazil

Platform: x86

Distribution: RedHat

Company Phone Number:

+55-41-333-5645

Company Fax:

+55-41-333-5645

Contact Email Address:

millennium@millennium.etc.br

Company URL:

http://www.millennium.etc.br

Primary Contact:

André A. C. Santos

8. VAR ID: Canada, Affinity Systems Inc.

```
Box 10, RR 1

Innisfail, AB T4G 1T6

Canada
```

Platform: x86 AXP

Distribution: Caldera RedHat

Company Phone Number:

1-403-886-4638

Company Fax:

1-403-886-2663

Contact Email Address:

sales@affinity-systems.ab.ca

Company URL:

http://www.affinity-systems.ab.ca

Primary Contact:

James Bourne

9. VAR ID: Canada, Centurion Services

9351 Blundell Road

Richmond, BC V6Y 1K5

Canada

Platform: x86 AXP

Distribution: RedHat Suse

Company Phone Number:

604-279-1857

Company Fax:

604-279-1800

Contact Email Address:

jwalter@home.com

Company URL:

http://www.ipipeline.net/centurion/

Primary Contact:

Mr.Jan Walter

10. VAR ID: Canada, Mediabase

```
3 Shenston rd.
look for new location soon!!
North York, ON m2r 3b3

Canada
```

Platform: x86

Distribution: Debian RedHat

Company Phone Number:

416-468-5400

Company Fax:

no

Contact Email Address:

mediabase@ihosts.net

Company URL:

http://www.mediabase.ihosts.net

Primary Contact:

Cyrus Hill

11. VAR ID: Canada, Net Direct Inc.

557 Havelock Dr.

Waterloo, ON N2L 4Z1

Canada

Platform: x86 AXP

Distribution: Caldera Debian RedHat

Company Phone Number:

519-579-5006

Company Fax:

519-745-9940

Contact Email Address:

john@netdirect.ca

Company URL:

http://www.netdirect.ca

Primary Contact:

John Van Ostrand

12. VAR ID: Canada, Softouch Computer Solutions

```
5151 3rd Street SE

Calgary, AB T2H2X6

Canada
```

Platform: x86

Distribution: RedHat

Company Phone Number:

403

Company Fax:

258-0267

Contact Email Address:

sales@softouchcomputer.com

Company URL:

http://www.softouchcomputer.com

Primary Contact:

Al Green

13. VAR ID: Canada, VL

221 Britannia

Ottawa, ON K2B 5X1

Canada

Platform: x86

Distribution: RedHat

Company Phone Number:

1 (613) 292-9818

Company Fax:

1 (613) 829-0967

Contact Email Address:

hungvu@netcom.ca

Company URL:

http://

Primary Contact:

Hung Vu

14. VAR ID: Finland, SOT Finnish Software Engineering Ltd.

```
Hermiankatu 8 E
FIN-33720
TAMPERE, NON United States

Finland
```

Platform: x86

Distribution: RedHat

Company Phone Number:

+358-3-316 5544

Company Fax:

+358-3-316 5959

Contact Email Address:

linux@sot.com

Company URL:

http://www.bestlinux.net

Primary Contact:

Kai Valijärvi

15. VAR ID: France, Paralline

```
71, avenue des Vosges
-
STRASBOURG, NON United States 67000

France
```

Platform: x86

Distribution: Suse

Company Phone Number:

33 3 88 14 17 40

Company Fax:

33 3 88 14 17 41

Contact Email Address:

sales@paralline.com

Company URL:

http://www.paralline.com

Primary Contact:

Pierre BRUA

16. VAR ID: Germany, IGEL GmbH

```
Aindlinger Str. 16
Augsburg, NON United States 86167

Germany
```

Platform: x86

Distribution: Suse

Company Phone Number:

+49-821-7200-200

Company Fax:

+49-821-7200-189

Contact Email Address:

sales@igel.de

Company URL:

http://www.igel.de

Primary Contact:

Mike Finckh

17. VAR ID: Germany, IGEL

Aindlinger Str. 16

Augsburg, NON United States 86167

Germany

Platform: x86

Distribution: Suse

Company Phone Number:

+49-821-7200-200

Company Fax:

+49-821-7200-189

Contact Email Address:

sales@igel.de

Company URL:

http://www.igel.de

Primary Contact:

Mike Finckh

18. VAR ID: Germany, dynamis EDV-Consulting

Bundschuhstrasse 24

76661 Philippsburg, NON United States

Germany

Platform: x86 AXP

Distribution: Caldera RedHat Suse

Company Phone Number:

+49 7256 9326-0

Company Fax:

+49 7256 9326-90

Contact Email Address:

info@dynamis.de

Company URL:

http://www.dynamis.de

Primary Contact:

Jochen Lillich

19. VAR ID: Hong Kong, Genesis Systems Int'l Ltd.

```
5/F., 7/F., Honytex Building
22 Ashley Road, T.S.T.
Kowloon, NON United States

Hong Kong
```

Platform: AXP

Distribution: RedHat

Company Phone Number:

+852 2815 0728

Company Fax:

+852 2815 0729

Contact Email Address:

info@genesis.com.hk

Company URL:

http://www.genesis.com.hk

Primary Contact:

Gyver Lo

20. VAR ID: India, G.T.Enterprises

```
# 912, 14th Main, 4th Cross
Maruthi Circle, Hanumanthanagar
Bangalore, NON United States 560019

India
```

Platform: x86

Distribution: RedHat

Company Phone Number:

+91-80-6606093

Company Fax:

+91-80-6671407

Contact Email Address:

gtcdrom@vsnl.com

Company URL:

http://www.gtcdrom.com

Primary Contact:

U.M. Taranath

21. VAR ID: Indonesia, PT Cakram DataLingga Duaribu (CDL2000)

Gunung Batu 131

Bogor, NON United States 16610

Indonesia

Platform: x86

Distribution: RedHat

Company Phone Number:

+62-251-332122

Company Fax:

+62-251-332122

Contact Email Address:

info@cdl2000.or.id

Company URL:

http://www.cdl2000.or.id

Primary Contact:

Maliana Harsoadi

22. VAR ID: Italy, Conecta Telematica srl


```
Viale Tricesimo 101/13

Udine, NON United States 33100

Italy
```


Platform: x86 AXP PPC

Distribution: Debian RedHat Suse LinuxPPC

Company Phone Number:

+39 432 548825

Company Fax:

+39 432 299141

Contact Email Address:

info@conecta.it

Company URL:

http://www.conecta.it/linux

Primary Contact:

Carlo Daffara

23. VAR ID: Italy, Nauta Srl - Internet per l'impresa

Viale Tricesimo 181

Udine (UD), NON United States I-33100

Italy

Platform: x86

Distribution: Debian RedHat LinuxRouter

Company Phone Number:

+39 0432 477588

Company Fax:

+39 0432 548314

Contact Email Address:

info@Nauta.it

Company URL:

http://www.Nauta.it

Primary Contact:

Antonio Pezzoni

24. VAR ID: Italy, Studio C

via Romeo Romei 23

Rome, NON United States 00136

Italy

Platform: x86 AXP Sparc PPC M68k

Distribution: Caldera Debian RedHat Suse LinuxPPC LinuxRouter UltraPenguin

Company Phone Number:

+39639722036

Company Fax:

+39639736598

Contact Email Address:

info@studioc-advertising.com

Company URL:

http://www.studioc-advertising.com

Primary Contact:

Marco Capecci

25. VAR ID: Japan, SoftWare Research Co.


```
Hayabusacho 2-11
Chiyodaku
Tokyo, NON United States 102-0092

Japan
```


Platform: x86

Distribution: Caldera Debian RedHat

Company Phone Number:

81-3-3264-6171

Company Fax:

81-3-3264-0647

Contact Email Address:

info@src.co.jp

Company URL:

http://www.src.co.jp

Primary Contact:

Yossihiro Hayama

26. VAR ID: Malaysia, Linux Resources Sdn. Bhd.

```
No. 33-1, Jalan 46A/26
Pusat Bandar Taman Sri Rampai
Setapak, NON United States 54000

Malaysia
```

Platform: x86 AXP Sparc PPC ARM M68k

Distribution: Caldera Debian RedHat Suse LinuxPPC LinuxRouter UltraPenguin

Company Phone Number:

+603-4139010

Company Fax:

+603-4139011

Contact Email Address:

root@ihsan.com

Company URL:

http://linux.ihsan.com

Primary Contact:

Liwauddin Mohamad

27. VAR ID: Malaysia, MAGNIFIX

```
2102 Wisma Rampai
Jln 34/26, Taman Seri Rampai, Setapak
Kuala Lumpur, NON United States 53300

Malaysia
```

Platform: x86

Distribution: RedHat

Company Phone Number:

603 412 1775

Company Fax:

603 412 1550

Contact Email Address:

izauddin@magnifix.com.my

Company URL:

http://www.magnifix.com.my

Primary Contact:

Izauddin Isa

28. VAR ID: Malaysia, Malaysian Linux Competency Center Sdn. Bhd.


```
32-A, Jalan Mamanda 7,
Ampang Point
Ampang, NON United States 68000 Selangor

Malaysia
```


Platform: x86 Sparc

Distribution: RedHat UltraPenguin

Company Phone Number:

+60 3 4571798

Company Fax:

+60 3 4523669

Contact Email Address:

linux@mlcc.com.my

Company URL:

http://www.mlcc.com.my

Primary Contact:

Abd Rahman Johari

29. VAR ID: Netherlands, CMG Trade, Transport & Industry

```
Kralingseweg 241-249
Postbus 8566
Rotterdam, NON United States 3009 AN

Netherlands
```

Platform: x86 AXP Sparc PPC ARM M68k

Distribution: Caldera Debian RedHat Suse LinuxPPC LinuxRouter UltraPenguin

Company Phone Number:

+31.10.2537000

Company Fax:

+31.10.2537032

Contact Email Address:

Henk.Drenth@cmg.nl

Company URL:

http://www.cmg.nl

Primary Contact:

Henk Drenth

30. VAR ID: Netherlands, X OS Experts in Open Systems BV

Kruislaan 419

Amsterdam, NON United States 1098 VA

Netherlands

Platform: x86 AXP Sparc

Distribution: Debian RedHat Suse

Company Phone Number:

+31 20 6938364

Company Fax:

+31 20 6948204

Contact Email Address:

info@xos.nl

Company URL:

http://www.xos.nl

Primary Contact:

Jos Vos

31. VAR ID: Poland, Emit Sp z o.o.

ul. Koszarowa 48/9

Wroclaw, NON United States

Poland

Platform: x86

Distribution:

Company Phone Number:

+48713267026

Company Fax:

+48723289733

Contact Email Address:

kontakt@emit.pl

Company URL:

http://www.emit.pl

Primary Contact:

Ian Carr-de Avelon

32. VAR ID: Romania, EVERCOMM Serv SRL


```
Str. Ciucea 8, bl.L18, ap.189
sector 3
Bucharest, NON United States 7400

Romania
```


Platform: x86

Distribution: RedHat

Company Phone Number:

+40-9-254-4355

Company Fax:

+40-1-345-4355

Contact Email Address:

Mihnea.Stoenescu@evercomm.ro

Company URL:

http://www.evercomm.ro

Primary Contact:

Mihnea Stoenescu

33. VAR ID: Spain, Innovacion y Futuro s.l.

```
Plaza de S'Antigor, 3

Manacor, NON United States 07500

Spain
```

Platform: x86

Distribution: Caldera

Company Phone Number:

+34 971846014

Company Fax:

+34 971846014

Contact Email Address:

info@i-futuro.com

Company URL:

http://www.i-futuro.com

Primary Contact:

Pascal Peregrina

34. VAR ID: Spain, LANGRE INGENIEROS S.L.

Castilla 95.

Santander., NON United States 39009

Spain

Platform: x86

Distribution: Suse

Company Phone Number:

34-942-33-56-33

Company Fax:

34-942-35-42-75

Contact Email Address:

arturo@langre.com

Company URL:

http://www.langre.com

Primary Contact:

Arturo Limon

35. VAR ID: Sweden, South Pole AB

Banvaktsvägen 12

Solna, NON United States 171 48

Sweden

Platform: x86 AXP

Distribution: Debian RedHat Suse

Company Phone Number:

+46 (0)8 566 106 50

Company Fax:

+46 (0)8 566 106 50

Contact Email Address:

info@southpole.se

Company URL:

http://southpole.se

Primary Contact:

Jakob Sandgren

36. VAR ID: Taiwan, Whinstone Info. Corp.

```
4f, No 15, Lane 2
Dong Shing Rd.
Taipei, NON United States 105

Taiwan
```

Platform: x86 AXP Sparc

Distribution: RedHat

Company Phone Number:

886-2-27492626

Company Fax:

886-2-27468790

Contact Email Address:

info@whinstone.com

Company URL:

http://www.whinstone.com

Primary Contact:

Anthony Hsia

37. VAR ID: Turkey, Gelecek Bilisim ve Iletisim A.S

```
Perpa Ticaret Merkezi B Blok
Kat 11 No:1932
Istanbul, NON United States

Turkey
```

Platform: x86

Distribution: Caldera Debian RedHat Suse

Company Phone Number:

+90 212 2107194

Company Fax:

+90 312 2100442

Contact Email Address:

info@gelecek.com.tr

Company URL:

http://www.gelecek.com.tr

Primary Contact:

Gorkem Cetin

38. VAR ID: US, AZ, Sunset Systems

1328 East Barbarita Avenue

Gilbert, AZ 85234

US

Platform: x86

Distribution: RedHat

Company Phone Number:

602-770-3161

Company Fax:

480-497-8012

Contact Email Address:

sales@sunsetsystems.com

Company URL:

http://www.sunsetsystems.com

Primary Contact:

Rod Roark

39. VAR ID: US, CA, ASL Workstations, Inc.

1757 Houret Court

Milpitas, CA 95035

US

Platform: x86

Distribution: Caldera RedHat Suse

Company Phone Number:

877 ASL 3535

Company Fax:

408 941 2071

Contact Email Address:

sales@aslab.com

Company URL:

http://www.aslab.com

Primary Contact:

Jim Truong

40. VAR ID: US, CA, Linux Laptops Ltd.

```
3583 Lynx Dr.
San Jose, CA 95136

US
```

Platform: x86

Distribution: Debian

Company Phone Number:

+1 888 546-8958

Company Fax:

+1 408 269 9872

Contact Email Address:

info@linuxlaptops.com

Company URL:

http://www.linuxlaptops.com

Primary Contact:

Nathan Myers

41. VAR ID: US, CA, PSSC Labs

27601 Forbes Rd Ste# 59

Laguna Niguel, CA 92677

US

Platform: x86 AXP

Distribution: Caldera RedHat

Company Phone Number:

949-582-7088

Company Fax:

949-582-7093

Contact Email Address:

4sales@pssclabs.com

Company URL:

http://www.pssclabs.com

Primary Contact:

Jason Smith

42. VAR ID: US, CO, Apache Digital Corporation

```
777 Main Avenue
#211
Durango, CO 81301

US
```

Platform: x86 AXP

Distribution: Caldera Debian RedHat Suse

Company Phone Number:

970-259-8153

Company Fax:

970-259-8145

Contact Email Address:

info@apache.com

Company URL:

http://www.apache.com

Primary Contact:

Steve Hauquitz

43. VAR ID: US, CT, GNUTEC

1655 North Colony Road, Suite 3002

Meriden, CT 06450

US

Platform: x86 AXP Sparc PPC ARM M68k

Distribution: Caldera Debian RedHat Suse LinuxPPC LinuxRouter UltraPenguin

Company Phone Number:

1-203-379-0125

Company Fax:

1-203-379-0126

Contact Email Address:

info@gnutec.com

Company URL:

http://www.gnutec.com

Primary Contact:

Kyle Amon

44. VAR ID: US, FL, eBuilders Network

```
8204 Crystal Clear Lane
Suite 1200
Orlando, FL 32809

US
```

Platform: x86

Distribution: Caldera RedHat

Company Phone Number:

407-857-6800

Company Fax:

407-857-2333

Contact Email Address:

pablo@ebuildernet.com

Company URL:

http://www.ebuildernet.com

Primary Contact:

Pablo Bryan

45. VAR ID: US, IL, Advanced Horizons, Inc

323 Ferndale Ave

Elmhurst, IL

US

Platform: x86

Distribution: Suse

Company Phone Number:

630-941-9332

Company Fax:

630-941-9338

Contact Email Address:

roger@ahinc.com

Company URL:

http://www.ahinc.com

Primary Contact:

Roger Petersen

46. VAR ID: US, IL, QLITech Linux COmputers

3717 43rd st SUite #112

Moline, IL 61265

US

Platform: x86

Distribution: Caldera Debian RedHat

Company Phone Number:

1-877-24-linux

Company Fax:

Contact Email Address:

sales@qlitech.net

Company URL:

http://www.qlitech.net

Primary Contact:

Ray Sanders

47. VAR ID: US, IL, QLITech Linux Computers

3717 43rd st. Suite #112

Moline, IL 61265

US

Platform: x86

Distribution: Caldera Debian RedHat Suse

Company Phone Number:

1-877-24-LINUX

Company Fax:

1-309-764-1142

Contact Email Address:

sales@qlitech.net

Company URL:

http://www.qlitech.net

Primary Contact:

Ray Sanders

48. VAR ID: US, IL, WellThot Inc.

305 Van Damin Ave.

Glen Ellyn, IL 60137-5215

US

Platform: x86

Distribution: RedHat LinuxRouter

Company Phone Number:

630-545-1470

Company Fax:

630-545-1471

Contact Email Address:

sales@wellthot.com

Company URL:

http://wellthot.com

Primary Contact:

Mark Schwenk

49. VAR ID: US, KY, IndyBox Systems, Inc.

PO Box 3564

Carmel, KY 46082-3564

US

Platform: x86 AXP Sparc PPC ARM M68k

Distribution: Caldera Debian RedHat Suse LinuxPPC LinuxRouter UltraPenguin

Company Phone Number:

+01 (317) 846.9762

Company Fax:

Contact Email Address:

info@indybox.com

Company URL:

http://www.indybox.com

Primary Contact:

Charles Martin

50. VAR ID: US, KY, Neville Technologies

4240 Frankfort Road

Shelbyville, KY 40065

US

Platform: x86

Distribution: RedHat

Company Phone Number:

502-633-3711

Company Fax:

502-633-7950

Contact Email Address:

info@neville.net

Company URL:

http://www.neville.net

Primary Contact:

Chuck Neville

51. VAR ID: US, MA, Network Concierge Inc.

350 Pleasant St.

Ashland, MA 01721

US

Platform: x86

Distribution:

Company Phone Number:

1- 877- 876-1169

Company Fax:

1-508-881-7374

Contact Email Address:

sales@nc4u.com

Company URL:

http://www.nc4u.com

Primary Contact:

Leonardo Salazar Sr.

52. VAR ID: US, MD, AmNet Computers

```
3618 Anne Hasthaway
Suite 1B
Randallstown, MD 21133

US
```

Platform: x86 AXP

Distribution: RedHat

Company Phone Number:

1 410 922-3554

Company Fax:

1 410 922-3552

Contact Email Address:

amnet@amnet-comp.com

Company URL:

http://amnet-comp.com

Primary Contact:

Joseph Volodarsky

53. VAR ID: US, MN, Cranston Software

3255 Hennepin Av.S.

Minneapolis, MN 55408

US

Platform: x86

Distribution: Debian

Company Phone Number:

612

Company Fax:

378-9052

Contact Email Address:

sales@cranston.com

Company URL:

http://cranston.com

Primary Contact:

M.A. al-Kabir

54. VAR ID: US, MN, Donnal Group Inc

19820 Cottagewood Road

Excelsior, MN 55331

US

Platform: x86

Distribution: Caldera

Company Phone Number:

612-474-7785

Company Fax:

612-474-7793

Contact Email Address:

info@donnalgroup.com

Company URL:

http://www.donnalgroup.com

Primary Contact:

Tim Burke

55. VAR ID: US, MO, Atipa Linux Solutions

2608 S. Main

Joplin, MO 64804

US

Platform: x86

Distribution: Caldera RedHat

Company Phone Number:

(800) 360-4346

Company Fax:

(417) 626-2692

Contact Email Address:

info@atipa.com

Company URL:

http://www.atipa.com

Primary Contact:

Brandon Fuhr

56. VAR ID: US, MO, Omnitec Corporatin

PO Box 411010

St. Louis, MO 63141

US

Platform: x86

Distribution: Debian RedHat Suse LinuxRouter

Company Phone Number:

3149974600

Company Fax:

3149970153

Contact Email Address:

info@omnitec.net

Company URL:

http://www.omnitec.net

Primary Contact:

Lee Lammert

57. VAR ID: US, NC, ApeX Systems Integration Corp.

PO Box 338

Apex, NC 27502

US

Platform: x86

Distribution: RedHat Suse

Company Phone Number:

919-468-8150

Company Fax:

919-468-5288

Contact Email Address:

apex@2boot.com

Company URL:

http://www.2boot.com

Primary Contact:

Joe Ferguson

58. VAR ID: US, NC, ComputerNet Inc.

```
8320 Univ Exec PK Dr
Suite 105
Charlotte, NC 28262

US
```

Platform: x86

Distribution: RedHat

Company Phone Number:

704-510-5292

Company Fax:

704-510-5296

Contact Email Address:

kbrust@computernetinc.com

Company URL:

http://http://www.computernetinc.com

Primary Contact:

Kurt Brust

59. VAR ID: US, NC, Indelible Blue

```
3209 Gresham Lake Road
Suite 135
Raleigh, NC 27615

US
```

Platform: x86

Distribution: Caldera Debian RedHat Suse LinuxRouter

Company Phone Number:

1.919.878.9700 or 1.800.776.8284

Company Fax:

1.919.878.7479

Contact Email Address:

sales@indelible-blue.com

Company URL:

http://www.indelible-blue.com

Primary Contact:

Mark Turner

60. VAR ID: US, NE, BrainStorm Technologies

1525 GrandView Ste#1

Papillion, NE

US

Platform: x86 AXP PPC

Distribution: RedHat

Company Phone Number:

1-800-598-6732

Company Fax:

1-800-598-6732

Contact Email Address:

info@bstc.net

Company URL:

http://www.bstc.net

Primary Contact:

Brian Roberson

61. VAR ID: US, NE, BrainStorm Technologies

1525 GrandView Ste#1

Papillion, NE

US

Platform: x86 AXP PPC

Distribution: RedHat

Company Phone Number:

1-800-598-6732

Company Fax:

1-800-598-6732

Contact Email Address:

info@bstc.net

Company URL:

http://www.bstc.net

Primary Contact:

Brian Roberson

62. VAR ID: US, NH, DCG Computers, Inc.

4 Sanborn Rd., # 2

Londonderry, NH 03053

US

Platform: x86 AXP Sparc

Distribution: Caldera Debian RedHat Suse

Company Phone Number:

+1-603-421-1800

Company Fax:

+1-603-421-0911

Contact Email Address:

sjg@dcginc.com

Company URL:

http://www.dcginc.com

Primary Contact:

Stephen Gaudet

63. VAR ID: US, NON United States, Microway

```
Research Park
Box 79
Kingston Ma, NON United States 02364

US
```

Platform: AXP

Distribution: RedHat

Company Phone Number:

508-746-7341

Company Fax:

508-746-4678

Contact Email Address:

info@microway.com

Company URL:

http://www.microway.com

Primary Contact:

Ann Fried

64. VAR ID: US, NON United States, Telenet System Solutions, Inc.

```
2480 Kruse Drive

San Jose, NON United States

US
```

Platform: x86

Distribution: RedHat

Company Phone Number:

1-877-BootLinux

Company Fax:

408-383-0334

Contact Email Address:

linux@tesys.com

Company URL:

http://www.tesys.com

Primary Contact:

Kamal Singh

65. VAR ID: US, NY, Logicept Corporation

41 Chenango Street, Suite 201

Binghamton, NY 13901-2901

US

Platform: x86

Distribution: RedHat

Company Phone Number:

888.875.1355

Company Fax:

Contact Email Address:

info@logicept.com

Company URL:

http://www.logicept.com

Primary Contact:

Jaron Rubenstein

66. VAR ID: US, OH, 800Linux

1357 W. Lane Ave suite 211

Columbus, OH 43221

US

Platform: x86

Distribution: RedHat

Company Phone Number:

614-324-1610

Company Fax:

614-324-1617

Contact Email Address:

sales@gtltd.com

Company URL:

http://800linux.com

Primary Contact:

67. VAR ID: US, OH, LANshark Systems, Inc.

784 Morrison Road

Columbus, OH 43230

US

Platform: x86

Distribution: Caldera Debian RedHat Suse

Company Phone Number:

614-751-1111

Company Fax:

614-751-1112

Contact Email Address:

info@lanshark.com

Company URL:

http://linux.lanshark.com

Primary Contact:

Scott Sharkey

68. VAR ID: US, OH, The Computer Underground

1357 West Lane Ave suite 210

Columbus, OH

US

Platform: x86

Distribution: Caldera Debian RedHat Suse

Company Phone Number:

614-485-0506

Company Fax:

Contact Email Address:

sales@tcu-inc.com

Company URL:

http://www.tcu-inc.com

Primary Contact:

Mark Nielsen

69. VAR ID: US, OH, Unique Systems, Inc.

```
6920 Spring Valley Dr.
Suite 106
Holland, OH 43528

US
```

Platform: x86 AXP

Distribution: Caldera RedHat Suse

Company Phone Number:

419 861-3331

Company Fax:

419 861-3340

Contact Email Address:

info@uniqsys.com

Company URL:

http://www.uniqsys.com

Primary Contact:

Glenn Jacobson

70. VAR ID: US, OR, 3D Technology, Inc.

```
9150 SW Pioneer Ct.,
Suite E
Wilsonville , OR 97070

US
```

Platform: x86

Distribution: Caldera RedHat

Company Phone Number:

1-503-582-1190

Company Fax:

1-503-582-9102

Contact Email Address:

netsales@3dtech.net

Company URL:

http://www.3dtech.net

Primary Contact:

Kerry Lafferty

71. VAR ID: US, PR, OG Consulting

```
Urb. Santa Clara
W-1 Anamu
Guaynabo, PR 00969

US
```

Platform: x86

Distribution: Caldera RedHat

Company Phone Number:

787-731-4332

Company Fax:

787-731-4331

Contact Email Address:

james@og-consulting.com

Company URL:

http://www.og-consulting.com

Primary Contact:

James O'Malley

72. VAR ID: US, TX, Computer Innovations International

2145 NW Military Highway, Suite 103

San Antonio, TX 78213

US

Platform: x86 AXP Sparc

Distribution: Caldera RedHat Suse LinuxRouter

Company Phone Number:

210-348-0850

Company Fax:

210-348-0851

Contact Email Address:

wvega@txcii.com

Company URL:

http://www.txcii.com

Primary Contact:

Waldemar Vega

73. VAR ID: US, TX, Heights Open Systems

517 Harvard

Houston, TX 77007

US

Platform: x86 AXP

Distribution: Debian RedHat LinuxRouter

Company Phone Number:

713-864-5676

Company Fax:

713-864-5676

Contact Email Address:

perry@open-systems.com

Company URL:

http://www.open-systems.com

Primary Contact:

Perry Piplani

74. VAR ID: US, TX, Parachute

```
3500 Oakmont
Suite 101
Austin, TX 78731

US
```

Platform: x86

Distribution: RedHat LinuxRouter

Company Phone Number:

(512)374-1898

Company Fax:

(512)374-9753

Contact Email Address:

kjones@parachute.com

Company URL:

http://www.parachute.com

Primary Contact:

Kristian Jones

75. VAR ID: US, VA, NetSource of Virginia, Inc.

```
7639 Hull Street Rd.
Suite 202
Richmond, VA 23235-6438

US
```

Platform: x86

Distribution: RedHat

Company Phone Number:

1-804-674-6040

Company Fax:

1-804-674-6041

Contact Email Address:

netsource@erols.com

Company URL:

http://www.erols.com/netsource

Primary Contact:

Jeff Davis

76. VAR ID: US, WA, Cheek Consulting

```
620 5th Ave W
Suite 309
Seattle, WA 98119

US
```

Platform: x86

Distribution: Caldera

Company Phone Number:

+1 (206) 282-2892

Company Fax:

+1 (206) 727-6797

Contact Email Address:

joseph@cheek.com

Company URL:

http://www.cheek.com

Primary Contact:

Joseph Cheek

77. VAR ID: US, WA, Lodgepole Technology

```
14578 NE 177th Drive
n/a
Redmond, WA 98072

US
```

Platform: AXP

Distribution: RedHat

Company Phone Number:

1(425)867-9091

Company Fax:

1(425)558-5752

Contact Email Address:

info@lodgepole.com

Company URL:

http://www.lodgepole.com

Primary Contact:

Geordy Rostad

78. VAR ID: United Arab Emirates, JM Infotech Solutions STPI


```
P.O. Box 692
AL MOTAMEM
Karama, NON United States 692

United Arab Emirates
```


Platform: x86

Distribution: RedHat

Company Phone Number:

971 50 6326822

Company Fax:

971 4 359295

Contact Email Address:

mammen@emirates.net.ae

Company URL:

http://linuxsolutions.hypermart.net

Primary Contact:

John Mammen

79. VAR ID: United Kingdom, DGC-NMS


```
7 Owen Road
Rainhill, Prescot
Merseyside, NON United States L35 0PJ

United Kingdom
```


Platform: x86 AXP Sparc PPC M68k

Distribution: Caldera Debian RedHat Suse LinuxPPC

Company Phone Number:

+44 151 289 9250

Company Fax:

+44 151 289 7895

Contact Email Address:

rich@dgc-nms.co.uk

Company URL:

http://www.dgc-nms.co.uk

Primary Contact:

Richard Bradshaw

80. VAR ID: United Kingdom, Definite Linux Systems Ltd

```
264 Ripon Road

Stevenage, NON United States SG1 4NF

United Kingdom
```

Platform: x86

Distribution: Caldera Debian RedHat Suse LinuxRouter

Company Phone Number:

+44 1438 223433

Company Fax:

Contact Email Address:

sales@definitelinux.com

Company URL:

http://www.definitelinux.com

Primary Contact:

Jason Clifford

81. VAR ID: United Kingdom, Digital Networks

```
Unit 12, New Mill
Park Road, Dukinfield
Cheshire, NON United States SK16 5LX

United Kingdom
```

Platform: x86

Distribution: Caldera RedHat Suse

Company Phone Number:

44 161 339 8555

Company Fax:

44 161 339 8666

Contact Email Address:

info@dnuk.com

Company URL:

http://dnuk.com

Primary Contact:

Lee Chisnall

82. VAR ID: United Kingdom, iCode Systems Ltd

```
Consort Court
High Street
Fareham, NON United States PO16 7AL

United Kingdom
```

Platform: x86

Distribution: Caldera Debian RedHat Suse

Company Phone Number:

+44 1329 829898

Company Fax:

+44 1329 829899

Contact Email Address:

linux@icode.co.uk

Company URL:

http://www.icode.co.uk

Primary Contact:

Darren Durbin

VME Howto

John Huggins and Michael Wyrick, vmelinux@va.net

v0.8a, 30 July 1998

This document came about to show the embedded system community how to run Linux on their VMEbus Pentium and other PCI local bus based VMEbus processor designs.

1. Introduction

- 1.1 Knowledge Required
- 1.2 Why use Linux on VMEbus systems?
- 1.3 Purpose
- 1.4 Feedback
- 1.5 VMELinux Revision History
- 1.6 Copyright/Distribution

2. Installation of the VMELinux Kernel Driver

- 2.1 Download the Source
- 2.2 Install the source to the software
- 2.3 Compile the VMELinux components
- 2.4 Load the VMELinux Kernel Module
- 2.5 Difficulties

3. How to talk to the VMEbus with the VMEUtils and the VMEShell Packages

- 3.1 What is the VMEUtils program
- 3.2 What are the VMEShell Scripts

- 3.3 The "vmemap" command.
- 3.4 Read Byte, Word or Long
- 3.5 Write Byte, Word or Long
- 3.6 Read the VMEbus to a file
- 3.7 Write a file to the VMEbus
- 3.8 Parameters
- 3.9 Options
- 3.10 A Note about DMA mode.

4. How to talk to the Tundra Universe PCI-VME bridge using the devices drivers.

- 4.1 The device drivers used with VMELinux
- 4.2 VMEMaster Device Drivers
- 4.3 VMESlave Device Drivers
- 4.4 Direct Control of the Universe Registers
- 4.5 read()
- 4.6 write()
- 4.7 lseek()
- 4.8 ioctl()
- 4.9 open() and close()

5. Advantages of the VMEbus, Linux and VMELinux

- 5.1 Pin and socket connectors
- 5.2 Eurocard assembly
- 5.3 Linux is Low Cost
- 5.4 Linux is Stable
- 5.5 Linux is Dynamic

6. Current and planned Board Support

- 6.1 Xycom XVME655 Pentium VMEbus Board
- 6.2 XyCom XVME656 Pentium VMEBus Board

- 6.3 Dynatem DPC1-0367
- 6.4 Planned Board Support

7. Conclusion

8. FAQ

- 8.1 The Shell utilities return a bunch of stars (*) when I access a board I know is there. What gives?
- 8.2 How does VMELinux handle interrupts?
- 8.3 I have RedHat 5.1 and can't get VMELinux programs to compile.

---

1. Introduction

1.1 Knowledge Required

Using Linux on an embedded VMEbus processor board is not very difficult. However, more than fundamental knowledge is required. This document is not a primer on how to fully configure a Linux machine.

In order to understand this HOWTO document it is assumed that you are thoroughly familiar with the following:

- Configuring and compiling a Linux kernel to operate the various peripherals on your board. Kernel-HOWTO
- Setting up and configuring of network devices NET-3 HOWTO
- Setting up of inetd NET-3 HOWTO
- Setting up and use of the Tundra Universe PCI to VME Bridge Chip Tundra Universe. The new VMEUtils program makes knowledge of the Universe unnecessary for those who do not wish to deal with register level Universe access.
- Compiling and installing various network packages like Apache Site Wu-Ftpd FAQ
- The VMEbus Rev. D and VME64. Excellent information may be found at the VMEbus International Trade Association (VITA).

If you are uncertain of how to proceed with any of the above it is STRONGLY recommended that you use the links provided to familiarize yourself with all packages. We may not reply to any mail regarding any of the above. Please direct any questions to the appropriate author of the HOWTO or consult the respective hardware manufacturer.

This document describes the installation and use of VMELinux on a Xycom XVME-655 6U VME processor board. Other brands of VME boards that use a Pentium and the Tundra Universe chip should be capable of running VMELinux. Please consult the Board Support Section of the VMELinux web site for tested boards. VMELinux Project Web Site

1.2 Why use Linux on VMEbus systems?

Operating systems for VMEbus computers are usually Real-time Operating Systems (RTOS) which have high cost and a significant learning curve. In return the RTOS offers quick response to real world events for control of machinery or response to a process.

The VMEbus provides a rugged computer enclosure and interconnection system. Many system integrators require this ruggedness and also need very fast real-time response. However, there are many

TOC

times when there is little need for real-time response, but the software still needs:

- networking capability,
- remote access via telnet or similar program,
- file transfer via FTP or similar programs,
- remote booting via BOOTP or similar method,
- a way to respond to system interrupts.

Linux has all these capabilities. Thus, the VMELinux Project exists.

1.3 Purpose

The purpose of VMELinux is to give the VME system integrator another choice in operating systems. Rich in features, high in reliability and low in cost, Linux offers benefits to the embedded computer industry. High cost operating systems economically prohibit the use of VME in many applications. With Linux and the VMELinux drivers, the rugged VMEbus has new possibilities.

The purpose of the VMELinux Project is to:

- Maintain and improve the free VMELinux Kernel Driver software,
- Offer added value software components such as the VMEUtils program and VMEShell utilities.
- Test the software on various makes and brands of manufacturer supplied VME processor boards,
- Maintain web based documentation on each tested brand and make of boards,
- Maintain this HOWTO.
- Integrate user suggested and user supplied improvements into the virgin code so we may all benefit from the programming talents of others.
- Become the original source for all the above software so VMELinux users can be assured of original code from the authors.

1.4 Feedback

As VMELinux is tested in the field, we encourage comments about how well or how bad it works. Please feel free to send comments to The VMELinux Project

As we get experience about each brand of VME CPU, we will list the different configurations in this HOWTO. For now we will describe only the Xycom board.

1.5 VMELinux Revision History

Linux Kernel Driver

- November, 1997, v0.2 - Initial version on Xycom Board
- December, 1997, v0.3 - Useable version used for actual work with project.
- February, 1998, v0.6 - DMA mode added to VME access modes.

TOC

- June, 1998, v0.8 - Fixed a few things to allow the new VMEUtils to work.
- June 24, 1998, v0.8a - Current version made available on the website.

VMEUtils Program

- February, 1998, v0.6 - Created a command line interpreter to access the VMEbus
- June, 1998, v0.8 - Fixed several issues to allow VMEShell Utilities to function
- June 24, 1998, v0.8a - Current version made available on the website.

VMEShell Utilities

- June, 1998, v0.8 - Created command line utilities that allow access to the VMEbus from the Linux shell prompt. These shell programs interface with the VMEUtils program.
- June 24, 1998, v0.8a - Changed the name of all the shell programs so they all begin with "vme." Current version made available on the website.

1.6 Copyright/Distribution

This document is Copyright 1997-1998 by John Huggins and the VMELinux Project.

A verbatim copy may be reproduced or distributed in any medium physical or electronic without permission of the author. Translations are similarly permitted without express permission if it includes a notice on who translated it. Commercial redistribution is allowed and encouraged; however please notify The VMELinux Project of any such distributions.

Excerpts from the document may be used without prior consent provided that the derivative work contains the verbatim copy or a pointer to a verbatim copy.

Permission is granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we wish to retain copyright on this HOWTO document, and would like to be notified of any plans to redistribute this HOWTO.

---

2. Installation of the VMELinux Kernel Driver

2.1 Download the Source

Download the distribution from the VMELinux Web Site.

2.2 Install the source to the software

Place the file in a directory reserved for VME usage; We suggest /universe. Untar the zipped/tarred file by typing...

```
tar -xzf VMELinux_08a.tar.gz
```

You should see three directories and one link to ca91c042

```
ca91c042
vmeshell
vmeutils
driver
```

In ca91c042 you should find:

```
ca91c042/
ca91c042/Makefile
ca91c042/ca91c042.c
ca91c042/ca91c042.h
ca91c042/README
ca91c042/e
ca91c042/ins
ca91c042/stat
ca91c042/uns
```

In vmeshell you should find:

```
vmeshell/vmer
vmeshell/README
vmeshell/vmeseek
vmeshell/cmd.vme
vmeshell/vmew
vmeshell/vmeregw
```

```
vmeshell/vmeregr
vmeshell/vmefa
vmeshell/vmecall
vmeshell/e
vmeshell/ec
vmeshell/fa.vme
vmeshell/map.vme
vmeshell/tmp.vme
vmeshell/vmedb
vmeshell/vmedl
vmeshell/vmedw
vmeshell/vmemap
vmeshell/vmerb
vmeshell/vmerf
vmeshell/vmerl
vmeshell/vmerw
vmeshell/vmewb
vmeshell/vmewf
vmeshell/vmewl
vmeshell/vmeww
```

In the vmeutils directory you should find:

```
vmeutils/commands.cpp
vmeutils/commands.h
vmeutils/universe.h
vmeutils/Makefile
vmeutils/vmeutils.h
vmeutils/unilib.h
vmeutils/unilib.cpp
vmeutils/vmeutils.cpp
vmeutils/README
```

2.3 Compile the VMELinux components

Enter the "ca91c042" directory and make the VMELinux device driver module.

```
make
```

Now you must create the several /dev driver files. Type:

```
make devices
```

Once made, you should see the file "ca91c042.o" in the directory. This is a loadable module. See below for loading information. Plus, you should find several "vme..." files in the /dev directory.

TOC

Once the devices are made in the /dev directory you must change their permissions. Type:

```
cd /dev
chmod 666 vme*
```

Here is how the files should look:

```
hostname:/dev# ls -l vme*
crw-rw-rw-   1 root     root      70,   8 Jul 30 10:51 vme_ctl
crw-rw-rw-   1 root     root      70,   0 Jul 30 10:51 vme_m0
crw-rw-rw-   1 root     root      70,   1 Jul 30 10:51 vme_m1
crw-rw-rw-   1 root     root      70,   2 Jul 30 10:51 vme_m2
crw-rw-rw-   1 root     root      70,   3 Jul 30 10:51 vme_m3
crw-rw-rw-   1 root     root      70,   4 Jul 30 10:51 vme_s0
crw-rw-rw-   1 root     root      70,   5 Jul 30 10:51 vme_s1
crw-rw-rw-   1 root     root      70,   6 Jul 30 10:51 vme_s2
crw-rw-rw-   1 root     root      70,   7 Jul 30 10:51 vme_s3
hostname:/dev#
```

Change to the "vmeutils" directory and type make there.

```
make
```

This will compile the "vmeutils" program. This program directly speaks to the kernel driver. It is a reference work for those of you who wish to write your own programs to directly speak with the driver.

Copy the program "vmeutils" to your user binary directory. On our system this is "/usr/local/bin." Alternatively, you can create a link in the user bin directory to the "vmeutils" program.

Change to the "vmeshell" directory. There are no files to be compiled here. These are shell programs that use the "vmeutils" program to access the VMEbus. All the files beginning with "vme" should with have a link made or be copied to the "/usr/local/bin" directory.

You are now ready to try the driver.

2.4 Load the VMELinux Kernel Module

Make sure you are root and insert "load" the VMELinux Kernel Module for the Universe chip by typing...

```
insmod ca91c042
```

Or just type "ins" to let the shell script do this for you. Once complete, type...

```
stat
```

You should see a list of registers displayed on your screen. Something like this...

TOC

```
Universe driver info:
  Control Pointer = 0000
  Stats  reads = 0  writes = 0  ioctls = 0
  LSI0_CTL = 00800000    LSI1_CTL = 00800000
  LSI0_BS  = C0000000    LSI1_BS  = 00000000
  LSI0_BD  = C0010000    LSI1_BD  = 00000000
  LSI0_TO  = 40009000    LSI1_TO  = 00000000
  LSI2_CTL = 00800000    LSI3_CTL = 00800000
  LSI2_BS  = 00000000    LSI3_BS  = 00000000
  LSI2_BD  = 00000000    LSI3_BD  = 00000000
  LSI2_TO  = 00000000    LSI3_TO  = 00000000
  image_va0   = 00000000     image_va1   = 00000000
  image_va2   = 00000000     image_va3   = 00000000

Driver Program Status:
  DMACTL 0    = 00000000 DMACTL 1    = 00000000
  DMACTL 2    = 00000000 DMACTL 3    = 00000000
  OkToWrite 0 = 0        OkToWrite 1 = 0
  OkToWrite 2 = 0        OkToWrite 3 = 0
  Mode 0      = 0        Mode 1      = 0
  Mode 2      = 0        Mode 3      = 0
```

If not, something went wrong.

2.5 Difficulties

The Universe driver does a good job of finding the Universe chip on a PCI bus, but differences in board design may prevent this. We tested all our routines on a Xycom XVME-655 board. There is little reason why this should not work on any other Intel board with a PCI bus and the Universe PCI-VME bridge chip. If you encounter problems, please let us know at the The VMELinux Project

---

3. How to talk to the VMEbus with the VMEUtils and the VMEShell Packages

3.1 What is the VMEUtils program

This program can be run as is. Once started, you will see a command prompt. Type ? And you will see a list of commands. While useful, I think you will find the VMEShell scripts a better way to go. They do use this program to speak with the kernel driver so it is necessary to have this program available in the current PATH.

The source code for "vmeutils" is also instruction on how to speak directly to the kernel driver. For those of you who wish to create programs that directly speak with the driver, these source files are good examples.

3.2 What are the VMEShell Scripts

The VMEShell programs are unix shell scripts. They offer the operator a simple way to access the data on a VMEbus. Using these commands creates temporary files in the user's working directory which store information on the last access you did. This is nice because it will be possible to log off the machine, log back in and proceed from where you left off without having to re-enter VMEbus information again. Plus, these files are stored in the current working directory, so you can have different VME access configuration just by setting up different directories for each VME board of interest.

Assuming you placed the shell programs and the "vmeutils" program in the /usr/local/bin directory, you should be able to log in as a regular user and run them. What follows assumes exactly this.

3.3 The "vmemap" command.

Login as a regular user and create a directory to experiment with. Once in this directory type:

```
vmemap
```

You should get a help screen like this...

```
Usage:  map address count space size type
  where address is VME Address to set Universe image to

    Space = 0 CR/CSR    Space = 1 A16
    Space = 2 A24       Space = 3 A32
```

TOC

```
Size  = 1 8 bit     Size  = 2 16 bit
Size  = 3 32 bit    Size  = 4 64 bit

Type  = 0 USR/DATA  Type  = 1 USR/PRG
Type  = 2 SUP/DATA  Type  = 3 SUP/PRG
```

This is where you tell VMELinux how you want to access the VMEbus. We assume you already know about the VMEbus' many modes of operation, but here is a short list to help you.

- **address** is the actual VMEbus address you wish to see. This should be set to the lower most value of the address range of interest.
- **count** is the number of bytes you consider a valid range to view. This is the number of bytes starting at the address specified above.
- **space** is the addressing space (mode). For those of you who do not know what we are talking about here, the VMEbus has four overlapping address spaces that can be called independant of each other. A16 is a 64 KiloByte space. A24 is a 16 MegaByte space. A32 is a 4 GigaByte space. There is an A64 space defined the VME specification, but the Universe does not support it.
- **Size** refers to the maximum datawidth allowed for the VME board you are accessing. Some VMEbus board only handle 8 bit data paths. Others transfer 32 bits (four bytes) at a time. Some can handle a special VME block mode which can move 64 bits per transaction. The Universe can handle all these modes allowing you to mix inexpensive serial port boards with hugh memory arrays.
- **Type** is the type of VME transaction performed. Some VME boards make a distinction between "User" access (USR) and "Supervisior" access (SUP). Also, some boards allow access to two "pages" of memory: Program (PRG) and Data. The Universe supports all modes.

Typing...

```
vmemap 0x8000 0x100 1 2 0
```

sets up the VMELinux driver to access an A16 board at base address 8000 Hex with a range of 100H bytes with 16 bit data width and USR/DATA mode.

You will find two new files in your current directory.

- fa.vme
- map.vme

fa.vme stores a "fixed adder" value that will be added to all subsequent accesses with the programs below.

map.vme store the parameters above so you do not have to enter them every time.

All the following shell utilities read values from these two files to performs VME accesses.

TOC

3.4 Read Byte, Word or Long

Syntax:

- vmerb -[options] address size
- vmerw -[options] address size
- vmerl -[options] address size

3.5 Write Byte, Word or Long

Syntax:

- vmewb -[options] address value
- vmeww -[options] address value
- vmewl -[options] address value

3.6 Read the VMEbus to a file

Syntax:

- vmerf -[options] address size filename

3.7 Write a file to the VMEbus

Syntax:

- vmewf -[options] address filename

3.8 Parameters

There are several parameters used with these commands: address, size and filename.

- address - The actual hexadecimal VMEbus address you wish to read. If the map command is set to access A16 VME address space, the address should be 0xABCD. If the space is A24 then use 0xABCDEF. For A32 space use 0xABCDEFGH.
- size - The number of bytes to read. This value is always the number of bytes regardless of the data word size read. For example, if you want to read 16 bytes of information and use vmerl, the display will show 16 bytes displayed as 4 long words.
- filename - The name of the file to send "read" VMEbus data to or "write" VMEbus data from.
- value - a hex value written as "0xXXXX."

TOC

3.9 Options

Available options are defined with a single dash with the any combination of the following:

- q - Hides details on the access to the vmeutils program (default)
- Q - Shows details on the access to the vmeutils program
- p - Single access PCI addressing mode (opposite of d) (default)
- d - DMA access PCI addressing mode (opposite of p) (very fast access to the VMEbus)
- 0, 1, 2, or 3 - Which Universe chip "Image" to use (defaults to 0)
- b - binary mode off (default)
- B - binary mode on
- v - turn off verbose parameter printing (default)
- V - turn on verbose parameter printing to see how the driver is begin used

3.10 A Note about DMA mode.

VMELinux offers access to all the features of the Universe Chip. Especially useful is access to the DMA engine on the chip. With this feature the Universe chip transfers data on the PCI bus by becoming a PCI master. This is nice, but the real benefit comes from the VMEbus accesses. Even if the VMEbus interface is not using block mode transfers, the Universe chip can complete VMEbus transfers under 400 nanoseconds sustained. This is the direct result of the Universe taking complete control of both the PCI bus and the VMEbus. Thus, it is possible to access non block mode VMEbus peripherals much faster than older technologies.

---

4. How to talk to the Tundra Universe PCI-VME bridge using the devices drivers.

4.1 The device drivers used with VMELinux

- /dev/vme_ioctl
- /dev/vme_m0
- /dev/vme_m1
- /dev/vme_m2
- /dev/vme_m3
- /dev/vme_s0
- /dev/vme_s1
- /dev/vme_s2
- /dev/vme_s3

4.2 VMEMaster Device Drivers

/dev/vme_m* are drivers used to access the VMEbus as a bus master.

The Universe chip offers the programmer four VMEMaster windows to the VMEbus. These windows are called Images. The details of the registers within these windows is beyond the scope of this Howto. Please refer to the Universe documentation for details. Tundra Universe

4.3 VMESlave Device Drivers

/dev/vme_s* are drivers used to allow another VMEbus master to access this device.

The Universe chip offers the programmer four VMESlave windows to the VMEbus. These windows are called Images. The details of the registers within these windows is beyond the scope of this Howto. Please refer to the Universe documentation for details. Tundra Universe

Slave VME modes are not yet supported by VMELinux.

4.4 Direct Control of the Universe Registers

/dev/vme_ioctl allows read and write access to the Universe registers.

For experienced users, this device allows direct access to the Universe chip's internal registers.

TOC

Explanation of these registers and what they do is beyond the scope of this howto. Please consult the Universe documentation available from Tundra Universe

4.5 read()

n = read(vme_handle,buf,len);

Where:

- vme_handle = The value returned by "open,"
- buf = pointer to data block,
- len = number of bytes to read from the VMEbus.

4.6 write()

write(vme_handle,buf,len);

Where:

- vme_handle = The value returned by "open,"
- buf = pointer to data block,
- len = number of bytes to write to the VMEbus.

4.7 lseek()

lseek(vme_handle,vme_pnt,Seek_Type);

Where:

- vme_handle = The value returned by "open,"
- vme_pnt = The actual VME address to access,
- Seek_Type = SEEK_SET or SEEK_CUR

4.8 ioctl()

ioctl(vme_handle, command, argument);

Where:

- vme_handle = The value returned by "open,"
- command = IOCTL_SET_CTL or IOCTL_SET_MODE or IOCTL_SET_BS or IOCTL_SET_BD or IOCTL_SET_TO
- argument to be sent

And:

- IOCTL_SET_CTL = Sets the image CTL register to argument. Argument must be 32 bits.

- IOCTL_SET_MODE = "MODE_DMA" or "MODE_PROGRAMMED" - Sets the mode by w the Universe chips communicates to the PCI bus (Not VME Block Mode)
- IOCTL_SET_BS = Sets the image BS register to arguments. NOTE: The BD register must already be set prior to making this call.
- IOCTL_SET_BD = Sets the image BD register to argument.
- IOCTL_SET_TO = Set the image TO register to argument.

4.9 open() and close()

Here is where you open and close the four VMELinux Master or Slave devices plus the Control device. Slave images are not yet supported.

- vme_handle = open("//dev//vme_m0",O_RDWR,0);
- uni_handle = open("//dev//vme_ctl",O_RDWR,0);
- close(vme_handle);
- close(uni_handle);

---

5. Advantages of the VMEbus, Linux and VMELinux

5.1 Pin and socket connectors

The VMEbus standard uses pin and socket connectors. This is superior to edge connections in that the connection is not exposed to humidity and other environmental conditions. It is a more expensive way of doing things, but offers longer times before failure.

5.2 Eurocard assembly

A VMEboard is either a 3U (160 x 100 mm) or a 6U size (160 x 233.35 mm). These sizes correspond to the Eurocard standard for board modules and card cages. Eurocard is a popular format used by many different busses including CompactPCI. This popularity makes the materials needed for cage assembly inexpensive and easy to obtain.

5.3 Linux is Low Cost

The nature of Linux is in its user supported and freely available format. The number of people using Linux is growing. The number of people contributing to the continued development of the Linux software base is growing. It is unfair to state that Linux is a good value because it is available for little to no charge. Linux is a good value because it works.

5.4 Linux is Stable

There are those who say that Linux us an unstable operating system. It is true that the new Linux kernels in development are experimental and should not be relied on for critical applications. However, stable versions of the Linux OS are always available and provide very stable operation. VMELinux is always based on the stable versions of the kernel source; Today's stable kernels are the 2.0.X series.

5.5 Linux is Dynamic

Because so many people are developing Linux, you do not have to wait long for improvements, fixes or new features to become part of the Linux distribution.

6. Current and planned Board Support

While the VMELinux driver should work with any PCI based design, the following boards have actually run our software.

6.1 Xycom XVME655 Pentium VMEbus Board

- This XyCom board is compatible with the standard VMELinux kernel driver package from VMELinux Project
- A prepared kernel will be available soon. It will be based on the newest version of the Linux kernel and will include appropriate drivers for the onboard NE2100 Ethernet interface. Check the website for details.

6.2 XyCom XVME656 Pentium VMEBus Board

- This XyCom board is compatible with the standard VMELinux kernel driver package from VMELinux Project
- A prepared kernel will be available soon. It will be based on the newest version of the Linux kernel and will include appropriate drivers for the onboard AHA2940/AIC7000 SCSI and 82558 Intel EtherExpress Ethernet peripherals. Check the website for details.

6.3 Dynatem DPC1-0367

- This board is compatible with the standard VMELinux kernel driver package from VMELinux Project
- A prepared kernel will be available soon. It will be based on the newest version of the Linux kernel and will include appropriate drivers for the onboard SCSI and Tulip Ethernet peripherals. Check the website for details.

6.4 Planned Board Support

If you do not see VMELinux support for your board let us know. Maybe the manufacture will lend us a board for development.

7. Conclusion

VMELinux offers the user a low cost way to implement a VMEbus system quickly, reliably and with all the advantages of a unix environment. We are using VMELinux in our projects so you can be sure future developments will come quick. On the drawing board for this year are:

- Implementation of Interrupts and Handling thereof,
- Porting to other brands of Intel VMEbus boards,
- Porting of VMELinux to other processors that use the Universe chip,
- A study of running the VMELinux kernel driver module as a RT-Linux task.

This document outlines the steps you need to install the VMELinux Kernel Driver into the example Xycom XVME-655 Pentium VME board. It is our hope that others will attempt installation of VMELinux into other boards and let us know their success.

Mail any responses to The VMELinux Project. If you have a question or an update to the document let us know and we will add it.

8. FAQ

8.1 The Shell utilities return a bunch of stars (*) when I access a board I know is there. What gives?

Check to be sure the /dev/vme... files have their permissions set to 666. If not, the shell utilities will return a * in place of data to indicate an error condition similar to a VME bus error.

8.2 How does VMELinux handle interrupts?

Right now it doesn't. However, we are planning to get that part going soon. Please be patient.

8.3 I have RedHat 5.1 and can't get VMELinux programs to compile.

RedHat 5.1 includes a new compiler. If you manually edit the Makefile in each directory to call up the new egcs compiler, things should compile. We fully intend to support RedHat 5.1 installations, but for now I suggest using 5.0 or Slackware.

From VMS to Linux HOWTO

By Guido Gonzato, guido@ibogfs.cineca.it

v1.0.2, 20 April 1998

This HOWTO is aimed at all those who have been using VMS and now need or want to switch to Linux, the free UNIX clone. The transition is made (hopefully) painless with a step--to--step comparison between commands and available tools.

1. Introduction

- 1.1 Why Linux?
- 1.2 Comparing Commands and Files

2. Short Intro

- 2.1 Files
- 2.2 Directories
- 2.3 Programs
- 2.4 Quick Tour

3. Editing Files

4. TeXing

5. Programming

- 5.1 Fortran
- 5.2 Using `make`
- 5.3 Shell Scripts
- 5.4 C

6. Graphics

7. Mail and Internet Tools

8. Advanced Topics

- 8.1 Permissions and Ownership
- 8.2 Multitasking: Processes and Jobs
- 8.3 Files, Revisited
- 8.4 Print Queues

9. Configuring

10. Useful Programs

- 10.1 Browsing Files: `less`
- 10.2 RCS in a Nutshell
- 10.3 Archiving: tar & gzip

11. Real Life Examples

12. Tips You Can't Do Without

13. The End

- 13.1 Copyright

14. Disclaimer

1. Introduction

1.1 Why Linux?

You've heard that UNIX is difficult and balk at the prospect of leaving VMS, don't you? Don't worry. Linux, one of the finest UNIX clones, is not more difficult to use than VMS; actually, I find it easier. Although VMS aficionados may not agree, in many people's opinion Linux is much more powerful and versatile.

Linux and VMS are both good operating systems and accomplish essentially the same tasks, but Linux' tools are (IMHO) superior, its syntax is often much more concise, and has some features missing in VMS that help save a lot of time. (You'll often hear that VMS and UNIX have a different `philosophy'.) Moreover, Linux is available for PCs while VMS is not, and modern Pentium-based Linux machines can outperform a VAX. The icing on the cake is the excellent performance of modern video cards, which turn an X11-based Linux box into a fast graphic workstation; nearly always, quicker than dedicated machines.

I have many reasons to believe that the combination Pentium + Linux is preferable to VAX--VMS, but preferences are a strictly personal matter and you may disagree. You'll decide by yourself after a few months.

I imagine you're a university researcher or a student, and that you use VMS for the following everyday tasks:

- writing papers with TeX/LaTeX;
- programming in Fortran;
- doing some graphics;
- using Internet services;
- et cetera.

In the following sections I'm going to explain to you how to do these tasks under Linux, exploiting your experience with VMS. Prerequisites:

- Linux and X Window System are properly installed;
- there's a system administrator to take care of the technical details (please get help from them, not from me ;-) ;
- your shell---the equivalent of `DCL`---is `bash` (ask your sysadm).

Please note that this HOWTO is not enough to acquaint you fully with Linux: it only contains the bare essential to get you started. You should learn more about Linux to make the most of it (advanced `bash` features, programming, regular expressions...). From now on, RMP means `please read the man pages for further details'. The man pages are the equivalent of the command `HELP`.

The Linux Documentation Project documents, available on `sunsite.unc.edu:/pub/Linux/docs/LDP`, are an important source of information. I suggest that you read Larry Greenfield's ``Linux User Guide''---it's invaluable for the novice user.

And now, go ahead.

1.2 Comparing Commands and Files

This table attempts to compare VMS' and Linux' most used commands. Please keep in mind that the syntax is often very different; for more details, refer to the following sections.

VMS	Linux	Notes

TOC

@COMMAND	command	must be executable
COPY file1 file2	cp file1 file2	
CREATE/DIR [.dirname]	mkdir dirname	only one at a time
CREATE/DIR [.dir1.dir2]	mkdirhier dir/name	
DELETE filename	rm filename	
DIFF file1 file2	diff -c file1 file2	
DIRECTORY	ls	
DIRECTORY [...]file	find . -name file	
DIRECTORY/FULL	ls -al	
EDIT filename	vi filename,	you won't like it
	emacs filename,	EDT compatible
	jed filename	ditto---my favourite
FORTRAN prog.for	g77 prog.f,	no need to do LINK
	f77 prog.f,	
	fort77 prog.f	
HELP command	man command	must specify `command'
	info command	ditto
LATEX file.tex	latex file.tex	
LOGIN.COM	.bash_profile,	`hidden' file
	.bashrc	ditto
LOGOUT.COM	.bash_logout	ditto
MAIL	mail,	crude
	elm,	much better
	pine	better still
PRINT file.ps	lpr file.ps	
PRINT/QUEUE=laser file.ps	lpr -Plaser file.ps	
PHONE user	talk user	
RENAME file1 file2	mv file1 file2	not for multiple files
RUN progname	progname	
SEARCH file "pattern"	grep pattern file	
SET DEFAULT [-]	cd ..	
SET DEFAULT [.dir.name]	cd dir/name	
SET HOST hostname	telnet hostname,	not exactly the same
	rlogin hostname	
SET FILE/OWNER_UIC=joe	chown joe file	completely different
SET NOBROADCAST	mesg	
SET PASSWORD	passwd	
SET PROT=(perm) file	chmod perm file	completely different
SET TERMINAL	export TERM=	different syntax
SHOW DEFAULT	pwd	
SHOW DEVICE	du, df	
SHOW ENTRY	lpq	
SHOW PROCESS	ps -ax	
SHOW QUEUE	lpq	
SHOW SYSTEM	top	
SHOW TIME	date	
SHOW USERS	w	
STOP	kill	
STOP/QUEUE	kill,	for processes
	lprm	for print queues
SUBMIT command	command &	

```
SUBMIT/AFTER=time command       at time command
TEX file.tex                    tex file.tex
TYPE/PAGE file                  more file
                                less file               much better
```

But of course it's not only a matter of different command names. Read on.

2. Short Intro

This is what you absolutely need to know before logging in the first time. Relax, it's not much.

2.1 Files

- Under VMS filenames are in the form `filename.extension.version;`. Under Linux, the version number doesn't exist (big limitation, but see Section RCS in a Nutshell); the filename has normally a limit of 255 characters and can have as many dots as you like. Example of filename: `This.is_a_FILEname.txt`.
- Linux distinguishes between upper case and lower case characters: `FILENAME.txt` and `filename.txt` are two different files; `ls` is a command, `LS` is not.
- A filename starting with a dot is a `hidden' file (that is, it won't normally show up in dir listings), while filenames ending with a tilde '~' represent backup files.

Now, a table to sum up how to translate commands from VMS to Linux:

```
VMS                                         Linux
-----------------------------------------------------------------------

$ COPY file1.txt; file2.txt;                $ cp file1.txt file2.txt
$ COPY [.dir]file.txt;1 []                  $ cp dir/file.txt .
$ COPY [.dir]file.txt;1 [-]                 $ cp dir/file.txt ..
$ DELETE *.dat.*                            $ rm *dat
$ DIFF file1 file2                          $ diff -c file1 file2
$ PRINT file                                $ lpr file
$ PRINT/queue=queuename file                $ lpr -Pprintername file
$ SEARCH *.tex.* "geology"                  $ grep geology *tex
```

For other examples involving directories, see below; for details about protections, ownership, and advanced topics, see Section Advanced Topics.

2.2 Directories

- Within the same node and device, directories names under VMS are in the form `[top.dir.subdir]`; under Linux, `/top/dir/subdir/`. On the top of the directory tree lies the so--called `root directory' called `/`; underneath there are other directories like `/bin`, `/usr`, `/tmp`, `/etc`, and others.
- The directory `/home` contains the so--called users' `home directories': e.g. `/home/guido`, `/home/warner`, and so on. When a user logs in, they start working in their home dir; it's the equivalent of `SYS$LOGIN`. There's a shortcut for the home directory: the tilde '~'. So, `cd ~/tmp` is the same as, say, `cd /home/guido/tmp`.
- Directory names follow the same rules as file names. Furthermore, each directory has two special entries: one is `.` and refers to the directory itself (like `[]`), and `..` that refers to the parent directory (like `[-]`).

And now for some other examples:

```
DOS                                          Linux
-----------------------------------------------------------------------

$ CREATE/DIR [.dirname]                      $ mkdir dirname
$ CREATE/DIR [.dir1.dir2.dir3]               $ mkdirhier dir1/dir2/dir3
   n/a                                       $ rmdir dirname
                                             (if dirname is empty)
                                             $ rm -R dirname
$ DIRECTORY                                  $ ls
$ DIRECTORY [...]file.*.*                    $ find . -name "file*"
$ SET DEF SYS$LOGIN                          $ cd
$ SET DEF [-]                                $ cd ..
$ SET DEF [top.dir.subdir]                   $ cd /top/dir/subdir
$ SET DEF [.dir.subdir]                      $ cd dir/subdir
$ SHOW DEF                                   $ pwd
```

For protections, ownership, and advanced topics, see Section Advanced Topics.

2.3 Programs

- Commands, compiled programs, and shell scripts (VMS' `command files') don't have compulsory extensions like `.EXE` or `.COM` and can be called whatever you like. Executable files are marked by an asterisk `'*'` when you issue `ls -F`.
- To run an executable file, just type its name (no `RUN PROGRAM.EXE` or `@COMMAND`). Caveat: it's essential that the file be located in a directory included in the *path of executables*, which is a list of directories. Typically, the path includes dirs like `/bin`, `/usr/bin`, `/usr/X11R6/bin`, and others. If you write your own programs, put them in a directory you have included in the path (see how in Section Configuring). As an alternative, you may run a program specifying its complete path: e.g., `/home/guido/data/myprog`; or `./myprog`, if the current directory isn't in the path.
- Command switches are obtained with `/OPTION=` under VMS, and with `-switch` or `--switch` under Linux, where `switch` is a letter, more letters combined, or a word. In particular, the switch `-R` (recursive) of many Linux commands performs the same action as `[...]` under VMS;
- You can issue several commands on the command line:

  ```
  $ command1 ; command2 ; ... ; commandn
  ```

- Most of the flexibility of Linux comes from two features awkwardly implemented or missing in VMS: I/O redirection and piping. (To be sincere, I have been told that recent versions of IDL support redirection and piping. I don't have that version.) Redirection is a side feature under VMS (remember the switch `/OUTPUT=` of many commands), or a fastidious process, like:

  ```
  $ DEFINE /USER SYS$OUTPUT OUT
  $ DEFINE /USER SYS$INPUT IN
  $ RUN PROG
  ```

  which has the simple Linux (UNIX) equivalent:

  ```
  $ prog < in > out
  ```

  Piping is simply impossible under VMS, but has a key role under UNIX. A typical example:

```
$ myprog < datafile | filter_1 | filter_2 >> result.dat 2> errors.log &
```

which means: the program `myprog` gets its input from the file `datafile` (via <), its output is piped (via |) to the program `filter_1` that takes it as input and processes it, the resulting output is piped again to `filter_2` for further processing, the final output is appended (via >>) to the file `result.dat`, and error messages are redirected (via 2>) onto the file `errors.log`. All this in background (& at the end of the command line). More about this in Section Examples.

For multitasking, `queues', and the like, see Section Advanced Topics.

2.4 Quick Tour

Now you are ready to try Linux out. Enter your login name and password *exactly* as they are. For example, if your login name and password are `john` and `My_PassWd`, *don't* type `John` or `my_passwd`. Remember, UNIX distinguishes between capital and small letters.

Once you've logged in, you'll see a prompt; chances are it'll be something like `machinename:$`. If you want to change the prompt or make some programs start automatically, you'll have to edit a `hidden' file called `.profile` or `.bash_profile` (see example in Section Configuring). This is the equivalent of `LOGIN.COM`.

Pressing ALT--F1, ALT--F2, ... ALT--F6 switches between `virtual consoles'. When one VC is busy with a full--screen application, you can flip over to another and continue to work. Try and log in to another VC.

Now you may want to start X Window System (from now on, X). X is a graphic environment very similar to DECWindows---actually, the latter derives from the former. Type the command `startx` and wait a few seconds; most likely you'll see an open `xterm` or equivalent terminal emulator, and possibly a button bar. (It depends on how your sysadm configured your Linux box.) Click on the desktop (try both mouse buttons) to see a menu.

While in X, to access the text mode (`console') sessions press CTRL--ALT--F1 ... CTRL--ALT--F6. Try it. When in console, go back to X pressing ALT--F7. To quit X, follow the menu instructions or press CTRL--ALT--BS.

Type the following command to list your home dir contents, including the hidden files:

```
$ ls -al
```

Press SHIFT--PAG UP to back-scroll. Now get help about the `ls` command typing:

```
$ man ls
```

pressing 'q' to exit. To end the tour, type `exit` to quit your session. If now you want to turn off your PC, press CTRL--ALT--DEL and wait a few seconds (*never* switch off the PC while in Linux! You could damage the filesystem.)

If you think you're ready to work, go ahead, but if I were you I'd jump to Section Advanced Topics.

3. Editing Files

Linux doesn't have `EDT`, but there are scores of editors available. The only one that's guaranteed to be included in every UNIX version is `vi`---forget it, your sysadm must have installed something better. Probably the most popular editor is `emacs`, which can emulate `EDT` to a certain degree; `jed` is another editor that provides `EDT` emulation.

These two editors are particularly useful for editing program sources, since they have two features unknown to `EDT`: syntax hilighting and automatic indentation. Moreover, you can compile your programs from within the editor (command `ESC-X compile`); in case of a syntax error, the cursor will be positioned on the offending line. I bet that you'll never want to use the true blue `EDT` again.

If you have `emacs`: start it, then type `ESC-X edt-emulation-on`. Pressing ALT--X or ESC-X is `emacs`' way of issuing commands, like `EDT`'s CTRL--Z. From now on, `emacs` acts like `EDT` apart from a few commands. Differences:

- *don't* press CTRL--Z to issue commands (if you did, you stopped `emacs`. Type `fg` to resume it);
- there's an extensive on-line help. Press CTRL-H ?, or CTRL-H T to start a tutorial;
- to save a file, press CTRL-X CTRL-S;
- to exit, press CTRL-X CTRL-C;
- to insert a new file in a buffer, press CTRL-X CTRL-F, then CTRL-X B to switch among buffers.

If you have `jed`: ask your sysadm to configure `jed` properly. Emulation is already on when you start it; use the normal keypad keys, and press CTRL--H CTRL--H or CTRL-? to get help. Commands are issued in the same way as `emacs`'. In addition, there are some handy key bindings missing in the original `EDT`; key bindings can also be tailored to your own taste. Ask your sysadm.

In alternative, you may use another editor with a completely different interface. `emacs` in native mode is an obvious choice; another popular editor is `joe`, which can emulate other editors like `emacs` itself (being even easier to use) or the DOS editor. Invoke the editor as `jmacs` or `jstar` and press, respectively, CTRL-X H or CTRL-J to get online help. `emacs` and `jed` are *much* more powerful than good ol' `EDT`.

4. TeXing

TeX and LaTeX are identical to their VMS counterparts---only quicker :-), but the tools to handle the `.dvi` and `.ps` files are superior:

- to run a TeX file through TeX, do as usual: `tex file.tex`;
- to turn a `.dvi` file into a `.ps` file, type `dvips -o filename.ps filename.dvi`;
- to visualize a `.dvi` file, type within an X session: `xdvi filename.dvi &`. Click on the page to magnify. This program is smart: if you edit and run TeX producing newer versions of the `.dvi` file, `xdvi` will update it automatically;
- to visualize a `.ps` file, type within an X session: `ghostview filename.ps &`. Click on the page to magnify. The whole document or selected pages can be printed. A newer and better program is `gv`.
- to print the `.ps`: usually the command `lpr mypaper.ps` will do, but if the PostScript printer is called, say, `ps' (ask your sysadm) you'll do: `lpr -Pps mypaper.ps`. For more information about print queues, go to Section Print Queues.

5. Programming

Programming under Linux is *much* better: there are lots of tools that make programming easier and quicker. For instance, the drudgery of editing--saving--exiting--compiling--re-editing can be cut short by using editors like `emacs` or `jed`, as seen above.

5.1 Fortran

Not substantial differences here, but note that at the time of writing the available (free) compilers are not 100% compatible with VMS'; expect some minor quirks. (It's actually the VMS compiler which has non-standard extensions.) See `/usr/doc/g77/DOC` or `/usr/doc/f2c/f2c.ps` for details.

Your sysadm has installed a native compiler called `g77` (good but, as of version 0.5.21, still not perfectly compatible with DEC Fortran) or possibly the Fortran to C translator, `f2c`, and one of the front-ends that make it mimic a native compiler. In my experience, the package `yaf77` is the one that provides best results.

To compile a Fortran program with `g77`, edit the source, save it with extension `.f`, then do:

```
$ g77 myprog.f
```

which creates by default an executable called `a.out` (you don't have to link anything). To give the executable a different name and do some optimisation:

```
$ g77 -O2 -o myprog myprog.f
```

Beware of optimisations! Ask your sysadm to read the documentation that comes with the compiler and tell you if there are any problems.

To compile a subroutine:

```
$ g77 -c mysub.f
```

This creates a file `mysub.o`. To link this subroutine to a program, you'll do

```
$ g77 -o myprog myprog.f mysub.o
```

If you have many external subroutines and you want to make a library, do the following:

```
$ cd subroutines/
$ cat *f >mylib.f ; g77 -c mylib.f
```

This will create `mylib.o` that you can link to your programs.

TOC

Finally, to link an external library called, say, `libdummy.so`:

```
$ g77 -o myprog myprog.f -ldummy
```

If you have `f2c`, you only have to use `f77` or `fort77` instead of `g77`.

Another useful programming tool is `make`, described below.

5.2 Using make

The utility `make` is a tool to handle the compilation of programs that are split into several source files.

Let's suppose you have source files containing your routines, `file_1.f, file_2.f, file_3.f`, and a source file of the main program that uses the routines, `myprog.f`. If you compile your program manually, whenever you modify one of the source files you have to figure out which file depends on which, which file to recompile first, and so on.

Instead of getting mad, you can write a `makefile'. This is a text file containing the dependencies between your sources: when one is modified, only the ones that depend on the modified file will be recompiled.

In our example, you'd write a makefile like this:

```
# This is makefile
# Press the <TAB> key where you see <TAB>!
# It's important: don't use spaces instead.

myprog: myprog.o file_1.o file_2.o file_3.o
<TAB>g77 -o myprog myprog.o file_1.o file_2.o file_3.o
# myprog depends on four object files

myprog.o: myprog.f
<TAB>g77 -c myprog.f
# myprog.o depends on its source file

file_1.o: file_1.f
<TAB>g77 -c file_1.f
# file_1.o depends on its source file

file_2.o: file_2.f file_1.o
<TAB>g77 -c file_2.f file_1.o
# file_2.o depends on its source file and an object file

file_3.o: file_3.f file_2.o
```

```
<TAB>g77 -c file_3.f file_2.o
# file_3.o depends on its source file and an object file

# end of makefile.
```

Save this file as `Makefile` and type `make` to compile your program; alternatively, save it as `myprog.mak` and type `make -f myprog.mak`. And of course, RMP.

5.3 Shell Scripts

Shell scripts are the equivalent of VMS' command files and, for a change, are much more powerful.

To write a script, all you have to do is write a standard ASCII file containing the commands, save it, then make it executable with the command `chmod +x <scriptfile>`. To execute it, type its name.

Writing scripts under `bash` is such a vast subject it would require a book by itself, and I will not delve into the topic any further. I'll just give you a more-or-less comprehensive and (hopefully) useful example you can extract some basic rules from.

EXAMPLE: sample.sh

```
#!/bin/sh
# sample.sh
# I am a comment
# don't change the first line, it must be there
echo "This system is: `uname -a`" # use the output of the command
echo "My name is $0" # built-in variables
echo "You gave me the following $# parameters: "$*
echo "First parameter is: "$1
echo -n "What's your name? " ; read your_name
echo notice the difference: "hi $your_name" # quoting with "
echo notice the difference: 'hi $your_name' # quoting with '
DIRS=0 ; FILES=0
for file in `ls .` ; do
  if [ -d ${file} ] ; then # if file is a directory
    DIRS=`expr $DIRS + 1`  # this means DIRS = DIRS + 1
  elif [ -f ${file} ] ; then
    FILES=`expr $FILES + 1`
  fi
  case ${file} in
    *.gif|*jpg) echo "${file}: graphic file" ;;
    *.txt|*.tex) echo "${file}: text file" ;;
    *.c|*.f|*.for) echo "${file}: source file" ;;
    *) echo "${file}: generic file" ;;
```

```
  esac
done
echo "there are ${DIRS} directories and ${FILES} files"
ls | grep "ZxY--!!!WKW"
if [ $? != 0 ] ; then # exit code of last command
  echo "ZxY--!!!WKW not found"
fi
echo "enough... type 'man bash' if you want more info."
```

5.4 C

Linux is an excellent environment to program in C. Taken for granted that you know C, here are a couple of guidelines. To compile your standard `hello.c` you'll use the `gcc` compiler, which comes as part of Linux and has the same syntax as `g77`:

```
$ gcc -O2 -o hello hello.c
```

To link a library to a program, add the switch `-l<libname>`. For example, to link the math library and optimize do

```
$ gcc -O2 -o mathprog mathprog.c -lm
```

(The `-l<libname>` switch forces `gcc` to link the library `/usr/lib/lib<libname>.a`; so `-lm` links `/usr/lib/libm.a`).

When your program is made of several source files, you'll need to use the utility `make` described above. Just use `gcc` and C source files in the makefile.

You can invoke some help about the C functions, that are covered by man pages, section 3; for example,

```
$ man 3 printf
```

There are lots of libraries available out there; among the first you'll want to use are `ncurses`, to handle text mode effects, and `svgalib`, to do graphics.

---

6. Graphics

Among the scores of graphic packages available, `gnuplot` stands out for its power and ease of use. Go to X and type `gnuplot`, and have two sample data files ready: `2D-data.dat` (two data per line), and `3D-data.dat` (three data per line).

Examples of 2-D graphs:

```
gnuplot> set title "my first graph"
gnuplot> plot '2D-data.dat'
gnuplot> plot '2D-data.dat' with linespoints
gnuplot> plot '2D-data.dat', sin(x)
gnuplot> plot [-5:10] '2D-data.dat'
```

Example of 3-D graphs (each `row' of X values is followed by a blank line):

```
gnuplot> set parametric ; set hidden3d ; set contour
gnuplot> splot '3D-data.dat' using 1:2:3 with linespoints
```

A single-column datafile (e.g., a time series) can also be plotted as a 2-D graph:

```
gnuplot> plot [-5:15] '2D-data-1col.dat' with linespoints
```

or as a 3-D graph (blank lines in the datafile, as above):

```
gnuplot> set noparametric ; set hidden3d
gnuplot> splot '3D-data-1col.dat' using 1 with linespoints
```

To print a graph: if the command to print on your Postscript printer is `lpr -Pps file.ps`, issue:

```
gnuplot> set term post
gnuplot> set out '| lpr -Pps'
gnuplot> replot
```

then type `set term x11` to restore. Don't get confused---the last print will come out only when you quit `gnuplot`.

For more info, type `help` or see the examples in directory `/usr/lib/gnuplot/demos/`, if you have it.

7. Mail and Internet Tools

Since Internet was born on UNIX machines, you find plenty of nice and easy-to-use applications under Linux. Here are just some:

- **Mail**: use `elm` or `pine` to handle your email; both programs have on-line help. For short messages, you could use `mail`, as in `mail -s "hello mate" user@somewhere < msg.txt`. You may like programs like `xmail` or some such.
- **Newsgroups**: use `tin` or `slrn`, both very intuitive and self-explanatory.
- **ftp**: apart from the usual character-based `ftp`, ask your sysadm to install the full-screen `ncftp` or a graphical ftp client like `xftp`.
- **WWW**: the ubiquitous `netscape`, or `xmosaic`, `chimera`, and `arena` are graphical web browsers; a character-based one is `lynx`, quick and effective.

8. Advanced Topics

Here the game gets tough. Learn these features, then you'll be ready to say that you `know something about Linux' ;-)

8.1 Permissions and Ownership

Files and directories have permissions (`protections') and ownership, just like under VMS. If you can't run a program, or can't modify a file, or can't access a directory, it's because you don't have the permission to do so, and/or because the file doesn't belong to you. Let's have a look at the following example:

```
$ ls -l /bin/ls
-rwxr-xr-x   1 root     bin         27281 Aug 15  1995 /bin/ls*
```

The first field shows the permissions of the file `ls` (owner root, group bin). There are three types of ownership: owner, group, and others (similar to VMS owner, group, world), and three types of permissions: read, write (and delete), and execute.

From left to right, `-` is the file type (`-` = ordinary file, `d` = directory, `l` = link, etc); `rwx` are the permissions for the file owner (read, write, execute); `r-x` are the permissions for the group of the file owner (read, execute); `r-x` are the permissions for all other users (read, execute).

To change a file's permissions:

```
$ chmod <whoXperm> <file>
```

where who is `u` (user, that is owner), `g` (group), `o` (other), X is either `+` or `-`, perm is `r` (read), `w` (write), or `x` (execute). Examples:

```
$ chmod u+x file
```

this sets the execute permission for the file owner. Shortcut: `chmod +x file`.

```
$ chmod go-wx file
```

this removes write and execute permission for everyone except the owner.

```
$ chmod ugo+rwx file
```

this gives everyone read, write, and execute permission.

A shorter way to refer to permissions is with numbers: `rwxr-xr-x` can be expressed as 755 (every letter corresponds to a bit: `---` is 0, `--x` is 1, `-w-` is 2...).

For a directory, `rx` means that you can `cd` to that directory, and `w` means that you can delete a file in the directory (according to the file's permissions, of course), or the directory itself. All this is only part of the matter---RMP.

To change a file's owner:

```
$ chown username file
```

To sum up, a table:

```
VMS                              Linux                    Notes
-----------------------------------------------------------------------------

SET PROT=(O:RW) file.txt         $ chmod u+rw file.txt
                                 $ chmod 600 file.txt
SET PROT=(O:RWED,W) file         $ chmod u+rwx file
                                 $ chmod 700 file
SET PROT=(O:RWED,W:RE) file      $ chmod 755 file
SET PROT=(O:RW,G:RW,W) file      $ chmod 660 file
SET FILE/OWNER_UIC=JOE file      $ chown joe file
SET DIR/OWNER_UIC=JOE [.dir]     $ chown joe dir/
```

8.2 Multitasking: Processes and Jobs

More about running programs. There are no `batch queues' under Linux as you're used to; multitasking is handled very differently. Again, this is what the typical command line looks like:

```
$ command -s1 -s2 ... -sn par1 par2 ... parn < input > output &
```

where `-s1`, ..., `-sn` are the program switches, `par1`, ..., `parn` are the program parameters.

Now let's see how multitasking works. Programs, running in foreground or background, are called `processes'.

- To launch a process in background:

  ```
  $ progname [-switches] [parameters] [< input] [> output] &
  [1] 234
  ```

  the shell tells you what the `job number' (the first digit; see below) and PID (Process IDentifier) of the process are. Each process is identified by its PID.
- To see how many processes there are:

  ```
  $ ps -ax
  ```

  This will output a list of currently running processes.
- To kill a process:

  ```
  $ kill <PID>
  ```

  You may need to kill a process when you don't know how to quit it the right way... ;-). Sometimes, a process will only be killed by one of the following:

  ```
  $ kill -15 <PID>
  $ kill -9 <PID>
  ```

In addition to this, the shell allows you to stop or temporarily suspend a process, send a process to background, and bring a process from background to foreground. In this context, processes are called `jobs'.

- To see how many jobs there are:

  ```
  $ jobs
  ```

  jobs are identified by the numbers the shell gives them, not by their PID.
- To stop a process running in foreground:

```
$ CTRL-C
```

(it doesn't always work)

- To suspend a process running in foreground:

```
$ CTRL-Z
```

(ditto)

- To send a suspended process into background (it becomes a job):

```
$ bg <job>
```

- To bring a job to foreground:

```
$ fg <job>
```

- To kill a job:

```
$ kill <%job>
```

8.3 Files, Revisited

More information about files.

- **stdin, stdout, stderr**: under UNIX, every system component is treated as if it were a file. Commands and programs get their input from a `file' called `stdin` (standard input; usually, the keyboard), put their output on a `file' called `stdout` (usually, the screen), and error messages go to a `file' called `stderr` (usually, the screen). Using < and > you redirect input and output to a different file. Moreover, >> appends the output to a file instead of overwriting it; 2> redirects error messages (stderr); 2>&1 redirects stderr to stdout, while 1>&2 redirects stdout to stderr. There's a `black hole' called `/dev/null`: everything redirected to it disappears;
- **wildcards**: '`*`' is almost the same. Usage: * matches all files except the hidden ones; .* matches all hidden files; *.* matches only those that have a '.' in the middle, followed by other characters; p*r matches both `peter' and `piper'; *c* matches both `picked' and `peck'. '%' becomes '?'. There is another wildcard: the `[]`. Usage: `[abc]*` matches files starting with a, b, c; `*[I-N,1,2,3]` matches files ending with I, J, K, L, M, N, 1, 2, 3;
- `mv` (`RENAME`) doesn't work for multiple files; that is, `mv *.xxx *.yyy` won't work;
- use `cp -i` and `mv -i` to be warned when a file is going to be overwritten.

8.4 Print Queues

Your prints are queued, like under VMS. When you issue a print command, you may specify a printer name. Example:

```
$ lpr file.txt          # this goes to the standard printer
$ lpr -Plaser file.ps   # this goes to the printer named 'laser'
```

To handle the print queues, you use the following commands:

```
VMS                                     Linux
-----------------------------------------------------------------------------

$ PRINT file.ps                         $ lpr file.ps
$ PRINT/QUEUE=laser file.ps             $ lpr -Plaser file.ps
$ SHOW QUEUE                            $ lpq
```

```
$ SHOW QUEUE/QUEUE=laser                    $ lpq -Plaser
$ STOP/QUEUE                                $ lprm <item>
```

9. Configuring

Your sysadm has already provided you with a number of configuration files like `.xinitrc`, `.bash_profile`, `.inputrc`, and many others. The ones you may want to edit are:

- `.bash_profile` or `.profile`: read by the shell at login time. It's like `LOGIN.COM`;
- `.bash_logout`: read by the shell at logout. It's like `LOGOUT.COM`;
- `.bashrc`: read by non--login shells.
- `.inputrc`: this file customises the key bindings and the behaviour of the shell.

To give you an example, I'll include my `.bash_profile` (abridged):

```
# $HOME/.bash_profile

# don't redefine the path if not necessary
echo $PATH | grep $LOGNAME > /dev/null
if [ $? != 0 ]
then
  export PATH="$PATH:/home/$LOGNAME/bin"  # add my dir to the PATH
fi

export PS1='LOGNAME:\w\$ '
export PS2='Continued...>'

# aliases

alias bin="cd ~/bin" ; alias cp="cp -i" ; alias d="dir"
alias del="delete" ; alias dir="/bin/ls $LS_OPTIONS --format=vertical"
alias ed="jed" ; alias mv='mv -i'
alias u="cd .." ; alias undel="undelete"

# A few useful functions

inst() # Install a .tar.gz archive in current directory.
{
  gzip -dc $1 | tar xvf -
}
cz() # List the contents of a .zip archive.
{
  unzip -l $*
}
ctgz() # List the contents of a .tar.gz archive.
{
  for file in $* ; do
    gzip -dc ${file} | tar tf -
```

```
  done
}
tgz() # Create a .tgz archive a la zip.
{
  name=$1 ; tar -cvf $1 ; shift
  tar -rf ${name} $* ; gzip -S .tgz ${name}
}
```

And this is my `.inputrc`:

```
# $HOME/.inputrc
# Last modified: 16 January 1997.
#
# This file is read by bash and defines key bindings to be used by the shell;
# what follows fixes the keys END, HOME, and DELETE, plus accented letters.
# For more information, man readline.

"\e[1~": beginning-of-line
"\e[3~": delete-char
"\e[4~": end-of-line

set bell-style visible
set meta-flag On
set convert-meta Off
set output-meta On
set horizontal-scroll-mode On
set show-all-if-ambiguous On

# (F1 .. F5) are "\e[[A" ... "\e[[E"

"\e[[A": "info "
```

---

10. Useful Programs

10.1 Browsing Files: `less`

You'll use this file browser every day, so I'll give you a couple of tips to use it at best. First of all, ask your sysadm to configure `less` so as it can display not only plain text files, but also compressed files, archives, and so on.

The main advantage of `less` over `TYPE` is that you can browse files in both directions. It also accepts several commands that are issued pressing a key. The most useful are:

- first of all, press `q` to leave the browser;
- `h` gives you extensive help;
- `g` to go to beginning of file, `G` to the end, number+`g` to go to line `number' (e.g. `125g`), number+% to move to that percentage of the file;
- `/pattern` searches forwards for `pattern'; `n` searches forwards for the next match; `?pattern` and `N` search backwards;
- `m`+letter marks current position (e.g. `ma`); `'`+letter go to the marked position.
- `:e` examines a new file;
- `!command` executes the shell command.

10.2 RCS in a Nutshell

The lack of version numbers in files can be easily overcome by using RCS (Revision Control System). This allows you to maintain several versions of the same file, and offers many more advantages. I'll only explain the very basics of this powerful version control system.

The most important commands are `ci` and `co`. The first (``check in'') is used to commit the changes you have done to your file, and create a new version. The second (``check out'') is used to obtain a working copy of your file from the RCS system, either to modify it or simply use it for browsing, printing, or whatever.

Let's see an example. First of all you create an *initial revision* of your file, using your favourite editor. Let's suppose that the file you'll have under RCS control is called `project.tex`. Follow these steps:

- make a subdirectory called `RCS/` in the directory containing `project.tex`. `RCS/` will contain the revision control file;
- to put `project.tex` under RCS control, issue the command

```
$ ci project.tex
RCS/project.tex,v  <-- project.tex
enter description, terminated with a single '.' or end of file:
```

```
NOTE: This is NOT the log message!
>>
```

- you will write a line or more containing a description of the contents of your file. End it with a line containing a '.' by itself, and you'll see

```
initial revision: 1.1
done
```

Now the file `project.tex` has been taken over by RCS.

Using the latest version

Whenever you want to use, but not modify, the latest version of project.tex, you issue the command

```
$ co project.tex
RCS/project.tex,v  --> project.tex
revision 1.1
done
```

This extracts the latest version (read only) of your file. Now you can browse it, or compile it with tex, but you can't modify it.

Creating a new version

When you want to modify your file, you must obtain a ``lock'' on it. This means that RCS knows that you're about to make a newer version. In this case, you use the command

```
$ co project.tex
RCS/project.tex,v  --> project.tex
revision 1.1 (locked)
done
```

You now have a working copy you can modify with your editor. When you're done editing it, you check it in again to commit the changes:

```
$ ci project.tex
RCS/project.tex,v  <-- project.tex
new revision 1.2; previous revision: 1.1
enter log message, terminated with a single '.' or end of file:
>> (enter your description here)
>> .
done
```

If you want to change the version number, type `ci -f2.0 project.tex`.

TOC

Comparing versions

If you want to see the history of the changes in project.tex, issue

```
$ rlog project.tex
```

Using an old version

To extract an older version of your file (say, version 1.2 when you're working on 1.6), issue

```
$ co -r1.2 project.tex
```

Be aware that this overwrites your existing working file, if you have one. You may do:

```
$ co -r1.2 -p project.tex > project.tex.1.2
```

10.3 Archiving: tar & gzip

Under UNIX there are some widely used applications to archive and compress files. `tar` is used to make archives, that is collections of files. To make a new archive:

```
$ tar -cvf <archive_name.tar> <file> [file...]
```

To extract files from an archive:

```
$ tar -xpvf <archive_name.tar> [file...]
```

To list the contents of an archive:

```
$ tar -tf <archive_name.tar> | less
```

Files can be compressed to save disk space using `compress`, which is obsolete and shouldn't be used any more, or `gzip`:

```
$ compress <file>
$ gzip <file>
```

that creates a compressed file with extension .Z (`compress`) or .gz (`gzip`). These programs don't make archives, but compress files individually. To decompress, use:

```
$ compress -d <file.Z>
$ gzip -d <file.gz>
```

RMP.

The `unarj`, `zip` and `unzip` utilities are also available. Files with extension `.tar.gz` or `.tgz`

(archived with `tar`, then compressed with `gzip`) are very common in the UNIX world. Here's how to the contents of a `.tar.gz` archive:

```
$ gzip -dc <file.tar.gz> | tar tf - | less
```

To extract the files from a `.tar.gz` archive:

```
$ gzip -dc <file.tar.gz> | tar xvf -
```

---

11. Real Life Examples

UNIX' core idea is that there are many simple commands that can linked together via piping and redirection to accomplish even really complex tasks. Look at the following examples; I'll only explain the most complex ones, for the others, please study the above sections and the man pages.

Problem: `ls` is too quick and the file names fly away.

Solution:

```
$ ls | less
```

Problem: I have a file containing a list of words. I want to sort it in reverse order and print it.

Solution:

```
$ cat myfile.txt | sort -r | lpr
```

Problem: my datafile has some repeated lines! How do I get rid of them?

Solution:

```
$ sort datafile.dat | uniq > newfile.dat
```

Problem: I have a file called 'mypaper.txt' or 'mypaper.tex' or some such somewhere, but I don't remember where I put it. How do I find it?

Solution:

```
$ find ~ -name "mypaper*"
```

Explanation: `find` is a very useful command that lists all the files in a directory tree (starting from ~ in this case). Its output can be filtered to meet several criteria, such as `-name`.

Problem: I have a text file containing the word 'entropy' in this directory, is there anything like `SEARCH`?

Solution: yes, try

```
$ grep -l 'entropy' *
```

Problem: somewhere I have text files containing the word 'entropy', I'd like to know which and where they are. Under VMS I'd use `search entropy [...]*.*.*`, but `grep` can't recurse subdirectories. Now what?

Solution:

```
$ find . -exec grep -l "entropy" {} \; 2> /dev/null
```

Explanation: `find .` outputs all the file names starting from the current directory, `-exec grep -l`

TOC

"entropy" is an action to be performed on each file (represented by {}), \ terminates the command. If you think this syntax is awful, you're right.

In alternative, write the following script:

```
#!/bin/sh
# rgrep: recursive grep
if [ $# != 3 ]
then
  echo "Usage: rgrep --switches 'pattern' 'directory'"
  exit 1
fi
find $3 -name "*" -exec grep $1 $2 {} \; 2> /dev/null
```

Explanation: `grep` works like `search`, and combining it with `find` we get the best of both worlds.

Problem: I have a data file that has two header lines, then every line has 'n' data, not necessarily equally spaced. I want the 2nd and 5th data of each line. Shall I write a Fortran program...?

Solution: nope. This is quicker:

```
$ awk 'NL > 2 {print $2, "\t", $5}' datafile.dat > newfile.dat
```

Explanation: the command `awk` is actually a programming language: for each line starting from the third in `datafile.dat`, print out the second and fifth field, separated by a tab. Learn some `awk`---it saves a lot of time.

Problem: I've downloaded an FTP site's `ls-lR.gz` to check its contents. For each subdirectory, it contains a line that reads "total xxxx", where xxxx is size in kbytes of the dir contents. I'd like to get the grand total of all these xxxx values.

Solution:

```
zcat ls-lR.gz | awk ' $1 == "total" { i += $2 } END {print i}'
```

Explanation: `zcat` outputs the contents of the `.gz` file and pipes to `awk`, whose man page you're kindly requested to read ;-)

Problem: I've written a Fortran program, `myprog`, to calculate a parameter from a data file. I'd like to run it on hundreds of data files and have a list of the results, but it's a nuisance to ask each time for the file name. Under VMS I'd write a lengthy command file, and under Linux?

Solution: a very short script. Make your program look for the data file '`mydata.dat`' and print the result on the screen (stdout), then write the following script:

```
#!/bin/sh
# myprog.sh: run the same command on many different files
# usage: myprog.sh *.dat
```

TOC

```
for file in $*  # for all parameters (e.g. *.dat)
do
  # append the file name to result.dat
  echo -n "${file}:    " >> results.dat
  # copy current argument to mydata.dat, run myprog
  # and append the output to results.dat
  cp ${file} mydata.dat ; myprog >> results.dat
done
```

Problem: I want to replace `geology' with `geophysics' in all my text files. Shall I edit them all manually?

Solution: nope. Write this shell script:

```
#!/bin/sh
# replace $1 with $2 in $*
# usage: replace "old-pattern" "new-pattern" file [file...]
OLD=$1          # first parameter of the script
NEW=$2          # second parameter
shift ; shift   # discard first two parameters: the next are the file names
for file in $*  # for all files given as parameters
do
# replace every occurrence of OLD with NEW, save on a temporary file
  sed "s/$OLD/$NEW/g" ${file} > ${file}.new
# rename the temporary file as the original file
  /bin/mv ${file}.new ${file}
done
```

Problem: I have some data files, I don't know their length and have to remove their last but one and last but two lines. Er... manually?

Solution: no, of course. Write this script:

```
#!/bin/sh
# prune.sh: removes n-1th and n-2th lines from files
# usage: prune.sh file [file...]
for file in $*   # for every parameter
do
  LINES=`wc -l $file | awk '{print $1}'`  # number of lines in file
  LINES=`expr $LINES - 3`                 # LINES = LINES - 3
  head -n $LINES $file > $file.new        # output first LINES lines
  tail -n 1 $file >> $file.new            # append last line
done
```

I hope these examples whetted your appetite...

---

12. Tips You Can't Do Without

- **Command completion**: pressing <TAB> when issuing a command will complete the command line for you. Example: you have to type `less this_is_a_long_name`; typing in `less thi<TAB>` will suffice. (If you have other files that start with the same characters, supply enough characters to resolve any ambiguity.)
- **Back-scrolling**: pressing SHIFT--PAG UP (the grey key) allows you to backscroll a few pages, depending on your PC's video memory.
- **Resetting the screen**: if you happen to `more` or `cat` a binary file, your screen may end up full of garbage. To fix things, blind type `reset` or this sequence of characters: `echo CTRL-V ESC c RETURN`.
- **Pasting text**: in console, see below; in X, click and drag to select the text in an `xterm` window, then click the middle button (or the two buttons together if you have a two-button mouse) to paste.
- **Using the mouse**: ask your sysadm to install `gpm`, a mouse driver for the console. Click and drag to select text, then right click to paste the selected text. It works across different VCs.

13. The End

13.1 Copyright

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu via email.

14. Disclaimer

This work was written following the experience we had at the Settore di Geofisica of Bologna University (Italy), where a VAX 4000 is being superseded and replaced by Linux-based Pentium PCs. Most of my colleagues are VMS users, and some of them have switched to Linux.

``From VMS to Linux HOWTO'' was written by Guido Gonzato, guido@ibogfs.cineca.it, 1997. Many thanks to my colleagues and friends who helped me define the needs and habits of the average VMS user, especially to Dr. Warner Marzocchi.

Please help me improve this HOWTO. I'm not a VMS expert and never will be, so your suggestions and bug reports are more than welcome.

Enjoy,

Guido =8-)

Linux VPN Masquerade HOWTO

John D. Hardin <jhardin@wolfenet.com>

v2.2, 9 September 1999

This document describes how to configure a Linux firewall to masquerade IPsec- and PPTP-based Virtual Private Network traffic, allowing you to establish a VPN connection without losing the security and flexibility of your Linux firewall's internet connection and allowing you to make available a VPN server that does not have a registered internet IP address. Information on configuring the VPN client and server is also given.

1. Introduction

- 1.1 Introduction
- 1.2 Feedback, Credits & Resources
- 1.3 Copyright & Disclaimer

2. Background Knowledge

- 2.1 What is a VPN?
- 2.2 What is IPsec?
- 2.3 What is PPTP?
- 2.4 Why masquerade a VPN client?
- 2.5 Can several clients on my local network use IPsec simultaneously?
- 2.6 Can several clients on my local network use PPTP simultaneously?
- 2.7 Can I access the remote network from my entire local network?
- 2.8 Why masquerade the VPN server?
- 2.9 Why patch the Linux kernel?
- 2.10 Current Status

TOC

3. Configuring the Linux firewall

- 3.1 Example network
- 3.2 Determining what needs to be done on the firewall
- 3.3 Patching and configuring the 2.0.x kernel for VPN Masquerade support
- 3.4 Patching and configuring the 2.2.x kernel for VPN Masquerade support
- 3.5 ipfwadm setup for a Private-IP VPN Client or Server
- 3.6 ipchains setup for a Private-IP VPN Client or Server
- 3.7 A note about dynamic IP addressing
- 3.8 Additional setup for a Private-IP VPN Server
- 3.9 ipfwadm setup for a Registered-IP VPN Server
- 3.10 ipfwadm setup for a Registered-IP VPN Client

4. Configuring the VPN client

- 4.1 Configuring a MS W'95 or W'98 client
- 4.2 Configuring a MS NT client

5. Troubleshooting

- 5.1 Testing
- 5.2 Possible problems
- 5.3 Troubleshooting
- 5.4 MS PPTP Clients and domain-name issues
- 5.5 MS PPTP Clients and Novell IPX
- 5.6 MS network password issues
- 5.7 If your IPsec session always dies after a certain amount of time

6. IPsec masquerade technical notes and special security considerations

- 6.1 Limitations and weaknesses of IPsec masquerade
- 6.2 Proper routing of inbound encrypted traffic

---

1. Introduction

1.1 Introduction

This document describes how to configure masquerading of IPsec and PPTP VPN traffic. SSH-based VPNs (such as that sold by F-Secure and outlined in the VPN mini-HOWTO) are based on standard TCP traffic and do not need any special kernel modifications.

VPN Masquerade allows you to establish one or more IPsec and/or PPTP sessions to internet-accessible VPN servers via your Linux internet firewall without forcing you to connect to your ISP directly from the VPN client system - thus retaining all of the benefits of your Linux internet firewall. It also allows you to set up a VPN server with a Private Network IP address (as described in RFC1918) behind a masquerading Linux firewall, permitting you to provide relatively secure access to a private network via only one registered IP address - even if that IP address represents a dynamic dial-up link.

It is strongly recommended that you understand, configure and test regular IP Masquerading before you attempt to set up VPN masquerading. Please see the IP Masquerade HOWTO and the IP Masquerade Resource page at http://ipmasq.cjb.net/ before proceeding.

The patch for the 2.0.x-series kernels works well on Linux kernel version 2.0.36, has been incorporated into the 2.0.37 release, may work on versions earlier than 2.0.36, and should work on Linux kernels up to about version 2.1.102. The IP masquerade code in the kernel was restructured at about version 2.1.103, requiring a different patch for the 2.1.105+ and 2.2.x series of kernels. A patch is available for kernels from 2.2.5 to 2.2.12, and it may work on earlier kernels.

1.2 Feedback, Credits & Resources

Please feel free to send any feedback or comments regarding this document to me at <jhardin@wolfenet.com>.

I personally have experience with masquerading a MS NT-Server-based PPTP client, configuring a registered-IP PPTP server, and using PPTP for network-to-network routing. The information on masquerading a Private-IP PPTP server is from discussions with Len Bayles <len@isdi.com>, Simon Cocking <simon@ibs.com.au> and C. Scott Ananian <cananian@lcs.mit.edu>.

The current version of this document can be found at ftp://ftp.rubyriver.com/pub/jhardin/masquerade/VPN-Masquerade.sgml - it and other formats (such as HTML) can be found via the Linux Documentation Project's HOWTO repository and in the `/usr/doc/HOWTO/` directory on your nearest Linux system.

The home page for the Linux VPN Masquerade kernel patch is

http://www.wolfenet.com/~jhardin/ip_masq_vpn.html

The home page for the PPTP-only Masquerade kernel patch for the 2.1.105+ and 2.2.x kernel series is http://bmrc.berkeley.edu/people/chaffee/linux_pptp.html.

The home page for the `ipportfw` port-forwarding kernel patch and configuration tool is http://www.ox.compsoc.org.uk/~steve/portforwarding.html.

The home page for the `ipfwd` generic IP redirector is http://www.pdos.lcs.mit.edu/~cananian/Projects/IPfwd/.

Profuse thanks to Gordon Chaffee <chaffee@cs.berkeley.edu> for coding and sharing a patch to traceroute that allows tracing GRE traffic. It should prove invaluable in troubleshooting if your GRE traffic is being blocked somewhere. The patch is available at http://www.wolfenet.com/~jhardin/pptp-traceroute.patch.gz

More thanks to Steve Chinatti <chinatti@alumni.Princeton.EDU> for contributing his original IPsec masquerade hack, from which I shamelessly stole some very important ideas...

More information on setting up firewall rules to run automatically - including how to automatically use the correct IP address in a dynamic-IP environment - can be found at http://www.wolfenet.com/~jhardin/ipfwadm/invocation.html

The home page for Linux FreeS/WAN (IPsec for Linux) is http://www.xs4all.nl/~freeswan

The home page for the Linux PPTP project is http://www.pdos.lcs.mit.edu/~cananian/Projects/PPTP and a patch to add PPTP server capability is available at http://debs.fuller.edu/cgi-bin/display?list=pptp&msg=222

A second Linux PPTP server called PoPToP is available at http://www.moretonbay.com/vpn/pptp.html

Paul Cadach <paul@odt.east.telecom.kz> has made patches that add MS-CHAP-v2, MPPE and Multilink support to Linux pppd. See ftp://ftp.east.telecom.kz/pub/src/networking/ppp/ppp-2.3.5-my.tgz for MS-CHAP and MPPE, and ftp://ftp.east.telecom.kz/pub/src/networking/ppp/multilink/ppp-2.3.5-mp.tgz for Multilink.

1.3 Copyright & Disclaimer

This document is copyright © 1999 by John D. Hardin. Permission is granted to redistribute it under the terms of the GNU General Public License.

The information presented in this document is correct to the best of my knowledge. IP Masquerading is *experimental*, and it is possible that I have made a mistake in writing or testing the kernel patch or composing the instructions in this document; you should determine for yourself if you want to make the changes outlined in this document.

THE AUTHOR IS NOT RESPONSIBLE FOR ANY DAMAGES INCURRED DUE TO ACTIONS TAKEN BASED ON THE INFORMATION IN THIS DOCUMENT.

BACK UP ANY AND ALL CRITICAL INFORMATION BEFORE IMPLEMENTING THE CHANGES OUTLINED IN THIS DOCUMENT. MAKE SURE YOU HAVE A WORKING, BOOTABLE KERNEL AVAILABLE BEFORE PATCHING AND RECOMPILING YOUR KERNEL AS OUTLINED IN THIS DOCUMENT.

In other words, take sensible precautions.

---

2. Background Knowledge

2.1 What is a VPN?

A Virtual Private Network, or "VPN", is a tunnel that carries private network traffic from one endpoint system to another over a public network (such as the Internet) without the traffic being aware that there are intermediate hops between the endpoints, or the intermediate hops being aware they are carrying the network packets that are traversing the tunnel. The tunnel may optionally compress and/or encrypt the data, providing enhanced performance and some measure of security.

The "Virtual" part stems from the fact that you are constructing a private link over a public network, rather than actually buying a direct hardwired link over leased lines. The VPN allows you to pretend you are using a leased line to communicate between the endpoints.

You may find the VPN FAQ at http://kubarb.phsx.ukans.edu/~tbird/FAQ.html informative.

2.2 What is IPsec?

IPsec is a set of standard protocols for implementing secure communications and encryption key exchange between computers. It can be used to implement a VPN.

An IPsec VPN generally consists of two communications channels between the the endpoints: a key-exchange channel over which authentication and encryption key information is passed, and one or more data channels over which private network traffic is carried.

The key-exchange channel is a standard UDP connection to and from port 500. The data channels carrying the traffic between the client and server use IP protocol number 50 (ESP).

More information is available in F-Secure's IPsec FAQ at http://www.data-fellows.com/f-secure/support/vpn-plus/faq/techfaq.htm, and in RFC2402 (the AH protocol), RFC2406 (the ESP protocol), and RFC2408 (the ISAKMP key-exchange protocol).

Important note: If your VPN is based on the AH protocol, it cannot be masqueraded. The AH protocol specifies a cryptographic checksum across portions of the IP header, including the IP addresses. IP Masquerade is implemented by modifying the source IP address for outbound packets and the destination IP address for inbound packets. Since the masquerading gateway cannot participate in the encryption key exchange, it cannot generate the correct cryptographic checksums for the modified IP headers. Thus the modified IP packets will be discarded by the recipient as invalid, because they fail the cryptographic checksum test.

TOC

2.3 What is PPTP?

PPTP stands for ***P***oint-to-***P***oint ***T***unnelling ***P***rotocol. It is a Microsoft-proposed protocol for implementing a VPN.

The PPTP VPN protocol consists of two communications channels between the client and server: a control channel over which link-management information is passed, and a data channel over which (possibly encrypted) private network traffic is carried.

The control channel is a standard TCP connection to port 1723 on the server. The data channel carrying the private network traffic uses IP protocol number 47 (GRE), a generic encapsulation protocol described in RFC1701. The transparent transmission of data over the data channel is achieved by negotiating a standard PPP connection over it, just as if it were a dialup connection directly from the client to the server. The options negotiated over the tunnel by PPP control whether the data is compressed and/or encrypted, thus PPTP itself has nothing to do with encryption.

The details of the PPTP protocol are documented at http://www.ietf.org/internet-drafts/draft-ietf-pppext-pptp-08.txt.

Microsoft's implementation of the PPTP protocol is not considered very secure. If you're interested in the details, here are three seperate analyses:

http://www.counterpane.com/pptp.html

http://www.geek-girl.com/bugtraq/1999_1/0664.html

http://oliver.efri.hr/~crv/security/bugs/NT/pptp2.html

2.4 Why masquerade a VPN client?

Most current VPN clients assume you will be connecting the client computer directly to the internet. Doing this when you have only a single connection for internet access bypasses your Linux firewall and the security and access-sharing capabilities that it provides. Extending the Linux firewall to also masquerade VPN traffic allows you to retain the firewalling security provided by the Linux firewall as well as permitting the other systems on your local network to access the internet regardless of whether or not the VPN network connection is active.

If your firewall is being used in a corporate setting you may also wish to require your VPN client users to go through that firewall for security reasons, rather than providing them with modems so they can dial out on their own when they need to use VPN. VPN Masquerade allows you to do so even if the desktops do not have registered IP addresses.

TOC

2.5 Can several clients on my local network use IPsec simultaneously?

Yes, though there may occasionally be minor problems.

The IPsec protocols define a method for identifying the traffic streams called the Security Parameters Index ("SPI"). Unfortunately the SPI used by outbound traffic is different from the SPI used by inbound traffic, and there is no other identifying information available that is not encrypted, so association of the inbound and outbound data streams is difficult and not perfectly reliable.

IPsec Masquerade attempts to associate inbound and outbound ESP traffic by serializing new connections. While this has worked well in testing, it cannot be guaranteed to be perfectly reliable, and the serialization of new traffic may result in some timeouts if the link is saturated or if many local IPsec hosts attempt to initiate communications or rekey with the same remote IPsec host simultaneously.

It is also assumed that should this association scheme fail to associate the traffic streams correctly, the IPsec hosts themselves will discard the incorrectly routed traffic because it will have the wrong SPI values. This is required by the IPsec RFC.

These problems could be eliminated if there was some way to sniff the new SPI values from the ISAKMP key exchange before any ESP traffic appears, but unfortunately that portion of the key exchange is encrypted.

To minimize the problems associated with this, it is recommended that you open a command window on your masqueraded IPsec host and run the "ping" program pinging a host on the remote network for as long as you have the tunnel up.

See the IPsec technical notes at the end of the document for more details.

2.6 Can several clients on my local network use PPTP simultaneously?

Yes.

You must enable PPTP Call ID masquerade when configuring your kernel in order to distinguish between multiple data streams from the same server. PPTP masq with Call ID masq enabled will support many concurrent masqueraded sessions with no restrictions on which server a client can call.

For another alternative, see the next question...

2.7 Can I access the remote network from my entire local network?

Yes. However, your VPN client must be able to forward IP traffic.

TOC

This means that you'll either have to use a Linux VPN client or a MS NT VPN client. The IP stack in W'95 and W'98 does not support IP forwarding. NT Workstation will work for this, and is less expensive than NT Server if you're only using it to route encrypted traffic.

If you cannot install a Linux or NT-based VPN client, then you'll have to enable PPTP Call-ID masquerade if you are using PPTP, and install VPN client software on every system you want to provide access for. This is inefficient, aesthetically revolting and a security weakness, but it's cheaper than licensing NT.

Network-to-network routing this way works very well. This is how I have my home network set up for telecommuting. It does require a little more networking knowhow than simply giving everybody their own VPN client.

In my experience, network-to-network routing in a pure-MS environment requires RRAS be installed at both ends of the tunnel.

2.8 Why masquerade the VPN server?

If your VPN server has a registered IP address you do not need to masquerade it, simply configure your firewall to route the VPN traffic properly as described below.

If your VPN server has a Private-Network IP address you will need to redirect the inbound traffic to it and masquerade the outbound traffic from it. Masquerading allows you to make a VPN server available to the internet even if you only have one assigned IP address. This should work even if your IP address is dynamically assigned: you would publicize the IP address for clients through the use of a third-party dynamic DNS service such as that provided by DDNS.ORG and configure the clients to connect to a system named `our-company.ddns.org` or something similar. Note that this is a security risk, because it is possible for an incorrect IP address to be retrieved from the dynamic DNS server through timing problems, a failure to properly register the current dynamic IP address, or a third party registering a different IP address under the system name.

2.9 Why patch the Linux kernel?

The largest problem in masquerading VPN traffic is that the stock 2.0.x Linux kernel has no special awareness of IP protocols other than TCP, UDP and ICMP.

All IP traffic may be forwarded and filtered by IP address, but filtering on or masquerading IP protocols other than TCP, UDP and ICMP requires modifying the kernel.

The PPTP control channel is plain TCP and requires no special setup beyond letting it through the firewall and masquerading it.

Masquerading the IPsec and PPTP data channels requires a modification that adds support for the ESP and GRE protocols to the masquerading code, and masquerading the ISAKMP key exchange protocol requires a modification that prevents masquerade from altering the UDP source port number and adds tracking of the ISAKMP cookie values.

2.10 Current Status

TOC

The 2.0.x kernel patch works on kernel 2.0.36 and is incorporated into the standard 2.0.37 kernel release. It may work on earlier kernels but I have not tested it, and I recommend you upgrade to the latest kernel anyway for security reasons if you are running a kernel that old.

The 2.2.x kernel patch works on kernels from 2.2.5 to 2.2.12 and may work on earlier kernels, but that has not been tested. It will be submitted for inclusion in the standard 2.2.13 release if all goes well.

The 2.0.x kernel patch has been tested and works on x86 and Sparc systems, and the 2.2.x kernel patch has been tested and works on x86 systems, but there should be no major problems in porting to other architectures. I believe the architecture dependencies are only in endian-ness within the bitmaps in the GRE header definition used to format debugging log messages. If anyone ports this to a non-Intel architecture I'd appreciate hearing about it so I can merge any changes into the master copy.

A PPTP-only kernel patch for the 2.1.105+ and 2.2.x kernel series is available at http://bmrc.berkeley.edu/people/chaffee/linux_pptp.html.

See the VPN Masquerade home page at http://www.wolfenet.com/~jhardin/ip_masq_vpn.html for the status of the VPN Masq patches, and http://bmrc.berkeley.edu/people/chaffee/linux_pptp.html for the status of the 2.1.105+/2.2.x PPTP-only Masq patch.

---

3. Configuring the Linux firewall

3.1 Example network

For the Private-IP configuration examples in this document we will use this sample network:

```
Internet-------- 200.200.200.*   ppp0 or  200.200.200.200 eth1
                                 Dual-Homed Linux Firewall
             .--- 10.0.0.1       eth0
             |
             |--- 10.0.0.2       VPN client or server
             |
```

For the registered-IP configuration examples in this document we will use this sample network:

```
Internet-------- 200.200.200.200 eth1
                                 Dual-Homed Linux Firewall
             .--- 222.0.0.1      eth0
             |
             |--- 222.0.0.2      VPN client or server
             |
```

The VPN server that the example clients connect to will be `199.0.0.1`

The VPN clients that the connect to the example server will be `199.0.0.2` and `199.0.0.3`

3.2 Determining what needs to be done on the firewall

If your VPN client or server has a registered internet IP address you do *not* need to masquerade or modify your kernel - the stock kernel will successfully route all VPN traffic. You can skip directly to the registered-IP setup sections below.

If your VPN client or server has a Private-Network IP address as described in RFC1918 you will need to patch your kernel (unless your kernel is 2.0.37 or higher in the 2.0.x series).

If you are setting up a masqueraded VPN server, you will also have to obtain and install the following two packages:

- To redirect the inbound TCP/UDP traffic (the 1723/tcp PPTP control channel or the 500/udp ISAKMP channel), you need the appropriate `ipportfw` port-forwarding kernel patch and configuration tool from http://www.ox.compsoc.org.uk/~steve/portforwarding.html. Port forwarding has been incorporated into the 2.2.x kernel. See `man ipmasqadm` for configuration details.

  If `ipmasqadm` is not included with your distribution it can be obtained at http://juanjox.linuxhq.com/
- To redirect the initial inbound tunnel traffic (GRE for PPTP and ESP for IPsec), you need the `ipfwd` generic-IP proxy from http://www.pdos.lcs.mit.edu/~cananian/Projects/IPfwd/.

You do not need port forwarding or ipfwd if you are only masquerading clients.

3.3 Patching and configuring the 2.0.x kernel for VPN Masquerade support

1. Install the kernel source (preferably version 2.0.37), which you can obtain from http://www.kernel.org/ or a mirror. The sources should be automatically extracted into a directory named `/usr/src/linux`.
2. Configure and test standard IP Masquerading (see the IP Masquerade HOWTO). Doing this will familiarize you with recompiling your kernel and introduce you to IP Masquerading in general.

TOC

3. *Back up your kernel sources.*
4. Obtain the kernel patch if necessary.

   If your kernel version is 2.0.36 or lower, obtain the 2.0.x VPN Masquerade kernel patch from the VPN Masquerade home page in the "Resources" section above.

   If your kernel version is 2.0.37 or higher in the 2.0.x series, you do not need to apply any patches. The VPN Masquerade code is included in the kernel. Skip the discussion of patching the kernel.

   For the purposes of this document we'll assume you've saved the appropriate patch in `/usr/src/ip_masq_vpn.patch.gz`.
5. Apply the VPN Masquerade patch to your kernel if necessary:
   - Change to the kernel source directory:

     ```
     cd /usr/src/linux
     ```
   - Apply the patch:

     ```
     gzcat ../ip_masq_vpn.patch.gz | patch -l -p0 > vpn-patch.log
     ```

     Note that the options are "dash lowercase L, dash lowercase P zero".

     Also, you may get odd results if you change the order of the arguments. Patch seems to be sensitive to the order they appear on the command line.
   - Check the `vpn-patch.log` file to see if any hunks failed. If you get failed hunks, then you probably either omitted the options or ran the patch program from the wrong directory. Restore your kernel from the backup and try again.
6. If you are masquerading a VPN server, obtain and install the `ipportfw` patch from the site given above.

   There is a known conflict between the VPN Masquerade patch and two other networking patches: the IP Firewall Chains patch and the ipportfw patch. They are all trying to add options at the same location in `net/ipv4/Config.in`, and the changes made by one patch alter the context that the other patches are looking for.

   If you're applying the VPN Masquerade patch and the IP Firewall Chains or ipportfw patches to your 2.0.x kernel, you will have to manually edit `net/ipv4/Config.in` and add the block of configuration options from the patch file that fails to work. Looking at the patch file should show you where in `net/ipv4/Config.in` the new options should be added.

   The syntax of patch files is simple. For each block of changes to make, there are two sections: the first shows the "before" state, with an indication of lines to be changed or deleted; the second shows the "after" state, with an indication of the lines that have been changed or added. Use the first section to find where to add the lines, and add the lines that are indicated in the second section.

   This should not be a problem once those patches are updated for 2.0.37+
7. Configure your kernel and select the following options - say *YES* to the following:

```
* Prompt for development and/or incomplete code/drivers
  CONFIG_EXPERIMENTAL
  - You must enable this to see the VPN Masq options.

* Networking support
  CONFIG_NET

* Network firewalls
  CONFIG_FIREWALL

* TCP/IP networking
  CONFIG_INET

* IP: forwarding/gatewaying
  CONFIG_IP_FORWARD

* IP: firewalling
  CONFIG_IP_FIREWALL
```

```
* IP: masquerading (EXPERIMENTAL)
  CONFIG_IP_MASQUERADE
  - This is required.

* IP: PPTP masq support (EXPERIMENTAL)
  CONFIG_IP_MASQUERADE_PPTP
  - Enables PPTP data channel masquerading, if you are
    masquerading a PPTP client or server.

* IP: PPTP Call ID masq support (EXPERIMENTAL)
  CONFIG_IP_MASQUERADE_PPTP_MULTICLIENT
  - Enables PPTP Call ID masquerading; only necessary if
    you will be masquerading more than one client trying
    to connect to the same remote server. DO NOT enable
    this option if you will be masquerading a PPTP server.

* IP: IPsec ESP & ISAKMP masq support (EXPERIMENTAL)
  CONFIG_IP_MASQUERADE_IPSEC
  - Enables IPsec masquerade, if you are masquerading an
    IPsec host.

* IP: IPSEC masq table lifetime (minutes)
  - See your network administrator to determine what the
    "rekey interval" or "key lifetime" is set to. The
    default lifetime of masq table entries is thirty minutes.
    If your rekey interval is greater than thirty minutes,
    then you should increase the lifetime to a value slightly
    greater than the rekey interval.

* IP: always defragment
  CONFIG_IP_ALWAYS_DEFRAG
  - Highly recommended for a firewall.
```

NOTE: These are just the settings you need for masquerading. Select whatever other options you need for your specific setup.

8. Recompile the kernel and install it for testing. Don't replace a known working kernel with your new kernel until you have proven it works.

To determine whether the running kernel includes VPN Masquerade support, run the following command:

```
grep masq /proc/ksyms
```

Look for the following entries:

- IPsec masquerade: `ip_masq_out_get_isakmp`, `ip_masq_in_get_isakmp`, `ip_fw_masq_esp` and `ip_fw_demasq_esp`
- PPTP masquerade: `ip_fw_masq_gre` and `ip_fw_demasq_gre`
- PPTP Call-ID masquerade: `ip_masq_pptp`

If you don't see these entries, VPN Masquerade support is probably not available. If you get complaints about `/proc/ksyms` not being available or `/proc` not being available, make sure that you have enabled the `/proc` filesystem in your kernel configuration.

See the Kernel HOWTO for more details on configuring and recompiling your kernel.

3.4 Patching and configuring the 2.2.x kernel for VPN Masquerade support

1. Install the kernel source (preferably version 2.2.10 or later), which you can obtain from http://www.kernel.org/ or a mirror. The

TOC

sources should be automatically extracted into a directory named `/usr/src/linux`.

2. Configure and test standard IP Masquerading (see the IP Masquerade HOWTO). Doing this will familiarize you with recompiling your kernel and introduce you to IP Masquerading in general.
3. *Back up your kernel sources.*
4. Obtain the kernel patch from the VPN Masquerade home page in the "Resources" section above.

   For the purposes of this document we'll assume you've saved the appropriate patch in `/usr/src/ip_masq_vpn.patch.gz`.
5. Apply the VPN Masquerade patch to your kernel if necessary:
   - Change to the source directory:

     ```
     cd /usr/src
     ```
   - Apply the patch:

     ```
     gzcat ip_masq_vpn.patch.gz | patch -l -p0 > vpn-patch.log
     ```

     Note that the options are "dash lowercase L, dash lowercase P zero".

     Also, you may get odd results if you change the order of the arguments. Patch seems to be sensitive to the order they appear on the command line.

     Also note that the directory you run the patch command in is different for the 2.2.x kernel patch
   - Check the `vpn-patch.log` file to see if any hunks failed. If you get failed hunks, then you probably either omitted the options or ran the patch program from the wrong directory. Restore your kernel from the backup and try again.
6. If you are masquerading a VPN server you do *not* need the `ipportfw` patch as port forwarding is now built-in. See the `ipmasqadm` man page for more details.
7. Configure your kernel and select the following options - say *YES* to the following:

```
* Prompt for development and/or incomplete code/drivers
  CONFIG_EXPERIMENTAL
  - You must enable this to see the VPN Masq options.

* Networking support
  CONFIG_NET

* Network firewalls
  CONFIG_FIREWALL

* TCP/IP networking
  CONFIG_INET

* IP: firewalling
  CONFIG_IP_FIREWALL

* IP: always defragment
  CONFIG_IP_ALWAYS_DEFRAG
  - Required for masquerading

* IP: masquerading (EXPERIMENTAL)
  CONFIG_IP_MASQUERADE
  - This is required.

* IP: masquerading special modules support
  CONFIG_IP_MASQUERADE_MOD
  - This is required.

* IP: ipportfw masq support (EXPERIMENTAL)
  CONFIG_IP_MASQUERADE_IPPORTFW
  - Enable this if you will be masquerading a VPN server.
```

```
* IP: PPTP masq support
  CONFIG_IP_MASQUERADE_PPTP
  - Enables PPTP data channel masquerading, if you are
    masquerading a PPTP client or server. This is now
    available as a module.
    Note that you no longer need to specify Call-ID masquerade.

* IP: IPsec ESP & ISAKMP masq support (EXPERIMENTAL)
  CONFIG_IP_MASQUERADE_IPSEC
  - Enables IPsec masquerade, if you are masquerading an
    IPsec host. This is now available as a module.

* IP: IPsec masq table lifetime (minutes)
  - See your network administrator to determine what the
    "rekey interval" or "key lifetime" is set to. The default
    lifetime of masq table entries is thirty minutes.
    If your rekey interval is greater than thirty minutes,
    then you should increase the lifetime to a value slightly
    greater than the rekey interval.

* IP: Enable parallel sessions (possible security risk - see help)
  CONFIG_IP_MASQUERADE_IPSEC_PAROK
  - See the IPsec masquerade technical notes and
    special security considerations section of the HOWTO for
    security considerations to be aware of when
    masquerading IPsec traffic. If you are only masquerading
    one IPsec client this setting has no effect.
```

NOTE: These are just the settings you need for masquerading. Select whatever other options you need for your specific setup.

8. Recompile the kernel and install it for testing. Don't replace a known working kernel with your new kernel until you have proven it works.

To determine whether the running kernel includes VPN Masquerade support, run the following command:

```
grep masq /proc/ksyms
```

Look for the following entries:

- IPsec masquerade: `ip_fw_masq_esp` and `ip_fw_demasq_esp`
- PPTP masquerade: `ip_fw_masq_gre` and `ip_fw_demasq_gre`

If you don't see these entries, VPN Masquerade support is probably not available (did you remember to `modprobe ip_masq_pptp` or `modprobe ip_masq_ipsec` if you compiled them as modules?). If you get complaints about `/proc/ksyms` not being available or `/proc` not being available, make sure that you have enabled the `/proc` filesystem in your kernel configuration.

See the Kernel HOWTO for more details on configuring and recompiling your kernel.

3.5 ipfwadm setup for a Private-IP VPN Client or Server

The firewall must now be configured to masquerade the outbound VPN traffic. You may wish to visit http://www.wolfenet.com/~jhardin/ipfwadm.html to take a look at a GUI wrapper around the ipfwadm command that automates a lot of security-related packet filtering setup.

The minimum firewall rules are:

```
# Set the default forwarding policy to DENY:
ipfwadm -F -p deny
# Allow local-network traffic
```

TOC

```
ipfwadm -I -a accept    -S 10.0.0.0/8 -D 0.0.0.0/0  -W eth0
ipfwadm -O -a accept    -S 0.0.0.0/0  -D 10.0.0.0/8 -W eth0
# Masquerade traffic for internet addresses and allow internet traffic
ipfwadm -F -a accept -m -S 10.0.0.0/8 -D 0.0.0.0/0  -W ppp0
ipfwadm -O -a accept    -S 0.0.0.0/0  -D 0.0.0.0/0  -W ppp0
ipfwadm -I -a accept    -S 0.0.0.0/0  -D 0.0.0.0/0  -W ppp0
```

or, if you have a permanent connection,

```
ipfwadm -F -a accept -m -S 10.0.0.0/8 -D 0.0.0.0/0  -W eth1
ipfwadm -O -a accept    -S 0.0.0.0/0  -D 0.0.0.0/0  -W eth1
ipfwadm -I -a accept    -S 0.0.0.0/0  -D 0.0.0.0/0  -W eth1
```

This is a completely open setup, though. It will masquerade *any* traffic from *any* host on the local network destined for *any* host on the internet, and provides *no* security at all.

A tight firewall setup would only allow traffic between the client and the server, and would block everything else:

```
# Set the default policy to DENY:
ipfwadm -I -p deny
ipfwadm -O -p deny
ipfwadm -F -p deny
# Allow local-network traffic
ipfwadm -I -a accept -S 10.0.0.0/8 -D 0.0.0.0/0  -W eth0
ipfwadm -O -a accept -S 0.0.0.0/0  -D 10.0.0.0/8 -W eth0
# Masquerade only VPN traffic between the VPN client and the VPN server
ipfwadm -F -a accept -m -P udp -S 10.0.0.2/32 500 -D 199.0.0.1/32 500  -W ppp0
ipfwadm -F -a accept -m -P tcp -S 10.0.0.2/32     -D 199.0.0.1/32 1723 -W ppp0
ipfwadm -F -a deny      -P tcp -S 10.0.0.2/32     -D 199.0.0.1/32      -W ppp0
ipfwadm -F -a deny      -P udp -S 10.0.0.2/32     -D 199.0.0.1/32      -W ppp0
ipfwadm -F -a accept -m -P all -S 10.0.0.2/32     -D 199.0.0.1/32      -W ppp0
ipfwadm -O -a accept    -P udp -S 200.200.200.0/24 500 -D 199.0.0.1/32 500  -W ppp0
ipfwadm -O -a accept    -P tcp -S 200.200.200.0/24     -D 199.0.0.1/32 1723 -W ppp0
ipfwadm -O -a deny      -P tcp -S 200.200.200.0/24     -D 199.0.0.1/32      -W ppp0
ipfwadm -O -a deny      -P udp -S 200.200.200.0/24     -D 199.0.0.1/32      -W ppp0
ipfwadm -O -a accept    -P all -S 200.200.200.0/24     -D 199.0.0.1/32      -W ppp0
ipfwadm -I -a accept    -P udp -S 199.0.0.1/32 500     -D 200.200.200.0/24 500 -W
ppp0
ipfwadm -I -a accept    -P tcp -S 199.0.0.1/32 1723    -D 200.200.200.0/24     -W
ppp0
ipfwadm -I -a deny      -P tcp -S 199.0.0.1/32         -D 200.200.200.0/24     -W
ppp0
ipfwadm -I -a deny      -P udp -S 199.0.0.1/32         -D 200.200.200.0/24     -W
ppp0
ipfwadm -I -a accept    -P all -S 199.0.0.1/32         -D 200.200.200.0/24     -W
ppp0
```

or, if you have a permanent connection,

```
ipfwadm -F -a accept -m -P udp -S 10.0.0.2/32 500 -D 199.0.0.1/32 500  -W eth1
ipfwadm -F -a accept -m -P tcp -S 10.0.0.2/32     -D 199.0.0.1/32 1723 -W eth1
ipfwadm -F -a deny      -P tcp -S 10.0.0.2/32     -D 199.0.0.1/32      -W eth1
ipfwadm -F -a deny      -P udp -S 10.0.0.2/32     -D 199.0.0.1/32      -W eth1
ipfwadm -F -a accept -m -P all -S 10.0.0.2/32     -D 199.0.0.1/32      -W eth1
ipfwadm -O -a accept    -P udp -S 200.200.200.200/32 500 -D 199.0.0.1/32 500  -W eth1
ipfwadm -O -a accept    -P tcp -S 200.200.200.200/32     -D 199.0.0.1/32 1723 -W eth1
ipfwadm -O -a deny      -P tcp -S 200.200.200.200/32     -D 199.0.0.1/32      -W eth1
ipfwadm -O -a deny      -P udp -S 200.200.200.200/32     -D 199.0.0.1/32      -W eth1
ipfwadm -O -a accept    -P all -S 200.200.200.200/32     -D 199.0.0.1/32      -W eth1
```

```
ipfwadm -I -a accept    -P udp -S 199.0.0.1/32 500  -D 200.200.200.200/32 500 -W
ipfwadm -I -a accept    -P tcp -S 199.0.0.1/32 1723 -D 200.200.200.200/32     -W
ipfwadm -I -a deny      -P tcp -S 199.0.0.1/32      -D 200.200.200.200/32     -W eth1
ipfwadm -I -a deny      -P udp -S 199.0.0.1/32      -D 200.200.200.200/32     -W eth1
ipfwadm -I -a accept    -P all -S 199.0.0.1/32      -D 200.200.200.200/32     -W eth1
```

Note: these rules only allow VPN traffic and block *everything else*. You will have to add rules for any other traffic you wish to permit, such as DNS, HTTP, POP, IMAP, etc.

3.6 ipchains setup for a Private-IP VPN Client or Server

The minimum ipchains firewall rules are:

```
# Set the default forwarding policy to DENY:
ipchains -P forward DENY
# Allow local-network traffic
ipchains -A input   -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0  -i eth0
ipchains -A output  -j ACCEPT -s 0.0.0.0/0  -d 10.0.0.0/8 -i eth0
# Masquerade traffic for internet addresses and allow internet traffic
ipchains -A forward -j MASQ   -s 10.0.0.0/8 -d 0.0.0.0/0  -i ppp0
ipchains -A output  -j ACCEPT -s 0.0.0.0/0  -d 0.0.0.0/0  -i ppp0
ipchains -A input   -j ACCEPT -s 0.0.0.0/0  -d 0.0.0.0/0  -i ppp0
```

or, if you have a permanent connection,

```
ipchains -A forward -j MASQ   -s 10.0.0.0/8 -d 0.0.0.0/0  -i eth1
ipchains -A output  -j ACCEPT -s 0.0.0.0/0  -d 0.0.0.0/0  -i eth1
ipchains -A input   -j ACCEPT -s 0.0.0.0/0  -d 0.0.0.0/0  -i eth1
```

This is a completely open setup, though. It will masquerade *any* traffic from *any* host on the local network destined for *any* host on the internet, and provides *no* security at all.

A tight firewall setup would only allow traffic between the client and the server, and would block everything else:

```
# Set the default policy to DENY:
ipchains -P input   DENY
ipchains -P output  DENY
ipchains -P forward DENY
# Allow local-network traffic
ipchains -A input  -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0  -i eth0
ipchains -A output -j ACCEPT -s 0.0.0.0/0  -d 10.0.0.0/8 -i eth0
# Masquerade only VPN traffic between the VPN client and the VPN server
# IPsec
ipchains -A forward -j MASQ   -p udp -s 10.0.0.2/32 500      -d 199.0.0.1/32 500
-i ppp0
ipchains -A output  -j ACCEPT -p udp -s 200.200.200.0/24 500 -d 199.0.0.1/32 500
-i ppp0
ipchains -A input   -j ACCEPT -p udp -s 199.0.0.1/32 500     -d 200.200.200.0/24 500
-i ppp0
ipchains -A forward -j MASQ   -p 50  -s 10.0.0.2/32          -d 199.0.0.1/32
-i ppp0
ipchains -A output  -j ACCEPT -p 50  -s 200.200.200.0/24     -d 199.0.0.1/32
-i ppp0
ipchains -A input   -j ACCEPT -p 50  -s 199.0.0.1/32         -d 200.200.200.0/24
-i ppp0
# PPTP
ipchains -A forward -j MASQ   -p tcp -s 10.0.0.2/32       -d 199.0.0.1/32 1723 -i
ppp0
ipchains -A output  -j ACCEPT -p tcp -s 200.200.200.0/24  -d 199.0.0.1/32 1723 -i
```

```
ppp0
ipchains -A input   -j ACCEPT -p tcp -s 199.0.0.1/32 1723 -d 200.200.200.0/24  -i
ppp0
ipchains -A forward -j MASQ   -p 47  -s 10.0.0.2/32       -d 199.0.0.1/32      -i
ppp0
ipchains -A output  -j ACCEPT -p 47  -s 200.200.200.0/24  -d 199.0.0.1/32      -i
ppp0
ipchains -A input   -j ACCEPT -p 47  -s 199.0.0.1/32      -d 200.200.200.0/24  -i
ppp0
```

or, if you have a permanent connection,

```
# IPsec
ipchains -A forward -j MASQ   -p udp -s 10.0.0.2/32 500        -d 199.0.0.1/32 500
-i eth1
ipchains -A output  -j ACCEPT -p udp -s 200.200.200.200/32 500 -d 199.0.0.1/32 500
-i eth1
ipchains -A input   -j ACCEPT -p udp -s 199.0.0.1/32 500       -d 200.200.200.200/32
500 -i eth1
ipchains -A forward -j MASQ   -p 50  -s 10.0.0.2/32            -d 199.0.0.1/32
-i eth1
ipchains -A output  -j ACCEPT -p 50  -s 200.200.200.200/32     -d 199.0.0.1/32
-i eth1
ipchains -A input   -j ACCEPT -p 50  -s 199.0.0.1/32           -d 200.200.200.200/32
-i eth1
# PPTP
ipchains -A forward -j MASQ   -p tcp -s 10.0.0.2/32        -d 199.0.0.1/32 1723  -i
eth1
ipchains -A output  -j ACCEPT -p tcp -s 200.200.200.200/32 -d 199.0.0.1/32 1723  -i
eth1
ipchains -A input   -j ACCEPT -p tcp -s 199.0.0.1/32 1723  -d 200.200.200.200/32 -i
eth1
ipchains -A forward -j MASQ   -p 47  -s 10.0.0.2/32        -d 199.0.0.1/32       -i
eth1
ipchains -A output  -j ACCEPT -p 47  -s 200.200.200.200/32 -d 199.0.0.1/32       -i
eth1
ipchains -A input   -j ACCEPT -p 47  -s 199.0.0.1/32       -d 200.200.200.200/32 -i
eth1
```

Note: these rules only allow VPN traffic. You will have to add rules for any other traffic you wish to permit, such as DNS, HTTP, POP, IMAP, etc.

Also note how there rules are much neater and easier to make sense of than the equivalent ipfwadm rules. This is because ipchains allows specification of all IP protocols, not just TCP, UDP, ICMP or ALL.

3.7 A note about dynamic IP addressing

If your firewall is assigned a dynamic IP address by your ISP (dialup accounts are this way, as are some cable internet services), then you should add the following to the startup script `/etc/rc.d/rc.local`:

```
echo 7 > /proc/sys/net/ipv4/ip_dynaddr
```

This enables dynamic IP address following, which means that should your connection drop and be reestablished, any active sessions will be updated to the new IP address rather than using the old IP address. This does not mean that the session will continue across the interruption, rather that it will be closed down quickly.

If you do not do this, then there may be a "dead period" after you redial and before old masq table entries expire where you're being masqueraded with the wrong IP address, which will prevent your establishing a connection.

This is particularly helpful if you are using a demand-dial daemon such as `diald` to manage your dialup connection.

TOC

See `/usr/src/linux/Documentation/networking/ip_dynaddr.txt` for more details.

3.8 Additional setup for a Private-IP VPN Server

If you are setting up VPN masquerade for a Private-IP VPN server (that is, you wish to provide for *inbound* connections as well as *outbound* connections), you also need to install two packet-forwarding utilities. One (`ipportfw`) forwards inbound TCP or UDP traffic addressed to a specific port on the firewall system to a system on the local network behind the firewall. This is used to redirect the initial inbound 1723/tcp PPTP control channel or 500/udp ISAKMP traffic to the VPN server. The other (`ipfwd`) is a more generic forwarding utility that allows you to do this for any IP protocol. It is used to forward the initial inbound 47/ip (GRE) or 50/ip (ESP) data channel traffic to the VPN server.

Outbound responses to the inbound 1723/tcp or 500/udp traffic are masqueraded using the normal IP-Masquerade facilities in the Linux kernel. The outbound 47/ip or 50/ip traffic is masqueraded using the VPN-Masquerade kernel patch you installed earlier.

Once these utilities are installed, you must configure them to forward the traffic to the VPN server.

- Configuring `ipportfw`

  The following commands will set up `ipportfw` to forward the initial inbound 500/udp traffic to the IPsec server:

  ```
  # Static-IP ipportfw setup for IPsec
  # Clear the ipportfw forwarding table
  /sbin/ipportfw -C
  # Forward traffic addressed to the firewall's 500/udp port
  # to the IPsec server's 500/udp port
  /sbin/ipportfw -A -u 200.200.200.200/500 -R 10.0.0.2/500
  ```

  The following commands will set up `ipportfw` to forward the initial inbound 1723/tcp traffic to the PPTP server:

  ```
  # Static-IP ipportfw setup for PPTP
  # Clear the ipportfw forwarding table
  /sbin/ipportfw -C
  # Forward traffic addressed to the firewall's 1723/tcp port
  # to the PPTP server's 1723/tcp port
  /sbin/ipportfw -A -t 200.200.200.200/1723 -R 10.0.0.2/1723
  ```

  Note that the ipportfw command line requires the internet IP address of the firewall, and you cannot specify the interface (e.g. `ppp0`) as you can with ipfwadm. This means that for a dynamic-IP connection (such as a typical dialup PPP connection) you have to run these commands every time you connect to the internet and are assigned a new IP address. You can do this quite easily - simply add the following to your `/etc/ppp/ip-up` or `/etc/ppp/ip-up.local` script:

  ```
  # Dynamic-IP ipportfw setup for IPsec
  # Clear the ipportfw forwarding table
  /sbin/ipportfw -C
  # Forward traffic addressed to the firewall's 500/udp port
  # to the IPsec server's 500/udp port
  /sbin/ipportfw -A -u ${4}/500 -R 10.0.0.2/500
  ```

  or:

  ```
  # Dynamic-IP ipportfw setup for PPTP
  # Clear the ipportfw forwarding table
  /sbin/ipportfw -C
  # Forward traffic addressed to the firewall's 1723/tcp port
  # to the PPTP server's 1723/tcp port
  /sbin/ipportfw -A -t ${4}/1723 -R 10.0.0.2/1723
  ```

  See http://www.wolfenet.com/~jhardin/ipfwadm/invocation.html for more information on firewalling with a dynamic IP.

- Configuring `ipfwd`

TOC

The following command will set up `ipfwd` to forward the initial inbound 50/ip traffic to the IPsec server:

```
/sbin/ipfwd --masq 10.0.0.2 50 &
```

The following command will set up `ipfwd` to forward the initial inbound 47/ip traffic to the PPTP server:

```
/sbin/ipfwd --masq 10.0.0.2 47 &
```

It should only be run once, from your `/etc/rc.d/rc.local` script.

The techniques described here can be generalized to allow masquerading of most any type of server - HTTP, FTP, SMTP, and so forth. Servers that are purely TCP- or UDP-based will not require `ipfwd`.

If you are masquerading a PPTP server you also need to make sure that you have not enabled PPTP Call ID masquerade in the kernel. Enabling PPTP Call ID masquerade builds in some assumptions that you're masquerading only PPTP clients, so enabling it will prevent proper masquerade of the PPTP server traffic. This also means that with this version of the patch you cannot simultaneously masquerade a PPTP server and PPTP clients. Contact me if this limitation causes you problems.

3.9 ipfwadm setup for a Registered-IP VPN Server

Setting up a registered-IP VPN server behind a Linux firewall is a simple matter of making sure the appropriate routing and packet-filter commands are in place. Masquerading is not required.

Unfortunately the 2.0.x-series kernels will not let us specify IP protocol 47 or 50 directly, so this firewall is less secure than it could be. If this is a problem for you, then install the IP Firewall Chains kernel patch or move to the 2.1.x or 2.2.x series kernel, where you can filter by IP protocol.

The firewall rules will look something like this:

```
# This section should follow your other firewall rules.

# Specify the acceptable clients explicitly for tighter security.
# Allow the IPsec ISAKMP traffic in and out.
ipfwadm -I -a accept -W eth1 -V 200.200.200.200 -P udp -S 199.0.0.2/32 500 -D
222.0.0.2/32 500
ipfwadm -O -a accept -W eth1 -V 200.200.200.200 -P udp -D 199.0.0.2/32 500 -S
222.0.0.2/32 500
ipfwadm -I -a accept -W eth1 -V 200.200.200.200 -P udp -S 199.0.0.3/32 500 -D
222.0.0.2/32 500
ipfwadm -O -a accept -W eth1 -V 200.200.200.200 -P udp -D 199.0.0.3/32 500 -S
222.0.0.2/32 500
# Allow the PPTP control channel in and out.
ipfwadm -I -a accept -W eth1 -V 200.200.200.200 -P tcp -S 199.0.0.2/32 -D
222.0.0.2/32 1723
ipfwadm -O -a accept -W eth1 -V 200.200.200.200 -P tcp -D 199.0.0.2/32 -S
222.0.0.2/32 1723
ipfwadm -I -a accept -W eth1 -V 200.200.200.200 -P tcp -S 199.0.0.3/32 -D
222.0.0.2/32 1723
ipfwadm -O -a accept -W eth1 -V 200.200.200.200 -P tcp -D 199.0.0.3/32 -S
222.0.0.2/32 1723

# Block all other TCP and UDP traffic from the internet.
# This is essentially a "default deny TCP/UDP" that
# only applies to the internet interface.
ipfwadm -I -a deny -W eth1 -V 200.200.200.200 -P tcp
ipfwadm -I -a deny -W eth1 -V 200.200.200.200 -P udp

# Specify the acceptable clients explicitly for tighter security.
# Note that this is too open since we're forced to
```

```
# specify "-P all" rather than "-P 47" or "-P 50"...
# Allow the PPTP data channel and IPsec ESP traffic in and out.
ipfwadm -I -a accept -W eth1 -V 200.200.200.200 -P all -S 199.0.0.2/32 -D
222.0.0.2/32
ipfwadm -O -a accept -W eth1 -V 200.200.200.200 -P all -D 199.0.0.2/32 -S
222.0.0.2/32
ipfwadm -I -a accept -W eth1 -V 200.200.200.200 -P all -S 199.0.0.3/32 -D
222.0.0.2/32
ipfwadm -O -a accept -W eth1 -V 200.200.200.200 -P all -D 199.0.0.3/32 -S
222.0.0.2/32

# Block all other traffic from the internet.
# This is essentially a "default deny" that
# only applies to the internet interface.
ipfwadm -I -a deny -W eth1 -V 200.200.200.200
```

If you are installing firewall rules on forwarding and/or output, you will have do do something similar. The above example only covers VPN traffic; you will have to merge it into your existing firewall setup to allow any other traffic you need.

3.10 ipfwadm setup for a Registered-IP VPN Client

Setting up a registered-IP VPN client behind a Linux firewall is similar to setting up a registered-IP VPN server.

The firewall rules will look something like this:

```
# Allow the IPsec ISAKMP traffic out and in.
ipfwadm -O -a accept -W eth1 -V 200.200.200.200 -P udp -S 222.0.0.2/32 500 -D
199.0.0.1/32 500
ipfwadm -I -a accept -W eth1 -V 200.200.200.200 -P udp -D 222.0.0.2/32 500 -S
199.0.0.1/32 500
# Allow the PPTP control channel out and in.
ipfwadm -O -a accept -W eth1 -V 200.200.200.200 -P tcp -S 222.0.0.2/32 -D
199.0.0.1/32 1723
ipfwadm -I -a accept -W eth1 -V 200.200.200.200 -P tcp -D 222.0.0.2/32 -S
199.0.0.1/32 1723

# Block all other TCP and UDP traffic from the internet.
# This is essentially a "default deny TCP/UDP" that
# only applies to the internet interface.
ipfwadm -I -a deny -W eth1 -V 200.200.200.200 -P tcp
ipfwadm -I -a deny -W eth1 -V 200.200.200.200 -P udp

# Note that this is too open since we're forced to
# specify "-P all" rather than "-P 47" or "-P 50"...
# Allow the PPTP data channel and IPsec ESP traffic out and in
ipfwadm -O -a accept -W eth1 -V 200.200.200.200 -P all -S 222.0.0.2/32 -D
199.0.0.1/32
ipfwadm -I -a accept -W eth1 -V 200.200.200.200 -P all -D 222.0.0.2/32 -S
199.0.0.1/32

# Block all other traffic from the internet.
# This is essentially a "default deny" that
# only applies to the internet interface.
ipfwadm -I -a deny -W eth1 -V 200.200.200.200
```

4. Configuring the VPN client

4.1 Configuring a MS W'95 or W'98 client

1. Set up your routing so that the Linux firewall is your default gateway:
    1. Open `Control Panel/Network` or right-click "Network Neighborhood" and click on `Properties`.
    2. Click on the `Configuration` tab.
    3. In the list of installed network components, double-click on the "TCP/IP -> whatever-NIC-you-have" line.
    4. Click on the `Gateway` tab.
    5. Enter the local-network IP address of your Linux firewall. Delete any other gateways.
    6. Click on the "OK" button.
2. Test masquerading. For example, run "`telnet` *`my.isp.mail.server`* `smtp`" and you should see the mail server's welcome banner.
3. Install and configure the VPN software. For IPsec software follow the manufacturer's instructions. For MS PPTP:
    1. Open `Control Panel/Network` or right-click "Network Neighborhood" and click on `Properties`.
    2. Click on the `Configuration` tab.
    3. Click on the "Add" button, then double-click on the "Adapter" line.
    4. Select "Microsoft" as the manufacturer and add the "Virtual Private Networking Adapter" adapter.
    5. Reboot when prompted to.
    6. If you need to use strong (128-bit) encryption, download the strong encryption DUN update from the MS secure site at http://mssecure.www.conxion.com/cgi-bin/ntitar.pl and install it, then reboot again when prompted to.
    7. Create a new dial-up phonebook entry for your PPTP server.
    8. Select the VPN adapter as the device to use, and enter the PPTP server's internet IP address as the telephone number.
    9. Select the `Server Types` tab, and check the compression and encryption checkboxes.
    10. Click on the "TCP/IP Settings" button.
    11. Set the dynamic/static IP address information for your client as instructed to by your PPTP server's administrator.
    12. If you wish to have access to your local network while the PPTP connection is up, uncheck the "Use default gateway on remote network" checkbox.

13. Reboot a few more times, just from habit... :)

TOC

4.2 Configuring a MS NT client

Note: this section may be incomplete as it's been a while since I've installed PPTP on an NT system.

1. Set up your routing so that the Linux firewall is your default gateway:
    1. Open `Control Panel/Network` or right-click "Network Neighborhood" and click on `Properties`.
    2. Click on the `Protocols` tab and double-click on the "TCP/IP" line.
    3. Enter the local-network IP address of your Linux firewall in the "Default Gateway" box.
    4. Click on the "OK" button.
2. Test masquerading. For example, run "`telnet` *`my.isp.mail.server`* `smtp`" and you should see the mail server's welcome banner.
3. Install and configure the VPN software. For IPsec software follow the manufacturer's instructions. For MS PPTP:
    1. Open `Control Panel/Network` or right-click "Network Neighborhood" and click on `Properties`.
    2. Click on the `Protocols` tab.
    3. Click on the "Add" button, then double-click on the "Point-to-Point Tunneling Protocol" line.
    4. When it asks for the number of Virtual Private Networks, enter the number of PPTP servers you could possibly be communicating with.
    5. Reboot when prompted to.
    6. If you need to use strong (128-bit) encryption, download the strong encryption PPTP update from the MS secure site at http://mssecure.www.conxion.com/cgi-bin/ntitar.pl and install it, then reboot again when prompted to.
    7. Create a new dial-up phonebook entry for your PPTP server.
    8. Select the VPN adapter as the device to use, and enter the PPTP server's internet IP address as the telephone number.
    9. Select the `Server Types` tab, and check the compression and encryption checkboxes.
    10. Click on the "TCP/IP Settings" button.
    11. Set the dynamic/static IP address information for your client as instructed to by your PPTP server's administrator.
    12. If you wish to have access to your local network while the PPTP connection is up, see MS Knowledge Base article Q143168 for a registry fix. (*Sigh*.)
    13. Make sure you reapply the most recent Service Pack, to ensure that your RAS and PPTP libraries are up-to-date for security and performance enhancements.

---

5. Troubleshooting

5.1 Testing

To test VPN Masquerade:

1. Bring up your ISP connection from your Linux box and verify that it still works properly.
2. Verify that regular masquerading still works properly by, for example, trying to browse a Web site or access an FTP server from a masqueraded box on your local network.
3. PPTP: Verify that you have masquerading of the PPTP control channel properly configured: try to telnet from the PPTP client system to port 1723 on your PPTP server. Don't expect to see anything, but if you get a timeout or an error saying the connection failed, take a look at the masquerade rules on your Linux box to ensure that you are indeed masquerading traffic from your PPTP client to TCP port 1723 on your PPTP server.
4. PPTP: Attempt to establish a PPTP connection. I recommend you also run `RASMON` if it is available, as this will give you a minimal amount of information about the status of the connection. If you establish a PPTP connection on the first try, congratulations! You're done!
5. IPsec: Attempt to establish an IPsec connection.

5.2 Possible problems

There are several things that may prevent a VPN session from being established. We'll work through them going from the client to the server and back again. We will assume you're using a Windows-based client for the examples, as that's the most common case.

1. Connect information: the "telephone number" in the VPN dialup configuration must be the Internet IP address of the VPN server, or the IP address of the firewall if the server is being masqueraded.
2. PPTP and strong encryption: unless both client and server have the 128-bit `NDISWAN.SYS` or W'95/'98 PPTP software, you will not be able to establish a strongly-encrypted session. Unfortunately in my experience this problem does not generate any obvious error messages, it just keeps trying and trying and trying... The strong encryption update can be obtained from the Microsoft secure site URL given int he "Configuring a MS Client" section.

   This may also affect IPsec clients, if they use the MS-supplied encryption libraries rather than using their own libraries.
3. Routing: verify that the default route on your VPN client is pointing at the Linux masquerade box. Run the `route print` command and look for an `0.0.0.0` entry.

   If other masqueraded services (such as HTTP, FTP, IRC, etc.) work from your VPN client system then this probably is not the problem.
4. Masquerading: there are two parts to the VPN session.

For IPsec, the authentication and key exchange service (ISAKMP), which is a normal UDP se to port 500 on the remote IPsec host, must be configured for masquerading as you would any other UDP service (such as DNS).

For PPTP, the control channel, which is a normal TCP session to port 1723 on the PPTP server, must be configured for masquerading as you would any other TCP service (such as HTTP).

The encrypted data channel in IPsec is carried over ESP, IP protocol 50. The encrypted data channel in PPTP is carried over GRE, IP protocol 47. (Note that these are *not* TCP or UDP port numbers!) Since the 2.0 Linux kernel only lets you specify `TCP`, `UDP`, `ICMP` and `ALL` IP protocols when creating masquerade rules, you must also masquerade `ALL` protocol traffic if you are masquerading only specific services. If you are masquerading everything, you don't need to worry about this.

In order to isolate the firewall rules from the kernel masquerade code, try establishing a VPN connection with your firewall completely open, then if it works, tighten the firewall rules.

2.0.x ipfwadm completely open firewall:

```
ipfwadm -I -p accept
ipfwadm -O -p accept
ipfwadm -F -a accept -m
```

2.2.x ipchains completely open firewall:

```
ipchains -P input   ACCEPT
ipchains -P output  ACCEPT
ipchains -P forward MASQ
```

Do *not* leave your firewall completely open for any longer than it takes to prove that a masqueraded VPN connection can be established!

5. Intermediary hops and the Internet: All routers between your Linux firewall and the remote IPsec host must forward packets carrying IP protocol 50. All routers between your Linux firewall and the PPTP server must forward packets carrying IP protocol 47. If you had IPsec or PPTP working when your VPN client system directly dialled your ISP then this probably is not the problem.

   To isolate whether an intermediary hop is blocking GRE traffic, use a patched `traceroute` to trace the progress of GRE packets. See the resources section for information on the traceroute patch. A similar patch for ESP is in the works.
6. The remote firewall: the firewall at the server end must allow a system with the IP address assigned to your Linux box by your ISP to connect to port 500/udp on the IPsec host or port 1723/tcp on the PPTP server. If you had the VPN working when your VPN client system directly dialled your ISP then this probably is not the problem.
7. The server firewall and ESP: the IPsec encrypted data is carried over IP protocol 50. If the firewall the remote IPsec host is behind does not forward ESP traffic in both directions, IPsec will not work. Again, if you had IPsec working when your IPsec client system directly dialled your ISP then this probably is not the problem.

8. The server firewall and GRE: the PPTP data channel is carried as a GRE-encapsulated (IP pro 47) PPP session. If the firewall your PPTP server is behind does not forward GRE traffic in bou directions, PPTP will not work. Again, if you had PPTP working when your PPTP client system directly dialed your ISP then this probably isn't the problem.
9. The patch: If your IPsec client successfully authenticates you but cannot establish a network connection, the patch may not be masquerading ESP traffic properly. If your PPTP client establishes the control channel (RASMON beeps and the little telephone lights up) and sends GRE traffic (the upper light in RASMON blinks) but gets no GRE traffic back (the lower light in RASMON does not blink in response) the patch may not be masquerading GRE traffic properly.

   Look in `/var/log/messages` for log entries showing that VPN traffic was seen. Turning on VPN debugging may help you to determine whether or not the patch is at fault. Also run a sniffer on your internet connection and look for outbound VPN traffic *(see below)*.
10. Multiple clients: the older PPTP patch does NOT support masquerading of multiple PPTP clients attempting to access the *same* PPTP server. If you're trying to do this, you should take a look at your network design and consider whether you should set up a PPTP router for your local clients. The 2.0 patch incorporates Call-ID masquerading, which allows multiple simultaneous sessions. *Note:* do not enable PPTP Call-ID masquerade if you are masquerading a PPTP Server. At the current time this will prevent the server's outbound traffic from being masqueraded.

5.3 Troubleshooting

Most problems can be localized by running a packet sniffer (e.g. `tcpdump` with the `-v` option) on your VPN firewall. If everything is working properly, you'll see the following traffic:

- Client local network:

  IPsec: UDP (destination UDP port 500) and ESP (IP protocol 50) traffic from your IPsec client local network IP to the remote IPsec host's Internet IP. If you don't see this, your IPsec client is misconfigured.

  PPTP: TCP (destination TCP port 1723) and GRE (IP protocol 47) traffic from your PPTP client local network IP to the PPTP server's Internet IP. If you don't see this, your PPTP client is misconfigured.
- ISP side of client firewall: UDP and ESP or TCP and GRE traffic from the client firewall Internet IP (remember - we're masquerading) to the VPN server's Internet IP. If you don't see this, your masquerade is misconfigured or the patch isn't working.
- ISP side of server firewall: UDP and ESP or TCP and GRE traffic from the client Internet IP to the VPN server's Internet IP. If you don't see this, the Internet is down `:)` or some intermediary is blocking ESP or GRE traffic.
- Boundary network (DMZ) side of server firewall: UDP and ESP or TCP and GRE traffic from the client internet IP to the server IP. If you don't see this, check your firewall rules for forwarding UDP port 500 and IP protocol 50 or TCP port 1723 and IP protocol 47, and the configuration of `ipportfw` and `ipfwd` if you're masquerading the server.
- Boundary network side of server firewall: UDP (source port 500) and ESP or TCP (source port

1723) and GRE traffic from the VPN server IP to the client internet IP. If you don't see this, cl the VPN server configuration, including the packet filtering rules on the VPN server.

- ISP side of server firewall: UDP and ESP or TCP and GRE traffic from the VPN server IP (or firewall IP if the server is masqueraded) to the client internet IP. If you don't see this, check your firewall rules for forwarding UDP port 500 and IP protocol 50 or TCP port 1723 and IP protocol 47.
- ISP side of client firewall: UDP and ESP or TCP and GRE traffic from the VPN server IP to the client firewall internet IP. If you don't see this, the Internet is acting up again.
- Client local network: UDP and ESP or TCP and GRE traffic from the VPN server internet IP to the VPN client local network IP. If you see the UDP traffic but not the ESP traffic, or the TCP traffic but not the GRE traffic, the patch isn't working or wasn't properly installed.

You may find it helpful to turn on VPN debugging and recompile your kernel. Add the following to `/etc/syslog.conf`

```
# debugging
*.=debug            /var/log/debug
```

and watch `/var/log/messages` and `/var/log/debug` for log messages about the VPN traffic. Note that logging - especially verbose logging - will cause a great deal of disk activity and will cause the log files to grow very large very quickly. Don't turn on debugging unless you need to, and turn it off when you're done.

5.4 MS PPTP Clients and domain-name issues

Thanks to Charles Curley <ccurley@trib.com> for the following:

> If you use PPTP (Point to Point Tunneling Protocol) to access a Microsoft Networking (SMB) environment and have your own Microsoft Networking environment in your local private network (Samba or Windows), give your local workgroup a name that does not show up in the remote environment. The reason is that while your PPTP client is logged into the remote environment, it will see the remote environment's domain name servers, and will only see the remote computers in that workgroup.
>
> You should avoid the lazy option. Microsoft ships Windows set up for a default workgroup name of WORKGROUP. Some people will be lazy and accept that as their workgroup when they set up their computers. So there is a good chance that the remote environment will have a workgroup called WORKGROUP, administrators willing or not.

I think that this will apply regardless of the VPN in use, as name services aren't dependent on the transport. If your client(s) can see the WINS servers on the remote network then you may experience this, PPTP or no PPTP.

TOC

5.5 MS PPTP Clients and Novell IPX

If you're having trouble with IPX traffic over your PPTP link, please see sections 3.5 and 5.2 in this MS Knowledge Base article: http://support.microsoft.com/support/kb/articles/q154/0/91.asp

Thanks to David Griswold <dgriswol@ix.netcom.com>

5.6 MS network password issues

When you are using a VPN to access a MS network you should remember that you will have to provide two different authentication tokens - one to connect to the VPN server (the VPN password) and the other to access resources on the remote network once the connection is established (the network password).

The VPN password - the username and password you enter into your VPN client when initiating the call to the VPN server - is only used by the VPN server to grant you permission to connect to the network via the VPN. It isn't used for anything else once you're connected.

The VPN password is *not* used to prove your identity to other computers on the remote network. You must provide another username/password pair - your network password - for that.

There are two ways to supply a network password. Your network password may be the same username/password pair you supplied when logging onto the local network when you started your computer up. If it is different, you can configure your VPN client to ask you for your network password for the remote network once the VPN connection is established.

If you are successfully connecting to the VPN server but you cannot access any of the resources provided by the remote network, then you aren't providing a valid network username/password pair for the remote network. Verify that the username and password for your local network will also work on the remote network, or set your VPN client to prompt you for a username and password for use on the remote network and "log on" to the remote network once the VPN connection is established.

5.7 If your IPsec session always dies after a certain amount of time

If you're having trouble with your IPsec tunnel regularly dying, particularly if checking the system logs on the firewall shows that ISAKMP packets with "zero cookie" values are being seen, here's what's happening:

Earlier versions of the IPsec Masq patch did not change the timeout for masq table entries for ISAKMP UDP packets. The masq table entries for the ISAKMP UDP traffic would time out fairly quickly (relative to the data channel) and be removed; if the remote IPsec host then decided to initiate rekeying before the local IPsec host did, the inbound ISAKMP traffic for the rekey couldn't be routed to the masqueraded host. The rekey traffic would be discarded, the remote IPsec host would think the link had failed, and the connection would eventually be terminated.

TOC

The 2.0.36 patch has been modified to increase the timeout on ISAKMP UDP masq table entries. Ge current version of the patch, available via the sites given in the Resources section, and repatch and recompile your kernel.

Also verify that your `IPsec Masq Table Lifetime` parameter is configured to be the same as or slightly longer than your rekey interval.

---

6. IPsec masquerade technical notes and special security considerations

6.1 Limitations and weaknesses of IPsec masquerade

Traffic that uses the AH protocol *cannot* be masqueraded. The AH protocol incorporates a cryptographic checksum across the IP addresses that the masquerade gateway cannot correctly regenerate. Thus, all masqueraded AH traffic will be discarded as having invalid checksums.

IPsec traffic using transport-mode ESP also cannot be reliably masqueraded. Transport mode ESP essentially encrypts everything after the IP header. Since, for example, the TCP and UDP checksums include the IP source and destination addresses, and the TCP/UDP checksum is within the encrypted payload and thus cannot be recalculated after the masquerade gateway alters the IP addresses, the TCP/UDP header will fail the checksum test at the remote gateway and the packet will be discarded. Protocols that do not include information about the source or destination IP addresses may successfully use masqueraded transport mode.

Apart from these limitations, IPsec masquerade is secure and reliable when only one IPsec host is being masqueraded at a time, or when each masqueraded host is communicating with a different remote host. When more than one masqueraded host is communicating with the same remote host, a few weaknesses show up:

- Transport-mode communications are subject to collisions.

  If two or more masqueraded hosts are using transport mode to communicate with the same remote host, and the security policy of the remote host permits multiple transport-mode sessions with the same peer, it is possible for sessions to experience collisions. This happens because the IP address of the *masquerading gateway* will be used to identify the sessions, and any other identifying information cannot be masqueraded because it is within the encrypted portion of the packet.

  If the remote host's security policy does not permit multiple transport-mode sessions with the same peer, the situation is even worse: the more-recently-negotiated transport mode session will likely completely take over *all* of the traffic from the older session, causing the older session to "go dead". While the established sessions from the older transport-mode IPsec session may be quickly reset if the remote host isn't expecting to receive the traffic, at least one packet of information will be sent to the wrong host. This information will probably be discarded by the recipient, but it will still be sent.

  Thus, a transport-mode collision may result in leaking of information between the two sessions or termination of one or both sessions. Using IPsec in transport mode via a masquerading gateway is *not recommended* if there is the possibility that other transport mode IPsec sessions will be

TOC

attempted via the same masquerading gateway to the same remote IPsec host.

IPsec using tunnel mode with extruded network addressing (where the masqueraded IPsec host is assigned an IP address from the remote host's network) is *not* subject to these problems, as the IP addresses assigned from the remote network will be used to identify the sessions instead of using the IP address of the masquerading host.

- ISAKMP communications are subject to cookie collisions.

  If two or more masqueraded hosts establishing a session to the same remote host happen to select the same initiator cookie when initiating ISAKMP traffic, the masquerading gateway will route all of the ISAKMP traffic to the second host. There is a 1 in 2^64 (i.e. very small) chance of this collision happening for each host, at the time of establishing the initial ISAKMP connection.

  Correcting this requires including the responder cookie in the key used to route inbound ISAKMP traffic. This modification will be incorporated into IPsec masquerade for the 2.2.x kernel, and the short window between the time the masqueraded host initiates the ISAKMP exchange and the remote host responds is covered by discarding any new ISAKMP traffic that would collide with the current outstanding traffic. This modification will be backported to the 2.0.x code soon.
- There may be a collision between SPI values on inbound traffic.

  Two or more masqueraded IPsec hosts communicating with the same remote IPsec host may negotiate to use the same SPI value for inbound traffic. If this happens the masquerading gateway will route all of the inbound traffic to the first host to receive any inbound traffic using that SPI. The possibility of this happening is about 1 in 2^32 for each outstanding ESP session, and may occur on any rekey.

  Since the SPI values refer to different SAs having different encryption keys the first host will not be able to decrypt the data intended for the other hosts, so no data leakage will occur. There is no way for the masquerading gateway to detect or prevent this collision. The only way to prevent this collision is for the remote IPsec host to check the SPI value proposed by the masqueraded host to see if that SPI value is already in use by another SA from the same IP address. It is not likely that this will be done, since it imposes more overhead on an already expensive operation (the rekey) to benefit a small percentage of users in case of a relatively rare event.
- Inbound and outbound SPI values may be misassociated.

  This is discussed in detail in the next section.

The 2.2.x code by default prevents the establishment of multiple connections to the same remote host. If the weaknesses exposed by multiple connections to the same remote host are acceptable, you can enable "parallel sessions".

6.2 Proper routing of inbound encrypted traffic

The portion of the ISAKMP key exchange where the ESP SPI values are communicated is encrypted, so the ESP SPI values must be determined by inspection of the actual ESP traffic. Also, the outbound ESP traffic does not contain any indication of what the inbound SPI will be. This means there is no perfectly reliable way to associate inbound ESP traffic with outbound ESP traffic.

IPsec masq attempts to associate inbound and outbound ESP traffic by serializing initial ESP traffic by-remote-host basis. What this means is:

- If an outbound ESP packet with an SPI value that has not previously been seen (or whose masquerade table entry has expired) is received (which shall hereafter be called an "initial packet"), a masquerade table entry for that SourceAddr+SPI+DestAddr combination is created. It is marked as "outstanding", that is, no inbound ESP traffic has been received for it yet. This is done by setting the "inbound SPI" value in the masq table entry to zero, which is a value reserved for uses such as this. This will happen at the initiation of a new ESP connection and at regular intervals when an existing ESP connection rekeys.
- As long as the masq table entry is outstanding, no other initial ESP packets for the *same remote host* will be processed. The packets are immediately discarded, and a system log entry is made saying the traffic is temporarily blocked. This also applies to initial traffic from the same masqueraded host going to the same remote host if the SPI values differ. Traffic to other remote hosts, and traffic where both SPI values are known ("established" traffic) is not affected by this.
- This could easily lead to a Denial of Service of the remote host, so this outstanding ESP masq table entry is given a short lifetime, and only a limited number of retries of the same traffic are allowed. This permits round-robin access to the remote host if several masqueraded hosts are attempting to initialize simultaneously and responses aren't coming back very quickly, for example due to network congestion or a slow remote host. The retry limitation begins once there is a collision, so the masqueraded IPsec host can wait as long as necessary for a reply until there's a need for serialization.
- When an ESP packet from the outstanding remote host is received and the SPI value does not appear in any masq table entry, it is assumed that the packet is the response to the outstanding initial packet. The SPI value is stored in that masq table entry, thus associating the SPI values, and the inbound ESP traffic is routed to the masqueraded host. At this point another initial packet for the remote server may be processed.
- Any ESP traffic with a zero SPI value is discarded as invalid, per the RFC requirements.

There are several ways this can fail to associate traffic properly:

- Network delays or a slow remote host can cause the response to the first initial packet to be delayed long enough that the init masq table entry expires and a different masqueraded host is given a chance to initialize. This could cause the response to be associated with the wrong outbound SPI, which would cause inbound traffic to be routed to the wrong masqueraded host. If this happens the masqueraded host receiving the traffic in error will discard it because it has an unexpected SPI value, and everybody will eventually time out, rekey and try again. This can be addressed by editing `/usr/src/linux/net/ipv4/ip_masq.c` and increasing the INIT lifetime or the number of INIT retries permitted, at the cost of increasing the blocking (and DoS) window.
- Sessions idle or semi-idle (with infrequent inbound traffic and no outbound traffic) for a long period of time may be idle long enough for the masq table entry to expire. If the remote host sends traffic to an established yet expired session while an outstanding init to the same remote host is underway, the traffic may be misrouted for the same reason as described above. This can be addressed by making sure the `IPsec Masq Table Lifetime` kernel configuration parameter is slightly longer than the rekey interval, which is the longest time any given SPI pair

will be used. The problem here is that you may not know all of the rekey intervals if you're masquerading for many remote servers, or some may have their rekey intervals set to unreasonably high values, such as several hours.

- If there is a delay between a rekey and the transmission of outbound ESP traffic using the new SPI, and during this delay inbound ESP traffic using the new SPI is received, there will be no masq table entry describing how to route the inbound traffic. If another masqueraded host has a pending init with the same remote host, the traffic will be misassociated. Note that serialization of ESP initial traffic *does not* affect ISAKMP rekey traffic.

The best solution is to have some way to preload the masq table with the properly associated out-SPI/in-SPI pair or some other mapping of remote_host + inbound_SPI to masqueraded_host. This cannot be done by inspecting the ISAKMP key exchange, as it is encrypted. It may be possible to use RSIP (a.k.a. Host-NAT) to communicate with the masqueraded IPsec host and request notification of SPI information once it has been negotiated. This is being investigated. If something is done to implement this it will be done no sooner than the 2.3.x series, as RSIP is a fairly complex client/server NAT protocol.

When an inbound ESP packet with a new SPI is received the masquerading firewall attempts to guess which masqueraded host(s) the unassociated inbound traffic is intended for. If the inbound ESP traffic is not matched to an established session or a pending session initialization, then the packet is sent to the masqueraded host(s) who most recently rekeyed with that remote host. The "incorrect" masqueraded hosts will discard the traffic as being improperly encrypted, and the "correct" host will get its data. When the "correct" host responds, the normal ESP init serialization process occurs.

Virtual Services Howto

Brian Ackerman, brian@nycrc.net

v2.1, 15 August 1998

This document came about to satisfy the ever increasing need to know how to virtualize a service.

1. Introduction

- 1.1 Knowledge Required
- 1.2 Purpose
- 1.3 Feedback
- 1.4 Revision History
- 1.5 Copyright/Distribution

2. IP Aliasing

3. Virtuald

- 3.1 Introduction
- 3.2 Inetd
- 3.3 Config File
- 3.4 Source

4. Shell Scripts

- 4.1 Virtfs
- 4.2 Virtexec
- 4.3 Notes

TOC

5. DNS

6. Syslogd

- 6.1 Problem
- 6.2 Solution
- 6.3 Multiple Syslogd's

7. Virtual FTP

- 7.1 Inetd
- 7.2 Anonymous FTP
- 7.3 Virtual FTP Users

8. Virtual Web

- 8.1 Running With Virtuald
- 8.2 Running With Apache VirtualHost
- 8.3 File Descriptor Overflow
- 8.4 Sharing Servers With One IP
- 8.5 More Information

9. Virtual Mail/Pop

- 9.1 Problem
- 9.2 Solution
- 9.3 Sendmail Solution
- 9.4 Qmail Solution
- 9.5 Acknowledgement

10. Virtual Samba

- 10.1 Setup
- 10.2 Inetd
- 10.3 Smb.init

11. Virtual Other

12. Conclusion

13. FAQ

1. Introduction

1.1 Knowledge Required

Creating a virtual services machine is not all that difficult, however, more than fundamental knowledge is required. This document is not a primer to how to fully configure a Linux machine.

In order to understand this HOWTO document it is assumed that you are thoroughly familiar with the following:

- Compiling a Linux kernel and adding IP aliasing support IP alias mini-HOWTO
- Setting up and configuring of network devices NET-3 HOWTO
- Setting up of inetd NET-3 HOWTO
- Various network packages like Sendmail Apache Qmail SAMBA
- Setting up DNS DNS HOWTO
- Understanding basic system administration Linux Systems Administrators's Guide
- Understanding how to setup a Web Server WWW HOWTO

If you are uncertain of how to proceed with any of the above it is STRONGLY recommended that you use the html links provided to familiarize yourself with all packages. I will NOT reply to mail regarding any of the above. Please direct your questions to the appropriate author of the HOWTO.

1.2 Purpose

The purpose of virtual services is to allow a single machine to recognize multiple IP addresses without multiple network cards. IP aliasing is a kernel option that allows you to assign each network device more than one IP address. The kernel then multiplexes (swaps between them very fast) in the background and to the user it appears like you have more than one server.

This multiplexing allows multiple domains (www.domain1.com, www.domain2.com, etc.) to be hosted by the same machine for the same cost as hosting one domain. Unfortunately, most services (FTP, web, mail) were not designed to handle muliple domains. In order to make them work properly you must modify both configuration files and source code. This document describes how to make these modifications in the setting up of a virtual machine.

A deamon is also required in order to make virtual services function. The source for this daemon (virtuald) is provided later in this document.

TOC

1.3 Feedback

This document will expand as packages are updated and source or configuration modifications change. If there are any portions of this document that are unclear please feel free to email me with your suggestions or questions. So that I do not have to go searching through the entire HOWTO please make certain that all comments are as specific as possible and include the section where the uncertainty lies. It is important that all mail be addressed with VIRTSERVICES HOWTO in the subject line. Any other mail will be considered personal and all my friends know that I do not ever read my personal mail so it will probably get discarded with theirs.

Please note that my examples are just that, examples and should not be copied verbatim. You may have to insert your own values. If you are having trouble, send me mail. Include all the pertinent configuration files and the error messages you get when installing and I will look them over and reply with my suggestions.

1.4 Revision History

V1.0

Initial version

V1.1

Fixed error in Virtual Web Section

V1.2

Fixed the date

V2.0

Updated html links.

Web updates.

New Sendmail option.

New Qmail section.

Syslogd updates.

FTP updates.

Virtuald default option.

New SAMBA section.

FAQ updates.

V2.1

Changed all paths to /usr/local.

Added virtuald VERBOSELOG compile option.

Fixed setuid/setgid bug in virtmailfilter.

Fixed execl bug in virtmailfilter.

Fixed capitialization bug in virtmailfilter.

Fixed environment variable sanity check in virtmailfilter.

Removed mbox code from virtmailfilter/virtmaildelivery.

Added tcpserver.init pop section for Qmail.

Added alias domain name question to the FAQ.

Fixed virtmailfilter to send home directory to virtmaildelivery.

1.5 Copyright/Distribution

This document is Copyright (c) 1997 by The Computer Resource Center Inc.

A verbatim copy may be reproduced or distributed in any medium physical or electronic without permission of the author. Translations are similiarly permitted without express permission if it includes a notice on who translated it. Commercial redistribution is allowed and encouraged; however please notify Computer Resource Center of any such distributions.

Excerpts from the document may be used without prior consent provided that the derivative work contains the verbatim copy or a pointer to a verbatim copy.

Permission is granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies.

In short, we wish to promote dissemination of this information through as many channels as possible. However, I do wish to retain copyright on this HOWTO document, and would like to be notified of any plans to redistribute this HOWTO.

---

2. IP Aliasing

IP aliasing is a kernel option that needs to be set up in order to run a virtual hosting machine. There is already a mini-HOWTO on IP aliasing. Consult that for any questions on how to set it up.

3. Virtuald

3.1 Introduction

Every network connection is made up of two IP address/port pairs. The API (Applications Program Interface) for network programming is called the Sockets API. The socket acts like an open file and by reading/writing to it you can send data over a network connection. There is a function call `getsockname` that will return the IP address of the local socket. Virtuald uses `getsockname` to determine which IP on the local machine is being accessed. Virtuald reads a config file to retrieve the directory associated with that IP. It will `chroot` to that directory and hand the connection off to the service. `Chroot` resets / or the root directory to a new point so everything higher in the directory tree is cut off from the running program. Therefore, each IP address gets their own virtual filesystem. To the network program this is transparent and the program will behave like nothing happened. Virtuald in conjunction with a program like inetd can then be used to virtualize any service.

3.2 Inetd

Inetd is a network super server that listens at multiple ports and when it receives a connection (for example, an incoming pop request), inetd performs the network negotiation and hands the network connection off to the specified program. This prevents services from running idly when they are not needed.

A standard /etc/inetd.conf file looks like this:

```
ftp stream tcp nowait root /usr/sbin/tcpd \
        wu.ftpd -l -a
pop-3 stream tcp nowait root /usr/sbin/tcpd \
        in.qpop -s
```

A virtual /etc/inetd.conf file looks like this:

```
ftp stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.ftp wu.ftpd -l -a
pop-3 stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.pop in.qpop -s
```

3.3 Config File

Each service gets a config file that will control what IPs and directories are allowed for that service. You can have one master config file or several config files if you want each service to get a different list of domains. A config file looks like this:

```
# This is a comment and so are blank lines

# Format IP SPACE dir NOSPACES
```

```
10.10.10.129 /virtual/domain1.com
10.10.10.130 /virtual/domain2.com
10.10.10.157 /virtual/domain3.com

# Default option for all other IPs
default /
```

3.4 Source

This is the C source code to the virtuald program. Compile it and install it in /usr/local/bin with permission 0755, user root, and group root. The only compile option is VERBOSELOG which will turn on/off logging of connections.

```
#include <netinet/in.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdarg.h>
#include <unistd.h>
#include <string.h>
#include <syslog.h>
#include <stdio.h>

#undef VERBOSELOG

#define BUFSIZE 8192

int getipaddr(char **ipaddr)
{
        struct sockaddr_in virtual_addr;
        static char ipaddrbuf[BUFSIZE];
        int virtual_len;
        char *ipptr;

        virtual_len=sizeof(virtual_addr);
        if (getsockname(0,(struct sockaddr *)&virtual_addr,&virtual_len)<0)
        {
                syslog(LOG_ERR,"getipaddr: getsockname failed: %m");
                return -1;
        }
        if (!(ipptr=inet_ntoa(virtual_addr.sin_addr)))
        {
                syslog(LOG_ERR,"getipaddr: inet_ntoa failed: %m");
                return -1;
        }
        strncpy(ipaddrbuf,ipptr,sizeof(ipaddrbuf)-1);
        *ipaddr=ipaddrbuf;
        return 0;
}
```

TOC

```
int iptodir(char **dir,char *ipaddr,char *filename)
{
        char buffer[BUFSIZE],*bufptr;
        static char dirbuf[BUFSIZE];
        FILE *fp;

        if (!(fp=fopen(filename,"r")))
        {
                syslog(LOG_ERR,"iptodir: fopen failed: %m");
                return -1;
        }
        *dir=NULL;
        while(fgets(buffer,BUFSIZE,fp))
        {
                buffer[strlen(buffer)-1]=0;
                if (*buffer=='#' || *buffer==0)
                        continue;
                if (!(bufptr=strchr(buffer,' ')))
                {
                        syslog(LOG_ERR,"iptodir: strchr failed");
                        return -1;
                }
                *bufptr++=0;
                if (!strcmp(buffer,ipaddr))
                {
                        strncpy(dirbuf,bufptr,sizeof(dirbuf)-1);
                        *dir=dirbuf;
                        break;
                }
                if (!strcmp(buffer,"default"))
                {
                        strncpy(dirbuf,bufptr,sizeof(dirbuf)-1);
                        *dir=dirbuf;
                        break;
                }
        }
        if (fclose(fp)==EOF)
        {
                syslog(LOG_ERR,"iptodir: fclose failed: %m");
                return -1;
        }
        if (!*dir)
        {
                syslog(LOG_ERR,"iptodir: ip not found in conf file");
                return -1;
        }
        return 0;
```

TOC

```
}

int main(int argc,char **argv)
{
        char *ipaddr,*dir;

        openlog("virtuald",LOG_PID,LOG_DAEMON);

#ifdef VERBOSELOG
        syslog(LOG_ERR,"Virtuald Starting: $Revision: 1.49 $");
#endif
        if (!argv[1])
        {
                syslog(LOG_ERR,"invalid arguments: no conf file");
                exit(0);
        }
        if (!argv[2])
        {
                syslog(LOG_ERR,"invalid arguments: no program to run");
                exit(0);
        }
        if (getipaddr(&ipaddr))
        {
                syslog(LOG_ERR,"getipaddr failed");
                exit(0);
        }
#ifdef VERBOSELOG
        syslog(LOG_ERR,"Incoming ip: %s",ipaddr);
#endif
        if (iptodir(&dir,ipaddr,argv[1]))
        {
                syslog(LOG_ERR,"iptodir failed");
                exit(0);
        }
        if (chroot(dir)<0)
        {
                syslog(LOG_ERR,"chroot failed: %m");
                exit(0);
        }
#ifdef VERBOSELOG
        syslog(LOG_ERR,"Chroot dir: %s",dir);
#endif
        if (chdir("/")<0)
        {
                syslog(LOG_ERR,"chdir failed: %m");
                exit(0);
        }
        if (execvp(argv[2],argv+2)<0)
```

```
        {
                syslog(LOG_ERR,"execvp failed: %m");
                exit(0);
        }

        closelog();

        exit(0);
}
```

4. Shell Scripts

4.1 Virtfs

Each domain should get their own directory structure. Since you are using `chroot` you will require duplicate copies of the shared libraries, binaries, conf files, etc. I use /virtual/domain1.com for each domain that I create.

I realize that you are taking up more disk space but it is cheaper than a whole new machine and network cards. If you really want to preserve space you can hard link the files together so only one copy of each binary exists. The filesystem that I use takes up a little over 2M. However, this script attempts to copy all the files from the main filesystem in order to be as generic as possible.

Here is a sample virtfs script:

```
#!/bin/sh

echo '$Revision: 1.49 $'

echo -n "Enter the domain name: "
read domain

if [ "$domain" = "" ]
then
        echo Nothing entered: aborting
        exit 0
fi

leadingdir=/virtual

echo -n "Enter leading dir: (Enter for default: $leadingdir): "
read ans

if [ "$ans" != "" ]
then
        leadingdir=$ans
fi

newdir=$leadingdir/$domain

if [ -d "$newdir" ]
then
        echo New directory: $newdir: ALREADY exists
        exit 0
else
        echo New directory: $newdir
```

TOC

```
fi

echo Create $newdir
mkdir -p $newdir

echo Create bin
cp -pdR /bin $newdir

echo Create dev
cp -pdR /dev $newdir

echo Create dev/log
ln -f /virtual/log $newdir/dev/log

echo Create etc
mkdir -p $newdir/etc
for i in /etc/*
do
        if [ -d "$i" ]
        then
                continue
        fi
        cp -pd $i $newdir/etc
done

echo Create etc/skel
mkdir -p $newdir/etc/skel

echo Create home
for i in a b c d e f g h i j k l m n o p q r s t u v w x y z
do
        mkdir -p $newdir/home/$i
done

echo Create home/c/crc
mkdir -p $newdir/home/c/crc
chown crc.users $newdir/home/c/crc

echo Create lib
mkdir -p $newdir/lib
for i in /lib/*
do
        if [ -d "$i" ]
        then
                continue
        fi
        cp -pd $i $newdir/lib
done
```

TOC

```
echo Create proc
mkdir -p $newdir/proc

echo Create sbin
cp -pdR /sbin $newdir

echo Create tmp
mkdir -p -m 0777 $newdir/tmp
chmod +t $newdir/tmp

echo Create usr
mkdir -p $newdir/usr

echo Create usr/bin
cp -pdR /usr/bin $newdir/usr

echo Create usr/lib
mkdir -p $newdir/usr/lib

echo Create usr/lib/locale
cp -pdR /usr/lib/locale $newdir/usr/lib

echo Create usr/lib/terminfo
cp -pdR /usr/lib/terminfo $newdir/usr/lib

echo Create usr/lib/zoneinfo
cp -pdR /usr/lib/zoneinfo $newdir/usr/lib

echo Create usr/lib/\*.so\*
cp -pdR /usr/lib/*.so* $newdir/usr/lib

echo Create usr/sbin
cp -pdR /usr/sbin $newdir/usr

echo Linking usr/tmp
ln -s /tmp $newdir/usr/tmp

echo Create var
mkdir -p $newdir/var

echo Create var/lock
cp -pdR /var/lock $newdir/var

echo Create var/log
mkdir -p $newdir/var/log

echo Create var/log/wtmp
cp /dev/null $newdir/var/log/wtmp
```

TOC

```
echo Create var/run
cp -pdR /var/run $newdir/var

echo Create var/run/utmp
cp /dev/null $newdir/var/run/utmp

echo Create var/spool
cp -pdR /var/spool $newdir/var

echo Linking var/tmp
ln -s /tmp $newdir/var/tmp

echo Create var/www/html
mkdir -p $newdir/var/www/html
chown webmast.www $newdir/var/www/html
chmod g+s $newdir/var/www/html

echo Create var/www/master
mkdir -p $newdir/var/www/master
chown webmast.www $newdir/var/www/master

echo Create var/www/server
mkdir -p $newdir/var/www/server
chown webmast.www $newdir/var/www/server

exit 0
```

4.2 Virtexec

To execute commands in a virtual environment you have to `chroot` to that directory and then run the command. I have written a special shell script called virtexec that handles this for any command:

```
#!/bin/sh

echo '$Revision: 1.49 $'

BNAME=`basename $0`
FIRST4CHAR=`echo $BNAME | cut -c1-4`
REALBNAME=`echo $BNAME | cut -c5-`

if [ "$BNAME" = "virtexec" ]
then
        echo Cannot run virtexec directly: NEED a symlink
        exit 0
fi

if [ "$FIRST4CHAR" != "virt" ]
then
```

```
        echo Symlink not a virt function
        exit 0
fi

list=""
num=1
for i in /virtual/*
do
        if [ ! -d "$i" ]
        then
                continue
        fi
        if [ "$i" = "/virtual/lost+found" ]
        then
                continue
        fi
        list="$list $i $num"
        num=`expr $num + 1`
done

if [ "$list" = "" ]
then
        echo No virtual environments exist
        exit 0
fi

dialog --clear --title 'Virtexec' --menu Pick 20 70 12 $list 2> /tmp/menu.$$
if [ "$?" = "0" ]
then
        newdir=`cat /tmp/menu.$$`
else
        newdir=""
fi
tput clear
rm -f /tmp/menu.$$

echo '$Revision: 1.49 $'

if [ ! -d "$newdir" ]
then
        echo New directory: $newdir: NOT EXIST
        exit 0
else
        echo New directory: $newdir
fi

echo bname: $BNAME

echo realbname: $REALBNAME
```

```
if [ "$*" = "" ]
then
        echo args: none
else
        echo args: $*
fi

echo Changing to $newdir
cd $newdir

echo Running program $REALBNAME

chroot $newdir $REALBNAME $*

exit 0
```

Please note that you must have the `dialog` program installed on your system for this to work. To use virtexec just symlink a program to it. For example,

```
ln -s /usr/local/bin/virtexec /usr/local/bin/virtpasswd
ln -s /usr/local/bin/virtexec /usr/local/bin/virtvi
ln -s /usr/local/bin/virtexec /usr/local/bin/virtpico
ln -s /usr/local/bin/virtexec /usr/local/bin/virtemacs
ln -s /usr/local/bin/virtexec /usr/local/bin/virtmailq
```

Then if you type virtvi or virtpasswd or virtmailq it will allow you to vi a program, change a user's password or check the mail queue on your virtual system. You can create as many virtexec symlinks as you want. Please note that if your program requires a shared library it has to be in the virtual filesystem as well as the binary.

4.3 Notes

I install all the scripts in /usr/local/bin. Anything that I do not want to put on the virtual filesystem I put in /usr/local. The script does not copy any of the files in /usr/local to the virtual filesystem. Any files that are important to not cross virtual filesystems should be removed. For example, ssh is installed on my system and I did not want the private key for the server available on all the virtual filesystems so I remove it from each virtual filesystem after I run virtfs. I also change resolv.conf and remove anything that has the name of another domain on it for legal reasons. For example, /etc/hosts and /etc/HOSTNAME.

The programs that I symlink to virtexec are:

- virtpasswd -- change a user password
- virtadduser -- create a user
- virtdeluser -- delete a user
- virtsmbstatus -- see SAMBA status
- virtvi -- edit a file
- virtmailq -- check out the mailq
- virtnewaliases -- rebuild alias tables

---

5. DNS

You can configure DNS normally. There is a HOWTO on DNS.

6. Syslogd

6.1 Problem

Syslogd is the system logging utility commonly used on UNIX systems. Syslogd is a daemon that opens a special file called a FIFO. A FIFO is a special file that acts like a pipe. Anything that is written to the write side will come out the read side. Syslogd waits for data from the read side. There are C functions that write to the write side. If your program uses these C functions your output will go to syslogd.

Remember that we have used a `chroot` environment and the FIFO that syslogd is reading from (/dev/log) is not present. That means all the virtual environments will not log to syslogd.

6.2 Solution

Setup Links

Syslogd can look to a different FIFO if you tell it on the command line so run syslogd with the argument:

```
syslogd -p /virtual/log
```

Then symlink /dev/log to /virtual/log by:

```
ln -sf /virtual/log /dev/log
```

Then hard link all the /dev/log copies to this file by running:

```
ln -f /virtual/log /virtual/domain1.com/dev/log
```

The virtfs script above already does this. Since /virtual is one contiguous disk and the /dev/log's are hard linked they have the same inode number and point to the same data. The `chroot` cannot stop this so all your virtual /dev/log's will now function. Note that all the messages from all the environments will be logged in one place. However, you can write separate programs to filter out the data.

Syslogd.init

This version of the syslogd.init file hard links the /dev/log's each time you start it because syslogd deletes and creates the /dev/log FIFO each time it runs. Here is a modified syslogd.init file:

```
#!/bin/sh

. /etc/rc.d/init.d/functions
```

TOC

```
case "$1" in
  start)
        echo -n "Starting dev log: "
        ln -sf /virtual/log /dev/log
        echo done
        echo -n "Starting system loggers: "
        daemon syslogd -p /virtual/log
        daemon klogd
        echo
        echo -n "Starting virtual dev log: "
        for i in /virtual/*
        do
                if [ ! -d "$i" ]
                then
                        continue
                fi
                if [ "$i" = "/virtual/lost+found" ]
                then
                        continue
                fi
                ln -f /virtual/log $i/dev/log
                echo -n "."
        done
        echo " done"
        touch /var/lock/subsys/syslogd
        ;;
  stop)
        echo -n "Shutting down system loggers: "
        killproc syslogd
        killproc klogd
        echo
        rm -f /var/lock/subsys/syslogd
        ;;
  *)
        echo "Usage: syslogd {start|stop}"
        exit 1
esac

exit 0
```

6.3 Multiple Syslogd's

TOC

One Per Disk

If you run out of space on one filesystem and you have to break up your virtual domains onto different disks remember that hard links will not cross disks. That means you will have to run a separate syslogd for each group of domains on a disk. For example, if you had thirteen domains on /virtual1 and fifteen domains on /virtual2, you would hard link thirteen domains to /virtual1/log and run one syslogd with `syslogd -p /virtual1/log` and hard link fifteen other domains to /virtual2/log with a syslogd running with `syslogd -p /virtual2/log` .

One Per Domain

If you do not want to centralize the logs to one place you could also run one syslogd per domain. This wastes process ID's so I do not recommend it but it is easier to implement. You would have to alter your syslogd.init file to run syslogd as `chroot /virtual/domain1.com syslogd` for each domain. This will run each syslogd within the `chroot` and the logs will be in /virtual/domain1.com/var/log rather than all combined in /var/log. Do not forget to run a syslogd normally `syslogd` for the main system and a kernel logger `klogd` .

---

7. Virtual FTP

7.1 Inetd

Wu-ftpd comes with built in support to make it virtual. However, you cannot maintain separate password files for each domain. For example, if `bob@domain1.com` and `bob@domain2.com` both want an account you would have to make one of them bob2 or have one of the users choose a different user name. Since you now have a virtual filesystem for each domain you have separate password files and this problem goes away. Just create a virtnewuser script and a virtpasswd script in the way mentioned above and you are all set.

The inetd.conf entries for wu-ftpd:

```
ftp stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.ftp wu.ftpd -l -a
```

7.2 Anonymous FTP

These are unaffected by the virtuald setup. For an anonymous user just create the FTP user in /virtual/domain1.com/etc/passwd like you would normally.

```
ftp:x:14:50:Anonymous FTP:/var/ftp:/bin/false
```

Then setup the anonymous FTP directory. You have separate password files for each domain so you can restrict which domain has an anonymous FTP account. Please note that since the FTP server is already `chrooted` into the /virtual/domain1.com directory you do not have to prefix any paths with it.

7.3 Virtual FTP Users

Wu-ftpd supports something called a guest group. This allows you to create different FTP areas for each user. The FTP server does a `chroot` to the specified area so the user cannot go outside that directory tree. If you create the users within a virtual domain this way they will not be able to view the system files.

Add the guest's group to the /virtual/domain1.com/etc/ftpaccess file.

Create an entry in /virtual/domain1.com/etc/passwd with the `chroot` dir and the starting home directory separated by `/./` :

```
guest1:x:8500:51:Guest FTP:/home/g/guest1/./incoming:/bin/false
```

Then setup guest's home like you would for anonymous FTP. You have separate password files for e domain so you can specifiy which domains have guest accounts and which users within a domain ar guest users. Please note that since the FTP server is already `chrooted` into the /virtual/domain1.com directory you do not have to prefix any paths with it.

---

8. Virtual Web

8.1 Running With Virtuald

Not recommended

Apache has their own support for virtual domains. This is the only program I recommend using the internal virtual domain mechanism. When you run something through inetd there is a cost, the program has to start up each time you run it. This results in slower response time, which is perfectly fine for most services but is completely unacceptable for web service. Apache also has a mechanism for stopping connections when too many come in, which can be critical for even medium volume sites.

Simply stated, virtualizing Apache with virtuald is a really bad idea. The whole point of virtuald is to fill the gap created when services DO NOT have their own internal mechanism to do the job. Virtuald is not meant to replace good code that already completes the task at hand.

The above not withstanding here is how to do it for those who are foolhardy enough to do so.

Inetd

Edit /etc/inetd.conf

```
vi /etc/inetd.conf # Add this line
www stream tcp nowait www /usr/local/bin/virtuald \
        virtuald /virtual/conf.www httpd -f /var/www/conf/httpd.conf
```

Httpd.conf

Edit /var/www/conf/httpd.conf

```
vi /var/www/conf/httpd.conf # Or wherever you put the Apache config files
It should say:
ServerType standalone

Replace it with:
ServerType inetd
```

Configuration

Then configure each instance of the Apache server like you would normally for single domain use.

Httpd.init

An httpd.init file is not needed since the server is run through inetd.

8.2 Running With Apache VirtualHost

Apache has three configuration files `access.conf` , `httpd.conf` , and `srm.conf` . Newer versions of Apache have made the three configuration files unnecessary. However, I find that breaking up the configuration into three sections makes it easier to manage so I will be keeping with that style in this HOWTO document.

Access.conf

This configuration file is used to control the accessibility of directories in the web directory structure. Here is a sample configuration file that shows how to have different options for each domain.

```
# /var/www/conf/access.conf: Global access configuration

# Options are inherited from the parent directory
# Set the main directory with default options
<Directory />
AllowOverride None
Options Indexes
</Directory>

# Give one domain a passwd protected directory
<Directory /virtual/domain1.com/var/www/html/priv>
AuthUserFile /var/www/passwd/domain1.com-priv
AuthGroupFile /var/www/passwd/domain1.com-priv-g
AuthName PRIVSECTION
AuthType Basic
<Limit GET PUT POST>
require valid-user
</Limit>
</Directory>

# Give another domain Server Side Includes
<Directory /virtual/domain2.com/var/www/html>
Options IncludesNOEXEC
</Directory>
```

Httpd.conf

This configuration file is used to control the main options for the Apache server. Here is a sample configuration file that shows how to have different options for each domain.

```
# /var/www/conf/httpd.conf: Main server configuration file

# Begin: main conf section
```

```
# Needed since not using inetd
ServerType standalone

# Port to run on
Port 80

# Log clients with names vs IP addresses
HostnameLookups on

# User to run server as
User www
Group www

# Where server config, error and log files are
ServerRoot /var/www

# Process Id of server in this file
PidFile /var/run/httpd.pid

# Internal server process info
ScoreBoardFile /var/www/logs/apache_status

# Timeout and KeepAlive options
Timeout 400
KeepAlive 5
KeepAliveTimeout 15

# Number of servers to run
MinSpareServers 5
MaxSpareServers 10
StartServers 5
MaxClients 150
MaxRequestsPerChild 30

# End: main conf section

# Begin: virtual host section

# Tell server to accept requests for ip:port
# I have one for each IP needed so you can explicitly ignore certain domains
Listen 10.10.10.129:80
Listen 10.10.10.130:80

# VirtualHost directive allows you to specify another virtual
# domain on your server.  Most Apache options can be specified
# within this section.
<VirtualHost www.domain1.com>
```

TOC

```
# Mail to this address on errors
ServerAdmin webmaster@domain1.com

# Where documents are kept in the virtual domain
DocumentRoot /virtual/domain1.com/var/www/html

# Name of the server
ServerName www.domain1.com

# Log files Relative to ServerRoot option
ErrorLog logs/domain1.com-error_log
TransferLog logs/domain1.com-access_log
RefererLog logs/domain1.com-referer_log
AgentLog logs/domain1.com-agent_log

# Use CGI scripts in this domain
ScriptAlias /cgi-bin/ /var/www/cgi-bin/domain1.com/
AddHandler cgi-script .cgi
AddHandler cgi-script .pl
</VirtualHost>

<VirtualHost www.domain2.com>

# Mail to this address on errors
ServerAdmin webmaster@domain2.com

# Where documents are kept in the virtual domain
DocumentRoot /virtual/domain2.com/var/www/html

# Name of the server
ServerName www.domain2.com

# Log files Relative to ServerRoot option
ErrorLog logs/domain2.com-error_log
TransferLog logs/domain2.com-access_log
RefererLog logs/domain2.com-referer_log
AgentLog logs/domain2.com-agent_log

# No CGI's for this host
</VirtualHost>
# End: virtual host section
```

Srm.conf

This configuration file is used to control how requests are serviced and how results are formatted. You do not have to edit anything here for the virtual domains. The sample config file from Apache should work.

TOC

Httpd.init

Nothing special has to be done to the httpd.init file. Use a standard one that comes with the Apache configuration.

8.3 File Descriptor Overflow

Warning

This only applies to the standalone style Apache server. A server run through inetd does not interact with the other domains so it has the whole file descriptor table.

Every log file that the Apache server opens is another file descriptor for the process. There is a limit of 256 file descriptors per process in Linux. Since you have multiple domains you are using a lot more file descriptors. If you have too many domains running off of one Apache web server process you can overflow this table. This would mean that certain logs would not work and CGI's would fail.

Multiple Apache Servers

If you assume five file descriptors per domain you can have 50 domains running on your Apache server without any problems. However, if you find your server having problems like this you could create /var/www1 with an Apache server in charge of domain1 - domain25 and /var/www2 with an Apache server in charge of domain26 - domain50 and so on. This would give each server their own configuration, error, and log directory. Each server should be configured separately with their own Listen and VirtualHost directives. Do not forget to run multiple servers in your httpd.init file.

8.4 Sharing Servers With One IP

Saving IPs

The HTTP (HyperText Transfer Protocol) version 1.1 added a feature that communicates the name of the server to the client. This means that the client does not need to look up the server from its IP address. Therefore, two virtual servers could have the same IP address and be different web sites. The Apache configuration is the same as above except that you do not have to put in a different Listen directive since the two domains will have the same IP.

Drawback

The only problem is that virtuald uses IP addresses to distinguish between domains. In its current form virtuald would not be able to `chroot` to different spool directories for each domain. Therefore, mail would only be able to respond as one IP and there would no longer be a unique spool directory for each domain. All the web sharing IP clients would have to share that IPs spool directory. That would mean duplicate usernames would be an issue again. However, that is the price you pay for sharing IPs.

8.5 More Information

This HOWTO only shows how to implement virtual support on the Apache web server. Most web servers use a similar interface. For more information on virtual web hosting consult the WWW HOWTO, the documentation for Apache at Apache's Site, or the documentation at ApacheWeek.

---

9. Virtual Mail/Pop

9.1 Problem

Virtual mail support is in ever increasing demand. Sendmail says it supports virtual mail. What it does support is listening for incoming mail from different domains. You can then specify to have the mail forwarded somewhere. However, if you forward it to the local machine and have incoming mail to bob@domain1.com and bob@domain2.com they will go to the same mail folder. This is a problem since both bob's are different people with different mail.

9.2 Solution

You can make sure that each user name is unique by using a numbering scheme: bob1, bob2, etc or prepending a few characters to each username dom1bob, dom2bob, etc. You could also hack mail and pop to do these conversions behind the scenes but that can get messy. Outgoing mail also has the banner maindomain.com and you want each subdomain's outgoing mail banner to be different.

I have two solutions. One works with sendmail and one works with Qmail. The solution with sendmail should work with a stock install of sendmail. However, it shares all the limitations built into sendmail. It also requires that one sendmail has to be run in queue mode for each domain. Having 50 or more sendmail queue processes that wake up every hour can put a little strain on a machine.

The solution offered with Qmail does not require multiple instances of Qmail and can run out of one queue directory. It does require an extra program since Qmail does not rely on virtuald. I believe a similar procedure can be done with sendmail. However, Qmail lends itself to this solution more readily.

I do not endorse any one program over the other. The sendmail install is a little more straight forward but Qmail is probably the more powerful of the two mail server packages.

9.3 Sendmail Solution

Introduction

Each virtual filesystem gives a domain its own /etc/passwd. This means that bob@domain1.com and bob@domain2.com are different users in different /etc/passwds so mail will be no problem. They also have their own spool directories so the mail folders will be different files on different virtual filesystems.

Create Sendmail Configuration File

Create /etc/sendmail.cf like you would normally through m4. I used:

```
divert(0)
VERSIONID(`tcpproto.mc')
OSTYPE(linux)
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
```

```
MAILER(local)
MAILER(smtp)
```

Edit Sendmail Configuration File

Edit /virtual/domain1.com/etc/sendmail.cf to respond as your virtual domain:

```
vi /virtual/domain1.com/etc/sendmail.cf # Approximately Line 86
It should say:

#Dj$w.Foo.COM

Replace it with:

Djdomain1.com
```

Sendmail Local Delivery

Edit /virtual/domain1.com/etc/sendmail.cw with the local hostnames.

```
vi /virtual/domain1.com/etc/sendmail.cw
mail.domain1.com
domain1.com
domain1
localhost
```

Sendmail Between Virtual Domains: The Hack (PRE8.8.6)

However, sendmail requires one minor source code modification. Sendmail has a file called /etc/sendmail.cw and it contains all machine names that sendmail will deliver mail to locally rather than forwarding to another machine. Sendmail does internal checking of all the devices on the machine to initialize this list with the local IPs. This presents a problem if you are mailing between virtual domains on the same machine. Sendmail will be fooled into thinking another virtual domain is a local address and spool the mail locally. For example, bob@domain1.com sends mail to fred@domain2.com. Since domain1.com's sendmail thinks domain2.com is local, it will spool the mail on domain1.com and never send it to domain2.com. You have to modify sendmail (I did this on v8.8.5 without a problem):

```
vi v8.8.5/src/main.c # Approximately Line 494
It should say:

load_if_names();

Replace it with:

/* load_if_names(); Commented out since hurts virtual */
```

Note only do this if you need to send mail between virtual domains which I think is probable.

This will fix the problem. However, the main ethernet device eth0 is not removed. Therefore, if you send mail from a virtual IP to the one on eth0 on the same box it will delivery locally. Therefore, I just use this as a dummy IP virtual1.maindomain.com (10.10.10.157). I never send mail to this host so neither will the virtual domains. This is also the IP I would use to ssh into the box to check if the system is ok.

Sendmail Between Virtual Domains: New Sendmail Feature (POST8.8.6)

TOC

As of Sendmail V8.8.6, there is a new option to disable loading of the extra network interfaces. This means you do NOT have to alter the code in any way. It is called `DontProbeInterfaces` .

Edit /virtual/domain1.com/etc/sendmail.cf

```
vi /virtual/domain1.com/etc/sendmail.cf # Add the line
O DontProbeInterfaces=True
```

Sendmail.init

Sendmail cannot be started stand alone anymore so you have to run it through inetd. This is inefficient and will result in lower start up time but if you had such a high hit site you would not share it on a virtual box with other domains. Note that you are NOT running with the `-bd` flag. Also note that you need a `sendmail -q` running for each domain to queue up undelivered mail. The new sendmail.init file:

```
#!/bin/sh

. /etc/rc.d/init.d/functions

case "$1" in
  start)
        echo -n "Starting sendmail: "
        daemon sendmail -q1h
        echo
        echo -n "Starting virtual sendmail: "
        for i in /virtual/*
        do
                if [ ! -d "$i" ]
                then
                        continue
                fi
                if [ "$i" = "/virtual/lost+found" ]
                then
                        continue
                fi
                chroot $i sendmail -q1h
                echo -n "."
        done
        echo " done"
        touch /var/lock/subsys/sendmail
        ;;
  stop)
        echo -n "Stopping sendmail: "
        killproc sendmail
        echo
        rm -f /var/lock/subsys/sendmail
        ;;
  *)
        echo "Usage: sendmail {start|stop}"
```

```
        exit 1
esac

exit 0
```

Inetd Setup

Pop should install normally with no extra effort. It will just need the inetd entry for it with the virtuald part added. The inetd.conf entries for sendmail and pop:

```
pop-3 stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.pop in.qpop -s
smtp stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.mail sendmail -bs
```

9.4 Qmail Solution

Introduction

This solution takes over the delivery responsibilities of qmail-local, so use of the .qmail files in the virtual home directories will not work. However, each domain will still get a domain master user that will control aliasing for the whole domain. Two external programs will be used for that domain masters .qmail-default file. The mail will be passed through these two programs in order to deliver mail for each domain.

Two programs are required since one of them is run setuid root. It is a small program that changes to a non-root user and then runs the second program. Consult your nearest security related site for a discussion as to why this is necessary.

This solution bypasses the need for using virtuald. Qmail is flexible enough to not require a general virtuald setup. Qmail's design utilizes the chaining of programs together to deliver mail. This design makes it very easy to insert the virtual section into the Qmail delivery process without altering a stock install of Qmail.

A note that since you are using one Qmail any unqualified domain name will be expanded with the domain of the main server. This is because you do not have a separate Qmail server for each domain. Therefore, make sure that your client (Eudora, elm, mutt, etc.) knows to expand all of your unqualified domain names.

Setup Virtual Domains

Qmail has to be configured to accept mail for each of the virtual domains you will be serving. Type the following commands.

```
echo "domain1.com:domain1" >> /var/qmail/control/virtualdomains
```

Setup Domain Master User

Add to your main /etc/passwd file the user domain1. I would make the shell /bin/false so that the domain master cannot log in. That user will be able to add .qmail files and all mail for domain1 will route through that account. Note that usernames can only be eight characters long and domain names can be longer. The remaining characters are truncated. That means that user domain12 and domain123 are going to be the same user and Qmail might get confused. So be careful in your master domain user naming convention.

Create the domain master's .qmail files with the following commands. Add any other system aliases at this point. For example, webmaster or hostmaster.

TOC

```
echo "user@domain1.com" > /home/d/domain1/.qmail-mailer-daemon
echo "user@domain1.com" > /home/d/domain1/.qmail-postmaster
echo "user@domain1.com" > /home/d/domain1/.qmail-root
```

Create the domain master's .qmail-default file. This will filter all mail to the virtual domain.

```
echo "| /usr/local/bin/virtmailfilter" > /home/d/domain1/.qmail-default
```

Tcpserver

Qmail requires a special pop that can support the Maildir format. The pop program has to be virtualized. The author of Qmail recommends using tcpserver (an inetd replacement) with Qmail so my examples use tcpserver and NOT inetd.

Tcpserver does not require a config file. All the information can be passed to it via the command line. Here is the tcpserver.init file that you would use for the mail daemon and popper:

```
#!/bin/sh

. /etc/rc.d/init.d/functions

QMAILDUSER=`grep qmaild /etc/passwd | cut -d: -f3`
QMAILDGROUP=`grep qmaild /etc/passwd | cut -d: -f4`

# See how we were called.
case "$1" in
  start)
        echo -n "Starting tcpserver: "
        tcpserver -u 0 -g 0 0 pop-3 /usr/local/bin/virtualdd \
                /virtual/conf.pop qmail-popup virt.domain1.com \
                /bin/checkpassword /bin/qmail-pop3d Maildir &
        echo -n "pop "
        tcpserver -u $QMAILDUSER -g $QMAILDGROUP 0 smtp \
                /var/qmail/bin/qmail-smtpd &
        echo -n "qmail "
        echo
        touch /var/lock/subsys/tcpserver
        ;;
  stop)
        echo -n "Stopping tcpserver: "
        killall -TERM tcpserver
        echo -n "killing "
        echo
        rm -f /var/lock/subsys/tcpserver
        ;;
  *)
        echo "Usage: tcpserver {start|stop}"
        exit 1
esac

exit 0
```

Qmail.init

You can use the standard Qmail init script provided. Qmail comes with very good documentation describing how to set this up.

Source

You require two other programs to get virtual mail working with Qmail. They are virtmailfilter and virtmaildelivery. This is the C source to virtmailfilter. It should be installed in /usr/local/bin with permissions 4750, user root, and group nofiles.

```
#include <sys/wait.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <ctype.h>
#include <pwd.h>

#define VIRTPRE                 "/virtual"

#define VIRTPWFILE              "etc/passwd"
#define VIRTDELIVERY            "/usr/local/bin/virtmaildelivery"
#define VIRTDELIVERY0           "virtmaildelivery"

#define PERM                    100
#define TEMP                    111
#define BUFSIZE                 8192

int main(int argc,char **argv)
{
        char *username,*usernameptr,*domain,*domainptr,*homedir;
        char virtpath[BUFSIZE];
        struct passwd *p;
        FILE *fppw;
        int status;
        gid_t gid;
        pid_t pid;

        if (!(username=getenv("EXT")))
        {
                fprintf(stdout,"environment variable EXT not set\n");
                exit(TEMP);
        }

        for(usernameptr=username;*usernameptr;usernameptr++)
        {
                *usernameptr=tolower(*usernameptr);
        }

        if (!(domain=getenv("HOST")))
```

TOC

```
        {
                fprintf(stdout,"environment variable HOST not set\n");
                exit(TEMP);
        }

        for(domainptr=domain;*domainptr;domainptr++)
        {
                if (*domainptr=='.' && *(domainptr+1)=='.')
                {
                        fprintf(stdout,"environment variable HOST has ..\n");
                        exit(TEMP);
                }
                if (*domainptr=='/')
                {
                        fprintf(stdout,"environment variable HOST has /\n");
                        exit(TEMP);
                }

                *domainptr=tolower(*domainptr);
        }

        for(domainptr=domain;;)
        {
                snprintf(virtpath,BUFSIZE,"%s/%s",VIRTPRE,domainptr);
                if (chdir(virtpath)>=0)
                        break;

                if (!(domainptr=strchr(domainptr,'.')))
                {
                        fprintf(stdout,"domain failed: %s\n",domain);
                        exit(TEMP);
                }

                domainptr++;
        }

        if (!(fppw=fopen(VIRTPWFILE,"r+")))
        {
                fprintf(stdout,"fopen failed: %s\n",VIRTPWFILE);
                exit(TEMP);
        }

        while((p=fgetpwent(fppw))!=NULL)
        {
                if (!strcmp(p->pw_name,username))
                        break;
        }

        if (!p)
        {
                fprintf(stdout,"user %s: not exist\n",username);
                exit(PERM);
```

TOC

```
        }

        if (fclose(fppw)==EOF)
        {
                fprintf(stdout,"fclose failed\n");
                exit(TEMP);
        }

        gid=p->pw_gid;
        homedir=p->pw_dir;

        if (setgid(gid)<0 || setuid(p->pw_uid)<0)
        {
                fprintf(stdout,"setuid/setgid failed\n");
                exit(TEMP);
        }

        switch(pid=fork())
        {
                case -1:
                        fprintf(stdout,"fork failed\n");
                        exit(TEMP);
                case 0:
                        if
(execl(VIRTDELIVERY,VIRTDELIVERY0,username,homedir,NULL)<0)
                        {
                                fprintf(stdout,"execl failed\n");
                                exit(TEMP);
                        }
                default:
                        if (wait(&status)<0)
                        {
                                fprintf(stdout,"wait failed\n");
                                exit(TEMP);
                        }
                        if (!WIFEXITED(status))
                        {
                                fprintf(stdout,"child did not exit normally\n");
                                exit(TEMP);
                        }
                        break;
        }

        exit(WEXITSTATUS(status));
}
```

Source

You require two other programs to get virtual mail working with Qmail. They are virtmailfilter and virtmaildelivery. This is the C source to virtmaildelivery. It should be installed in /usr/local/bin with permissions 0755, user root, and group root.

TOC

```
#include <sys/stat.h>
#include <sys/file.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <stdio.h>
#include <errno.h>
#include <time.h>

#define TEMP                    111
#define BUFSIZE                 8192
#define ATTEMPTS                10

int main(int argc,char **argv)
{
        char *user,*homedir,*dtline,*rpline,buffer[BUFSIZE],*p,mail[BUFSIZE];
        char maildir[BUFSIZE],newmaildir[BUFSIZE],host[BUFSIZE];
        int fd,n,nl,i,retval;
        struct stat statp;
        time_t thetime;
        pid_t pid;
        FILE *fp;

        retval=0;

        if (!argv[1])
        {
                fprintf(stdout,"invalid arguments: need username\n");
                exit(TEMP);
        }

        user=argv[1];

        if (!argv[2])
        {
                fprintf(stdout,"invalid arguments: need home directory\n");
                exit(TEMP);
        }

        homedir=argv[2];

        if (!(dtline=getenv("DTLINE")))
        {
                fprintf(stdout,"environment variable DTLINE not set\n");
                exit(TEMP);
        }

        if (!(rpline=getenv("RPLINE")))
        {
                fprintf(stdout,"environment variable RPLINE not set\n");
                exit(TEMP);
```

TOC

```
        }

        while (*homedir=='/')
                homedir++;
        snprintf(maildir,BUFSIZE,"%s/Maildir",homedir);
        if (chdir(maildir)<0)
        {
                fprintf(stdout,"chdir failed: %s\n",maildir);
                exit(TEMP);
        }

        time(&thetime);
        pid=getpid();
        if (gethostname(host,BUFSIZE)<0)
        {
                fprintf(stdout,"gethostname failed\n");
                exit(TEMP);
        }

        for(i=0;i<ATTEMPTS;i++)
        {
                snprintf(mail,BUFSIZE,"tmp/%u.%d.%s",thetime,pid,host);
                errno=0;
                stat(mail,&statp);
                if (errno==ENOENT)
                        break;

                sleep(2);
                time(&thetime);
        }
        if (i>=ATTEMPTS)
        {
                fprintf(stdout,"could not create %s\n",mail);
                exit(TEMP);
        }

        if (!(fp=fopen(mail,"w+")))
        {
                fprintf(stdout,"fopen failed: %s\n",mail);
                retval=TEMP; goto unlinkit;
        }

        fd=fileno(fp);

        if (fprintf(fp,"%s",rpline)<0)
        {
                fprintf(stdout,"fprintf failed\n");
                retval=TEMP; goto unlinkit;
        }

        if (fprintf(fp,"%s",dtline)<0)
        {
```

```
        fprintf(stdout,"fprintf failed\n");
        retval=TEMP; goto unlinkit;
}

while(fgets(buffer,BUFSIZE,stdin))
{
        for(p=buffer;*p=='>';p++)
                ;

        if (!strncmp(p,"From ",5))
        {
                if (fputc('>',fp)<0)
                {
                        fprintf(stdout,"fputc failed\n");
                        retval=TEMP; goto unlinkit;
                }
        }

        if (fprintf(fp,"%s",buffer)<0)
        {
                fprintf(stdout,"fprintf failed\n");
                retval=TEMP; goto unlinkit;
        }
}

p=buffer+strlen(buffer);
nl=2;
if (*p=='\n')
        nl=1;

for(n=0;n<nl;n++)
{
        if (fputc('\n',fp)<0)
        {
                fprintf(stdout,"fputc failed\n");
                retval=TEMP; goto unlinkit;
        }
}

if (fsync(fd)<0)
{
        fprintf(stdout,"fsync failed\n");
        retval=TEMP; goto unlinkit;
}

if (fclose(fp)==EOF)
{
        fprintf(stdout,"fclose failed\n");
        retval=TEMP; goto unlinkit;
}

snprintf(newmaildir,BUFSIZE,"new/%u.%d.%s",thetime,pid,host);
```

```
        if (link(mail,newmaildir)<0)
        {
                fprintf(stdout,"link failed: %s %s\n",mail,newmaildir);
                retval=TEMP; goto unlinkit;
        }

unlinkit:
        if (unlink(mail)<0)
        {
                fprintf(stdout,"unlink failed: %s\n",mail);
                retval=TEMP;
        }

        exit(retval);
}
```

9.5 Acknowledgement

Thank you Vicente Gonzalez (vince@nycrc.net) for helping make the Qmail solution possible. You can certainly mail your thanks to Vince, however all questions and comments including issues regarding Qmail, about this HOWTO should continue to be directed to me.

10. Virtual Samba

10.1 Setup

Virtual SAMBA is very simple to install. Make sure that the following files are setup properly:

- /virtual/domain1.com/etc/smb.conf FILE
- /virtual/domain1.com/var/lock/samba DIRECTORY
- /virtual/domain1.com/var/log DIRECTORY
- /usr/local/bin/virtsmbstatus SYMLINK /usr/local/bin/virtexec

10.2 Inetd

Edit /etc/inetd.conf

```
vi /etc/inetd.conf # Add this line
netbios-ssn stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.smbd smbd
```

10.3 Smb.init

An smb.init file is not needed since the server is run through inetd.

11. Virtual Other

Any other service should be a similar procedure.

- Run virtfs to add the binaries and libraries to the virtual filesystem.
- Add it to /etc/inetd.conf.
- Create a /virtual/conf.service file.
- Create any virtual scripts that need to be made.

12. Conclusion

Those are all the steps you need. Again mail any responses to Computer Resource Center. If you have a question or an update to the document let me know and I will add it.

The document has met with a very good response. I thank all the people who sent me questions as they are helping to shape the document to meet the needs of users everywhere. Before you ask a question I urge you to read the FAQ to see if it has been already asked and answered. Thanks again. Brian

13. FAQ

Q1. I created sendmail.init and syslogd.init. I put them in /usr/local/bin and tried to run them but I got errors.

A1. These files are called init scripts. They are run by the program init when your computer boots. They do not go with the /usr/local binaries. Consult the Linux System Administrators Guide or the Linux Getting Started Guide for information on how to use the init scripts system.

Q2. I put these lines into /etc/sendmail.cf

```
divert(0)
VERSIONID(`tcpproto.mc')
OSTYPE(linux)
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(local)
MAILER(smtp)
```

And I got really stange output. Why?

A2. You do not put these lines directly in /etc/sendmail.cf. The sendmail.cf file was written to be easy for sendmail to understand and hard for humans to read. Therefore, to make it easy to configure we use a program called m4 and its macro capabilities to create the sendmail.cf file. The FEATURE lines are actually macros that expand to sendmail configuration statements. See the sendmail docs on how to configure sendmail through this method. Also note that you create a main /etc/sendmail.cf file and the virtfs script then copies this to /virtual/domain1.com/etc/sendmail.cf. Then you edit that sendmail.cf file to respond as your domain.

Q3. Where do I get virtuald, what is it, and how do I use it?

A3. Virtuald is C source that I wrote to run a virtual service. It is included with this HOWTO. You compile it like a normal C program `make virtuald` . The resulting binary is placed into /usr/local/bin. Add lines to /etc/inetd.conf that use virtuald as a wrapper to a normal network server program.

Q4. I do not have dialog installed on my system?

A4. Dialog is a program that allows you to put dialog pop up windows into your shell scripts. It is required for my virtual shell script examples to work. You can get a copy of dialog at sunsite. It compiles very easily and should be no problem to install.

Q5. How can I know if virtual syslogd is working?

A5. When virtuald runs it should output the following messages to syslogd (/var/log/messages):

```
Nov 19 17:21:07 virtual virtuald[10223]: Virtuald Starting: $Revision: 1.49 $
Nov 19 17:21:07 virtual virtuald[10223]: Incoming ip: 204.249.11.136
Nov 19 17:21:07 virtual virtuald[10223]: Chroot dir: /virtual/domain1.com
```

TOC

The `Chroot dir` message is sent by virtuald after the `chroot` system call is performed. If this message ap virtual syslogd is working. If the service you are virtualizing logs messages to syslogd and you see them that i also a sign that virtual syslogd is correctly setup.

Note that if you have not turned on the compile time option VERBOSELOG, virtuald will not log at all. The only way to tell if virtual syslogd is working at that point is if the daemon you are virtualizing independently logs something to syslogd.

Q6. How can I setup quotas across virtual filesystems?

A6. You setup quotas like you would normally. See the Quota mini-HOWTO. However, you have to make sure there are no uid conflicts across domains. If there are conflicts you will have users sharing a quota. Set aside a range of uid's that you know will have quota's enabled and tell your domains that they cannot have any users in that range except the ones registered to have a quota.

Q7. What is this \ notation in all the inetd.conf entries?

A7. That is just a method of breaking up config files across two lines. I did that so the line would word wrap in a nice place. You can just ignore the \ and join the two lines back together.

Q8. When I run passwd or other login programs I get `permission denied` . When I run FTP or su I get `no modules loaded for service XXX` . Why?

A8. Those are PAM error messages. I wrote these scripts before PAM was out. My virtfs script does not copy /etc/pam.d, /usr/lib/cracklib_dict.*, /lib/security or any of the other files PAM requires. PAM needs these to function. If you edit my virtfs script to copy these files the problem will go away.

Q9. Can virtuald work with tcpd hosts.allow and hosts.deny files?

A9. Yes it can with some modifications.

First the source has to be changed in two places.

This has to be inserted where the arguments are checked.

```
        if (!argv[3])
        {
                syslog(LOG_ERR,"invalid arguments: no program to run");
                exit(0);
        }
```

The exec line has to be changed from:

```
        if (execvp(argv[2],argv+2)<0)
```

to:

```
        if (execvp(argv[2],argv+3)<0)
```

Second the inetd.conf lines have to be changed from:

```
ftp stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.ftp wu.ftpd -l -a
```

TOC

to:

```
ftp stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.ftp tcpd wu.ftpd -l -a
```

Third edit the /virtual/domain1.com/etc/hosts.allow and /virtual/domain1.com/etc/hosts.deny files accordingly.

Q10. Can my virtual hosts run CGI's?

A10. Yes they can but I recommend putting the /cgi-bin in a place outside of the `chroot` that only you have access to. For example, /var/www/cgi-bin/domain1.com. Giving clients access to /cgi-bin is giving them the opportunity to run programs on your sever. This is a big security hole. Be careful. I do not let any cgi run on my systems that I have not personally inspected for bugs.

Q11. My configuration files are different from your examples. What do I do?

A11. There are two basic configuration styles: SystemV and BSD. The examples provided in the HOWTO are based on SystemV style configuration files. Virtual services works equally well on either system. For information on BSD style configuration files consult the origin of your distribution or the nearest LDP site.

Q12. I sent you mail and have not heard a response from you or your response took a long time. Why?

A12. Probably because you did not put VIRTSERVICES HOWTO in your subject header. Please bear in mind that I am a network administrator and that among the other things I do in my 20 hour days is administering my own virtual boxes and those of my clients. Mail that is properly addressed is always responded to within two or three days. Mail that is improperly addressed does not get filtered into my VIRTSERVICES mailbox and can lie around unnoticed for days or weeks.

Q13. Does virtuald work under 100Mbit?

A13 The speed of the network card is unrelated to whether virtuald will work or not. Try making sure that your server works under 10Mbit and that your 100Mbit network card works normally without a virtual server.

Q14. Should I use sendmail's virthost table?

A14. No. That is sendmail's feature to accept info for multiple domains. Virtuald gives each sendmail its own separate `chroot` environment. Install virtuald and then configure sendmail like you would normally for each domain.

Q15. Can I setup virtual telnet on my machine? What about creating a virtual root account so clients can administer their own domains?

A15. These questions come to me quite often and to be honest, I am getting a bit tired of them. The answer, as stated numerous times in the documentation, is that any service run through inetd can be virtualized using virtuald so there is nothing to stop you from doing either of the above. Nothing except common sense. Whatever benefits you might derive from allowing telnet are heavily outweighed by the cost to the virtual box (and thus the sites you are supposed to be hosting in a responsible manner) in terms of security. Here are just a few issues involved:

- In order to completely fool an incoming telnet session you have to hack the kernel to get multiple procs working, reset your source IP address for outgoing connections, fool gethostname so it uses the virtual hostname and not the system hostname, etc. If you are an advanced user then by all means hack the kernel. For the newbie I do not recommend it.
- By allowing users to come into your box via telnet you allow them to run arbitrary programs. Through known hacks you can get root and cause damage to the system.
- Giving a root telnet account on a virtual box is very bad. A root virtual user can still read raw device files

which nullifies the `chroot` , shutdown the system, and can kill other processes on the system.

- The programs that these telnet sessions are running take up valuable CPU time that the network services could be using.
- Telnet is an insecure network service. Plain text passwords are sent out over the net. If a malicious user gets this password he/she can use the above mentioned attacks to harm your system.
- Your virtual environments will have to be bigger. You will need more shared libraries, more configuration files, and more binaries. A six gigabyte disk can run out of space really fast.

The bottom line is that allowing login's on a virtual box is a really bad idea. If permitted, every site hosted on that machine is at risk. If you want to allow a site holder to administer users then you are advised to write (not script) the code necessary to run the virtual processes that allow them to add, delete or modify users upon login through ssh. This should be completely menu driven, should never allow a console and should not run as root. In order to accomplish this you will have to change ownership of the pertinent files from root to some other user. If done in this manner it is marginally safe to incorporate into a virtual machine. There is never an acceptable time to allow root login's either through telnet or ssh. Doing so is simply an invitation to disaster. If there is an overwhelming reason to run telnet then the site should be hosted on a dedicated machine where the only risk is to the individual site. No responsible administrator would ever do otherwise and so I will waste no more time on this issue.

Q16. Is there an rpm, tar, web site, mailing list, etc. associated with virtuald and the Virtual-Services HOWTO?

A16. Currently there is nothing like that available. This HOWTO is the only source of information to everything I do concerning this project. I find the HOWTO to be fairly self contained making the need for other pieces of information superfluous.

Q17. When I try to run virtexec as a regular user I get `chroot: operation not permitted` . Why?

A17. `Chroot` is a root restricted system call. Only the superuser can execute it. The virtexec script runs the `chroot` program which is why you need to be root in order to run it.

Q18. I setup pop and sendmail but popping mail does not seem to work. How come?

A18. Some pop programs come with /usr/spool/mail as their place for mail files. I know that qpop has to be manually editted to fix this. Either recompile the source to your program or symlink /virtual/domain1.com/usr/spool to /virtual/domain1.com/var/spool.

Q19. I did not use the program mentioned in your HOWTO, I used program XXX. It does not work. Why?

A19. I tried to make sure to use the most generic of each server in my examples. However, I know that everyone has their favorite version of each server. Send me as much information as possible and I will try to figure out how to solve your problem and document it in the FAQ. The most important piece of information to send me is where to get the version of the software you are running (in the form ftp://ftp.domain1.com/subdir/subdir/file.tgz).

Q20. When I run virtexec is says `symlink not a virt function` . What does this mean and how do I fix it?

A20. Virtexec is a program that will take its zero argument, strip off the first four characters, and run the remaining name in the virtual environment. For example, virtpasswd runs passwd. If the first four characters that it strips off are not `virt` it complains and outputs that error message. Virtexec is written in shell script and should be fairly simple to follow. Refer to the manual pages on bash or whatever shell you run for questions about shell script programming.

Q21. I have a question about Qmail, SAMBA, Apache, etc. that is unrelated to the virtuald setup or how the package interfaces to virtuald.

TOC

A21. All the packages described here are fully documented. Some even have full web sites like www.packagename.org dedicated to them. Please consult them about questions dealing with the package that unrelated to their virtual hosting functionality.

Q22. I have several domain aliases to domain1.com but mail keeps bouncing from the aliases. How come?

A22. Virtmaildelivery relies on the environment variables passed to it to determine which /virtual/domain1.com directory to deliver to. It does not perform any DNS lookups to determine the address of the mail. However, if the address is submail.mail.domain1.com, virtmaildelivery will first try that address and then mail.domain1.com and then domain1.com and then com in that order until either a match happens or there is no domain name left.

However, if you have domain aliases that are not subdomains of one another you have to create symlinks like so:

```
cd /virtual
ln -s domain1.com domain1alias.com
```

That way virtmaildelivery will be fooled into thinking that both directories exist even though one is a symlink and mail will be able to be delivered to user@domain1.com or user@domain1alias.com. Note that virtexec will list both of the domains in the dialog box when your run it. You can choose either one since they will be the same virtual filesystem.

Linux WWW HOWTO

by Mr. Poet, poet@linuxports.com

v0.85, 21 August 1999

This document contains information about setting up WWW services under Linux (both server and client). It tries not to be a in detail manual but an overview and a good pointer to further information.

1. Introduction

- 1.1 Copyright
- 1.2 Feedback
- 1.3 New versions of this Document

2. Setting up WWW client software (Antiquated)

- 2.1 Overview

3. Lynx

- 3.1 Where to get

4. Emacs-W3

- 4.1 Where to get

5. Netscape Navigator/Communicator

- 5.1 Different versions and options.
- 5.2 Where to get
- 5.3 Installing

TOC

6. Setting up WWW server systems

- 6.1 Overview

7. Apache

- 7.1 Where to get
- 7.2 Compiling and Installing
- 7.3 Configuring
- 7.4 Hosting virtual websites
- 7.5 CGI scripts
- 7.6 Users Web Directories
- 7.7 Daemon mode vs. Inetd mode
- 7.8 Allowing put and delete commands
- 7.9 User Authentication/Access Control
- 7.10 su-exec
- 7.11 Imagemaps
- 7.12 SSI/XSSI
- 7.13 Module system

8. Web Server Add-ons

9. Intranet Section

- 9.1 What is required
- 9.2 New versions of this document
- 9.3 Feedback

10. Install the HTTP server

- 10.1 Preparation before downloading
- 10.2 Compiling HTTPd

11. Testing HTTPd

12. Connecting to the Linux Server

- 12.1 Setup the Linux server
- 12.2 Setup the Netware server
- 12.3 Setup the Netware Client
- 12.4 Setup Microsoft Client
- 12.5 Setup TCP/IP on Macintosh

13. Setting up the Intranet

- 13.1 NCPFS
- 13.2 SMBFS
- 13.3 NFS

14. Accessing the Web

- 14.1 Accessing the mounted filesystems
- 14.2 Connecting to the Internet
- 14.3 Other uses

15. More things to do

16. Credits and Legalities

- 16.1 Thanks
- 16.2 Copyright information

17. For further reading

- 17.1 O'Reilly & Associates Books
- 17.2 Internet Request For Comments (RFC)

1. Introduction

Many people are trying Linux because they are looking for a really good *Internet capable* operating system. Also, there are institutes, universities, non-profits, and small businesses which want to set up Internet sites on a small budget. This is where the WWW-HOWTO comes in. This document explains how to set up clients and servers for the largest part of the Internet - *The World Wide Web*.

All prices in this document are stated in US dollars. This document assumes you are running Linux on an Intel platform. Instructions and product availability my vary from platform to platform. There are many links for downloading software in this document. Whenever possible use a mirror site for faster downloading and to keep the load down on the main server.

1.1 Copyright

This document is Copyright (c) 1999 by Mr. Poet

This document is Copyright (c) 1997 by Wayne Leister. The original author of this document was Peter Dreuw.(All versions prior to 0.8)

This HOWTO is free documentation; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details.

You can obtain a copy of the GNU General Public License by writing to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Trademarks are owned by there respective owners.

1.2 Feedback

Any feedback is welcome. I do not claim to be an expert. Some of this information was taken from badly written web sites; there are bound to be errors and omissions. But make sure you have the latest version before you send corrections; It may be fixed in the next version (see the next section for where to get the latest version). Send feedback to poet@linuxports.com.

1.3 New versions of this Document

New versions of this document are always available at the LinuxPorts Website.

2. Setting up WWW client software (Antiquated)

Please note that the following section is very outdated and was last updated in 1997. This section is being kept here for historical purposes only. All major distributions of Linux come with Netscape and Lynx. These are the two most popular browsers for the Linux Operating System. If you do not have these browsers installed currently please go to their respective sections for download information.

The following chapter is dedicated to the setting up web browsers. Please feel free to contact me, if your favorite web browser is not mentioned here. In this version of the document only a few of the browsers have there own section, but I tried to include all of them (all I could find) in the overview section. In the future those browsers that deserve there own section will have it.

The overview section is designed to help you decide which browser to use, and give you basic information on each browser. The detail section is designed to help you install, configure, and maintain the browser.

However I use Lynx when I don't feel like firing up the X-windows/Netscape monster.

2.1 Overview

Navigator/Communicator

Netscape Navigator is the only Linux browser mentioned here, which is capable of advanced HTML features. Some of these features are frames, Java, Javascript, automatic update, and layers. It also has news and mail capability. But it is a resource hog; it takes up lots of CPU time and memory. It also sets up a separate cache for each user wasting disk space.

Netscape is currently an OpenSource product and can be downloaded from ftp.netscape.com=>. <tag><ref id= name="Lynx"> Lynx is the one of the smallest web browsers. It is the king of text based browsers. It's free and the source code is available under the GNU public license. It's text based, but it has many special features. Lynx now supports tables, color (via curses) and frames.

Note on frame support for lynx:

The frame support for lynx is limited, it will notice the frames and show the title of the frames for your to select as hot links. Since, frame titles are usually very undescriptive for coding simplicity this can be confusing.

Kfm

Kfm is part of the K Desktop Environment (KDE). KDE is a system that runs on top of X-windows. It gives you many features like drag an drop, sounds, a trashcan and a unified look and feel. Kfm is the K File Manager, but it is also a web browser. It is very usable as a web browser and it supports frames, tables, ftp downloads, looking into tar files, and more. The current

release of KDE is 1.1.1 with 1.1.2 very near by. Kfm can be used without KDE, but you still n the librarys that come with KDE. For more information about KDE and Kfm visit the KDE website at http://www.kde.org.

TOC

Emacs

Emacs is the one program that does everything. It is a word processor, news reader, mail reader, and web browser. It has a steep learning curve at first, because you have to learn what all the keys do. The X-windows version is easier to use, because most of the functions are on menus. Another drawback is that it's mostly text based. (It can display graphics if you are running it under X-windows). It is also free, and the source code is available under the GNU public license.

NCSA Mosaic

Mosaic is an X-windows browser developed by the National Center for Supercomputing Applications (NCSA) at the University of Illinois. NCSA spent four years on the project and has now moved on to other things. Again, Mosaic is no longer supported. However since the source is free for non-commercial use it might make an interesting project for someone who wants to develop a new browser.

Amaya

Amaya is the X-windows concept browser for the W3C for HTML 3.2. Therefore it supports all the HTML 3.2 standards. It also supports some of the features of HTML 4.0. It supports tables, forms, client side image maps, put publishing, gifs, jpegs, and png graphics. It is both a browser and authoring tool. The latest public release is 1.0 beta. Version 1.1 beta is in internal testing and is due out soon. For more information visit the Amaya web site at http://www.w3.org/Amaya/. It can be downloaded from ftp://ftp.w3.org/pub/Amaya-LINUX-ELF-1.0b.tar.gz.

Qweb

Qweb is yet another basic X-windows browser. It supports tables, forms, and server site image maps. The latest version is 1.3. For more information visit the Qweb website at http://sunsite.auc.dk/qweb/ The source is available from http://sunsite.auc.dk/qweb/qweb-1.3.tar.gz The binaries are available in a Red Hat RPM from http://sunsite.auc.dk/qweb/qweb-1.3-1.i386.rpm

It is the reccomendation of this author that users of web browsers use either Netscape 4.x, Lynx or Netscape 5.xAlpha. They are the only one currently available for Linux that support most features. Personally I suggest Netscape5.xAlpha, even though it is Alpha Software it is quite stable and frankly implements the standards better than the 4.x versions do.

---

3. Lynx

Lynx is one of the smaller (around 600 K executable) and faster web browsers available. It does not eat up much bandwidth nor system resources as it only deals with text displays. It can display on any console, terminal or xterm. You will not need an *X Windows system* or additional system memory to run this little browser.

3.1 Where to get

Most distributions have Lynx in them. Therefore I will not bore you with the details of compiling and installing Lynx.

The latest version is 2.8.2 and can be retrieved from http://www.slcc.edu/lynx/fote/ or from almost any friendly Linux FTP server like ftp://sunsite.unc.edu under /pub/Linux/apps/www/broswers/ or mirror site.

For more information on Lynx try these locations:

Lynx Links

> http://www.crl.com/~subir/lynx.html

Lynx Pages

> http://lynx.browser.org

Lynx Help Pages

> http://www.crl.com/~subir/lynx/lynx_help/lynx_help_main.html (the same pages you get from lynx --help and typing ? in lynx)

Note: The Lynx help pages have recently moved. If you have an older version of Lynx, you will need to change your lynx.cfg (in /usr/lib) to point to the new address(above).

I think the most special feature of Lynx against all other web browsers is the capability for batch mode retrieval. One can write a shell script which retrieves a document, file or anything like that via *http*, *FTP*, *gopher*, *WAIS*, *NNTP* or *file://* - url's and save it to disk. Furthermore, one can fill in data into HTML forms in batch mode by simply redirecting the standard input and using the *-post_data* option.

For more special features of Lynx just look at the help files and the man pages. If you use a special feature of Lynx that you would like to see added to this document, let me know.

4. Emacs-W3

There are several different flavors of Emacs. The two most popular are GNU Emacs and XEmacs. GNU Emacs is put out by the Free Software Foundation, and is the original Emacs. It is mainly geared toward text based terminals, but it does run in X-Windows. XEmacs (formerly Lucid Emacs) is a version that only runs on X-Windows. It has many special features that are X-Windows related (better menus etc).

4.1 Where to get

Most distributions include GNU Emacs.

The most recent GNU emacs is 19.34. It doesn't seem to have a web site. The FTP site is at ftp://ftp.gnu.ai.mit.edu/pub/gnu/.

The latest version of XEmacs is 20.2. The XEmacs FTP site is at ftp://ftp.xemacs.org/pub/xemacs. For more information about XEmacs goto see its web page at http://www.xemacs.org.

Both are available from the Linux archives at ftp://sunsite.unc.edu under /pub/Linux/apps/editors/emacs/

If you got GNU Emacs or XEmacs installed, you probably got the W3 browser running to.

The Emacs W3 mode is a nearly fully featured web browser system written in the Emacs Lisp system. It mostly deals with text, but can display graphics, too - at least - if you run the emacs under the X Window system.

To get XEmacs in to W3 mode, goto the apps menu and select browse the web.

I don't use Emacs, so if someone will explain how to get it into the W3 mode I'll add it to this document. Most of this information was from the original author. If any information is incorrect, please let me know. Also let me know if you think anything else should be added about Emacs.

5. Netscape Navigator/Communicator

5.1 Different versions and options.

Netscape Navigator is the King of WWW browsers. Netscape Navigator can do almost everything. But on the other hand, it is one of the most memory hungry and resource eating program I've ever seen.

There are 3 different versions of the program:

Netscape Navigator includes the web browser, netcaster (push client) and a basic mail program.

Netscape Communicator includes the web browser, a web editor, an advanced mail program, a news reader, netcaster (push client), and a group conference utility.

Netscape Communicator Pro includes everything Communicator has plus a group calendar, IBM terminal emulation, and remote administration features (administrators can update thousands of copies of Netscape from their desk).

In addition to the three versions there are two other options you must pick.

The first is full install or base install. The full install includes everything. The base install includes enough to get you started. You can download the additional components as you need them (such as multimedia support and netcaster). These components can be installed by the Netscape smart update utility (after installing goto help->software updates). At this time the full install is not available for Linux.

The second option is import or export. If you are from the US are Canada you have the option of selecting the import version. This gives you the stronger 128 bit encryption for secure transactions (SSL). The export version only has 40 bit encryption, and is the only version allowed outside the US and Canada.

The latest version of the Netscape Navigator/Communicator/Communicator Pro is 4.6.1. There are two different versions for Linux. One is for the old 1.2 series kernels and one for the new 2.x kernels. If you don't have at least a 2.0 kernel I suggest you upgrade; there are many improvements in the new kernel.

Beta versions are also available. If you try a beta version, they usually expire in a month or so!

You can also try the 5.0 alpha at the Mozilla project located at www.mozilla.org. I suggest this highly, the new Netscape 5 is the best browser I have seen in a long time. It is extremely flexible and mostly stable even for an Alpha release.

5.2 Where to get

The best way to get Netscape software is to go through their web site at http://www.netscape.com/download/. They have menu's to guide you through the selection. When it ask for the Linux version, it is referring to the kernel (most people should be using 2.0 by now). If your not sure which version kernel you have run 'cat /proc/version'. Going through the web site is the only way to get the import versions.

If you want an export version you can download them directly from the Netscape FTP servers. The FTP servers are also more up to date. For example when I first wrote this the web interface did not have the non-beta 4.03 for Linux yet, but it was on the FTP site. Here are the links to the export Linux 2.0 and 2.2 versions:

Netscape Navigator 4.6.1 is at
ftp://ftp.netscape.com/pub/communicator/4.6/shipping/english/unix/linux20/navigator_standalone/navigator-v403-export.x86-unknown-linux2.0.tar.gz

Netscape Communicator 4.6.1 for Linux 2.0 (kernel) is at
ftp://ftp.netscape.com/pub/communicator/4.6/shipping/english/unix/linux20/base_install/communicator-v403-export.x86-unknown-linux2.0.tar.gz

These url's will change as new versions come out. If these links break you can find them by fishing around at the FTP site ftp://ftp.netscape.com/pub/communicator/.

These servers are heavily loaded at times. Its best to wait for off peak hours or select a mirror site. Be prepared to wait, these archives are large. Navigator is almost 8megs, and Communicator base install is 10megs.

5.3 Installing

This section explains how to install version 4 of Netscape Navigator, Communicator, and Communicator Pro.

First unpack the archive to a temporary directory. Then run the `ns-install` script (type `./ns-install`). Then make a symbolic link from the `/usr/local/netscape/netscape` binary to `/usr/local/bin/netscape` (type `ln -s /usr/local/netscape/netscape /usr/local/bin/netscape`). Finally set the system wide environment variable `$MOZILLA_HOME` to `/usr/local/netscape` so Netscape

can find its files. If you are using bash for your shell edit your `/etc/profile` and add the lines:

```
MOZILLA_HOME="/usr/local/netscape"
export MOZILLA_HOME
```

After you have it installed the software can automatically update itself with smart update. Just run Netscape as root and goto help->software updates. If you only got the base install, you can also install the Netscape components from there.

Note: This will not remove any old versions of Netscape, you must manually remove them by deleting the Netscape binary and Java class file (for version 3).

---

6. Setting up WWW server systems

This section contains information on different http server software packages and additional server side tools like script languages for CGI programs etc. There are several dozen web servers, I only covered those that are fully functional. As some of these are commercial products, I have no way of trying them. Most of the information in the overview section was pieced together from various web sites. If there is any incorrect or missing information please let me know.

For a technical description on the http mechanism, take a look at the RFC documents mentioned in the chapter "For further reading" of this HOWTO.

I prefer to use the Apache server. It has almost all the features you would ever need and its free! I will admit that this section is heavily biased toward Apache. I decided to concentrate my efforts on the Apache section rather than spread it out over all the web servers. I may cover other web servers in the future.

6.1 Overview

Cern httpd

This was the first web server. It was developed by the European Laboratory for Particle Physics (CERN). CERN httpd is no longer supported. The CERN httpd server is reported to have some ugly bugs, to be quite slow and resource hungry. The latest version is 3.0. For more information visit the CERN httpd home page at http://www.w3.org/Daemon/Status.html. It is available for download at ftp://sunsite.unc.edu/pub/Linux/apps/www/servers/httpd-3.0.term.tpz (no it is not a typo, the extension is actually .tpz on the site; probably should be .tgz)

NCSA HTTPd

The NCSA HTTPd server is the father to Apache (The development split into two different servers). Therefore the setup files are very similar. NCSA HTTPd is free and the source code is available. This server not covered in this document, although reading the Apache section may give you some help. The NCSA server was once popular, but most people are replacing it with Apache. Apache is a drop in replacement for the NCSA server(same configuration files), and it fixes several shortcomings of the NCSA server. NCSA HTTPd accounts for 4.9% (and falling) of all web servers. (source September 1997 Netcraft survey). The latest version is 1.5.2a. For more information see the NCSA website at http://hoohoo.ncsa.uiuc.edu.

Apache

Apache is the king of all web servers. Apache and its source code is free. Apache is modular, therefore it is easy to add features. Apache is very flexible and has many, many features. Apache and its derivatives makes up 55% of all web domains. There are over 3,928,112 Apache servers in operation (source August 1999 Netcraft survey).

The official Apache is missing SSL, but there are two derivatives that fill the gap. Stronghold commercial product that is based on Apache. It retails for $995; an economy version is available for $495 (based on an old version of Apache). Stronghold is the number two secure server behind Netscape (source C2 net and Netcraft survey). For more information visit the Stronghold website at http://www.c2.net/products/stronghold/. It was developed outside the US, so it is available with 128 bit SSL everywhere.

Apache-SSL is a free implementation of SSL, but it is not for commercial use in the US (RSA has US patents on SSL technology). It can be used for non-commercial use in the US if you link with the free RSAREF library. For more information see the website at http://www.algroup.co.uk/Apache-SSL/.

Netscape Fast Track Server

Fast Track was developed by Netscape, but the Linux version is put out by Caldera. The Caldera site lists it as Fast Track for OpenLinux. I'm not sure if it only runs on Caldera OpenLinux or if any Linux distribution will do (E-mail me if you have the answer). Netscape servers account for 6.11% (and RISING!) of all web servers (source August 1999 http://www.netcraft.com/survey/). The server sells for $295. It is also included with the Caldera OpenLinux Standard distribution which sells for $399 ($199.50 educational). The web pages tell of a nice administration interface and a quick 10 minute setup. The server has support for 40-bit SSL. To get the full 128-bit SSL you need Netscape Enterprise Server. Unfortunately that is not available for Linux :(The latest version available for Linux is 2.0 (Version 3 is in beta, but its not available for Linux yet). To buy a copy goto the Caldera web site at http://www.caldera.com/products/netscape/netscape.html For more information goto the Fast Track page at http://www.netscape.com/comprod/server_central/product/fast_track/

WN

WN has many features that make it attractive. First it is smaller than the CERN, NCSA HTTPd, an Apache servers. It also has many built-in features that would require CGI's. For example site searches, enhanced server side includes. It can also decompress/compress files on the fly with its filter feature. It also has the ability to retrieve only part of a file with its ranges feature. It is released under the GNU public license. The current version is 2.08. For more information see the WN website at http://hopf.math.nwu.edu/.

AOLserver

AOLserver is made by America Online. I'll admit that I was surprised by the features of a web server coming from AOL. In addition to the standard features it supports database connectivity. Pages can query a database by Structured Query Language (SQL) commands. The database is access through Open Database Connectivity (ODBC). It also has built-in search engine and TCL scripting. If that is not enough you can add your own modules through the c Application Programming Interface (API). I almost forgot to mention support for 40 bit SSL. And you get all this for free! For more information visit the AOLserver site at http://www.aolserver.com/server/

CL-HTTP

CL-HTTP stands for Common Lisp Hypermedia Server. If you are a Lisp programmer this server

is for you. You can write your CGI scripts in Lisp. It has a web based setup function. It also supports all the standard server features. CL-HTTP is free and the source code is available. Fo more information visit the CL-HTTP website at http://www.ai.mit.edu/projects/iiip/doc/cl-http/home-page.html

If you have a commercial purpose (company web site, or ISP), I would strongly recommend that you use Apache. I've also heard that the Netscape Server is easy to setup. If you have an internal use you can be a bit more flexible. But unless one of them has a feature that you just have to use,

This is only a partial listing of all the servers available. For a more complete list visit Netcraft at http://www.netcraft.com/survey/servers.html or Web Compare at http://webcompare.internet.com.

---

7. Apache

The current version of Apache is 1.3.9. The main Apache site is at http://www.apache.org/. Another good source of information is Apacheweek at http://www.apacheweek.com/. The Apache documentation is ok, so I'm not going to go into detail in setting up apache. The documentation is on the website and is included with the source (in HTML format). There are also text files included with the source, but the HTML version is better. The documentation should get a whole lot better once the Apache Documentation Project gets under way. Right now most of the documents are written by the developers. Not to discredit the developers, but they are a little hard to understand if you don't know the terminology.

7.1 Where to get

Apache is included in the Red Hat, Slackware, and OpenLinux distributions. Although they may not be the latest version, they are very reliable binaries. The bad news is you will have to live with their directory choices (which are totally different from each other and the Apache defaults).

The source is available from the Apache web site at http://www.apache.org/dist/ Binaries are are also available at apache at the same place. You can also get binaries from sunsite at ftp://sunsite.unc.edu/pub/Linux/apps/www/servers/. And for those of us running Red Hat the latest binary RPM file can usually be found in the contrib directory at ftp://ftp.redhat.com/pub/contrib/i386/

If your server is going to be used for commercial purposes, it is highly recommended that you get the source from the Apache website and compile it yourself. The other option is to use a binary that comes with a major distribution. For example Slackware, Red Hat, or OpenLinux distributions. The main reason for this is security. An unknown binary could have a back door for hackers, or an unstable patch that could crash your system. This also gives you more control over what modules are compiled in, and allows you to set the default directories. It's not that difficult to compile Apache, and besides you not a real Linux user until you compile your own programs ;)

7.2 Compiling and Installing

First untar the archive to a temporary directory. Next change to the src directory. Then edit the Configuration file if you want to include any special modules. The most commonly used modules are already included. There is no need to change the rules or makefile stuff for Linux. Next run the Configure shell script (`./Configure`). Make sure it says Linux platform and gcc as the compiler. Next you may want to edit the httpd.h file to change the default directories. The server home (where the config files are kept) default is `/usr/local/etc/httpd/`, but you may want to change it to just `/etc/httpd/`. And the server root (where the HTML pages are served from) default is `/usr/local/etc/httpd/htdocs/`, but I like the directory `/home/httpd/html` (the Red Hat default for Apache). If you are going to be using su-exec (see special features below) you may want to change that directory too. The server root can also be changed from the config files too. But it is also good to compile it in, just encase Apache can't find or read the config file. Everything else should be changed from the config files. Finally run make to compile Apache.

If you run in to problems with include files missing, check the following things. Make sure you have the kernel headers (include files) installed for your kernel version. Also make sure you have these symbolic links in place:

```
/usr/include/linux should be a link to /usr/src/linux/include/linux
/usr/include/asm should be a link to /usr/src/linux/include/asm
/usr/src/linux should be a link to the Linux source directory (ex.linux-2.0.30)
```

Links can be made with `ln -s`, it works just like the cp command except it makes a link (`ln -s source-dir destination-link`)

When make is finished there should be an executable named httpd in the directory. This needs to be moved in to a bin directory. `/usr/sbin` or `/usr/local/sbin` would be good choices.

Copy the conf, logs, and icons sub-directories from the source to the server home directory. Next rename 3 of the files files in the conf sub-directory to get rid of the `-dist` extension (ex. `httpd.conf-dist` becomes `httpd.conf`)

There are also several support programs that are included with Apache. They are in the `support` directory and must be compiled and installed separately. Most of them can be make by using the makefile in that directory (which is made when you run the main `Configure` script). You don't need any of them to run Apache, but some of them make the administrators job easier.

7.3 Configuring

Now you should have four files in your `conf` sub-directory (under your server home directory). The `httpd.conf` sets up the server daemon (port number, user, etc). The `srm.conf` sets the root document tree, special handlers, etc. The `access.conf` sets the base case for access. Finally `mime.types` tells the server what mime type to send to the browser for each extension.

The configuration files are pretty much self-documented (plenty of comments), as long as you understand the lingo. You should read through them thoroughly before putting your server to work. Each configuration item is covered in the Apache documentation.

The `mime.types` file is not really a configuration file. It is used by the server to translate file extensions into mime-types to send to the browser. Most of the common mime-types are already in the file. Most people should not need to edit this file. As time goes on, more mime types will be added to support new programs. The best thing to do is get a new mime-types file (and maybe a new version of the server) at that time.

Always remember when you change the configuration files you need to restart Apache or send it the SIGHUP signal with `kill` for the changes to take effect. Make sure you send the signal to the parent process and not any of the child processes. The parent usually has the lowest process id number. The process id of the parent is also in the `httpd.pid` file in the log directory. If you accidently send it to one of the child processes the child will die and the parent will restart it.

I will not be walking you through the steps of configuring Apache. Instead I will deal with specific issues, choices to be made, and special features.

I highly recommend that all users read through the security tips in the Apache documentation. It is also available from the Apache website at http://www.apache.org/docs/mics/security_tips.html.

7.4 Hosting virtual websites

Virtual Hosting is when one computer has more than one domain name. The old way was to have each virtual host have its own IP address. The new way uses only one IP address, but it doesn't work correctly with browsers that don't support HTTP 1.1.

My recommendation for businesses is to go with the IP based virtual hosting until most people have browsers that support HTTP 1.1 (give it a year or two). This also gives you a more complete illusion of virtual hosting. While both methods can give you virtual mail capabilities (can someone confirm this?), only IP based virtual hosting can also give you virtual FTP as well.

If it is for a club or personal page, you may want to consider shared IP virtual hosting. It should be cheaper than IP based hosting and you will be saving precious IP addresses.

You can also mix and match IP and shared IP virtual hosts on the same server. For more information on virtual hosting visit Apacheweek at http://www.apacheweek.com/features/vhost.

IP based virtual hosting

In this method each virtual host has its own IP address. By determining the IP address that the request was sent to, Apache and other programs can tell what domain to serve. This is an incredible waste of IP space. Take for example the servers where my virtual domain is kept. They have over 35,000 virtual accounts, that means 35,000 IP addresses. Yet I believe at last count they had less than 50 servers running.

Setting this up is a two part process. The first is getting Linux setup to accept more than one IP address. The second is setting up apache to serve the virtual hosts.

The first step in setting up Linux to accept multiple IP addresses is to make a new kernel. This works best with a 2.0 series kernel (or higher). You need to include IP networking and IP aliasing support. If you need help with compiling the kernel see the kernel howto.

Next you need to setup each interface at boot. If you are using the Red Hat Distribution then this can be done from the control panel. Start X-windows as root, you should see a control panel. Then double click on network configuration. Next goto the interfaces panel and select your network card. Then click alias at the bottom of the screen. Fill in the information and click done. This will need to be done for each virtual host/IP address.

If you are using other distributions you may have to do it manually. You can just put the commands in the `rc.local` file in `/etc/rc.d` (really they should go in with the networking stuff). You need to have a `ifconfig` and `route` command for each device. The aliased addresses are given a sub device of the main one. For example eth0 would have aliases eth0:0, eth0:1, eth0:2, etc. Here is an example of configuring a aliased device:

```
ifconfig eth0:0 192.168.1.57
route add -host 192.168.1.57 dev eth0:0
```

You can also add a broadcast address and a netmask to the ifconfig command. If you have alot of aliases you may want to make a for loop to make it easier. For more information see the IP alias mini howto.

Then you need to setup your domain name server (DNS) to serve these new domains. And if you don't already own the domain names, you need to contact the Internic to register the domain names. See the DNS-howto for information on setting up your DNS.

Finally you need to setup Apache to server the virtual domain correctly. This is in the `httpd.conf` configuration file near the end. They give you an example to go by. All commands specific to that virtual host are put in between the `virtualhost` directive tags. You can put almost any command in there. Usually you set up a different document root, script directory, and log files. You can have almost unlimited number of virtual hosts by adding more `virtualhost` directive tags.

In rare cases you may need to run separate servers if a directive is needed for a virtual host, but is not allowed in the virtual host tags. This is done using the bindaddress directive. Each server will have a different name and setup files. Each server only responds to one IP address, specified by the bindaddress directive. This is an incredible waste of system resources.

Shared IP virtual hosting

This is a new way to do virtual hosting. It uses a single IP address, thus conserving IP addresses for real machines (not virtual ones). In the same example used above those 30,000 virtual hosts would only take 50 IP addresses (one for each machine). This is done by using the new HTTP 1.1 protocol. The browser tells the server which site it wants when it sends the request. The problem is browsers that don't support HTTP 1.1 will get the servers main page, which could be setup to provide a menu of virtual hosts available. That ruins the whole illusion of virtual hosting. The illusion that you have your own server.

The setup is much simpler than the IP based virtual hosting. You still need to get your domain from the Internic and setup your DNS. This time the DNS points to the same IP address as the original domain. Then Apache is setup the same as before. Since you are using the same IP address in the virtualhost tags, it knows you want Shared IP virtual hosting.

There are several work arounds for older browsers. I'll explain the best one. First you need to make your main pages a virtual host (either IP based or shared IP). This frees up the main page for a link list to all your virtual hosts. Next you to make a back door for the old browsers to get in. This is done using the `ServerPath` directive for each virtual host inside the `virtualhost` directive. For example by adding `ServerPath /mysite/` to www.mysite.com old browsers would be able to access the site by www.mysite.com/mysite/. Then you put the default page on the main server that politely tells them to get a new browser, and lists links to all the back doors of all the sites you host on that machine. When an old browser accesses the site they will be sent to the main page, and get a link to the correct page. New browsers will never see the main page and will go directly to the virtual hosts. You must remember to keep all of your links relative within the web sites, because the pages will be accessed from two different URL's (www.mysite.com and www.mysite.com/mysite/).

I hope I didn't lose you there, but its not an easy workaround. Maybe you should consider IP based hosting after all. A very similar workaround is also explained on the apache website at http://www.apache.org/manual/host.html.

If anyone has a great resource for Shared IP hosting, I would like to know about it. It would be nice to know what percent of browsers out there support HTTP 1.1, and to have a list of which browsers and versions support HTTP 1.1.

7.5 CGI scripts

There are two different ways to give your users CGI script capability. The first is make everything ending in `.cgi` a CGI script. The second is to make script directories (usually named `cgi-bin`). You could also use both methods. For either method to work the scripts must be world executable (`chmod 711`). By giving your users script access you are creating a big security risk. Be sure to do your homework to minimize the security risk.

I prefer the first method, especially for complex scripting. It allows you to put scripts in any directory. I like to put my scripts with the web pages they work with. For sites with allot of scripts it looks much better than having a directory full of scripts. This is simple to setup. First uncomment the `.cgi` handler at the end of the `srm.conf` file. Then make sure all your directories have the `option ExecCGI` or `All` in the `access.conf` file.

Making script directories is considered more secure. To make a script directory you use the ScriptAlias directive in the `srm.conf` file. The first argument is the Alias the second is the actual directory. For example `ScriptAlias /cgi-bin/ /usr/httpd/cgi-bin/` would make `/usr/httpd/cgi-bin` able to execute scripts. That directory would be used whenever someone asked for the directory `/cgi-bin/`. For security reasons you should also change the properties of the directory to `Options none, AllowOveride none` in the `access.conf` (just uncomment the example that is there). Also do not make your script directories subdirectories of your web page directories. For example if you are serving pages from `/home/httpd/html/`, don't make the script directory `/home/httpd/html/cgi-bin`; Instead make it `/home/httpd/cgi-bin`.

If you want your users to have there own script directories you can use multiple `ScriptAlias` commands. Virtual hosts should have there `ScriptAlias` command inside the `virtualhost` directive tags. Does anyone know a simple way to allow all users to have a cgi-bin directory without individual ScriptAlias commands?

7.6 Users Web Directories

There are two different ways to handle user web directories. The first is to have a subdirectory under the users home directory (usually `public_html`). The second is to have an entirely different directory tree for web directories. With both methods make sure set the access options for these directories in the `access.conf` file.

The first method is already setup in apache by default. Whenever a request for `/~bob/` comes in it looks for the `public_html` directory in bob's home directory. You can change the directory with the `UserDir` directive in the `srm.conf` file. This directory must be world readable and executable. This method creates a security risk because for Apache to access the directory the users home directory must be world executable.

The second method is easy to setup. You just need to change the `UserDir` directive in the `srm.conf` file. It has many different formats; you may want to consult the Apache documentation for clarification. If you want each user to have their own directory under `/home/httpd/`, you would use `UserDir /home/httpd`. Then when the request `/~bob/` comes

in it would translate to `/home/httpd/bob/`. Or if you want to have a subdirectory under bob's directory you would `UserDir /home/httpd/*/html`. This would translate to `/home/httpd/bob/html/` and would allow you to have a script directory too (for example `/home/httpd/bob/cgi-bin/`).

TOC

7.7 Daemon mode vs. Inetd mode

There are two ways that apache can be run. One is as a daemon that is always running (Apache calls this standalone). The second is from the inetd super-server.

Daemon mode is far superior to inetd mode. Apache is setup for daemon mode by default. The only reason to use the inetd mode is for very low use applications. Such as internal testing of scripts, small company Intranet, etc. Inetd mode will save memory because apache will be loaded as needed. Only the inetd daemon will remain in memory.

If you don't use apache that often you may just want to keep it in daemon mode and just start it when you need it. Then you can kill it when you are done (be sure to kill the parent and not one of the child processes).

To setup inetd mode you need to edit a few files. First in `/etc/services` see if http is already in there. If its not then add it:

```
http    80/tcp
```

Right after 79 (finger) would be a good place. Then you need to edit the `/etc/inetd.conf` file and add the line for Apache:

```
http    stream  tcp     nowait  root    /usr/sbin/httpd httpd
```

Be sure to change the path if you have Apache in a different location. And the second httpd is not a typo; the inet daemon requires that. If you are not currently using the inet daemon, you may want to comment out the rest of the lines in the file so you don't activate other services as well (FTP, finger, telnet, and many other things are usually run from this daemon).

If you are already running the inet deamon (`inetd`), then you only need to send it the SIGHUP signal (via kill; see kill's man page for more info) or reboot the computer for changes to take effect. If you are not running `inetd` then you can start it manually. You should also add it to your init files so it is loaded at boot (the `rc.local` file may be a good choice).

7.8 Allowing put and delete commands

The newer web publishing tools support this new method of uploading web pages by http (instead of FTP). Some of these products don't even support FTP anymore! Apache does support this, but it is lacking a script to handle the requests. This script could be a big security hole, be sure you know what you are doing before attempting to write or install one.

If anyone knows of a script that works let me know and I'll include the address to it here.

For more information goto Apacheweek's article at http://www.apacheweek.com/features/put.

7.9 User Authentication/Access Control

This is one of my favorite features. It allows you to password protect a directory or a file without using CGI scripts. It also allows you to deny or grant access based on the IP address or domain name of the client. That is a great feature for keeping jerks out of your message boards and guest books (you get the IP or domain name from the log files).

To allow user authentication the directory must have `AllowOverrides AuthConfig` set in the `access.conf` file. To allow access control (by domain or IP address) AllowOverrides Limit must be set for that directory.

Setting up the directory involves putting an `.htaccess` file in the directory. For user authentication it is usually used with an `.htpasswd` and optionally a `.htgroup` file. Those files can be shared among multiple `.htaccess` files if you wish.

TOC

For security reasons I recommend that everyone use these directives in there access.conf file:

```
<files ~ "/\.ht">
order deny,allow
deny from all
</files>
```

If you are not the administrator of the system you can also put it in your .htaccess file if AllowOverride Limit is set for your directory. This directive will prevent people from looking into your access control files (.htaccess, .htpasswd, etc).

There are many different options and file types that can be used with access control. Therefore it is beyond the scope of this document to describe the files. For information on how to setup User Authentication see the Apacheweek feature at http://www.apacheweek.com/features/userauth or the NCSA pages at http://hoohoo.ncsa.uiuc.edu/docs-1.5/tutorials/user.html.

7.10 su-exec

The su-exec feature runs CGI scripts as the user of the owner. Normally it is run as the user of the web server (usually nobody). This allows users to access there own files in CGI scripts without making them world writable (a security hole). But if you are not careful you can create a bigger security hole by using the su-exec code. The su-exec code does security checks before executing the scripts, but if you set it up wrong you will have a security hole.

The su-exec code is not for amateurs. Don't use it if you don't know what you are doing. You could end up with a gaping security hole where your users can gain root access to your system. Do not modify the code for any reason. Be sure to read all the documentation carefully. The su-exec code is hard to setup on purpose, to keep the amateurs out (everything must be done manually, no make file no install scripts).

The su-exec code resides in the `support` directory of the source. First you need to edit the `suexec.h` file for your system. Then you need to compile the su-exec code with this command:

```
gcc suexec.c -o suexec
```

Then copy the suexec executable to the proper directory. The Apache default is `/usr/local/etc/httpd/sbin/`. This can be changed by editing `httpd.h` in the Apache source and recompiling Apache. Apache will only look in this directory, it will not search the path. Next the file needs to be changed to user root (`chown root suexec`) and the suid bit needs to be set (`chmod 4711 suexec`). Finally restart Apache, it should display a message on the console that su-exec is being used.

CGI scripts should be set world executable like normal. They will automaticaly be run as the owner of the CGI script. If you set the SUID (set user id) bit on the CGI scripts they will not run. If the directory or file is world or group writable the script will not run. Scripts owned by system users will not be run (root, bin, etc.). For other security conditions that must be met see the su-exec documentation. If you are having problems see the su-exec log file named `cgi.log`.

Su-exec does not work if you are running Apache from inetd, it only works in daemon mode. It will be fixed in the next version because there will be no inetd mode. If you like playing around in source code, you can edit the http_main.c. You want to get rid of the line where Apache announces that it is using the su-exec wrapper (It wrongly prints this in front of the output of everything).

Be sure and read the Apache documentation on su-exec. It is included with the source and is available on the Apache web site at http://www.apache.org/docs/suexec.html

TOC

7.11 Imagemaps

Apache has the ability to handle server side imagemaps. Imagemaps are images on webpages that take users to different locations depending on where they click. To enable imagemaps first make sure the imagemap module is installed (its one of the default modules). Next you need to uncomment the `.map` handler at the end of the `srm.conf` file. Now all files ending in `.map` will be imagemap files. Imagemap files map different areas on the image to separate links. Apache uses map files in the standard NCSA format. Here is an example of using a map file in a web page:

```
<a href="/map/mapfile.map">
<img src="picture.gif" ISMAP>
</a>
```

In this example `mapfile.map` is the mapfile, and `picture.gif` is the image to click on.

There are many programs that can generate NCSA compatible map files or you can create them yourself. For a more detailed discussion of imagemaps and map files see the Apacheweek feature at http://www.apacheweek.com/features/imagemaps.

7.12 SSI/XSSI

Server Side Includes (SSI) adds dynamic content to otherwise static web pages. The includes are embedded in the web page as comments. The web server then parses these includes and passes the results to the web server. SSI can add headers and footers to documents, add date the document was last updated, execute a system command or a CGI script. With the new eXtended Server Side Includes (XSSI) you can do a whole lot more. XSSI adds variables and flow control statements (if, else, etc). Its almost like having an programming language to work with.

Parsing all HTML files for SSI commands would waste allot of system resources. Therefore you need to distinguish normal HTML files from those that contain SSI commands. This is usually done by changing the extension of the SSI enhanced HTML files. Usually the `.shtml` extension is used.

To enable SSI/XSSI first make sure that the includes module is installed. Then edit `srm.conf` and uncomment the `AddType` and `AddHandler` directives for `.shtml` files. Finally you must set `Options Includes` for all directories where you want to run SSI/XSSI files. This is done in the `access.conf` file. Now all files with the extension `.shtml` will be parsed for SSI/XSSI commands.

Another way of enabling includes is to use the `XBitHack` directive. If you turn this on it looks to see if the file is executable by user. If it is and `Options Includes` is on for that directory, then it is treated as an SSI file. This only works for files with the mime type text/html (`.html .htm` files). This is not the preferred method.

There is a security risk in allowing SSI to execute system commands and CGI scripts. Therefore it is possible to lock that feature out with the `Option IncludesNOEXEC` instead of Option Includes in the `access.conf` file. All the other SSI commands will still work.

For more information see the Apache mod_includes documentation that comes with the source. It is also available on the website at http://www.apache.org/docs/mod/mod_include.html.

For a more detailed discussion of SSI/XSSI implementation see the Apacheweek feature at http://www.apacheweek.com/features/ssi.

For more information on SSI commands see the NCSA documentation at http://hoohoo.ncsa.uiuc.edu/docs/tutorials/includes.html.

For more information on XSSI commands goto ftp://pageplus.com/pub/hsf/xssi/xssi-1.1.html.

7.13 Module system

Apache can be extended to support almost anything with modules. There are allot of modules already in existence. Only the general interest modules are included with Apache. For links to existing modules goto the

Apache Module Registry at http://www.zyzzyva.com/module_registry/.

For module programming information goto http://www.zyzzyva.com/module_registry/reference/

---

8. Web Server Add-ons

Sorry this section has not been written yet.

Coming soon: mSQL, PHP/FI, cgiwrap, Fast-cgi, MS frontpage extentions, and more.

9. Intranet Section

In simple terms, the **Intranet** is the descriptive term being used for the implementation of Internet technologies within a corporate organisation, rather than for external connection to the global Internet. This implementation is performed in such a way as to transparently deliver the immense informational resources of an organisation to each individuals desktop with minimal cost, time and effort. This document attempts to explain in simple terms how to setup an Intranet using tools which are readily available and are generally costing little or are free.

This document assumes that you already know how to install TCP/IP on your Linux server and connect it physically to your LAN using an Ethernet network card. This also assumes you have some basic knowledge of Netware, WinNT and Mac systems. The configuration of the Netware server has been shown using version 3.1x as the basis. You can also use INETCFG to achieve the same result. On the client side the discussion is with respect to Windows 3.1x, Windows for Workgroups and Win95, WinNT and the Apple PowerMac.

I am using the private network addresses (RFC-1918) of 172.16.0.0 and 172.17.0.0 only as examples. You may choose suitable addresses depending on your configuration.

```
     Linux        Netware       WFWG/WinNT
     Server       Server        Server
    172.16.0.1   172.16.0.2    172.16.0.3
       |            |            |    172.16.0.0
------+-----+-----+--------+--+--------------
            |               | 172.16.0.254
          W/S 1           Router
        172.16.0.5          | 172.17.0.254
                            |
                  ----------+-------+--------
              172.17.0.0            |
                                  W/S 2
                                172.17.0.5
```

9.1 What is required

You will need the following software before attempting the installation.

- the HTTP server software which can be downloaded from OneStep NCSA HTTPd Downloader at http://hoohoo.ncsa.uiuc.edu/docs/setup/OneStep.html page.
- The Novell Netware Client available from HTTP://support.novell.com/ (The TCP/IP files are included with the client).

- The Microsoft TCP/IP client available from HTTP://www.microsoft.com/
- The Apple MacTCP client available from HTTP://www.apple.com/
- WWW Browsers like Netscape at HTTP://home.netscape.com/ or MS Internet Explorer at HTTP://www.microsoft.com/ or NCSA Mosaic from http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html

9.2 New versions of this document

New versions of the Linux Intranet Server HOWTO will be periodically posted to comp.os.linux.announce and comp.os.linux.help. They will also be uploaded to various Linux FTP sites, including sunsite.unc.edu.

The Latest version of this document is available in HTML format at http://www.linuxports.com/

9.3 Feedback

If you have questions or comments about this document, please feel free to mail Mr. Poet, at poet@linuxports.com. Suggestions, criticism and mail are always welcome. If you find a mistake with this document, please let me know so I can correct it in the next version. Thanx.

---

10. Install the HTTP server

When you download the server you have two options: To get the source and compile it yourself, or get the precompiled binaries. The precompiled binaries for Linux (ELF) version are available at NCSA but not the older versions.

10.1 Preparation before downloading

The server at NCSA will guide you through the steps for configuration options and prepare the various files for you. But before you attempt to download HTTPd be prepared with answers to the following questions

The Operating System

First, you must choose whether to download the source or a pre-compiled version of the software. If your particular system doesn't appear in the menu, then you will have to get the default source, and compile it yourself.

To check the version of your Linux go to the command prompt on your Linux machine and type

```
linux:~$  uname -a
```

which will respond with a line which looks similar to this

```
linux:~$  uname -a
Linux linux 2.0.29 #4 Tue Sep 13 04:05:51 CDT 1994 i586
linux:~$
```

The version of Linux is 2.0.29.

The remaining parameters can be specified before downloading or configured later by modifying the file `srm.conf` in the `/usr/local/etc/httpd/conf` directory. The names of the actual directives that appear in the file `httpd.conf` are shown in brackets. The only exception is the directive DocumentRoot which appears in the file `srm.conf`

Process type (ServerType)

This specifies how your machine will run your HTTPd server. The preferred method is "standalone". This makes the HTTP daemon to be running constantly. If you choose to load HTTPd under "inetd", the server binary will be reloaded into memory for every request, which may slow your server down.

TOC

Binding Port (Port)

This specifies which port of your machine that the HTTPd daemon will bind to and listen for HTTP requests. If you can login as "root", use the default setting of 80. Otherwise choose a setting between 1025 and 65535.

Server user identity (User)

This is the user id the server will change to when answering requests and acting on files.This question needs to be answered only if you are running the server as "standalone". If you are someone without root permissions, just use your own login name. If you are system administrator, you might want to create a special user so you can control file permissions.

Server group identity (Group)

This is the group id the server will change to when answering requests and acting on files. This is similar to Server User identity and is applicable only if you are running the server as standalone.

If you do not have root permissions, just use the name of your primary group. You can find out your group by typing **groups** at the Linux command prompt.

Server administrator email address (ServerAdmin)

This is the email address that the user should send an email message to when reporting a problem with the server. You can put your personal e-mail address.

Location of server directory (ServerRoot)

This is where the server resides on your system. If you have root permissions leave it in its recommended location `/usr/local/etc/httpd`. If you cannot login as root, choose a subdirectory in your home path. You can find out the path of your home directory with the **pwd** command.

Location of HTML files (DocumentRoot)

This is where the HTML files to be served are located. The default location is `/usr/local/etc/httpd/htdocs`. You could however set it to be the home directory of the special user you chose in Server user identity, or a subdirectory in your home directory if you can't login as root.

When in doubt, use the default settings. Now that you have answers to the above questions you can Download NCSA HTTPd at http://hoohoo.ncsa.uiuc.edu/docs/setup/OneStep.html. You should read the HTTPd Documentation at http://hoohoo.ncsa.uiuc.edu/docs/ before you attempt installation. If you are planning to compile the code then you need to modify the makefiles in each of the th ree directories `support, src, cgi-src`. If your version of Linux is already supported then you just have to type **make linux** at the top level directory (i.e. `/usr/local/etc/httpd`)

10.2 Compiling HTTPd

TOC

Compiling is simple, just type `make linux` at the prompt in the server root directory. **Note:** Users of pre-ELF Linux have to uncomment `#define NO_PASS` in file `portability.h` and set `DBM_LIBS= -ldbm` in the `Makefile` before compiling HTTPd.

---

11. Testing HTTPd

After you have installed HTTPd, login as root and start it by typing **httpd &** . (assuming you have installed as standalone) You should now be able to see it in the list generated by **ps**. The simplest way to test HTTPd is by Telnet. At the Linux command prompt type

```
linux:~$  telnet 172.16.0.1 80
```

where 80 is the default port for HTTP. If you have configured "Port" as something different then type that number instead. You should get a response which looks like this

```
Trying 172.16.0.1...
Connected to linux.mydomain.
Escape character is '^]'.
```

Now if you type in any character and press Enter you should get a response similar to the one shown below.

```
HTTP/1.0 400 Bad Request
Date: Wed, 10 Jan 1996 10:24:37 GMT
Server: NCSA/1.5
Content-type: text/html

<HEAD><TITLE>400 Bad Request < /TITLE> < /HEAD>
<BODY><H1>400 Bad Request < /H1>
Your client sent a query that this server could
not understand.<P>
Reason: Invalid or unsupported method.<P>
< /BODY>
```

Now we are ready to connect to this server using another PC and a WWW Browser.

12. Connecting to the Linux Server

Please refer to the diagram shown in the chapter <@@ref>IntroIntroduction for the addressing scheme used. Workstation 1 (W/S1) is on network 172.16.0.0 and can access the Linux server directly whereas Workstation 2 (W/S2) is on network 172.17.0.0 and needs to use the gateway (router) 172.17.0.254 to access the Linux box. This gateway information needs to be provided while configuring the clients only on W/S2. Netware refers to the gateway as 'ip_router'.

I am using W/S2 to illustrate the client setup. To setup W/S1 just change the address 172.17.0.5 to 172.16.0.5 and ignore all references to the gateway/router.

If you do not have a router you can skip the next section and proceed to

- Setup Netware Server if you use a Netware server.
- Setup MS Windows Client if you use the Microsoft Client.

12.1 Setup the Linux server

You may skip this section if you do not have a router.

You have to configure the Linux server to recognise the router thus allowing Workstation 2 to connect to the Web server. In order to setup the Linux server you should login as root. At the server prompt type

```
route add gw default 172.16.0.254
```

To use this gateway everytime you boot the Linux server edit the file `/etc/rc.d/rc.inet1` and change the line containing the gateway definition to `GATEWAY = "172.16.0.254"`. Make sure the line for adding the gateway is not commented out.

ALT: You can add routes to the networks on the other side of the router. This would be done as

```
route add -net 172.17.0.0 gw 172.16.0.254
```

To add this route everytime you boot Linux add the command to your `/etc/rc.d/rc.local` file.

12.2 Setup the Netware server

In order to setup the Netware server you should have Supervisor permissions or atleast Console operator permissions. If these cannot be got, try asking your Network Administrator to help you with the setup. At the Server enable the Ethernet_II frame type on the LAN by typing these commands or include them in the AUTOEXEC.ncf file.

```
load NE2000 frame=Ethernet_II name=IPNET
load TCPIP
bind IP to IPNET addr=172.16.0.2 mask=FF.FF.FF.0
```

You might have to specify the slot or board number while loading the NE2000 driver depending on your machine configuration. (eg: load NE2000 slot=3 frame=.....)

12.3 Setup the Netware Client

TOC

On the PC you have the choice of Win3.1,WFWG or Win95. The installation procedure differs between Win95 and the older windows if you are using the 32bit client from Microsoft or Novell. If you are going to use the 16bit client, the procedure is the same and you can refer to the Windows 3.x installation instructions. For installing the 32bit client for Win95 skip to Windows 95 installation.

Windows 3.x

If you are using Win3.1 or WFWG you can install the Netware Client (VLMs) and some additional files which are provided with the TCP/IP diskette, namely

TCPIP.exe, VTCPIP.386, WINSOCK.dll and WLIBSOCK.dll

Note that the WINSOCK.dll file is different from the ones provided with Win95 and Trumpet. Install the Netware Client with the support for windows. Copy VTCPIP.386, WINSOCK.dll and WLIBSOCK.dll to the SYSTEM directory and TCPIP.exe to the NWCLIENT directory. Now modify the STARTNET.bat in the NWCLIENT directory to

```
lsl
ne2000     ---> your network card driver
c:\windows\odihlp.exe     ---->if you are using WFWG
ipxodi
tcpip        ---> add this line
nwip         ---> if you use Netware/IP
vlm
```

Create a subdirectory (say) `\NET\TCP` and copy the files HOSTS, NETWORKS, PROTOCOLS and SERVICES from `/etc` on your Linux server or the directory `SYS:ETC` on your Netware server. Edit the copied HOSTS file to add the line for your new Linux server. This will enable you to refer to the Linux server as `http://linux.mydomain/` instead of `http://172.16.0.1/`in your WWW browser

```
127.0.0.1      localhost
172.16.0.1      linux.mydomain
```

Edit the NET.cfg file in NWCLIENT directory

```
Link Driver NE2000
    port 300
    int 3
    MEM D0000
    FRAME Ethernet_802.2

; ---- add these lines ----

   FRAME Ethernet_II

Protocol TCPIP
   PATH TCP_CFG C:\NET\TCP
   ip_address  172.17.0.5
   ip_netmask  255.255.255.0
   ip_router   172.17.0.254     ---> add the address of your gateway only
                                ---> if you have to use this
```

```
                                        ---> gateway to reach your HTTP server

      Link Support
         MemPool 6192       ---> the minimum is 1024. Try with different values
         Buffers 10 1580    ---> this again can be fine tuned

   ;-----------------------------------
   ; You may need to add lines like these if you are using Netware/IP
   ;
      NWIP
         NWIP_DOMAIN_NAME  mydomain
         NSQ_BROADCAST   ON
         NWIP1_1 COMPATIBILITY   OFF
         AUTORETRIES     1
         AUTORETRY SECS  10
```

Edit the SYSTEM.ini file in the WINDOWS directory and add this entry for VTCPIP.386

```
      [386Enh]
      .....
      network=*vnetbios, vipx.386, vnetware.386, VTCPIP.386
      .....
```

Reboot your PC, run STARTNET.bat and you can now use your favorite WWW browser to access your Web pages. You need not login to Netware and you don't have to run TCPMAN (if you use Trumpet Winsock).

Windows 95

This section explains how to install the 32bit client on Win95. Firstly you must install the following

```
      Client for Netware Networks (from Microsoft or Novell)
      Microsoft TCP/IP Protocol
      Network Adapter
```

To install these items, click on My Computer, Control Panel, Networks. Click Add. You will now be in a window that displays Client, Adapter, Protocol and Service. To install the Client for Netware Networks:

```
     1. DoubleClick on Client
     2. Click on Microsoft or Novell
     3. DoubleClick on Client for Netware Networks
```

To install the TCP/IP Protocol:

```
     1. DoubleClick on Protocol
     2. Click on Microsoft
     3. DoubleClick on TCP/IP
```

Windows 95 by default installs several other protocols automatically. Remove them by clicking on them and clicking the Remove button. Typically Win95 installs the Microsoft NetBeui protocol, and IPX/SPX compatible protocol. You can delete the NetBEUI protocol, but you will need the IPX/SPX protocol if you wish to login to the Netware Server.

To setup TCP/IP click on TCP/IP, click on Properties, click on the tab IP address

```
Enter your IP address in the "Specify an IP address "
                      box as 172.17.0.5
In the Subnet Mask box enter 255.255.255.0
```

select the tab Gateway

```
Enter your gateway (router) address in the box New gateway
   as 172.17.0.254
Click the Add button
```

The gateway address should now appear under the installed gateways box. Now Click OK.

You should get a message to reboot. Do so. You should now be able to use the Browser to connect to your HTTP Server.

12.4 Setup Microsoft Client

If you are using the Microsoft Client for accessing your network, then this section details how to install TCP/IP for

- Windows for Workgroups (WFWG)
- Windows 95
- Windows NT 4.0

Note: To enable you to refer to the Linux server as `http://linux.mydomain/` instead of `http://172.16.0.1/` in the WWW browser and all your intranet commands you need to edit the `hosts` file. You can add more entries for each of your other hosts (Netware, Unix, WinNT) as well. The Windows family keeps its HOSTS file in `\WINDOWS` or in `\WINDOWS\SYSTEM` depending on the version. Edit this file and add a line for your Linux server as:

```
127.0.0.1       localhost
172.16.0.1      linux.mydomain

172.16.0.2      netware.mydomain
172.16.0.3      winNT.mydomain
172.16.0.5      ws_1
```

Windows for Workgroups

This section explains how to install the 32bit client on WFWG. Firstly you must download the TCP/IP drivers for Windows from Microsoft. The current version is 3.11b and is available at ftp://ftp.microsoft.com or other sites as `tcp32b.exe`. Make sure that you have load Win32s before trying to load the TCP/IP-32bit driver.

Having expanded the TCP/IP files into a temporary directory (say `C:\TEMP`), check your `\WINDOWS\SYSTEM` directory for copies of `OEMSETUP.INF`. If there are any, rename them. Now copy the `OEMSETUP.INF` file from the TEMP directory to the `\WINDOWS\SYSTEM` directory. If you have loaded any other TCP/IP stacks on your system, please remove them before you proceed.

Start Network Setup or Windows Setup/Change Network settings

```
Click the Networks button
Click Install Microsoft Windows Network.
  Choose support for additional networks (if required)
Click OK
```

TOC

You should be prompted for your network adapter - select the appropriate one. If you are not prompted, then

```
Click the Adapter button
    select an adapter (say NE2000)
    Click OK
Click the Protocol button
    select the MS TCP/IP-32 protocol
    click OK
```

You will now be prompted to configure the TCP/IP protocol stack. You can always reconfigure this by highlighting the TCP/IP protocol shown in the box Adapters and clicking the Setup button.

```
In the IP address box enter 172.17.0.5
In the Subnet Mask box enter 255.255.255.0
Enter your gateway (router) address in the box default gateway
  as 172.17.0.254
```

Click OK. The computer will ask you to restart. Do so. You should now be able to use the Browser to connect to your HTTP Server.

Windows 95

This section explains how to install the 32bit client for Microsoft on Win95. Firstly you must install the following

```
Client for Microsoft Networks
Microsoft TCP/IP Protocol
Network Adapter
```

To install these items, click on My Computer, Control Panel, Networks. Click Add. You will now be in a window that displays Client, Adapter, Protocol and Service. To install the Client for Microsoft Networks:

```
1. DoubleClick on Client
2. Click on Microsoft
3. DoubleClick on Client for Microsoft Networks
```

To install the TCP/IP Protocol:

```
1. DoubleClick on Protocol
2. Click on Microsoft
3. DoubleClick on TCP/IP
```

Windows 95 by default installs several protocols automatically. Remove them by clicking on them and clicking the Remove button. Typically Win95 installs the Microsoft NetBeui protocol.

To setup TCP/IP click on TCP/IP, click on Properties, click on the tab IP address

```
Enter your IP address in the "Specify an IP address "
                      box as 172.17.0.5
In the Subnet Mask box enter 255.255.255.0
```

select the tab Gateway

```
Enter your gateway (router) address in the box New gateway
   as 172.17.0.254
```

```
Click the Add button
```

The gateway address should now appear under the installed gateways box. Now Click OK.

You should get a message to reboot. Do so. You should now be able to use the Browser to connect to your HTTP Server.

Windows NT

This section details how to Install the TCP/IP client for WinNT 4.0. Start Control Panel/ Network

```
Select the Adapter tab.
    Click Add to add a new adapter (if you don't have one)
```

You should be prompted for your network adapter - select the appropriate one. To add the protocols.

```
Select the Protocols tab
    Click Add
    Select the TCP/IP protocol
    Click OK
```

You will now be prompted to configure the TCP/IP protocol stack. You can always reconfigure this by highlighting the TCP/IP protocol and clicking the Properties button.

```
Select the tab IP Address
   Mark the checkbox 'Specify an IP address'
   In the IP address box enter 172.17.0.5
   In the Subnet Mask box enter 255.255.255.0
   Enter your gateway (router) address in the box Default Gateway
       as 172.17.0.254
```

Click OK. The computer will ask you to restart. You can now use any Browser to connect to your HTTP Server.

12.5 Setup TCP/IP on Macintosh

If you are using the Macintosh for accessing your network, then this section details how to install MacTCP for the PowerMacs.

Note: To enable you to refer to the Linux server as `http://linux.mydomain/` instead of `http://172.16.0.1/` in the WWW browser and all your intranet commands you need to edit the `hosts` file. The format of the hosts file is different from the one used in Unix. The Mac hosts file is based on RFC-1035. You can add more entries for each of your other hosts (Netware, Unix, WinNT) as well. The MacOS keeps its HOSTS file in the `Preferences folder` under the `System folder`. Edit this file and add a line for your Linux server as:

```
linux.mydomain    A  172.16.0.1

netware.mydomain  A  172.16.0.2
winNT.mydomain    A  172.16.0.3
ws_1              A  172.16.0.5
```

MacTCP

This section explains how to install MacTCP. Firstly you must download the MacTCP files from Apple or install it from the Internet Connection CD. To configure MacTCP, click the Apple Menu/ Control Panels/ TCP/IP. In the screen change the setting for 'Connect via:' to 'Ethernet'

Change the 'Configure' setting to 'Manually'

```
In the IP address box enter 172.17.0.5
In the Subnet Mask box enter 255.255.255.0
Enter your gateway (router) address in the box
  Router address as 172.17.0.254
```

Click OK. You should now be able to use the Browser to connect to your HTTP Server.

13. Setting up the Intranet

An Intranet cannot be complete without sharing the resources on the different platforms. You will need support for other filesystems, so that you can access the data available on them. This document provides instructions to connect Linux to the following popular filesystems.

- NCP filesystem for Netware
- SMB filesystem for Windows
- NFS filesystem for Unix

These filesystems can be compiled into the Linux kernel or added as modules, depending on the version of Linux. If you are not familiar with compiling the kernel you can refer to the Kernel HOWTO http://sunsite.unc.edu/mdw/HOWTO/Kernel-HOWTO.html and the Module HOWTO http://sunsite.unc.edu/mdw/HOWTO/Module-HOWTO.html for compiling the kernel with modules.

13.1 NCPFS

To share the files on the Netware server you will need support for NCP (ncpfs). NCPFS works with kernel version 1.2.x and 1.3.71 upwards. It does not work with any earlier 1.3.x kernel. It cannot access the NDS database in Netware 4.x, but can make use of the bindery. If you are using Netware 4.x you can enable bindery support for specific containers using the command `Set Bindery Context` at the console as:

```
set Bindery Context = CORP.MYDOM;WEBUSER.MYDOM
```

In the above example two containers have bindery support enabled.

You will need to download the NCP filesystem utilities using the URL ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/ncpfs.tgz (currently ncpfs-2.0.10) from Sunsite.

Installation

To install the ncpfs utilities, type

```
zcat ncpfs.tgz | tar xvf -
```

to expand the files into its own directory. In this case you will get a directory `ncpfs-2.0.10` Change your directory to this ncpfs directory before proceeding with the installation. Read the README and edit the Makefile if necessary.

The installation of ncpfs depends on the kernel version you are using. For kernel 1.2, you should simply type 'make'. Subsequently typing 'make install' will install the executables and man pages.

If you use Kernel 1.3.71 or later, you might have to recompile your kernel. With these kernels, the kernel part of ncpfs is already included in the main source tree. To check if the kernel needs to be recompiled type

```
cat /proc/filesystems
```

It should show you a line saying that the kernel knows ncpfs.

If ncpfs is not there, you can either recompile the kernel or add ncpfs as a module. For recompiling the kernel you should type 'make config' and when it asks you for

```
The IPX protocol (CONFIG_IPX) [N/y/?]
```

simply answer 'y'. Probably you do not need the full internal net that you are asked for next. Once the kernel is successfully installed, reboot, check `/proc/filesystems` and if everything is OK proceed with the installation of the ncpfs utilities. Change directory to the location holding your downloaded ncpfs files, and type 'make'. After the compilation is finished type 'make install' to install the various utilities and man pages.

Mounting NCPFS

To check the installation type

```
        ipx_configure --auto_interface=on --auto_primary=on

    ....wait for 10 seconds and type

        slist
```

You should be able to see a list of your Netware servers. Now we are ready to share files from the Netware server.

Suppose we need to access HTML files from directory `\home\htmldocs` on volume VOL1: on the server MYDOM_NW, I recommend that you create a new user (say) 'EXPORT' with password 'EXP123' on this server to whom you grant appropriate access rights to this directory using SYSCON or NWADMIN.

On the Linux machine create a new directory `/mnt/MYDOM_NW`. Now type the command

```
        ncpmount -S MYDOM_NW -U EXPORT -P EXP123 /mnt/MYDOM_NW
```

to mount the netware file system. Typing the command

```
        ls /mnt/MYDOM_NW/vol1/home/htmldocs
```

will show you a list of all the files in `MYDOM_NW/VOL1:\HOME\HTMLDOCS` (using Netware file notation). If you have any problems please read the IPX HOWTO at http://sunsite.unc.edu/mdw/HOWTO/IPX-HOWTO.html for more insights into the IPX system.

13.2 SMBFS

To share the files on the Windows server you will need support for SMB (smbfs).

You will need to download the SMB filesystem utilities from ftp://sunsite.unc.edu/pub/Linux/system/filesystems/smbfs/smbfs.tgz (currently smbfs-2.0.1) from Sunsite.

Installation

To install the smbfs utilities, type

```
        zcat smbfs.tgz | tar xvf -
```

to expand the files into its own directory. In this case you will get a directory `smbfs-2.0.1` Change your directory to this smbfs directory before proceeding with the installation. Read the README and edit the Makefile if

necessary.

The installation of smbfs depends on the kernel version you are using. For kernel 1.2, you should simply type 'make'. Subsequently typing 'make install' will install the executables and man pages.

If you use Kernel 2.0 or later, you might have to recompile your kernel. With these kernels, the kernel part of smbfs is already included in the main source tree. To check if the kernel needs to be recompiled type

```
cat /proc/filesystems
```

It should show you a line saying that the kernel knows smbfs.

If smbfs is not there, you can either recompile the kernel or add smbfs as a module. For recompiling the kernel you should type 'make config' and when it asks you for adding SMB filesystem support simply answer yes. Once the kernel is successfully installed, reboot, check `/proc/filesystems` and if everything is OK proceed with the installation of the smbfs utilities. Change directory to the location holding your downloaded smbfs files, and type 'make'. After the compilation is finished type 'make install' to install the various utilities and man pages.

Mounting SMBFS

In our example let us assume that the WinNT server is called 'MYDOM_NT' and is sharing its directory `C:\PUB\HTMLDOCS` with a share name of 'HTMLDOCS' without a password. On the Linux machine create a new directory `/mnt/MYDOM_NT`. Now type the command

```
smbmount //MYDOM_NT/HTMLDOCS /mnt/MYDOM_NT -n
```

to mount the SMB (windows share) file system. If this does not work try

```
smbmount //MYDOM_NT/COMMON /mnt/MYDOM_NT -n -I 172.16.0.3
```

Typing the command

```
ls /mnt/MYDOM_NT
```

will show you a list of all the files in `bsol;bsol;MYDOM_NT\PUB\HTMLDOCS` (using Windows file notation).

13.3 NFS

First you will need a kernel with the NFS file system either compiled in or available as a module.

Suppose you have a Unix host running NFS with the name MYDOM_UNIX and an IP address of 172.16.0.4. You can check the directories that are being exported (shared) by this host by typing the command

```
showmount -e 172.16.0.4
```

Once we know the exported directories you can mount them by entering a appropriate mount command. I recommend that you create a subdirectory under '/mnt' (say) 'MYDOM_UNIX' and use that as your mount point.

```
mount -o rsize=1024,wsize=1024 172.16.0.4:/pub/htmldocs /mnt/MYDOM_UNIX
```

The rsize and wsize may have to be changed depending on your environment.

If you have any problems please read the NFS HOWTO at http://sunsite.unc.edu/mdw/HOWTO/NFS-HOWTO for more insights into the NFS system.

14. Accessing the Web

Now that we have setup the HTTP server, the clients and interconnected the Linux server with the other servers, we need to make some small adjustments on the Linux server to be able to access these mounted filesystems from the Web Browser.

14.1 Accessing the mounted filesystems

To access the mounted directories in your HTML pages you have two methods:

- Create a link in DocumentRoot (`/usr/local/etc/httpd/htdocs`) to refer to the mounted directory as

```
ln -s /mnt/MYDOM_NW/vol1/home/htmldocs netware
                  or

ln -s /mnt/MYDOM_NT    winNT
                  or

ln -s /mnt/MYDOM_UNIX    unix
```

- to edit the file `srm.conf` in your `/usr/local/etc/httpd/conf` directory and add a new alias.

```
# Alias fakename realname
Alias /icons/     /usr/local/etc/httpd/icons/

# alias for netware server
Alias /netware/   /mnt/MYDOM_NW/vol1/home/htmldocs/
Alias /winNT/     /mnt/MYDOM_NT/
Alias /unix/      /mnt/MYDOM_UNIX
```

And restart your HTTPd. You can access the documents on the netware server by referring to them as `http://linux.mydomain/netware/index.htm` for the netware files and similar notations for the others.

14.2 Connecting to the Internet

You can finally connect your Intranet to the Internet to access E-Mail and all the wonderful information out there. I propose to write a brief note on how to do this in a future revision. Detailed explanations are available in the ISP Hookup HOWTO from http://sunsite.unc.edu/mdw/HOWTO/ISP-Hookup-HOWTO.html and Diald mini HOWTO at

http://sunsite.unc.edu/mdw/HOWTO/mini/Diald for setting up these connections.

TOC

14.3 Other uses

The HTTP server can be used in the office to provide transparent access to information residing on different servers, at several locations and directories. The data can be simple documents in Word, Lotus spreadsheets, or complex databases.

The application of this technology is being typically used as follows:-

- Publishing corporate documents

  These documents can include newsletters, annual reports, maps, company facilities, price lists, product information literature, and any document which is of value within the corporate entity.
- Access into searchable directories

  Rapid access to corporate phone books and the like. This data can be mirrored at a Web site or, via CGI scripts, the Web server can serve as a gateway to back-end pre-existing or new applications. This means that, using the same standard access mechanisms, information can be made more widely available and in a simpler manner. This means that it can be used to create an interface with RDBMS like ORACLE and SYBASE for generating real-time information. Here is a list of links to such sites on the Web.

  - Web Access - http://cscsun1.larc.nasa.gov/~beowulf/db/web_access.html - CGI gateways - HTTP://www.w3.org/hypertext/WWW/RDBGate/Overview.html
- Corporate/Department/Individual pages

  As cultures change within organistions to the point where even each department moves towards their own individual mission statements, the Intranet technology provides the ideal medium to communicate current information to the Department or Individual. Powerful search engines provide the means for people to find the group or individual who has the answers to the continuous questions which arise in the normal day-to-day course of doing business.
- Simple Groupware applications

  With HTML forms support, sites can provide sign-up sheets, surveys and simple scheduling.
- Software distribution

  Administrators can use the Intranet to deliver software and up-dates on-demand to users across the corporate network . This can be done with 'Java' which allows the creation and transparent distribution of objects on-demand rather than just data or applications. This is indeed possible more easily with the newer versions of Linux which has builtin support for Java.
- Mail

  With the move to the use of Intranet mail products with standard and simple methods for attachment of documents, sound, vision and other multimedia between individuals, mail is being pushed further forward as a simple, de facto communications method. Mail is essentially individual to individual, or individual to small group, communication. Several utilities are

TOC

available on the Linux platform to setup an E-mail system like **sendmail, pop3d, imapd**.

- User Interface

  The Intranet technology is evolving so rapidly that the tools available, in particular HTML, can be used to dramatically change the way we interface with systems. With HTML you can build an Interface which is only limited by the creators imagination. The beauty about using Intranet technologies for this is that it is so simple. Clicking a hyperlink from HTML can take you to another page, it could ring an alarm, run a yearend procedure or anything else that a computer program can do.

---

15. More things to do

Here is a list of other interesting things to do with your Linux Intranet server. All the software mentioned below is freeware or shareware.

- Browse the Linux server using Network Neighbourhood in Win95/ NT; Setup a WINS like NBT server. Check out the SAMBA Web page at http://lake.canberra.edu.au/pub/samba/samba.html
- Implement a search engine on your Intranet. Connect to ht://Dig at http://htdig.sdsu.edu/
- Use CUSeeMe by setting up a local reflector. Refer to their page at Cornell http://cu-seeme.cornell.edu/
- Setup Web Conferencing. Use COW from http://thecity.sfsu.edu/COW/
- Deploy a SQL database. Refer to the mSQL Home page at http://Hughes.com.au/
- Setup FTP,Gopher,Finger,Bootp servers on the Netware server. Get them at http://mft.ucs.ed.ac.uk/
- Emulate a Netware server. Check out the NCP Utilities at ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/

If you find other interesting things to do with your Linux Intranet server, please feel free to mail me.

16. Credits and Legalities

16.1 Thanks

Thanks to the people at NCSA for providing such excellent documentation, David Anderson and all others for trying out this HOWTO and sending in their comments. The details on Netware/IP are courtesy Romel Flores (rom@mnl.sequel.net).

16.2 Copyright information

This document is copyrighted © 1999 {Poet} and LinuxPorts, it is distributed under the following terms:

- Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.
- All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.
- If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu Finger for phone number and snail mail address.

17. For further reading

17.1 O'Reilly & Associates Books

In my humble opinion O'Reilly & Associates make the best technical books on the planet. They focus mainly on Internet, Unix and programming related topics. They start off slow with plenty of examples and when you finish the book your an expert. I think you could get by if you only read half of the book. They also add some humor to otherwise boring subjects.

They have great books on HTML, PERL, CGI Programming, Java, JavaScript, C/C++, Sendmail, Linux and much much more. And the fast moving topics (like HTML) are updated and revised about every 6 months or so. So visit the O'Reilly & Associates web site or stop by your local book store for more info.

And remember if it doesn't say O'Reilly & Associates on the cover, someone else probably wrote it.

17.2 Internet Request For Comments (RFC)

- RFC1866 written by T. Berners-Lee and D. Connolly, "Hypertext Markup Language - 2.0", 11/03/1995
- RFC1867 writtenm by E. Nebel and L. Masinter, "Form-based File Upload in HTML", 11/07/1995
- RFC1942 written by D. Raggett, "HTML Tables", 05/15/1996
- RFC1945 by T. Berners-Lee, R. Fielding, H. Nielsen, "Hypertext Transfer Protocol -- HTTP/1.0", 05/17/1996.
- RFC1630 by T. Berners-Lee, "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web", 06/09/1994
- RFC1959 by T. Howes, M. Smith, "An LDAP URL Format", 06/19/1996

A mSQL and perl Web Server Mini HOWTO

Oliver Corff, corff@zedat.fu-berlin.de

v0.1, 17 September 1997

This Mini HOWTO, highly inspired by Michael Schilli's article Gebunkert: Datenbankbedienung mit Perl und CGI, published in the german computer magazine iX 8/1997, describes how to build a SQL client/server database using WWW and HTML for the user interface.

1. About this Document

- 1.1 Intended Audience
- 1.2 Conventions used in this text

2. Introduction

3. Installation Procedure

- 3.1 Hardware Requirements
- 3.2 Software Requirements
- 3.3 Installing the Operating System
- 3.4 The http Server
- 3.5 The Browsers
- 3.6 Cooperation of Apache and Browsers
- 3.7 The Database Engine and its Installation
- 3.8 Choice of Interfaces: DBI/mSQL, MsqlPerl, and Lite
- 3.9 Going the generic way: DBI and DBD-msql
- 3.10 The MsqlPerl Interface
- 3.11 perl's CGI library
- 3.12 Installation Checklist

TOC

4. Running an Example Database

- 4.1 Adapting the sample script for MsqlPerl
- 4.2 Adapting the sample script for msql-2

5. Conclusion and Outlook

---

1. About this Document

1.1 Intended Audience

Everybody who wants to install a web server database but does not know which software is necessary and how it is installed should benefit from reading this text. This text provides all information necessary to get a SQL database for a web server going; it does *not* go into any detail of CGI programming, nor does it explain the SQL database language. Excellent books are available on both topics, and it is the intention of this text to provide a working platform based on which a user can then study CGI programming and SQL.

For getting a small scale SQL system running (not the notorious example of a major airline booking system, or space mission management database) it will be sufficient to have the software described in this text and the documentation accompanying it. The user manual of msql (a database introduced in this text) provides sufficient information on SQL for building your own database.

The reader of this text should have a working knowledge of how to obtain files via `ftp` if he has no access to CD-ROMs, and a basic understanding of how to build binaries from sources. Anyway, all steps explained in this text were tested on a real life system and should also work on the reader's system.

1.2 Conventions used in this text

A user command:

```
# make install
```

Screen output from a program:

```
    Program installed. Read README for details on how to start.
```

Sample code of a file:

```
# My comment
char letter;
```

2. Introduction

It can be safely assumed that databases with a high volume of data or a complicated relational setup (like, perhaps, a lexical database for a living language) must be accessible to many users and operators at the same time. Ideally, it should be possible to use existing different hardware and software platforms that can be combined into the actual system. In order to reduce the implementation cost, only one system, the database server, needs to be powerful; the user stations typically just display data and accept user commands, but the processing is done on one machine only which led to the name client-server database. In addition, the user interface should be easy to maintain and should require as little as possible on the client side.

A system which meets these criteria can be built around the following items of protocols, concepts and software:

Linux

supplies the operating system. It is a stable Unix implementation providing true multi-user multi-tasking services with full network (TCP/IP e. a.) support. Except from the actual media and transmission cost, it is available free of charge and comes in form of so-called distributions which usually include everything needed from the basic OS to text processing, scripting, software development, interface builders, etc.

HTML

is the Hypertext Markup Language used to build interfaces to network systems like Intranets and the WWW, the World Wide Web. HTML is very simple and can be produced with any ASCII-capable text editor.

Browsers

are text-based (e. g. Lynx) or graphical (e. g. Mosaic, Netscape, Arena etc.) applications accepting, evaluating and displaying HTML documents. They are the only piece of software which is directly operated by the database user. Using browsers, it is possible to display various types of data (text, possibly images) and communicate with http servers (see next) on about every popular computer model for which a browser has been made available.

http servers

provide access to the area of a host computer where data intended for public use in a network are stored. They understand the http protocol and procure the information the user requests.

SQL

Structured Query Language is a language for manipulating data in relational databases. It has a very simple grammar and is a standard with wide industry support. SQL-based databases have become the core of the classical client/server database concept. There are many famous SQL systems available, like Oracle, Informix etc., and then there is also msql which comes with a very low or even zero price tag if it is used in academical and educational environments.

CGI

Common Gateway Interface is the programming interface between the system holding the data (in our case an SQL-based system) and the network protocol (HTML, of course). CGIs can be built around many programming languages, but a particularly popular language is perl.

perl

is an extremely powerful scripting language which combines all merits of C, various shell languages, and stream manipulation languages like awk and sed. Perl has a lot of modularized interfaces and can be used to control SQL databases, for example.

---

3. Installation Procedure

3.1 Hardware Requirements

No general statement can be made about the hardware requirements of a database server. Too much depends on the expected number of users, the kind of application, the network load etc. In a small environment with only a few users and little network traffic a i486-equivalent machine with 16 MB of RAM can be completely sufficient. Linux, the operating system, is very efficient in terms of resources, and can supply enough horse-power for running a broad variety of applications at the same time. Of course, faster processors and more RAM mean more speed, but much more important than the processor is the amount of RAM. The more RAM the system has the less it is forced to swap memory intensive processes to disk in case a bottleneck occurs.

Given anything like 32 MB RAM and a PCI bus, searches and sorting operations can be done without much resorting to swap files etc., resulting in lightening fast speed.

The model installation described in this article was made on a IBM 686 (133Mhz) with 32 MB RAM and a 1.2 GB IDE hard disk. Assuming that the installation process starts from scratch, here is a list of the necessary steps.

3.2 Software Requirements

The software described in this article is available from the Internet or from CD-ROM. The following products were used:

- Red Hat Linux PowerTools: 6 CD's Complete Easy-to-Use Red Hat 4.2, Summer '97; alternatively from `http://www.redhat.com`;
- msql SQL database server: it is now available in two versions. The versions have differences in the number of transactions they can handle, the administration interface, etc. The elder version, 1.0.16, is available from Sunsite mirrors. The ELF executable can be found at `sunsite:apps/database/sql/msql-1.0.16` or on CD-ROM (here: disc 4 of InfoMagic Linux Developer's Resource, 6-CD set, December 1996) or alternatively from the following URL: `http://www.infomagic.com`. The newer version, 2.0.1, can be directly obtained from Hughes' homepage in Australia (`http://www.hughes.com.au`) or from numerous mirror sites around the world;
- perl from CPAN: The Comprehensive Perl Archive Network. Walnut Creek CDROM, ISBN 1-57176-077-6, May 1997;
- Michael Schilli's CGI example program from computer journal iX 8/1997, pages 150--152, available via ftp from `ftp.uni-paderborn.de:/doc/magazin/iX`;

3.3 Installing the Operating System

Linux is installed in form of the Red Hat Linux Distribution 4.2. In order to install successfully, the machine must either have a DOS-accessible CD-ROM drive, a bootable CD-ROM drive, or else a boot disk must be made following the instructions on the Linux CD.

During installation the user has the choice to select and configure numerous software packages. It is convenient to select the following items now:

- TCP/IP network support,
- the http server Apache, and
- the scripting language perl, and

- the X Window System, as well as
- the browsers Arena (graphical) and Lynx (text-based).

All these packages are provided with the Linux distribution. If you do not install these packages now you still have the chance to do this later with the assistance of glint, the graphical and intuitive software package installation manager. Be sure to be root when installing these packages.

It is beyond the scope of this article to describe the network installation and initialization procedure. Please consult the online (manpages, HTML, texinfo) and printed (Linux Bible, etc. etc.) documentation.

The installation procedure of Red Hat is very mature and requires only little user attention besides the usual choices (like providing host names, etc.). Once the installation ends successfully, the system is basically ready to go.

Installing the X Window System is not mandatory for a pure server but it makes local access and testing much easier. The X installation procedure is done by any of several programs; XF86Setup offers the most extensive self-testing facilities and needs the least handling of hairy details (like video clock programming, etc.). The only requirement is that the software can detect the video adapter. A cheap accelerated graphics adapter (like Trio S64 based cards prior to S64UV+) usually works ``out of the box''.

At this point we assume that our system is up and running and that Apache, Perl and the X Window System have been successfully installed. We further assume that all standard structures like the file and directory structure are kept as they are defined in the installation. Last but not least we leave the host name as it is, and do at this moment accept the name `localhost`. We'll use this name for testing the installation; once the whole system works the true name can be added. Please note that the network setup also requires editing the files `/etc/hosts`, among others. Ideally this should be done with the administration tools provided to user root.

3.4 The http Server

The http server supplied with Linux is known as Apache to humans and as httpd to the system. The manpage (man httpd) explains how to install and start the http daemon (hence http*d*) but, as mentioned, if the installation went without problems then the server should be running. You can verify the directory tree: there must be a directory `/home/httpd/` with three subdirectories: `../cgi-bin/`, `../html/` and `../icons/`. In `../html/` there must be a file `index.html`. Later we will manipulate or replace this file by our own `index.html`. All configuration information is stored in`/etc/httpd/conf/`. The system is well preconfigured and does not need further setup provided the installation went without error.

3.5 The Browsers

There are essentially three types of browsers available for Linux: pure text-based systems like Lynx, experimental and simple ones like Arena (free!) and commercial ones like Netscape (shareware!) with Java support. While Lynx and Arena come with Linux, Netscape must be procured from other sources. Netscape is available as a precombiled binary for Linux on ix86 architectures and will run ``out of the box'' as soon as the archive is unpacked.

Configuring Lynx

Once Lynx is started it will look for a `default URL' which is usually not very meaningful if the system does not have permanent Internet access. In order to change the default URL (and lots of other configuration details) the system administrator should edit `/usr/lib/lynx.cfg`. The file is big, around 57000 bytes and contains occasionally contradicting information. It states its own home as `/usr/local/lib/`. Not far from top is a line beginning with `STARTFILE`. We replace this line by the following entry: `STARTFILE:http://localhost` and make sure that no spacing etc. is inserted:

```
# STARTFILE:http://www.nyu.edu/pages/wsn/subir/lynx.html
STARTFILE:http://localhost
```

After saving the file, Lynx should now reveal our `index.html` document if started without arguments.

Configuring Arena

Arena first looks for its own default URL when started without arguments. This URL is hard-wired into the executable but can be overrun by the environment variable `WWW_HOME`. The system administrator can place a line saying `WWW_HOME="http://localhost"` in `/etc/profile`. The variable must then be exported, either by a separate statement (`export WWW_HOME`) or by appending `WWW_HOME` to the existing export statement:

```
WWW_HOME="http://localhost"
export WWW_HOME
```

After relaunching a login shell, the new default URL is now system-wide known to Arena.

Installing and Configuring Netscape

Netscape is a commercial product and thus not included with the Linux distributions. It is either downloadable from the Internet or available from software collections on CDROM. Netscape comes in form of precompiled binaries for every important hardware platform. For installation purposes, it is useful to create a directory `/usr/local/Netscape/` where the archive is unpacked. The files can be kept in place (except for the Java library: follow the instructions in the `README` file that comes with the Netscape binary), and it is sufficient to create a soft link in `/usr/local/bin/` by issuing the command

```
# ln -s /usr/local/Netscape/netscape .
```

from within `/usr/local/bin/`.

Netscape is now ready for use and can be configured via the ``Options'' menu. In ``General Preferences'' there is a card ``Appearance'' with the entry ``Home Page Location''. Enter `http://localhost` here and do not forget to save the options (via ``Options'' --- ``Save Options'') before exiting Netscape. At the next startup, Netscape will now show the Apache `homepage'.

3.6 Cooperation of Apache and Browsers

You can now conduct the first real test of both the browser and the http server: simply start any of the available browsers and the `Apache: Red Hat Linux Web Server` page will pop up. This page shows the file locations and other basics of http server installation. If this page is not displayed please check whether the files mentioned above are in place and whether the browser configuration is correct. Close edited configuration files before you start the browser again. If all files are in place and the browsers seem to be configured correctly then examine the network setup of your machine. Either the host name is different from what was entered in the configuration, or the network setup as such is not correct. It is utterly important that `/etc/hosts` contains at least a line like

```
127.0.0.1               localhost localhost.localdomain
```

which implies that you can connect locally to your machine. One can verify this by issuing any network-sensitive command requiring a host name as argument, like `telnet localhost` (provided `telnet` is installed). If that does not work then the network setup must be verified before continuing with the main task.

3.7 The Database Engine and its Installation

TOC

Installing the database requires only little more preparation than the previous installation steps. There are a few SQL database engines available with different runtime and administrative requirements, and possibly one of the most straightforward systems is msql, or ``Mini-SQL'' by David Hughes. msql is shareware. Depending on the version used, commercial sites are charged USD 250.00 and more, private users are charged USD 65.00 and more, and only educational institutions and registered non-profit organizations can use this software free of charge. Please note that the exact figures are provided in the licence notes of the database documentation. The figures given here serve as a rough indicator only.

A few words are in place here why the author chose msql. First of all, there is personal experience. While searching for a database engine the author found msql to be about the easiest to install and maintain, and it provides enough coverage of the SQL language to meet general needs. Only when writing these lines, the author discovered the following words of praise in Alligator Descartes' DBI FAQ (perl database interface FAQ):

> From the current author's point of view, if the dataset is relatively small, being tables of less than 1 million rows, and less than 1000 tables in a given database, then mSQL is a perfectly acceptable solution to your problem. This database is extremely cheap, is wonderfully robust and has excellent support. ...

Msql is available in two versions now, msql-1.0.16 and msql-2.0.1, which differ in performance (not noticeable in small scale projects) and accompanying software (the newer version comes with more tools, its own scripting language, etc.). We will describe both versions of msql since their installion differs in a few points.

Installing msql-1.0.16

msql is available as source and as compiled executable with ELF support. Using the ELF binaries makes installation easy since the archive file `msql-1.0.16.ELF.tgz` contains a complete absolute directory tree so that all directories are generated properly when unpacked from `/`.

If you decide to compile msql-1.0.16 yourself and are going to use the MsqlPerl package rather than the DBI interface (see a detailed discussion on the difference between these two further down) then be prepared that MsqlPerl might complain during the test suites that some instruction inside msql failed. In this case a patch may be necessary which is described in the MsqlPerl documentation (file `patch.lost.tables`). Notably, this demands including three lines in `msqldb.c` after line 1400 which says `entry->def = NULL;`:

```
*(entry->DB) = 0;
*(entry->table) = 0;
entry->age = 0;
```

The code fragment should now look like

```
freeTableDef(entry->def);
safeFree(entry->rowBuf);
safeFree(entry->keyBuf);
entry->def = NULL;
*(entry->DB) = 0;
*(entry->table) = 0;
entry->age = 0;
```

Compiling msql involves several steps. After unpacking the source archive, it is necessary to build a target directory. This is done by saying

```
# make target
```

TOC

If successful, the system will then answer with

```
Build of target directory for Linux-2.0.30-i486 complete
```

You must now change into this newly created directory and run a

```
# ./setup
```

command first. The `./` sequence is necessary to make sure that really the command `setup` in this directory and not another command which happens to have the same name is executed. You will then be asked questions on the location of the source directory and whether a root installation is desired. These questions answered, the system should then run a number of tests checking for available software (compilers, utilities etc.) and finally say

```
Ready to build mSQL.

You may wish to check "common/site.h" although the defaults should be
fine.  When you're ready, type  "make all" to build the software
```

We say

```
# make all
```

If everything went as intended, we'll read:

```
make[2]: Leaving directory `/usr/local/Minerva/src/msql'
<-- [msql] done

Make of mSQL complete.
You should now mSQL using make install

NOTE : mSQL cannot be used free of charge at commercial sites.
       Please read the doc/License file to see what you have to do.

make[1]: Leaving directory `/usr/local/Minerva/src'
```

All binaries must then be made visible to the search paths by creating soft links in `/usr/local/bin/`. Change to that directory and issue the command

```
# ln -s /usr/local/Minerva/bin/* .
```

after which the links will be properly set.

Testing msql-1

After the installation it is now possible to test whether the database works. Before anything else is done, the server daemon must be started. The system administrator holding root privileges issues the command

```
# msqld &
```

(do not forget to add the &, otherwise msql won't run in the background.) after which the following screen message appears:

```
mSQL Server 1.0.16 starting ...

Warning : Couldn't open ACL file: No such file or directory
```

```
Without an ACL file global access is Read/Write
```

This message tells us that everything so far worked since we did not set up any access restrictions. For the moment it is sufficient to start the msql daemon from within a shell but later we may want to have the system startup automatically execute this command for us. The command must then be mentioned in a suitable `rc.d` script. Only now the administrator can issue the first genuine database command:

```
# msqladmin create inventur
```

msql replies by saying `Database "inventur" created.`. As a further proof, we find that the directory `/usr/local/Minerva/msqldb/` contains now the empty subdirectory `../inventur/`. We could manipulate the newly created database with the administration tools; these procedures are all covered in detail in the msql documentation.

Installing msql-2.0.1

There is now a newer, more powerful version of Hughes' mSQL server available the installation of which is different in a few points. Installing msql-2 from scratch involves the following steps. Copy the archive to your extraction point, e. g. `/usr/local/msql-2/`, then untar the archive:

```
# tar xfvz msql-2.0.1.tar.gz
```

Change to the root direction of the install tree and issue a

```
# make target
```

Change to `targets` and look for your machine type. There should be a new subdirectory `Linux-`*(your version)-(your cpu)/*. Change to that directory and start the setup facility located here:

```
# ./setup
```

There is also a file `site.mm` which can be edited. Maybe you have got used to the directory name `/usr/local/Minerva/` and want to preserve it? In this case change the `INST_DIR=...` line to your desired target directory. Otherwise, leave everything as it is.

Now you can start building the database:

```
# make
# make install
```

If everything went successfully, we'll see a message like:

```
[...]

Installation of mSQL-2 complete.

*********
**   This is the commercial, production release of mSQL-2.0
**   Please see the README file in the top directory of the
**   distribution for license information.
*********
```

After all is installed properly we have to take care of the administration details. Here, the real differences from msql-1 begin. First, a user `msql` is created which is responsible for database administration.

```
# adduser msql
```

Then we have to change all ownerships in the mSQL directory to `msql` by saying:

```
# cd /usr/local/Minerva
# chown -R msql:msql *
```

Then we create soft links for all database binaries in `/usr/local/bin/` by saying:

```
# ln -s /usr/local/Minerva/bin/* .
```

Testing msql-2

We can now start the database server by issuing the command `msql2d &` and should get a response similar to this one:

```
    Mini SQL Version 2.0.1
    Copyright (c) 1993-4 David J. Hughes
    Copyright (c) 1995-7 Hughes Technologies Pty. Ltd.
    All rights reserved.

            Loading configuration from '/usr/local/Minerva/msql.conf'.
            Server process reconfigured to accept 214 connections.
            Server running as user 'msql'.
            Server mode is Read/Write.

    Warning : No ACL file.  Using global read/write access.
```

That looks perfect. The database is compiled and in place, and we can now continue with the perl modules since these rely partially on the presence of a working database server for testing.

Accidentally, this is also a good moment to print the complete manual that comes with msql-2.0.1:

```
# gzip -d manual.ps.gz
# lpr manual.ps
```

We can proceed to building the interfaces now, but it is a good idea to keep the newly created SQL server up and running since that makes testing the interface libraries somewhat simpler.

3.8 Choice of Interfaces: DBI/mSQL, MsqlPerl, and Lite

A frequently quoted saying in the Camel Book (the authorative perl documentation) states that there is more than one way to achieve a result when using perl. This, alas, holds true for our model application, too. Basically there are three ways to access an msql database via CGI. First of all the question is whether or not perl shall be used. If we use perl (on which this article focuses) then we still have the choice between two completely different interface models. Besides using perl, we can also employ msql's own scripting language, called Lite, which is reasonably simple and a close clone of C.

DBI and DBD-mSQL

By the time of this writing, using perl's generic database interface called DBI is the method of choice. DBI has a few advantages: It provides unified access control to a number of commercial databases with a single command set. The actual database in use on a given system is then contacted through a driver which effectively hides the pecularities of that database from the programmer. Being such, using DBI provides for a smooth transition between different databases by

different makers. In one single script it is even possible to contact several different databases. Please refer to the DBI-FAQ for details. There is, however, one drawback: The DBI interface is still under development and shows rapid galloping version numbers (sometimes with updates taking place within less than a month). Similarly, the individual database drivers are also frequently updated and may rely on specific versions of the database interface. Users making first-time installations should stick to the version numbers given in this article since other versions may cause compilation and testing problems the trouble shooting of which is nothing for the faint-hearted.

MsqlPerl

MsqlPerl is a library for directly accessing msql from perl scripts. It bypasses the DBI interface and is fairly compact. Though it works fine with both versions of msql, its usage is not promoted anymore in favour of the generalized DBI interface. Nonetheless, in a given installation it may prove to be the interface of choice since it is small and easy to install. Notably, it has less version dependencies than revealed by the interaction of DBI and particular database drivers.

msql's own scripting language: Lite

Last but not least msql-2 comes with its own scripting language: Lite. The language is a close relative of C stripped of its oddities with additional shell-like features (in a way, something like a very specialized version of perl). Lite is a simple language and is well documented in the msql-2 manual. The msql-2 package also comes with a sample application sporting Lite.

We will not describe Lite here because it is well documented but fairly specific to msql-2, and because it is assumed that the readers of this article have a basic interest in and a basic understanding of perl. Nonetheless it is highly recommended to have a closer look at Lite: it may well be the case that Lite offers the solution of choice in an exclusive msql-2 environment (implying no other databases are involved) due to its simplicity and straightforward concept.

3.9 Going the generic way: DBI and DBD-msql

We assume that perl was installed during the system setup or via the package manager mentioned above. No further details will be given here. Nonetheless we first test whether our version of perl is up to date:

```
# perl -v
```

perl should respond with the following message:

```
This is perl, version 5.003 with EMBED
        Locally applied patches:
          SUIDBUF - Buffer overflow fixes for suidperl security

        built under linux at Apr 22 1997 10:04:46
        + two suidperl security patches

Copyright 1987-1996, Larry Wall
[...]
```

So far, everything is fine. The next step includes installing the perl libraries for databases in general (DBI), the msql driver (DBD-mSQL) and CGI. The CGI driver is necessary in any case. The following archives are necessary:

1. DBI-0.81.tar.gz
2. DBD-mSQL-0.65.tar.gz
3. CGI.pm-2.31.tar.gz (or higher)

A caveat is necessary here for beginners: the test installation described here works fine using software with *exactly* these

version numbers, and combinations of other versions failed in one or the other way. Debugging flawed version combinations is nothing for those who are not very familiar with the intimate details of the calling conventions etc. interfaces. Sometimes only a method is renamed while performing the same task, but sometimes the internal structure changes significantly. So, again, stick with these version numbers if you want to be on the safe side even if you discover that version numbers have increased in the meantime. Frequent updates of these interfaces are the rule rather than the exception, so you should really anticipate problems when installing other versions than those indicated here.

It is very important that the database driver for mSQL (DBD-mSQL) is installed *after* the generic interface DBI.

We start by creating the directory `/usr/local/PerlModules/` as it is very important to keep the original perl directory tree untouched. We could also choose a different directory name since the name is completely uncritical, and unfortunately that is not really mentioned in the README files of the verious perl modules. Having copied the above-mentioned archives to `/usr/local/PerlModules/` we unpack them saying

```
# tar xzvf [archive-file]
```

for every single of the three archives. Do not forget to supply the real archive name to `tar`. The installation process for the three modules is essentially stardardized; only the screen messages showing important steps of individual packages are reproduced here.

Installing perl's Database Interface DBI

The database interface must always be installed before installing the specific database driver. Unpacking the DBI archive creates the directory `/usr/local/PerlModules/DBI-0.81/`. Change to that directory. There are a `README` file (you should read it) and a perl-specific makefile. Now issue the command

```
# perl Makefile.PL
```

The system should answer with a lengthy message of which the most important part is shown here::

```
[...]
MakeMaker (v5.34)
Checking if your kit is complete...
Looks good
        NAME => q[DBI]
        PREREQ_PM => {  }
        VERSION_FROM => q[DBI.pm]
        clean => { FILES=>q[$(DISTVNAME)/] }
        dist => { DIST_DEFAULT=>q[clean distcheck disttest [...]
Using PERL=/usr/bin/perl

WARNING! By default new modules are installed into your 'site_lib'
directories. Since site_lib directories come after the normal library
directories you MUST delete old DBI files and directories from your
'privlib' and 'archlib' directories and their auto subdirectories.

Writing Makefile for DBI
```

This looks good, as the program says, and we can proceed with the next step:

```
# make
```

If no error message occurs (the detailed protocol dumped on screen is *not* an error message) we test the newly installed library with the command

```
# make test
```

Watch the output for the following lines (you can always scroll back with `[Shift]-[PgUp]`):

```
[...]
t/basics............ok
t/dbidrv............ok
t/examp.............ok
All tests successful.
[...]
DBI test application $Revision: 1.20 $
Switch: DBI-0.81 Switch by Tim Bunce, 0.81
Available Drivers: ExampleP, NullP, Sponge
ExampleP: testing 2 sets of 5 connections:
Connecting... 1 2 3 4 5
Disconnecting...
Connecting... 1 2 3 4 5
Disconnecting...
Made 10 connections in  0 secs ( 0.00 usr  0.00 sys =  0.00 cpu)

test.pl done
```

The final step is to install all files in their proper directories. The following command will take care of it:

```
# make install
```

No more duties are left. If for some reason the installation failed and you want to redo it do not forget to issue

```
# make realclean
```

first. This will remove stale leftovers of the previous installation. You can also remove the files which were installed by copying the screen contents (shown abbreviated)

```
Installing /usr/lib/perl5/site_perl/i386-linux/./auto/DBI/DBIXS.h
Installing /usr/lib/perl5/site_perl/i386-linux/./auto/DBI/DBI.so
Installing /usr/lib/perl5/site_perl/i386-linux/./auto/DBI/DBI.bs
[...]
Writing /usr/lib/perl5/site_perl/i386-linux/auto/DBI/.packlist
Appending installation info to /usr/lib/perl5/i386-linux/5.003/perllocal.pod
```

into a file, replacing every `Installing` with `rm`. Provided you named the file `uninstall` you can then say

```
# . uninstall
```

which will remove the recently installed files.

perl's msql Driver DBD-mSQL

The msql driver can only be installed *after* a successful installation of perl's generic database interface.

The basic steps are the same as above; so first go through

```
# perl Makefile.PL
```

Here, the system should answer with an urgent warning to read the accompanying documentation. It will then detect

where msql resides, and asks which version you use:

```
$MSQL_HOME not defined. Searching for mSQL...
Using mSQL in /usr/local/Hughes

 -> Which version of mSQL are you using [1/2]?
```

State your correct version number. Quite a few lines of text will follow. Watch for the following ones:

```
Splendid! Your mSQL daemon is running. We can auto-detect your configuration!

I've auto-detected your configuration to be running on port: 1114
```

You can now test the driver by saying

```
# make test
```

Again, a lengthy output follows. If it ends with

```
Testing: $cursor->func( '_ListSelectedFields' ). This will fail.
        ok: not a SELECT in msqlListSelectedFields!
Re-testing: $dbh->do( 'DROP TABLE testaa' )
        ok
*** Testing of DBD::mSQL complete! You appear to be normal! ***
```

you are on the safe side of life and can install your driver by saying

```
# make install
```

You are now ready to go and can skip the next paragraph.

3.10 The MsqlPerl Interface

If you decide to use the exclusive MsqlPerl interface then no generic database driver is needed, only `MsqlPerl-1.15.tar.gz`, since, as mentioned earlier, MsqlPerl provides a direct interface between perl and the database server without using the DBI interface. Installing and testing is straightforward.

After saying `perl Makefile.PL` the make utility can be started. First you have to answer the question where mSQL resides. If it resides in `/usr/local/Minerva/` the default answer can be confirmed.

Then do a `make test`. Before doing so you must ensure that you have a database named `test` and that you have read and write permissions for it. This can be done by

```
# msqladmin create test
```

3.11 perl's CGI library

Installing perl's CGI part is the simpliest of the three steps. Execute the following commands in the given order and everything is done:

```
# perl Makefile.PL
# make
# make install
```

TOC

Unlike the previous drivers this interface does not have a test option (`# make test`) whereas the other modules s be tested in any case.

A subdirectory with CGI example scripts is also created. You can copy the contents of this directory into `/home/http/cgi-bin/` and use the browser to experiment with the scripts.

3.12 Installation Checklist

We went through the following steps, in this order:

1. Install Linux with networking support
2. Install a http server, e. g. Apache
3. Install a browser, e. g. Arena, lynx or Netscape
4. Install an SQL server, e. g. msql
5. Install a suitable perl SQL interface
6. Install the CGI files

Finally, you can do some clean-up. All source trees for msql and the perl modules can be safely deleted (however, you should not delete your archive files!) since the binaries and documentation are now based in different directories.

---

4. Running an Example Database

After completing the system installation we can now finally run a model application. Depending on the version of msql installed and the perl database interface used, we have to modify the sample programs in a few points.

First however, the file `index.html` residing in `/home/httpd/html/` must be modified to allow calling a sample database application. We can place our database (which we call `database.cgi` or `inventur.cgi` here despite its archive name `perl.lst.ck`) in `/home/httpd/html/test/`.

We add one line (of course, depending on your installation choices) similar to the following to `index.html`:

```
<LI>Test the <A HREF="test/database.cgi">Database, DBI:DBD-mSQL style!</A>
<LI>Test the <A HREF="test/inventur.cgi">Database, MsqlPerl style!</A>
```

Usually you should only pick one of these two choices but if you have both types of database interface installed you can leave both lines here as they are. You can then compare performance, etc.

4.1 Adapting the sample script for MsqlPerl

Our sample script has to be told to use the MsqlPerl interface. The modification takes place in several locations. First, near the beginning of the file, we change the `use` clause:

```
#
# use DBI;            # Generisches Datenbank-Interface
use Msql;
```

Then, near line 27, the MsqlPerl syntax does not require the mentioning of a specific driver:

```
# $dbh = DBI->connect($host, $database, '', $driver) ||
$dbh = Msql->connect($host, $database) ||
```

Then, from line 33 onward throughout the whole script, we have to change all instances of `do` against `query`:

```
# $dbh->do("SELECT * FROM hw") || db_init($dbh);
$dbh->query("SELECT * FROM hw") || db_init($dbh);
```

Finally, in MsqlPerl speak, line 207 can be commented out:

```
# $sth->execute || msg("SQL Error:", $sth->errstr);
```

In addition, it may become necessary to swap all `errstr` calls like the one in the preceding code fragment against `errmsg`. This is also version dependent.

After these modifications, the script should run smoothly.

4.2 Adapting the sample script for msql-2

The SQL syntax was redefined during the development of mslq-2. The original script will fail to execute the table initialization statements in lines 45 -- 58. The `primary key` modifier is no longer supported by msql-2, and should simply be skipped:

```
    $dbh->do(<<EOT) || die $dbh->errstr; # Neue Personen-Tabelle
        create table person (
# We do not need the 'primary key' modifier anymore in msql-2!
#           pn        int primary key,   # Personalnummer
            pn        int,               # Personalnummer
            name      char(80),          # Nachname, Vorname
            raum      int                # Raumnummer
        )
EOT
    $dbh->do(<<EOT) || die $dbh->errstr; # Neue Hardware-Tabelle
        create table hw (
# We do not need the 'primary key' modifier anymore in msql-2!
#           asset int primary key,       # Inventurnummer
            asset int,                   # Inventurnummer
            name   char(80),             # Bezeichnung
            person int                   # Besitzer
        )
EOT
```

Unfortunately, this specific script will then accept new entries with identical personnel numbers; the msql-1 modifier `primary key` intends to prevent exactly this behaviour. The msql-2 documentation shows how to use the `CREATE INDEX` clause to create unique entries.

5. Conclusion and Outlook

If you have installed msql-2 on your system then you can have a look at the sample programs written in Lite, msql-2's own scripting language.

Either version of msql comes with a basic set of administration tools which allow the user to create and drop tables (`msqladmin`) and examine database structures (`relshow`).

The second generation msql (i.e. msql-2) has a few more genuinely useful utilities: `msqlimport` and `msqlexport`. These allow the dumping of flat line data files into and out of the SQL database. They can be used for loading quantities of existing data *d'un coup* into existing tables, or extract flat data from tables, and the user does not have to deal with writing a *single* line of perl or SQL or whatever code for this task.

If you want to write your own perl scripts dealing with databases you'll find sufficient support in the example files and the extensive on-line documentation that comes with the DBI module.

Anyway, you are now ready to go and present your data to the users of your own network, or even the WWW.

The Linux XFree86 HOWTO

by Eric S. Raymond

v5.12, 12 May 1999

This document describes how to obtain, install, and configure version 3.3 of the XFree86 version of the X Window System (X11R6) for Linux systems. It is a step-by-step guide to configuring XFree86 on your system.

1. Introduction

- 1.1 Other sources of information
- 1.2 New versions of this document
- 1.3 Feedback and Corrections

2. Hardware requirements

3. Installing XFree86

4. Configuring XFree86

- 4.1 Normal Configuration
- 4.2 Troubleshooting
- 4.3 Custom Configuration
- 4.4 Using 16-bit Color

5. Running XFree86

- 5.1 Terms of Use
- 5.2 Acknowledgements

6. Translations

1. Introduction

The X Window System is a large and powerful (some might say excessively large and overly complex) graphics environment for UNIX systems. The original X Window System code was developed at MIT; commercial vendors have since made X the industry standard for UNIX platforms. Virtually every UNIX workstation in the world runs some variant of the X Window system.

A freely redistributable port of the MIT X Window System version 11, release 6 (X11R6) for 80386/80486/Pentium UNIX systems has been developed by a team of programmers originally headed by David Wexelblat <dwex@XFree86.org>. The release, known as XFree86, is available for System V/386, 386BSD, and other x86 UNIX implementations, including Linux. It includes all of the required binaries, support files, libraries, and tools.

Complete information on XFree86 is available at the XFree86 web site, http://www.XFree86.org.

In this document, we'll give a step-by-step description of how to install and configure XFree86 for Linux, but you will have to fill in some of the details yourself by reading the documentation released with XFree86 itself. (This documentation is discussed below.) However, using and customizing the X Window System is far beyond the scope of this document---for this purpose you should obtain one of the many good books on using the X Window System.

1.1 Other sources of information

If you have never heard of Linux before, there are several sources of basic information about the system. The best place to find these is at the Linux Documentation Project home page at http://metalab.unc.edu/LDP. You can find the latest, up-to-date version of this document there, as http://metalab.unc.edu/LDP/HOWTO/XFree86-HOWTO.html

1.2 New versions of this document

New versions of the Linux XFree86 HOWTO will be periodically posted to comp.os.linux.help and news:comp.os.linux.announce and news.answers. They will also be uploaded to various Linux WWW and FTP sites, including the LDP home page.

You can always view the latest version of this on the World Wide Web via the URL http://metalab.unc.edu/LDP/HOWTO/XFree86-HOWTO.html.

1.3 Feedback and Corrections

TOC

If you have questions or comments about this document, please feel free to mail Eric S. Raymond, at esr@thyrsus.com. I welcome any suggestions or criticisms. If you find a mistake with this document, please let me know so I can correct it in the next version. Thanks.

Please do *not* mail me questions about how to make your video card and monitor work with X. This HOWTO is intended to be a rapid, painless guide to *normal* installation using the new interactive configurator. If you run into problems, browse the XFree86 Video Timings HOWTO, http://metalab.unc.edu/LDP/HOWTO/XFree86-Video-Timings-HOWTO.html. (This is the up-to-date HTML version of XFree86's `Videomodes.doc' file.) That document tells everything I know about configuration troubleshooting. If it can't help you, I can't either.

---

2. Hardware requirements

XFree86 3.3.3 presently supports the following chipsets: The documentation included with your video adaptor should specify the chipset used. If you are in the market for a new video card, or are buying a new machine that comes with a video card, have the vendor find out exactly what the make, model, and chipset of the video card is. This may require the vendor to call technical support on your behalf; in general vendors will be happy to do this. Many PC hardware vendors will state that the video card is a ``standard SVGA card'' which ``should work'' on your system. Explain that your software (mention Linux and XFree86!) does not support all video chipsets and that you must have detailed information.

You can also determine your videocard chipset by running the `SuperProbe` program included with the XFree86 distribution. This is covered in more detail below.

Ark Logic

ARK1000PV, ARK1000VL, ARK2000PV, ARK2000MT

Alliance

AP6422, AT24

ATI

18800, 18800-1, 28800-2, 28800-4, 28800-5, 28800-6, 68800-3, 68800-6, 68800AX, 68800LX, 88800GX-C, 88800GX-D, 88800GX-E, 88800GX-F, 88800CX, 264CT, 264ET, 264VT, 264GT, 264VT-B, 264VT3, 264GT-B, 264GT3 (this list includes the Mach8, Mach32, Mach64, 3D Rage, 3D Rage II and 3D Rage Pro)

Avance Logic

ALG2101, ALG2228, ALG2301, ALG2302, ALG2308, ALG2401

Chips & Technologies

65520, 65525, 65530, 65535, 65540, 65545, 65546, 65548, 65550, 65554, 65555, 68554, 69000, 64200, 64300

Cirrus Logic

CLGD5420, CLGD5422, CLGD5424, CLGD5426, CLGD5428, CLGD5429, CLGD5430, CLGD5434, CLGD5436, CLGD5440, CLGD5446, CLGD5462, CLGD5464, CLGD5465, CLGD5480, CLGD6205, CLGD6215, CLGD6225, CLGD6235, CLGD6410, CLGD6412, CLGD6420, CLGD6440, CLGD7541(*), CLGD7543(*), CLGD7548(*), CLGD7555(*)

Cyrix

MediaGX, MediaGXm

Compaq

AVGA

TOC

Digital Equipment Corporation

TGA

Epson

SPC8110

Genoa

GVGA

IBM

8514/A (and true clones), XGA-2

IIT

AGX-014, AGX-015, AGX-016

Matrox

MGA2064W (Millennium), MGA1064SG (Mystique and Mystique 220), MGA2164W (Millennium II PCI and AGP), G100, G200

MX

MX68000(*), MX680010(*)

NCR

77C22(*), 77C22E(*), 77C22E+(*)

NeoMagic

2200, 2160, 2097, 2093, 2090, 2070

Number Nine

I128 (series I and II), Revolution 3D (T2R)

NVidia/SGS Thomson

NV1, STG2000, RIVA128, Riva TNT

OAK

OTI067, OTI077, OTI087

RealTek

RTG3106(*)

Rendition

V1000, V2x00

S3

86C911, 86C924, 86C801, 86C805, 86C805i, 86C928, 86C864, 86C964, 86C732, 86C764, 86C765, 86C767, 86C775, 86C785, 86C868, 86C968, 86C325, 86C357, 86C375, 86C375, 86C385, 86C988, 86CM65, 86C260

TOC

SiS

86C201, 86C202, 86C205, 86C215, 86C225, 5597, 5598, 6326

3DLabs

GLINT 500TX, GLINT MX, Permedia, Permedia 2, Permedia 2v

Tseng

ET3000, ET4000AX, ET4000/W32, ET4000/W32i, ET4000/W32p, ET6000, ET6100

Trident

TVGA8800CS, TVGA8900B, TVGA8900C, TVGA8900CL, TVGA9000, TVGA9000i, TVGA9100B, TVGA9200CXR, Cyber9320(*), TVGA9400CXi, TVGA9420, TGUI9420DGi, TGUI9430DGi, TGUI9440AGi, TGUI9660XGi, TGUI9680, ProVidia 9682, ProVidia 9685(*), Cyber 9382, Cyber 9385, Cyber 9388, 3DImage975, 3DImage985, Cyber 9397, Cyber 9520

Video 7/Headland Technologies

HT216-32(*)

Weitek

P9000, P9100

Western Digital/Paradise

PVGA1

Western Digital

WD90C00, WD90C10, WD90C11, WD90C24, WD90C24A, WD90C30, WD90C31, WD90C33

(*) Note, chips marked in this way have either limited support or the drivers for them are not actively maintained.

All of the above are supported in both 256 color, and some are supported in mono and 16 color modes, and some are supported an higher color depths.

The monochrome server also supports generic VGA cards, using 64k of video memory in a single bank, the Hercules monochrome card, the Hyundai HGC1280, Sigma LaserView, Visa and Apollo monochrome cards.

The VGA16 server supports memory banking with the ET4000, Trident, ATI, NCR, OAK and Cirrus 6420 chipsets allowing virtual display sizes up to about 1600x1200 (with 1MB of video memory). For other chipsets the display size is limited to approximately 800x600.

You can find an up-to-date list of supported cards at http://www.xfree86.org/3.3.3/README3.html.

Video cards using these chipsets are supported on all bus types, including VLB and PCI.

One problem faced by the XFree86 developers is that some video card manufacturers use non-standard mechanisms for determining clock frequencies used to drive the card. Some of these manufacturers either don't release specifications describing how to program the card, or they require developers to sign a

non-disclosure statement to obtain the information. This would obviously restrict the free distribution of the XFree86 software, something that the XFree86 development team is not willing to do. For a long time, this has been a problem with certain video cards manufactured by Diamond, but as of release 3.1 of XFree86, Diamond has started to work with the development team to release free drivers for these cards.

The suggested setup for XFree86 under Linux is a 486 or better with at least 8 megabytes of RAM, and a video card with a chipset listed above. For optimal performance, we suggest using an accelerated card, such as an S3-chipset card. You should check the documentation for XFree86 and verify that your particular card is supported before taking the plunge and purchasing expensive hardware.

As a side note, the personal Linux system of Matt Welsh (this FAQ's originator) was a 486DX2-66, 20 megabytes of RAM, equipped with a VLB S3-864 chipset card with 2 megabytes of DRAM. He ran X benchmarks on this machine as well as on Sun Sparc IPX workstations. The Linux system was roughly 7 times faster than the Sparc IPX (for the curious, XFree86-3.1 under Linux, with this video card, runs at around 171,000 xstones; the Sparc IPX at around 24,000). In general, XFree86 on a Linux system with an accelerated SVGA card will give you much greater performance than that found on commercial UNIX workstations (which usually employ simple framebuffers for graphics).

Your machine will need at least 4 megabytes of physical RAM, and 16 megabytes of virtual RAM (for example, 8 megs physical and 8 megs swap). Remember that the more physical RAM that you have, the less that the system will swap to and from disk when memory is low. Because swapping is inherently slow (disks are very slow compared to memory), having 8 megabytes of RAM or more is necessary to run XFree86 comfortably. 16 is better. A system with 4 megabytes of physical RAM could run *much* (up to 10 times) more slowly than one with 8 megs or more.

---

3. Installing XFree86

It's quite likely that you obtained XFree86 as part of a Linux distribution, in which case downloading the software separately is not necessary. Or you may be able to get RPM binary packages built for your machine, in which case you can just install those using rpm(1). In either case you can skip this the rest of this section.

The Linux binary distribution of XFree86 can be found on a number of FTP sites. On the XFree86 site it's under ftp://ftp.xfree86.org/pub/XFree86/current/binaries/.

Before doing anything else, download and run the `preinst.sh' shell script first. This may tell you about prerequisites you'll need to have in place before contibuing your installation.

If you are downloading XFree86 directly, this table lists the files in the XFree86-3.3 distribution.

One of the following servers is required:

X8514.tgz

Server for 8514-based boards.

XAGX.tgz

Server for AGX-based boards.

XI128.tgz

Server for the Number Nine Imagine 128.

XMach32.tgz

Server for Mach32-based boards.

XMach64.tgz

Server for Mach64-based boards.

XMach8.tgz

Server for Mach8-based boards.

XMono.tgz

Server for monochrome video modes.

XP9K.tgz

Server for P9000-based boards.

XS3.tgz

Server for S3-based boards.

XS3V.tgz

TOC

Server for the S3 ViRGE and ViRGE/VX (considered beta)

XSVGA.tgz

Server for Super VGA-based boards.

XW32.tgz

Server for ET4000/W32-based boards.

If you don't know which one to take, take the VGA16 server, XVG16.tgz. You want to download this one anyway, because you'll need it to run the auto-configuration utility in the next step.

All of the following files are required:

preinst.sh

Pre-installation script

postinst.sh

Post-installation script

Xbin.tgz

The rest of the X11R6 binaries.

Xcfg.tgz

Config files for `xdm`, `xinit` and `fs`.

Xdoc.tgz

Documentation.

Xman.tgz

Manual pages.

Xfnts.tgz

75dpi, misc and PEX fonts

Xlib.tgz

Shared X libraries and support files.

Xset.tgz

XF86Setup utility.

XVG16.tgz

Server for VGA/EGA-based boards.

The following files are optional:

Xf100.tgz

100dpi fonts

Xfcyr.tgz

TOC

Cyrillic fonts

Xfnon.tgz

Other fonts (Chinese, Japanese, Korean, Hebrew)

Xfscl.tgz

Scalable fonts (Speedo and Type1)

Xfsrv.tgz

Font server and config files

Xprog.tgz

X header files, config files and compile-time libs

Xlkit.tgz

X server LinkKit

Xlk98.tgz

PC98 X server LinkKit

Xnest.tgz

Nested X server

Xprt.tgz

X print server

Xvfb.tgz

Virtual framebuffer X server

Xps.tgz

PostScript version of the documentation

Xhtml.tgz

HTML version of the documentation

The XFree86 directory should contain release notes for the current version in `RELNOTES`. Consult those for installation details

All that is required to install XFree86 is to obtain the above files, create the directory `/usr/X11R6` (as `root`), and unpack the files from `/usr/X11R6` with a command such as:

```
gzip -dc Xbin.tgz | tar xfB -
```

Remember that these tar files are packed relative to `/usr/X11R6`, so it's important to unpack the files there.

You need to make sure that `/usr/X11R6/bin` is on your path. This can be done by editing your system default `/etc/profile` or `/etc/csh.login` (based on the shell that you, or other users on

your system, use). Or you can simply add the directory to your personal path by modifying `.bashr` `.cshrc`, based on your shell.

You also need to make sure that `/usr/X11R6/lib` can be located by `ld.so`, the runtime linker. To do this, add the line

```
/usr/X11R6/lib
```

to the file `/etc/ld.so.conf`, and run `/sbin/ldconfig`, as `root`.

---

4. Configuring XFree86

4.1 Normal Configuration

Configuring XFree86 to use your mouse, keyboard, monitor, and video card correctly used to be something of a black art, requiring extensive hand-hacking of a complex configuration file. No more; the 3.2/3.3 release made the process nearly trivial. All you do is fire up the program `XF86Setup`.

This program depends on the fact that all new PC hardware these days ships with EGA/VGA capable monitors. It invokes the VGA16 server and uses it to bring up X in a lowest-common-denominator 640x480 mode. Then it runs an interactive program that walks you through a series of five configuration panels -- mouse, keyboard, (video) card, monitor, and `other' (miscellaneous server options). The whole process is quite painless.

(If you're running Red Hat Linux, you may see a different program called `xf86config`. This works fairly similarly to `XF86Setup` but does not itself use an X interface and the VGA16 server.)

One minor point to keep in mind is that, if you're like most people using a current PC, your keyboard is actually what XF86Setup calls `Generic 102-key PC (intl)' rather than the default `Generic 101-key PC'. If you pick the default (101) the key cluster on the extreme right of your keyboard (numeric keypad and friends) may stop working.

If you're not sure of your monitor type, you can try the listed ones in succession. Work your way from top down (upper choices involve lower dot-clock speeds and are less demanding on the hardware). Back off if you get hash or a seriously distorted picture. Minor distortions (picture slightly too large, slightly too small, or slightly off-center) are no problem; you'll get a chance to correct those immediately by fine-tuning the mode.

And, when the program brings up `xvidtune` to allow you to tweak your video mode, don't let the initial warning box make you nervous. Modern multisync monitors (unlike their fixed-frequency predecessors) are not easy to damage this way.

XF86Config may assume that your mouse device is /dev/mouse. If you find this doesn't work, you may need to link /dev/mouse to whatever /dev/ca[01] the mouse is on. If you find that XFree86 gives you a "mouse busy" error when `gpm` is running, you may need to link to /dev/ttyS[01] instead.

The configuration process operates by selecting a server corresponding to to the general type of your video card (such as `XF86_VGA16`, `XF86_Mach64`, or `XF86_S3`) and configuring an `XF86Config` that the server will read on startup to get specific parameters for your installation. The location of `XF86Config` varies depending on your OS, but one place to look for it is `/etc/X11`.

On older versions of XFree86, `XF86Setup` links the `X' command directly to the chosen server. In recent versions, `X' is instead linked to a set-user-id wrapper program called Xwrapper. The idea is that all the setuid root stuff gets localized in Xwrapper so the server doesn't have to run setuid root.

4.2 Troubleshooting

Occasionally, something will not be quite right when you initially fire up the X server. This is almost always caused by a problem in your configuration file. Usually, the monitor timing values are off, or the video card dot clocks set incorrectly. Minor problems can be fixed with `xvidtune`; a really garbled screen usually means you need to go back into XF86Setup and choose a less capable monitor type.

If your display seems to roll, or the edges are fuzzy, this is a clear indication that the monitor timing values or dot clocks are wrong. Also be sure that you are correctly specifying your video card chipset, as well as other options in the `Device` section of `XF86Config`. Be absolutely certain that you are using the right X server and that `/usr/X11R6/bin/X` is a symbolic link to this server.

If all else fails, try to start X ``bare''; that is, use a command such as:

```
X > /tmp/x.out 2>&1
```

You can then kill the X server (using the `ctrl-alt-backspace` key combination) and examine the contents of `/tmp/x.out`. The X server will report any warnings or errors---for example, if your video card doesn't have a dot clock corresponding to a mode supported by your monitor.

Remember that you can use `ctrl-alt-numeric +` and `ctrl-alt-numeric -` to switch between the video modes listed on the `Modes` line of the `Screen` section of `XF86Config`. If the highest resolution mode doesn't look right, try switching to lower resolutions. This will let you know, at least, that those parts of your X configuration are working correctly.

Also, check the vertical and horizontal size/hold knobs on your monitor. In many cases it is necessary to adjust these when starting up X. For example, if the display seems to be shifted slightly to one side, you can usually correct this using the monitor controls.

The USENET newsgroup `comp.windows.x.i386unix` is devoted to discussions about XFree86, as is `comp.os.linux.x`. It might be a good idea to watch that newsgroup for postings relating to your video configuration---you might run across someone with the same problems as your own.

4.3 Custom Configuration

You will need to hand-hack your X configuration to get optimal performance if your monitor can support 1600x1200 -- the highest canned resolution XF86Setup supports is 1280x1024.

If you want to hand-hack your video configuration for this or any other reason, go see the LDP's XFree86 Video Timings HOWTO, http://metalab.unc.edu/LDP/HOWTO/XFree86-Video-Timings-HOWTO.html. (This is the up-to-date HTML version of XFree86's `Videomodes.doc' file.)

4.4 Using 16-bit Color

By default, X uses 8-bit color depth giving 256 colors. To circumvent this restriction, many applications allocate their own colormaps, resulting in sudden color jumps when the cursor moves between two windows each having a color map of its own. The Arena WWW browser does it this way.

If you want to use advanced graphic applications 256 colors may be not sufficient. You may need to go to 16-bit color depth (65,536 colors). But beware, not all applications will work with 16bit colors.

You can use 16bit color depth with 65K different colors simply by starting X with

---

```
startx -- -bpp 16
```

---

or putting

```
exec X :0 -bpp 16
```

into your .xserverrc file. In order for this to work, however, you need to have a

```
screen
```

section in your XF86Config with

```
DefaultColorDepth 16
```

If you're using xdm, you may need to change the Xservers file, which is probably located in /etc/X11/xdm/. A typical configuration has just one uncommented line, looking something like

```
:0 local /usr/X11R6/bin/X
```

Add

```
-bpp 16
```

to the startup options:

```
:0 local /usr/X11R6/bin/X -bpp 16
```

Also, you'll need to add to the `screen' section of your X configuration a Display part with a Depth value of 16. Generally you can just duplicate the 8-bit Display section and change the Depth field.

More colors makes your video card transfer more data during the same time. If your video card cannot cope, then either the resolution has or the refresh rate has to be reduced. By default, XFree reduces the resolution. If you want to keep the resolution and reduce the refresh rate, you must insert a new appropriate Modeline into your XF86Config file which defines that resolution with a lower refresh rate. For instance replace the old value

```
Modeline "1024x768"  75  1024 1048 1184 1328 768 771 777 806 -hsync -vsync
```

with

```
Modeline "1024x768"  65  1024 1032 1176 1344 768 771 777 806 -hsync -vsync.
```

The magic numbers 75 and 65 are the respective clock rates which you find reported by X in your .X.err file. Consult the monitors file in the XF86 documentation for Modelines suitable to the maximum clock rate your video card can deliver under 16bit color depth.

5. Running XFree86

With your `XF86Config` file configured, you're ready to fire up the X server and give it a spin. First, be sure that `/usr/X11R6/bin` is on your path.

The command to start up XFree86 is

```
startx
```

This is a front-end to `xinit` (in case you're used to using `xinit` on other UNIX systems).

This command will start the X server and run the commands found in the file `.xinitrc` in your home directory. `.xinitrc` is just a shell script containing X clients to run. If this file does not exist, the system default `/usr/X11R6/lib/X11/xinit/xinitrc` will be used.

A standard `.xinitrc` file looks like this:

```
#!/bin/sh

xterm -fn 7x13bold -geometry 80x32+10+50 &
xterm -fn 9x15bold -geometry 80x34+30-10 &
oclock -geometry 70x70-7+7 &
xsetroot -solid midnightblue &

exec fvwm2
```

This script will start up two `xterm` clients, an `oclock`, and set the root window (background) color to `midnightblue`. It will then start up `fvwm2`, the window manager. Note that `fvwm2` is executed with the shell's `exec` statement; this causes the `xinit` process to be replaced with `fvwm2`. Once the `fvwm2` process exits, the X server will shut down. You can cause `fvwm2` to exit by using the root menus: depress mouse button 1 on the desktop background---this will display a pop up menu which will allow you to `Exit Fvwm2`.

Be sure that the last command in `.xinitrc` is started with `exec`, and that it is not placed into the background (no ampersand on the end of the line). Otherwise the X server will shut down as soon as it has started the clients in the `.xinitrc` file.

Alternately, you can exit X by pressing `ctrl-alt-backspace` in combination. This will kill the X server directly, exiting the window system.

The above is a very, very simple desktop configuration. Many wonderful programs and configurations are available with a bit of work on your `.xinitrc` file.

If you are new to the X Window System environment, we strongly suggest picking up a book such as *The Joy of X: An Overview of the X Window System* by Niall Mansfield (Addison-Wesley 1993, ISBN 0201-565129). Using and configuring X is far too in-depth to cover here. See the man pages for `xterm`,

`oclock`, and `fvwm2` for clues on getting started.

5.1 Terms of Use

This document is copyright 1996 by Eric S. Raymond. You may use, disseminate, and reproduce it freely, provided you:

- Do not omit or alter this copyright notice.
- Do not omit or alter the version number and date.
- Do not omit or alter the document's pointer to the current WWW version.
- Clearly mark any condensed or altered versions as such.

These restrictions are intended to protect potential readers from stale or mangled versions. If you think you have a good case for an exception, ask me.

5.2 Acknowledgements

This document was originated by Matt Welsh in the dim and backward abysm of time. Thanks, Matt!

---

6. Translations

- Italian
- Slovenian
- Croatian
- Dutch

XFree86 Video Timings HOWTO

Eric S. Raymond <esr@thyrsus.com>

v3.6, 13 June 1999

How to compose a mode line for your card/monitor combination under XFree86. The XFree86 distribution now includes good facilities for configuring most standard combinations; this document is mainly useful if you are tuning a custom mode line for a high-performance monitor or very unusual hardware. It may also help you in using kvideogen to generate mode lines, or xvidtune to tweak a standard mode that is not quite right for your monitor.

1. Disclaimer

2. Introduction

3. Tools for Automatic Computation

4. How Video Displays Work

5. Basic Things to Know about your Display and Adapter

- 5.1 The monitor sync frequencies
- 5.2 The monitor's video bandwidth
- 5.3 The card's dot clock
- 5.4 What these basic statistics control

6. Interpreting the Basic Specifications

- 6.1 About Bandwidth
- 6.2 Sync Frequencies and the Refresh Rate:

TOC

7. Tradeoffs in Configuring your System

8. Memory Requirements

9. Computing Frame Sizes

10. Black Magic and Sync Pulses

- 10.1 Horizontal Sync:
- 10.2 Vertical Sync:

11. Putting it All Together

12. Overdriving Your Monitor

13. Using Interlaced Modes

14. Questions and Answers

15. Fixing Problems with the Image.

- 15.1 The image is displaced to the left or right
- 15.2 The image is displaced up or down
- 15.3 The image is too large both horizontally and vertically
- 15.4 The image is too wide (too narrow) horizontally
- 15.5 The image is too deep (too shallow) vertically

16. Plotting Monitor Capabilities

17. Credits

1. Disclaimer

You use the material herein SOLELY AT YOUR OWN RISK. It is possible to harm both your monitor and yourself when driving it outside the manufacturer's specs. Read Overdriving Your Monitor for detailed cautions. Any damage to you or your monitor caused by overdriving it are your problem.

The most up-to-date version of this HOWTO can be found at the Linux Documentation Project web page.

Please direct comments, criticism, and suggestions for improvement to esr@snark.thyrsus.com. Please do *not* send email pleading for a magic solution to your special monitor problem, as doing so will only burn up my time and frustrate you -- everything I know about the subject is already in here.

2. Introduction

The XFree86 server allows users to configure their video subsystem and thus encourages best use of existing hardware. This tutorial is intended to help you learn how to generate your own timing numbers to make optimum use of your video card and monitor.

We'll present a method for getting something that works, and then show you how you can experiment starting from that base to develop settings that optimize for your taste.

If you already have a mode that almost works (in particular, if one of predefined VESA modes gives you a stable display but one that's displaced right or left, or too small, or too large) you can go straight to the section on Fixing Problems with the Image. This will enlighten you on ways to tweak the timing numbers to achieve particular effects.

Don't assume that you need to get all the way into mode tuning just because your X comes up with a scrambled display first time after installation; it may be that most of the factory mode lines are OK and you just happened to default to one that doesn't fit your hardware. Instead, cycle through all your installed modes with CTRL-ALT-KP+. If some of the modes look OK, try commenting out all but a 640x480 and check that that mode works. If it does then uncomment a couple of other modes, e.g. an 800x600 and a 1024x768 at a frequency that your monitor should be able to handle.

3. Tools for Automatic Computation

Starting with XFree86 3.2, XFree86 provides an **XF86Setup**(1) program that makes it easy to generate a working monitor mode interactively, without messing with video timing number directly. So you shouldn't actually need to calculate a base monitor mode in most cases. Unfortunately, **XF86Setup**(1) has some limitations; it only knows about standard video modes up to 1280x1024. If you have a very high-performance monitor capable of 1600x1200 or more you will still have to compute your base monitor mode yourself.

There is a KDE tool called KVideoGen that computes modelines from basic monitor and card statistics. I've experimented with generating modelines from it, but haven't tried them live. Note that its horizontal and vertical `refresh rate' parameters are the same as the sync frequencies HSF and VSF we describe below. The `horizontal sync pulse' number seems to be a sync pulse width in microseconds, HSP (with the tool assuming fixed `front porch' HGT1 and `back porch' HGT2 values). If you don't know the `horizontal sync pulse' number it's safe to use the default.

Recent versions of XFree86 provide a tool called **xvidtune**(1) which you will probably find quite useful for testing and tuning monitor modes. It begins with a gruesome warning about the possible consequences of mistakes with it. If you pay careful attention to this document and learn what is behind the pretty numbers in xvidtune's boxes, you will become able to use xvidtune effectively and with confidence.

If you have **xvidtune**(1), you'll be able to test new modes on the fly, without modifying your X configuration files or even rebooting your X server. Otherwise, XFree86 allows you to hot-key between different modes defined in Xconfig (see XFree86.man for details). Use this capabilty to save yourself hassles! When you want to test a new mode, give it a unique mode label and add it to the *end* of your hot-key list. Leave a known-good mode as the default to fall back on if the test mode doesn't work.

Towards the end of this document, we include a `modeplot' script that you can use to produce an analog graph of available modes. This is not directly helpful for generating modelines, but it may help you better understand the relationships that define them.

4. How Video Displays Work

Knowing how the display works is essential to understanding what numbers to put in the various fields in the file Xconfig. Those values are used in the lowest levels of controlling the display by the XFree86 server.

The display generates a picture from a series of dots. The dots are arranged from left to right to form lines. The lines are arranged from top to bottom to form the picture. The dots emit light when they are struck by the electron beam inside the display. To make the beam strike each dot for an equal amount of time, the beam is swept across the display in a constant pattern.

The pattern starts at the top left of the screen, goes across the screen to the right in a straight line, and stops temporarily on the right side of the screen. Then the beam is swept back to the left side of the display, but down one line. The new line is swept from left to right just as the first line was. This pattern is repeated until the bottom line on the display has been swept. Then the beam is moved from the bottom right corner of the display to the top left corner, and the pattern is started over again.

There is one variation of this scheme known as interlacing: here only every second line is swept during one half-frame and the others are filled in in during a second half-frame.

Starting the beam at the top left of the display is called the beginning of a frame. The frame ends when the beam reaches the the top left corner again as it comes from the bottom right corner of the display. A frame is made up of all of the lines the beam traced from the top of the display to the bottom.

If the electron beam were on all of the time it was sweeping through the frame, all of the dots on the display would be illuminated. There would be no black border around the edges of the display. At the edges of the display the picture would become distorted because the beam is hard to control there. To reduce the distortion, the dots around the edges of the display are not illuminated by the beam even though the beam may be pointing at them. The viewable area of the display is reduced this way.

Another important thing to understand is what becomes of the beam when no spot is being painted on the visible area. The time the beam would have been illuminating the side borders of the display is used for sweeping the beam back from the right edge to the left and moving the beam down to the next line. The time the beam would have been illuminating the top and bottom borders of the display is used for moving the beam from the bottom-right corner of the display to the top-left corner.

The adapter card generates the signals which cause the display to turn on the electron beam at each dot to generate a picture. The card also controls when the display moves the beam from the right side to the left and down a line by generating a signal called the horizontal sync (for synchronization) pulse. One horizontal sync pulse occurs at the end of every line. The adapter also generates a vertical sync pulse which signals the display to move the beam to the top-left corner of the display. A vertical sync pulse is generated near the end of every frame.

The display requires that there be short time periods both before and after the horizontal and vertical sync pulses so that the position of the electron beam can stabilize. If the beam can't stabilize, the picture will

not be steady.

In a later section, we'll come back to these basics with definitions, formulas and examples to help you use them.

---

5. Basic Things to Know about your Display and Adapter

There are some fundamental things you need to know before hacking an Xconfig entry. These are:

- your monitor's horizontal and vertical sync frequency options
- your monitor's bandwidth
- your video adapter's driving clock frequencies, or "dot clocks"

5.1 The monitor sync frequencies

The horizontal sync frequency is just the number of times per second the monitor can write a horizontal scan line; it is the single most important statistic about your monitor. The vertical sync frequency is the number of times per second the monitor can traverse its beam vertically.

Sync frequencies are usually listed on the specifications page of your monitor manual. The vertical sync frequency number is typically calibrated in Hz (cycles per second), the horizontal one in KHz (kilocycles per second). The usual ranges are between 50 and 150Hz vertical, and between 31 and 135KHz horizontal.

If you have a multisync monitor, these frequencies will be given as ranges. Some monitors, especially lower-end ones, have multiple fixed frequencies. These can be configured too, but your options will be severely limited by the built-in monitor characteristics. Choose the highest frequency pair for best resolution. And be careful --- trying to clock a fixed-frequency monitor at a higher speed than it's designed for can easily damage it.

Earlier versions of this guide were pretty cavalier about overdriving multisync monitors, pushing them past their nominal highest vertical sync frequency in order to get better performance. We have since had more reasons pointed out to us for caution on this score; we'll cover those under Overdriving Your Monitor below.

5.2 The monitor's video bandwidth

Your monitor's video bandwidth should be included on the manual's spec page. If it's not, look at the monitor's higest rated resolution. As a rule of thumb, here's how to translate these into bandwidth estimates (and thus into rough upper bounds for the dot clock you can use):

```
640x480                 25
800x600                 36
1024x768                65
1024x768 interlaced     45
1280x1024               110
1600x1200               185
```

BTW, there's nothing magic about this table; these numbers are just the lowest dot clocks per resolution in the standard XFree86 Modes database (except for the last, which I interpolated). The bandwidth of your monitor may actually be higher than the minimum needed for its top resolution, so don't be afraid to try a dot clock a few MHz higher.

Also note that bandwidth is seldom an issue for dot clocks under 65MHz or so. With an SVGA card and most

TOC

hi-res monitors, you can't get anywhere near the limit of your monitor's video bandwidth. The following are examples:

Brand	Video Bandwidth
NEC 4D	75Mhz
Nano 907a	50Mhz
Nano 9080i	60Mhz
Mitsubishi HL6615	110Mhz
Mitsubishi Diamond Scan	100Mhz
IDEK MF-5117	65Mhz
IOCOMM Thinksync-17 CM-7126	136Mhz
HP D1188A	100Mhz
Philips SC-17AS	110Mhz
Swan SW617	85Mhz
Viewsonic 21PS	185Mhz
PanaSync/Pro P21	220Mhz

Even low-end monitors usually aren't terribly bandwidth-constrained for their rated resolutions. The NEC Multisync II makes a good example --- it can't even display 800x600 per its spec. It can only display 800x560. For such low resolutions you don't need high dot clocks or a lot of bandwidth; probably the best you can do is 32Mhz or 36Mhz, both of them are still not too far from the monitor's rated video bandwidth of 30Mhz.

At these two driving frequencies, your screen image may not be as sharp as it should be, but definitely of tolerable quality. Of course it would be nicer if NEC Multisync II had a video bandwidth higher than, say, 36Mhz. But this is not critical for common tasks like text editing, as long as the difference is not so significant as to cause severe image distortion (your eyes would tell you right away if this were so).

5.3 The card's dot clock

Your video adapter manual's spec page will usually give you the card's maximum dot clock (that is, the total number of pixels per second it can write to the screen).

If you don't have this information, the X server will get it for you. Recent versions of the X servers all support a --probeonly option that prints out this information and exits without actually starting up X or changing the video mode.

If you don't have -probeonly, don't depair. Even if your X locks up your monitor, it will emit a line of clock and other info to standard error. If you redirect this to a file, it should be saved even if you have to reboot to get your console back.

The probe result or startup message should look something like one of the following examples:

If you're using XFree86:

```
Xconfig: /usr/X11R6/lib/X11/Xconfig
(**) stands for supplied, (--) stands for probed/default values
(**) Mouse: type: MouseMan, device: /dev/ttyS1, baudrate: 9600
Warning: The directory "/usr/andrew/X11fonts" does not exist.
         Entry deleted from font path.
(**) FontPath set to "/usr/lib/X11/fonts/misc/,/usr/lib/X11/fonts/75dpi/"
```

TOC

```
(--) S3: card type: 386/486 localbus
(--) S3: chipset:   924
                    ---
    Chipset -- this is the exact chip type; an early mask of the 86C911

(--) S3: chipset driver: s3_generic
(--) S3: videoram:  1024k
                    -----
         Size of on-board frame-buffer RAM

(**) S3: clocks:  25.00  28.00  40.00   3.00  50.00  77.00  36.00  45.00
(**) S3: clocks:   0.00   0.00  79.00  31.00  94.00  65.00  75.00  71.00
                  -------------------------------------------------------
                              Possible driving frequencies in MHz

(--) S3: Maximum allowed dot-clock: 110MHz
                                    ------
                                  Bandwidth
(**) S3: Mode "1024x768": mode clock =  79.000, clock used =  79.000
(--) S3: Virtual resolution set to 1024x768
(--) S3: Using a banksize of 64k, line width of 1024
(--) S3: Pixmap cache:
(--) S3: Using 2 128-pixel 4 64-pixel and 8 32-pixel slots
(--) S3: Using 8 pages of 768x255 for font caching
```

If you're using SGCS or X/Inside X:

```
WGA: 86C911 (mem: 1024k clocks: 25 28 40 3 50 77 36 45 0 0 79 31 94 65 75 71)
---  ------       -----         ---------------------------------------------
 |     |            |                 Possible driving frequencies in MHz
 |     |            +-- Size of on-board frame-buffer RAM
 |     +-- Chip type
 +-- Server type
```

Note: do this with your machine unloaded (if at all possible). Because X is an application, its timing loops can collide with disk activity, rendering the numbers above inaccurate. Do it several times and watch for the numbers to stabilize; if they don't, start killing processes until they do. Your mouse daemon process, if you have one, is particularly likely to trip you up (that's gpm for Linux users, mousemgr for SVr4 users).

In order to avoid the clock-probe inaccuracy, you should clip out the clock timings and put them in your Xconfig as the value of the Clocks property --- this suppresses the timing loop and gives X an exact list of the clock values it can try. Using the data from the example above:

```
wga
        Clocks  25 28 40 3 50 77 36 45 0 0 79 31 94 65 75 71
```

On systems with a highly variable load, this may help you avoid mysterious X startup failures. It's possible for X to come up, get its timings wrong due to system load, and then not be able to find a matching dot clock in its config database --- or find the wrong one!

5.4 What these basic statistics control

The sync frequency ranges of your monitor, together with your video adapter's dot clock, determine the ultimate resolution that you can use. But it's up to the driver to tap the potential of your hardware. A superior hardware combination without an equally competent device driver is a waste of money. On the other hand, with a versatile device driver but less capable hardware, you can push the hardware's envelope a little. This is the design philosophy of XFree86.

You should match the dot clock you use to the monitor's video bandwidth. There's a lot of give here, though --- some monitors can run as much as 30% over their nominal bandwidth. The risks here have to do with exceeding the monitor's rated vertical-sync frequency; we'll discuss them in detail below.

Knowing the bandwidth will enable you to make more intelligent choices between possible configurations. It may affect your display's visual quality (especially sharpness for fine details).

---

6. Interpreting the Basic Specifications

This section explains what the specifications above mean, and some other things you'll need to know. First, some definitions. Next to each in parens is the variable name we'll use for it when doing calculations

horizontal sync frequency (HSF)

Horizontal scans per second (see above).

vertical sync frequency (VSF)

Vertical scans per second (see above). Mainly important as the upper limit on your refresh rate.

dot clock (DCF)

More formally, `driving clock frequency'; The frequency of the crystal or VCO on your adaptor --- the maximum dots-per-second it can emit.

video bandwidth (VB)

The highest frequency you can feed into your monitor's video input and still expect to see anything discernible. If your adaptor produces an alternating on/off pattern, its lowest frequency is half the DCF, so in theory bandwidth starts making sense at DCF/2. For tolerately crisp display of fine details in the video image, however, you don't want it much below your highest DCF, and preferably higher.

frame length (HFL, VFL)

Horizontal frame length (HFL) is the number of dot-clock ticks needed for your monitor's electron gun to scan one horizontal line, *including the inactive left and right borders*. Vertical frame length (VFL) is the number of scan lines in the *entire* image, including the inactive top and bottom borders.

screen refresh rate (RR)

The number of times per second your screen is repainted (this is also called "frame rate"). Higher frequencies are better, as they reduce flicker. 60Hz is good, VESA-standard 72Hz is better. Compute it as

```
RR = DCF / (HFL * VFL)
```

Note that the product in the denominator is *not* the same as the monitor's visible resolution, but typically somewhat larger. We'll get to the details of this below.

The rates for which interlaced modes are usually specified (like 87Hz interlaced) are actually the half-frame rates: an entire screen seems to have about that flicker frequency for typical displays, but every single line is refreshed only half as often.

TOC

For calculation purposes we reckon an interlaced display at its full-frame (refresh) rate, i.e. 43.5Hz. The quality of an interlaced mode is better than that of a non-interlaced mode with the same full-frame rate, but definitely worse then the non-interlaced one corresponding to the half-frame rate.

6.1 About Bandwidth

Monitor makers like to advertise high bandwidth because it constrains the sharpness of intensity and color changes on the screen. A high bandwidth means smaller visible details.

Your monitor uses electronic signals to present an image to your eyes. Such signals always come in in wave form once they are converted into analog form from digitized form. They can be considered as combinations of many simpler wave forms each one of which has a fixed frequency, many of them are in the Mhz range, eg, 20Mhz, 40Mhz, or even 70Mhz. Your monitor video bandwidth is, effectively, the highest-frequency analog signal it can handle without distortion.

For our purposes, video bandwidth is mainly important as an approximate cutoff point for the highest dot clock you can use.

6.2 Sync Frequencies and the Refresh Rate:

Each horizontal scan line on the display is just the visible portion of a frame-length scan. At any instant there is actually only one dot active on the screen, but with a fast enough refresh rate your eye's persistence of vision enables you to "see" the whole image.

Here are some pictures to help:

```
 _________________________
|                         |     The horizontal sync frequency
|->->->->->->->->->->->->-> |     is the number of times per
|                        )|     second that the monitor's
|<-----<-----<-----<--- |     electron beam can trace
|                         |     a pattern like this
|                         |
|                         |
|                         |
|_________________________|

 _________________________
|          ^              |     The vertical sync frequency
|         ^ |             |     is the number of times per
|         | v             |     second that the monitor's
|         ^ |             |     electron beam can trace
|         | |             |     a pattern like this
|         ^ |             |
|         | v             |
```

```
     |        ^ |                |
     |________|_v_______________|
```

Remember that the actual raster scan is a very tight zigzag pattern; that is, the beam moves left-right and at the same time up-down.

Now we can see how the dot clock and frame size relates to refresh rate. By definition, one hertz (hz) is one cycle per second. So, if your horizontal frame length is HFL and your vertical frame length is VFL, then to cover the entire screen takes (HFL * VFL) ticks. Since your card emits DCF ticks per second by definition, then obviously your monitor's electron gun(s) can sweep the screen from left to right and back and from bottom to top and back DCF / (HFL * VFL) times/sec. This is your screen's refresh rate, because it's how many times your screen can be updated (thus *refreshed*) per second!

You need to understand this concept to design a configuration which trades off resolution against flicker in whatever way suits your needs.

For those of you who handle visuals better than text, here is one:

```
         RR                                   VB
          |   min HSF                     max HSF |
          |    |              R1        R2  |      |
max VSF -+----|------------/----------/---|------+----- max VSF
          |    |:::::::::::/:::::::::::/:::::\     |
          |    \:::::::::::/:::::::::::/:::::::\    |
          |     |::::::::::/:::::::::::/::::::::::|   |
          |     |:::::::::/:::::::::::/::::::::::\   |
          |     \::::::::/:::::::::::/:::::::::::::\  |
          |      \::::::/:::::::::::/:::::::::::::::| |
          |       |::/:::::::::::/:::::::::::::::::| |
          |        \/::::::::::/::::::::::::::::::::\|
          |        /\::::::::::/::::::::::::::::::::|
          |       /  \::::::::/::::::::::::::::::::::|\
          |      /    |::::::/::::::::::::::::::::::| |
          |     /     \::::/::::::::::::::::::::::::| \
min VSF -+----/-------\--/-----------------------|--\--- min VSF
          |   /         \/                       |   \
          +--/----------/\-----------------------+----\- DCF
            R1        R2  \                      |     \
                             min HSF             |    max HSF
                                                 VB
```

This is a generic monitor mode diagram. The x axis of the diagram shows the clock rate (DCF), the y axis represents the refresh rate (RR). The filled region of the diagram describes the monitor's capabilities: every point within this region is a possible video mode.

The lines labeled `R1' and `R2' represent a fixed resolutions (such as 640x480); they are meant to illustrate how one resolution can be realized by many different combinations of dot clock and refresh rate. The R2 line would represent a higher resolution than R1.

TOC

The top and bottom boundaries of the permitted region are simply horizontal lines representing the limiting values for the vertical sync frequency. The video bandwidth is an upper limit to the clock rate and hence is represented by a vertical line bounding the capability region on the right.

Under Plotting Monitor Capabilities) you'll find a program that will help you plot a diagram like this (but much nicer, with X graphics) for your individual monitor. That section also discusses the interesting part; the derivation of the boundaries resulting from the limits on the horizontal sync frequency.

---

7. Tradeoffs in Configuring your System

Another way to look at the formula we derived above is

```
DCF = RR * HFL * VFL
```

That is, your dot clock is fixed. You can use those dots per second to buy either refresh rate, horizontal resolution, or vertical resolution. If one of those increases, one or both of the others must decrease.

Note, though, that your refresh rate cannot be greater than the maximum vertical sync frequency of your monitor. Thus, for any given monitor at a given dot clock, there is a minimum product of frame lengths below which you can't force it.

In choosing your settings, remember: if you set RR too low, you will get mugged by screen flicker.

You probably do not want to pull your refresh rate below 60Hz. This is the flicker rate of fluorescent lights; if you're sensitive to those, you need to hang with 72Hz, the VESA ergonomic standard.

Flicker is very eye-fatiguing, though human eyes are adaptable and peoples' tolerance for it varies widely. If you face your monitor at a 90% viewing angle, are using a dark background and a good contrasting color for foreground, and stick with low to medium intensity, you *may* be comfortable at as little as 45Hz.

The acid test is this: open a xterm with pure white back-ground and black foreground using `xterm -bg white -fg black` and make it so large as to cover the entire viewable area. Now turn your monitor's intensity to 3/4 of its maximum setting, and turn your face away from the monitor. Try peeking at your monitor sideways (bringing the more sensitive peripheral-vision cells into play). If you don't sense any flicker or if you feel the flickering is tolerable, then that refresh rate is fine with you. Otherwise you better configure a higher refresh rate, because that semi-invisible flicker is going to fatigue your eyes like crazy and give you headaches, even if the screen looks OK to normal vision.

For interlaced modes, the amount of flicker depends on more factors such as the current vertical resolution and the actual screen contents. So just experiment. You won't want to go much below about 85Hz half frame rate, though.

So let's say you've picked a minimum acceptable refresh rate. In choosing your HFL and VFL, you'll have some room for maneuver.

8. Memory Requirements

Available frame-buffer RAM may limit the resolution you can achieve on color or gray-scale displays. It probably isn't a factor on displays that have only two colors, white and black with no shades of gray in between.

For 256-color displays, a byte of video memory is required for each visible dot to be shown. This byte contains the information that determines what mix of red, green, and blue is generated for its dot. To get the amount of memory required, multiply the number of visible dots per line by the number of visible lines. For a display with a resolution of 1024x768, this would be 1024 x 768 = 786432, which is the number of visible dots on the display. This is also, at one byte per dot, the number of bytes of video memory that will be necessary on your adapter card.

Thus, your memory requirement will typically be (HR * VR)/1024 Kbytes of VRAM, rounded up (it would come to 768K exactly in this example). If you have more memory than strictly required, you'll have extra for virtual-screen panning.

However, if you only have 512K on board yor video card, then you won't be able to use this resolution. Even if you have a good monitor, without enough video RAM, you can't take advantage of your monitor's potential. On the other hand, if your SVGA has one meg, but your monitor can display at most 800x600, then high resolution is beyond your reach anyway (see Using Interlaced Modes for a possible remedy).

Don't worry if you have more memory than required; XFree86 will make use of it by allowing you to scroll your viewable area (see the Xconfig file documentation on the virtual screen size parameter). Remember also that a card with 512K bytes of memory really doesn't have 512,000 bytes installed, it has 512 x 1024 = 524,288 bytes.

If you're running X/Inside using an S3 card, and are willing to live with 16 colors (4 bits per pixel), you can set depth 4 in Xconfig and effectively double the resolution your card can handle. S3 cards, for example, normally do 1024x768x256. You can make them do 1280x1024x16 with depth 4.

9. Computing Frame Sizes

Warning: this method was developed for multisync monitors. It will probably work with fixed-frequency monitors as well, but no guarantees!

Start by dividing DCF by your highest available HSF to get a horizontal frame length.

For example; suppose you have a Sigma Legend SVGA with a 65MHz dot clock, and your monitor has a 55KHz horizontal scan frequency. The quantity (DCF / HSF) is then 1181 (65MHz = 65000KHz; 65000/55 = 1181).

Now for our first bit of black magic. You need to round this figure to the nearest multiple of 8. This has to do with the VGA hardware controller used by SVGA and S3 cards; it uses an 8-bit register, left-shifted 3 bits, for what's really an 11-bit quantity. Other card types such as ATI 8514/A may not have this requirement, but we don't know and the correction can't hurt. So round the usable horizontal frame length figure down to 1176.

This figure (DCF / HSF rounded to a multiple of 8) is the minimum HFL you can use. You can get longer HFLs (and thus, possibly, more horizontal dots on the screen) by setting the sync pulse to produce a lower HSF. But you'll pay with a slower and more visible flicker rate.

As a rule of thumb, 80% of the horizontal frame length is available for horizontal resolution, the visible part of the horizontal scan line (this allows, roughly, for borders and sweepback time -- that is, the time required for the beam to move from the right screen edge to the left edge of the next raster line). In this example, that's 944 ticks.

Now, to get the normal 4:3 screen aspect ratio, set your vertical resolution to 3/4ths of the horizontal resolution you just calculated. For this example, that's 708 ticks. To get your actual VFL, multiply that by 1.05 to get 743 ticks.

The 4:3 is not technically magic; nothing prevents you from using a different ratio if that will get the best use out of your screen real estate. It does make figuring frame height and frame width from the diagonal size convenient, you just multiply the diagonal by by 0.8 to get width and 0.6 to get height.

So, HFL=1176 and VFL=743. Dividing 65MHz by the product of the two gives us a nice, healthy 74.4Hz refresh rate. Excellent! Better than VESA standard! And you got 944x708 to boot, more than the 800 by 600 you were probably expecting. Not bad at all!

You can even improve the refresh rate further, to almost 76 Hz, by using the fact that monitors can often sync horizontally at 2khz or so higher than rated, and by lowering VFL somewhat (that is, taking less than 75% of 944 in the example above). But before you try this "overdriving" maneuver, if you do, make *sure* that your monitor electron guns can sync up to 76 Hz vertical. (the popular NEC 4D, for instance, cannot. It goes only up to 75 Hz VSF). (See Overdriving Your Monitor for more general discussion of this issue.)

So far, most of this is simple arithmetic and basic facts about raster displays. Hardly any black magi
all!

10. Black Magic and Sync Pulses

OK, now you've computed HFL/VFL numbers for your chosen dot clock, found the refresh rate acceptable, and checked that you have enough VRAM. Now for the real black magic -- you need to know when and where to place synchronization pulses.

The sync pulses actually control the horizontal and vertical scan frequencies of the monitor. The HSF and VSF you've pulled off the spec sheet are nominal, approximate maximum sync frequencies. The sync pulse in the signal from the adapter card tells the monitor how fast to actually run.

Recall the two pictures above? Only part of the time required for raster-scanning a frame is used for displaying viewable image (ie. your resolution).

10.1 Horizontal Sync:

By previous definition, it takes HFL ticks to trace the a horizontal scan line. Let's call the visible tick count (your horizontal screen resolution) HR. Then Obviously, HR < HFL by definition. For concreteness, let's assume both start at the same instant as shown below:

```
|___ __ __ __ __ __ __ __ __ __ __ __ __
|_ _ _ _ _ _ _ _ _ _ _ _                |
|________________________|______________|_____
0                        ^              ^     unit: ticks
                         |   ^       ^   |
                         HR  |       |  HFL
                         |   |<----->|   |
                         |<->|  HSP  |<->|
                         HGT1         HGT2
```

Now, we would like to place a sync pulse of length HSP as shown above, ie, between the end of clock ticks for display data and the end of clock ticks for the entire frame. Why so? because if we can achieve this, then your screen image won't shift to the right or to the left. It will be where it supposed to be on the screen, covering squarely the monitor's viewable area.

Furthermore, we want about 30 ticks of "guard time" on either side of the sync pulse. This is represented by HGT1 and HGT2. In a typical configuration HGT1 != HGT2, but if you're building a configuration from scratch, you want to start your experimentation with them equal (that is, with the sync pulse centered).

The symptom of a misplaced sync pulse is that the image is displaced on the screen, with one border excessively wide and the other side of the image wrapped around the screen edge, producing a white edge line and a band of "ghost image" on that side. A way-out-of-place vertical sync pulse can actually cause the image to roll like a TV with a mis-adjusted vertical hold (in fact, it's the same phenomenon at

work).

If you're lucky, your monitor's sync pulse widths will be documented on its specification page. If not, here's where the real black magic starts...

You'll have to do a little trial and error for this part. But most of the time, we can safely assume that a sync pulse is about 3.5 to 4.0 microsecond in length.

For concretness again, let's take HSP to be 3.8 microseconds (which btw, is not a bad value to start with when experimenting).

Now, using the 65Mhz clock timing above, we know HSP is equivalent to 247 clock ticks (= 65 * 10**6 * 3.8 * 10^-6) [recall M=10^6, micro=10^-6]

Some makers like to quote their horizontal framing parameters as timings rather than dot widths. You may see the following terms:

active time (HAT)

Corresponds to HR, but in milliseconds. HAT * DCF = HR.

blanking time (HBT)

Corresponds to (HFL - HR), but in milliseconds. HBT * DCF = (HFL - HR).

front porch (HFP)

This is just HGT1.

sync time

This is just HSP.

back porch (HBP)

This is just HGT2.

10.2 Vertical Sync:

Going back to the picture above, how do we place the 247 clock ticks as shown in the picture?

Using our example, HR is 944 and HFL is 1176. The difference between the two is 1176 - 944=232 < 247! Obviously we have to do some adjustment here. What can we do?

The first thing is to raise 1176 to 1184, and lower 944 to 936. Now the difference = 1184-936= 248. Hmm, closer.

Next, instead using 3.8, we use 3.5 for calculating HSP; then, we have 65*3.5=227. Looks better. But 248 is not much higher than 227. It's normally necessary to have 30 or so clock ticks between HR and the start of SP, and the same for the end of SP and HFL. AND they have to be multiple of eight! Are we stuck?

No. Let's do this, 936 % 8 = 0, (936 + 32) % 8 = 0 too. But 936 + 32 = 968, 968 + 227 = 1195, 1195 + 32 = 1227. Hmm.. this looks not too bad. But it's not a multiple of 8, so let's round it up to 1232.

TOC

But now we have potential trouble, the sync pulse is no longer placed right in the middle between h a H any more. Happily, using our calculator we find 1232 - 32 = 1200 is also a multiple of 8 and (1232 - 32) - 968 = 232 corresponding using a sync pulse of 3.57 microseconds long, still reasonable.

In addition, 936/1232 0.76 or 76%, still not far from 80%, so it should be all right.

Furthermore, using the current horizontal frame length, we basically ask our monitor to sync at 52.7khz (= 65Mhz/1232) which is within its capability. No problems.

Using rules of thumb we mentioned before, 936*75%=702, This is our new vertical resolution. 702 * 1.05 = 737, our new vertical frame length.

Screen refresh rate = 65Mhz/(737*1232)=71.6 Hz. This is still excellent.

Figuring the vertical sync pulse layout is similar:

```
|___ __ __ __ __ __ __ __ __ __ __ __ __
|_ _ _ _ _ _ _ _ _ _ _ _                |
|________________________|______________|______
0                       VR             VFL     unit: ticks
                         ^   ^       ^
                         |   |       |
                         |<->|<----->|
                          VGT    VSP
```

We start the sync pulse just past the end of the vertical display data ticks. VGT is the vertical guard time required for the sync pulse. Most monitors are comfortable with a VGT of 0 (no guard time) and we'll use that in this example. A few need two or three ticks of guard time, and it usually doesn't hurt to add that.

Returning to the example: since by the defintion of frame length, a vertical tick is the time for tracing a complete HORIZONTAL frame, therefore in our example, it is 1232/65Mhz=18.95us.

Experience shows that a vertical sync pulse should be in the range of 50us and 300us. As an example let's use 150us, which translates into 8 vertical clock ticks (150us/18.95us 8).

Some makers like to quote their vertical framing parameters as timings rather than dot widths. You may see the following terms:

active time (VAT)

Corresponds to VR, but in milliseconds. VAT * VSF = VR.

blanking time (VBT)

Corresponds to (VFL - VR), but in milliseconds. VBT * VSF = (VFL - VR).

front porch (VFP)

This is just VGT.

sync time

TOC

This is just VSP.

back porch (VBP)

This is like a second guard time after the vertical sync pulse. It is often zero.

---

11. Putting it All Together

The Xconfig file Table of Video Modes contains lines of numbers, with each line being a complete specification for one mode of X-server operation. The fields are grouped into four sections, the name section, the clock frequency section, the horizontal section, and the vertical section.

The name section contains one field, the name of the video mode specified by the rest of the line. This name is referred to on the "Modes" line of the Graphics Driver Setup section of the Xconfig file. The name field may be omitted if the name of a previous line is the same as the current line.

The dot clock section contains only the dot clock (what we've called DCF) field of the video mode line. The number in this field specifies what dot clock was used to generate the numbers in the following sections.

The horizontal section consists of four fields which specify how each horizontal line on the display is to be generated. The first field of the section contains the number of dots per line which will be illuminated to form the picture (what we've called HR). The second field of the section (SH1) indicates at which dot the horizontal sync pulse will begin. The third field (SH2) indicates at which dot the horizontal sync pulse will end. The fourth field specifies the toal horzontal frame length (HFL).

The vertical section also contains four fields. The first field contains the number of visible lines which will appear on the display (VR). The second field (SV1) indicates the line number at which the vertical sync pulse will begin. The third field (SV2) specifies the line number at which the vertical sync pulse will end. The fourth field contains the total vertical frame length (VFL).

Example:

```
#Modename    clock  horizontal timing  vertical timing

"752x564"     40    752 784  944 1088  564 567 569 611
              44.5  752 792  976 1240  564 567 570 600
```

(Note: stock X11R5 doesn't support fractional dot clocks.)

For Xconfig, all of the numbers just mentioned - the number of illuminated dots on the line, the number of dots separating the illuminated dots from the beginning of the sync pulse, the number of dots representing the duration of the pulse, and the number of dots after the end of the sync pulse - are added to produce the number of dots per line. The number of horizontal dots must be evenly divisible by eight.

Example horizontal numbers: 800 864 1024 1088

This sample line has the number of illuminated dots (800) followed by the number of the dot when the sync pulse starts (864), followed by the number of the dot when the sync pulse ends (1024), followed by the number of the last dot on the horizontal line (1088).

Note again that all of the horizontal numbers (800, 864, 1024, and 1088) are divisible by eight! This is

not required of the vertical numbers.

The number of lines from the top of the display to the bottom form the frame. The basic timing signal for a frame is the line. A number of lines will contain the picture. After the last illuminated line has been displayed, a delay of a number of lines will occur before the vertical sync pulse is generated. Then the sync pulse will last for a few lines, and finally the last lines in the frame, the delay required after the pulse, will be generated. The numbers that specify this mode of operation are entered in a manner similar to the following example.

Example vertical numbers: 600 603 609 630

This example indicates that there are 600 visible lines on the display, that the vertical sync pulse starts with the 603rd line and ends with the 609th, and that there are 630 total lines being used.

Note that the vertical numbers don't have to be divisible by eight!

Let's return to the example we've been working. According to the above, all we need to do from now on is to write our result into Xconfig as follows:

```
<name>   DCF     HR  SH1 SH2   HFL   VR  SV1 SV2 VFL
```

where SH1 is the start tick of the horizontal sync pulse and SH2 is its end tick; similarly, SV1 is the start tick of the vertical sync pulse and SV2 is its end tick.

To place these, recall the discussion of black magic and sync pulses given above. SH1 is the dot that begins the leading edge of the horiziontal sync pulse; thus, SH1 = HR + HGT1. SH2 is the trailing edge; thus, SH2 = SH1 + HSP. Similarly, SV1 = VR + VGT (but VGT is usually zero) and SV2 = SV1 + VSP.

```
#name    clock   horizontal timing   vertical timing    flag
936x702  65      936 968 1200 1232   702 702 710 737
```

No special flag necessary; this is a non-interlaced mode. Now we are really done.

12. Overdriving Your Monitor

You should absolutely *not* try exceeding your monitor's scan rates if it's a fixed-frequency type. You can smoke your hardware doing this! There are potentially subtler problems with overdriving a multisync monitor which you should be aware of.

Having a pixel clock higher than the monitor's maximum bandwidth is rather harmless, in contrast. (Note: the theoretical limit of discernable features is reached when the pixel clock reaches double the monitor's bandwidth. This is a straightforward application of Nyquist's Theorem: consider the pixels as a spatially distributed series of samples of the drive signals and you'll see why.)

It's exceeding the rated maximum sync frequencies that's problematic. Some modern monitors might have protection circuitry that shuts the monitor down at dangerous scan rates, but don't rely on it. In particular there are older multisync monitors (like the Multisync II) which use just one horizontal transformer. These monitors will not have much protection against overdriving them. While you necessarily have high voltage regulation circuitry (which can be absent in fixed frequency monitors), it will not necessarily cover every conceivable frequency range, especially in cheaper models. This not only implies more wear on the circuitry, it can also cause the screen phosphors to age faster, and cause more than the specified radiation (including X-rays) to be emitted from the monitor.

Another importance of the bandwidth is that the monitor's input impedance is specified only for that range, and using higher frequencies can cause reflections probably causing minor screen interferences, and radio disturbance.

However, the basic problematic magnitude in question here is the slew rate (the steepness of the video signals) of the video output drivers, and that is usually independent of the actual pixel frequency, but (if your board manufacturer cares about such problems) related to the maximum pixel frequency of the board.

So be careful out there...

13. Using Interlaced Modes

(This section is largely due to David Kastrup <dak@pool.informatik.rwth-aachen.de>)

At a fixed dot clock, an interlaced display is going to have considerably less noticable flicker than a non-interlaced display, if the vertical circuitry of your monitor is able to support it stably. It is because of this that interlaced modes were invented in the first place.

Interlaced modes got their bad repute because they are inferior to their non-interlaced companions at the same vertical scan frequency, VSF (which is what is usually given in advertisements). But they are definitely superior at the same horizontal scan rate, and that's where the decisive limits of your monitor/graphics card usually lie.

At a fixed *refresh rate* (or half frame rate, or VSF) the interlaced display will flicker more: a 90Hz interlaced display will be inferior to a 90Hz non-interlaced display. It will, however, need only half the video bandwidth and half the horizontal scan rate. If you compared it to a non-interlaced mode with the same dot clock and the same scan rates, it would be vastly superior: 45Hz non-interlaced is intolerable. With 90Hz interlaced, I have worked for years with my Multisync 3D (at 1024x768) and am very satisfied. I'd guess you'd need at least a 70Hz non-interlaced display for similar comfort.

You have to watch a few points, though: use interlaced modes only at high resolutions, so that the alternately lighted lines are close together. You might want to play with sync pulse widths and positions to get the most stable line positions. If alternating lines are bright and dark, interlace will *jump* at you. I have one application that chooses such a dot pattern for a menu background (XCept, no other application I know does that, fortunately). I switch to 800x600 for using XCept because it really hurts my eyes otherwise.

For the same reason, use at least 100dpi fonts, or other fonts where horizontal beams are at least two lines thick (for high resolutions, nothing else will make sense anyhow).

And of course, never use an interlaced mode when your hardware would support a non-interlaced one with similar refresh rate.

If, however, you find that for some resolution you are pushing either monitor or graphics card to their upper limits, and getting dissatisfactorily flickery or outwashed (bandwidth exceeded) display, you might want to try tackling the same resolution using an interlaced mode. Of course this is useless if the VSF of your monitor is already close to its limits.

Design of interlaced modes is easy: do it like a non-interlaced mode. Just two more considerations are necessary: you need an odd total number of vertical lines (the last number in your mode line), and when you specify the "interlace" flag, the actual vertical frame rate for your monitor doubles. Your monitor needs to support a 90Hz frame rate if the mode you specified looks like a 45Hz mode apart from the "Interlace" flag.

As an example, here is my modeline for 1024x768 interlaced: my Multisync 3D will support up to 90Hz vertical and 38kHz horizontal.

```
ModeLine "1024x768" 45 1024 1048 1208 1248 768 768 776 807 Interlace
```

Both limits are pretty much exhausted with this mode. Specifying the same mode, just without the "Interlace" flag, still is almost at the limit of the monitor's horizontal capacity (and strictly speaking, a bit under the lower limit of vertical scan rate), but produces an intolerably flickery display.

Basic design rules: if you have designed a mode at less than half of your monitor's vertical capacity, make the vertical total of lines odd and add the "Interlace" flag. The display's quality should vastly improve in most cases.

If you have a non-interlaced mode otherwise exhausting your monitor's specs where the vertical scan rate lies about 30% or more under the maximum of your monitor, hand-designing an interlaced mode (probably with somewhat higher resolution) could deliver superior results, but I won't promise it.

---

14. Questions and Answers

Q. The example you gave is not a standard screen size, can I use it?

A. Why not? There is NO reason whatsover why you have to use 640x480, 800x600, or even 1024x768. The XFree86 servers let you configure your hardware with a lot of freedom. It usually takes two to three tries to come up the right one. The important thing to shoot for is high refresh rate with reasonable viewing area. not high resolution at the price of eye-tearing flicker!

Q. It this the only resolution given the 65Mhz dot clock and 55Khz HSF?

A. Absolutely not! You are encouraged to follow the general procedure and do some trial-and-error to come up a setting that's really to your liking. Experimenting with this can be lots of fun. Most settings may just give you nasty video hash, but in practice a modern multi-sync monitor is usually not damaged easily. Be sure though, that your monitor can support the frame rates of your mode before using it for longer times.

Beware fixed-frequency monitors! This kind of hacking around can damage them rather quickly. Be sure you use valid refresh rates for *every* experiment on them.

Q. You just mentioned two standard resolutions. In Xconfig, there are many standard resolutions available, can you tell me whether there's any point in tinkering with timings?

A. Absolutely! Take, for example, the "standard" 640x480 listed in the current Xconfig. It employes 25Mhz driving frequency, frame lengths are 800 and 525 => refresh rate 59.5Hz. Not too bad. But 28Mhz is a commonly available driving frequency from many SVGA boards. If we use it to drive 640x480, following the procedure we discussed above, you would get frame lengths like 812 (round down to 808) and 505. Now the refresh rate is raised to 68Hz, a quite significant improvement over the standard one.

Q. Can you summarize what we have discussed so far?

A. In a nutshell:

1. for any fixed driving frequency, raising max resolution incurs the penalty of lowering refresh rate and thus introducing more flicker.
2. if high resolution is desirable and your monitor supports it, try to get a SVGA card that provides a matching dot clock or DCF. The higher, the better!

15. Fixing Problems with the Image.

OK, so you've got your X configuration numbers. You put them in Xconfig with a test mode label. You fire up X, hot-key to the new mode, ... and the image doesn't look right. What do you do? Here's a list of common video image distortions and how to fix them.

(Fixing these minor distortions is where **xvidtune**(1) really shines.)

You *move* the image by changing the sync pulse timing. You *scale* it by changing the frame length (you need to move the sync pulse to keep it in the same relative position, otherwise scaling will move the image as well). Here are some more specific recipes:

The horizontal and vertical positions are independent. That is, moving the image horizontally doesn't affect placement vertically, or vice-versa. However, the same is not quite true of scaling. While changing the horizontal size does nothing to the vertical size or vice versa, the total change in both may be limited. In particular, if your image is too large in both dimensions you will probably have to go to a higher dot clock to fix it. Since this raises the usable resolution, it is seldom a problem!

15.1 The image is displaced to the left or right

To fix this, move the horizontal sync pulse. That is, increment or decrement (by a multiple of 8) the middle two numbers of the horizontal timing section that define the leading and trailing edge of the horizontal sync pulse.

If the image is shifted left (right border too large, you want to move the image to the right) decrement the numbers. If the image is shifted right (left border too large, you want it to move left) increment the sync pulse.

15.2 The image is displaced up or down

To fix this, move the vertical sync pulse. That is, increment or decrement the middle two numbers of the vertical timing section that define the leading and trailing edge of the vertical sync pulse.

If the image is shifted up (lower border too large, you want to move the image down) decrement the numbers. If the image is shifted down (top border too large, you want it to move up) increment the numbers.

15.3 The image is too large both horizontally and vertically

Switch to a higher card clock speed. If you have multiple modes in your clock file, possibly a

lower-speed one is being activated by mistake.

15.4 The image is too wide (too narrow) horizontally

To fix this, increase (decrease) the horizontal frame length. That is, change the fourth number in the first timing section. To avoid moving the image, also move the sync pulse (second and third numbers) half as far, to keep it in the same relative position.

15.5 The image is too deep (too shallow) vertically

To fix this, increase (decrease) the vertical frame length. That is, change the fourth number in the second timing section. To avoid moving the image, also move the sync pulse (second and third numbers) half as far, to keep it in the same relative position.

Any distortion that can't be handled by combining these techniques is probably evidence of something more basically wrong, like a calculation mistake or a faster dot clock than the monitor can handle.

Finally, remember that increasing either frame length will decrease your refresh rate, and vice-versa.

Occasionally you can fix minor distortions by fiddling with the picture controls on your monitor. The disadvantage is that if you take your controls too far off the neutral (factory) setting to fix graphics-mode problems, you may end up with a wacky image in text mode. It's better to get your modeline right.

---

16. Plotting Monitor Capabilities

To plot a monitor mode diagram, you'll need the gnuplot package (a freeware plotting language for UNIX-like operating systems) and the tool `modeplot`, a shell/gnuplot script to plot the diagram from your monitor characteristics, entered as command-line options.

Here is a copy of `modeplot`:

```
#!/bin/sh
#
# modeplot -- generate X mode plot of available monitor modes
#
# Do `modeplot -?' to see the control options.
#

# Monitor description. Bandwidth in MHz, horizontal frequencies in kHz
# and vertical frequencies in Hz.
TITLE="Viewsonic 21PS"
BANDWIDTH=185
MINHSF=31
MAXHSF=85
MINVSF=50
MAXVSF=160
ASPECT="4/3"
vesa=72.5       # VESA-recommended minimum refresh rate

while [ "$1" != "" ]
do
        case $1 in
        -t) TITLE="$2"; shift;;
        -b) BANDWIDTH="$2"; shift;;
        -h) MINHSF="$2" MAXHSF="$3"; shift; shift;;
        -v) MINVSF="$2" MAXVSF="$3"; shift; shift;;
        -a) ASPECT="$2"; shift;;
        -g) GNUOPTS="$2"; shift;;
        -?) cat <<EOF
modeplot control switches:

-t "<description>"   name of monitor            defaults to "Viewsonic 21PS"
-b <nn>              bandwidth in MHz           defaults to 185
-h <min> <max>    min & max HSF (kHz)        defaults to 31 85
-v <min> <max>    min & max VSF (Hz)         defaults to 50 160
-a <aspect ratio>    aspect ratio               defaults to 4/3
-g "<options>"       pass options to gnuplot

The -b, -h and -v options are required, -a, -t, -g optional.  You can
use -g to pass a device type to gnuplot so that (for example) modeplot's
```

TOC

```
output can be redirected to a printer.  See gnuplot(1) for  details.

The modeplot tool was created by Eric S. Raymond <esr@thyrsus.com> based on
analysis and scratch code by Martin Lottermoser <Martin.Lottermoser@mch.sni.de>

This is modeplot $Revision: 1.9 $
EOF
                exit;;
        esac
        shift
done

gnuplot $GNUOPTS <<EOF
set title "$TITLE Mode Plot"

# Magic numbers.  Unfortunately, the plot is quite sensitive to changes in
# these, and they may fail to represent reality on some monitors.  We need
# to fix values to get even an approximation of the mode diagram.  These come
# from looking at lots of values in the ModeDB database.
F1 = 1.30       # multiplier to convert horizontal resolution to frame width
F2 = 1.05       # multiplier to convert vertical resolution to frame height

# Function definitions (multiplication by 1.0 forces real-number arithmetic)
ac = (1.0*$ASPECT)*F1/F2
refresh(hsync, dcf) = ac * (hsync**2)/(1.0*dcf)
dotclock(hsync, rr) = ac * (hsync**2)/(1.0*rr)
resolution(hv, dcf) = dcf * (10**6)/(hv * F1 * F2)

# Put labels on the axes
set xlabel 'DCF (MHz)'
set ylabel 'RR (Hz)' 6  # Put it right over the Y axis

# Generate diagram
set grid
set label "VB" at $BANDWIDTH+1, ($MAXVSF + $MINVSF) / 2 left
set arrow from $BANDWIDTH, $MINVSF to $BANDWIDTH, $MAXVSF nohead
set label "max VSF" at 1, $MAXVSF-1.5
set arrow from 0, $MAXVSF to $BANDWIDTH, $MAXVSF nohead
set label "min VSF" at 1, $MINVSF-1.5
set arrow from 0, $MINVSF to $BANDWIDTH, $MINVSF nohead
set label "min HSF" at dotclock($MINHSF, $MAXVSF+17), $MAXVSF + 17 right
set label "max HSF" at dotclock($MAXHSF, $MAXVSF+17), $MAXVSF + 17 right
set label "VESA $vesa" at 1, $vesa-1.5
set arrow from 0, $vesa to $BANDWIDTH, $vesa nohead # style -1
plot [dcf=0:1.1*$BANDWIDTH] [$MINVSF-10:$MAXVSF+20] \
  refresh($MINHSF, dcf) notitle with lines 1, \
  refresh($MAXHSF, dcf) notitle with lines 1, \
  resolution(640*480,   dcf) title "640x480  " with points 2, \
  resolution(800*600,   dcf) title "800x600  " with points 3, \
  resolution(1024*768,  dcf) title "1024x768 " with points 4, \
  resolution(1280*1024, dcf) title "1280x1024" with points 5, \
```

```
 resolution(1600*1280, dcf) title "1600x1200" with points 6

pause 9999
EOF
```

Once you know you have `modeplot` and the gnuplot package in place, you'll need the following monitor characteristics:

- video bandwidth (VB)
- range of horizontal sync frequency (HSF)
- range of vertical sync frequency (VSF)

The plot program needs to make some simplifying assumptions which are not necessarily correct. This is the reason why the resulting diagram is only a rough description. These assumptions are:

1. All resolutions have a single fixed aspect ratio AR = HR/VR. Standard resolutions have AR = 4/3 or AR = 5/4. The `modeplot` programs assumes 4/3 by default, but you can override this.
2. For the modes considered, horizontal and vertical frame lengths are fixed multiples of horizontal and vertical resolutions, respectively:

```
HFL = F1 * HR
VFL = F2 * VR
```

As a rough guide, take F1 = 1.30 and F2 = 1.05 (see frame "Computing Frame Sizes").

Now take a particular sync frequency, HSF. Given the assumptions just presented, every value for the clock rate DCF already determines the refresh rate RR, i.e. for every value of HSF there is a function RR(DCF). This can be derived as follows.

The refresh rate is equal to the clock rate divided by the product of the frame sizes:

```
RR = DCF / (HFL * VFL)          (*)
```

On the other hand, the horizontal frame length is equal to the clock rate divided by the horizontal sync frequency:

```
HFL = DCF / HSF                 (**)
```

VFL can be reduced to HFL be means of the two assumptions above:

```
VFL = F2 * VR
    = F2 * (HR / AR)
    = (F2/F1) * HFL / AR        (***)
```

Inserting (**) and (***) into (*) we obtain:

```
RR = DCF / ((F2/F1) * HFL**2 / AR)
   = (F1/F2) * AR * DCF * (HSF/DCF)**2
   = (F1/F2) * AR * HSF**2 / DCF
```

For fixed HSF, F1, F2 and AR, this is a hyperbola in our diagram. Drawing two such curves for minimum and maximum horizontal sync frequencies we have obtained the two remaining boundaries of the permitted region.

The straight lines crossing the capability region represent particular resolutions. This is based on (*) and the second assumption:

```
RR = DCF / (HFL * VFL) = DCF / (F1 * HR * F2 * VR)
```

By drawing such lines for all resolutions one is interested in, one can immediately read off the possible relations between resolution, clock rate and refresh rate of which the monitor is capable. Note that these lines do not depend on monitor properties, but they do depend on the second assumption.

The `modeplot` tool provides you with an easy way to do this. Do `modeplot -?` to see its control options. A typical invocation looks like this:

```
modeplot -t "Swan SW617" -b 85 -v 50 90 -h 31 58
```

The -b option specifies video bandwidth; -v and -h set horizontal and vertical sync frequency ranges.

When reading the output of `modeplot`, always bear in mind that it gives only an approximate description. For example, it disregards limitations on HFL resulting from a minimum required sync pulse width, and it can only be accurate as far as the assumptions are. It is therefore no substitute for a detailed calculation (involving some black magic) as presented in Putting it All Together. However, it should give you a better feeling for what is possible and which tradeoffs are involved.

---

17. Credits

The original ancestor of this document was by Chin Fang <fangchin@leland.stanford.edu>.

Eric S. Raymond <esr@snark.thyrsus.com> reworked, reorganized, and massively rewrote Chin Fang's original in an attempt to understand it. In the process, he merged in most of a different how-to by Bob Crosson <crosson@cam.nist.gov>.

The material on interlaced modes is largely by David Kastrup <dak@pool.informatik.rwth-aachen.de>

Martin Lottermoser <Martin.Lottermoser@mch.sni.de> contributed the idea of using gnuplot to make mode diagrams and did the mathematical analysis behind `modeplot`. The distributed `modeplot` was redesigned and generalized by ESR from Martin's original gnuplot code for one case.

The X Window User HOWTO

by Ray Brigleb, ray@aracnet.com

v2.0, 1 September 1999

This document contains information on configuring the X Window environment for the Linux user, as well as for the beginning system administrator attempting to sort through the many configuration options and details of X Window. A basic knowledge of software configuration and installation is assumed, as is the presence of X on the users system.

1. Introduction

- 1.1 Other Sources Of Information
- 1.2 Versions Of This Document
- 1.3 Feedback And Corrections
- 1.4 Acknowledgments
- 1.5 Copyright

2. Getting Started

- 2.1 The X Window System: History and Architecture
- 2.2 Anatomy of Your Desktop
- 2.3 Invoking X Window
- 2.4 The X Display Manager

3. Choosing a Window Manager

- 3.1 FVWM And Its Ancestors
- 3.2 The Wide World of Window Systems
- 3.3 The X Graphical Interfaces
- 3.4 The X Desktop Environments
- 3.5 The Flashy Window Managers

TOC

4. Working In X

- 4.1 Command Line Options
- 4.2 Display Names
- 4.3 XTerm Versus Rxvt, or, Know Thy Terminal Emulator

5. X Startup

- 5.1 A Sample Starting Configuration
- 5.2 A More Intelligent Startup
- 5.3 Getting The Windows Where You Want Them

6. Configuring the Window Manager

- 6.1 Basic FVWM2 Configuration
- 6.2 Advanced FVWM2 Configuration
- 6.3 FVWM2 Configuration Shortcuts
- 6.4 FVWM2 Themes

7. Fonts and Colors

- 7.1 Fonts Demystified
- 7.2 Font Aliases and Configuration
- 7.3 Using Type 1 Fonts in X
- 7.4 Using TrueType Fonts in X
- 7.5 Colors

8. The X Resources

- 8.1 X Resources: The Basics
- 8.2 Inside The X Resource Database With *editres*
- 8.3 The Anatomy of X Resources
- 8.4 Making Your Changes Last With *.Xdefaults*
- 8.5 Your Own User Resource Directory

9. Clients and Application Tips

- 9.1 Screen Savers for X
- 9.2 Emacs and XEmacs
- 9.3 Some Useful Programs and Tricks

10. Advanced X Usage

- 10.1 Libraries and Compiling X Applications
- 10.2 Basic X Security
- 10.3 More About X Authority

11. Bibliography and Other Resources

---

1. Introduction

The *X Window System* is an advanced, network transparent, windowing, graphical environment, developed at the *Massachusetts Institute of Technology*, and first released in 1984. This document assumes that you have installed X and it is functional. We intend to learn how to use X productively, not so much how to compile the programs; most Linux distributions come with X as an option during installation, compiled and ready to go.

1.1 Other Sources Of Information

If you are just starting out, you may find the *XFree86 HOWTO* and *XFree86 Video Timings HOWTO*s to be more helpful, and you should be able to find that in the same place you found this. At the end of the document you will also find a Bibliography And Resources section, to find even more information. Oh, and don't forget to read the **man pages**.

1.2 Versions Of This Document

New versions of this HOWTO may be periodically posted to comp.os.linux.help They will also be uploaded to various Linux WWW and FTP sites, including Linux Documentation Project web site. The latest version can usually be found at http://www.croftj.net/~ray/howto, but the Metalab repository is the canonical distribution point.

Changes in version 1.1 include more coverage of **rxvt**, expanded coverage of X Resources, more trivial history of X, corrected references to SunSite to point to MetaLab, and instructions on using Type 1 fonts with X.

Changes in version 1.2 include much-requested coverage of using TrueType fonts in X, more details about setting up `xdm`, and a few more added resources. There is also a correction to my discussion of screen blanking features (thanks to Heinrich Langos for the correction!).

Changes in version 1.3 include some fixes, info for corrupted font paths in some recent distributions, and lots more info about basic X security. I've also added some tips about KDE and an e-mail address update.

Changes in version 1.4 include corrections and additions from Anthony J., and some very good security tips from Tomasz Motylewski.

Changes in this version (2.0) includes corrections from Guus Bosch, Brian J. Miller, and myself, as well as lots of new updates and info, and a plea for a new maintainer!

TOC

1.3 Feedback And Corrections

If you have questions or comments about this document, please feel free to email Ray Brigleb at my current (`ray@aracnet.com`) or ``permanent'' (`rayola@mindless.com`) address. I readily welcome any suggestions or criticisms. If you find any mistakes with this document, please let me know so I can correct them in the next revision. If you have information you would like to see in future revisions, or you would like to contribute to a future revision, drop me a line. I'm also looking for more resources to add to the sections and bibliography. While we're at it, I'm also looking for someone interested in taking over maintenance of this document, please let me know if you're interested. I think another maintainer might be able to breathe life into this project that I have not had as of late, and judging from the feedback I receive this is still a very viable and useful text.

1.4 Acknowledgments

A special thanks to the HOWTO coordinator Tim Bynum for help and suggestions, and the XFree86 Group for the man pages and help files that come with XFree86. I really need to thank all of those who have written documentation and descriptions of their offerings in the past, I have just gleaned a selection of the most relevant items from their material for this document. Also, thanks are due to Matt Chapman, author of the Window Managers Guide website, and Scott Scriven, for the FVWM2 Themes tips and tricks I stole from some of his hacks. Thanks are also due to Joe Croft for the Internet service, invaluable to the ongoing development of this HOWTO, and Claire Galper, for moral support and miscellaneous tips. Last but not least, thanks to the whole GNU/Linux community, for everything, and Mr. Stallman, for this text editor (and philosophy).

1.5 Copyright

Copyright (c) 1998, 1999 by Ray Brigleb.

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator for more information.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would very much like to be notified of any plans to redistribute the HOWTOs, this one in particular!

Many of the terms mentioned in this document are trade names. Unless otherwise stated, all trademarks are property of their respective owners.

---

2. Getting Started

2.1 The X Window System: History and Architecture

The *X Window System* was developed in the Laboratory for Computer Science at the *Massachusetts Institute of Technology*, as part of project Athena in cooperation with DEC, and first released in 1984. The project lead of the main development was Robert Scheifler, and the origins of X owe much debt to the ``*W*'' Windowing package, developed by Paul Asente at *Stanford*. In September of 1987, MIT issued the first release of the X11 that we know and use today. As of X11R2, control passed from MIT to the *X Consortium*, formed in January of 1988.

Many of the ideas that went into X Window also came from research at Xerox Corporation's *Palo Alto Research Center* (PARC), where they were working on computers like the Parc and the Star in the late seventies. None of these computers made it to market, but when Xerox demonstrated a window system custom built to run *Smalltalk 80*, people were hooked. These series of three computers demonstrated the WIMP (Windows, Icons, Menus, Pointer) interface so well that it spawned a revolution in computing almost overnight. Within a few years many computer users got a taste of a windowing system of some kind, and you might say they never looked back.

X Window is currently developed and distributed by the X Consortium, however, a liberal license permits the existence of free and low-cost implementations. The version of X used on Linux is XFree86. XFree86 is a collection of X servers for UNIX-like OSs on Intel x86 platforms. The work is derived from X386, and much of it is contributed back into X11R6 thereafter. We can think of XFree86, for all intents and purposes, to be X Window for Linux, unless you have purchased another X server.

X Window is built upon a great many toolkits, or libraries. It is built upon the X Toolkit Intrinsics and the Athena Widgets. Many programs use XView or Motif tools. More still are part of a newer, unified windowing and communication system, like GNOME or KDE. If you find many of your programs not compiling, or are getting strange and inexplicable errors, you may wish to make sure that you have installed X correctly, because most of these libraries (with the exception of Motif) are free and most likely preinstalled with X on your system.

And remember, it's called X Window, *not* X Windows!

2.2 Anatomy of Your Desktop

There are a few basic principles and terms you should familiarize yourself with to make using X much more straightforward. These terms will appear over and over again in the manual pages and help files, which it is suggested that you consult whenever necessary.

The **screen** is your whole ``desktop'', and the words may be used interchangeably. Technically it means the primary video display you view X with, and you can have more than one screen, in fact you can have

TOC

more than one computer running off a single X server. This is beyond the scope of this humble document, but you should be aware of the distinction.

The **root window** is the background of your screen. It is referred to a window in name alone, it does not behave like any other window, but rather you run your applications on the root window, or put a picture on it, or just a solid color.

The **window manager** is the main interface between the X Window system and the user. Without the window manager, the system would be rather difficult to use, and would certainly not be a very productive tool. The window manager provides such functionality as window borders, menus, icons, virtual desktops, button bars, tool bars, and allows the user to customize it at will, often adding to its functionality in the process.

The **pointer** is the arrow or indicator of any given shape which represents the location your mouse (or other pointing device) corresponds to on the screen. The pointer often changes to give you contextual feedback as to what will happen when you use the mouse at that point on the screen.

The **window** is a frame in which any given application resides which is ``managed'' by the window manager. This includes pretty much anything except the so-called root window. Even windows which do not appear to have frames, titles, or normal borders of any kind are being managed by your window manger. The **active window** is the window you are currently using, the window that will receive text when you type, and is traditionally denoted by the fact that your mouse cursor is pointing at it, though this is not always the case. The active window is said to have ``focus,'' the rest of the windows on your display being ``unfocused.''

Menus and **icons** behave in X similar to the way they behave in other windowing systems, and the same general principles apply. Windows with text only are called **terminal emulators**, an example would be xterm, and these basically emulate a console text-only display, but let you multiplex and use more than one at a time, and have many other advantages available due to their being used in X. We shall discuss many of these later on.

2.3 Invoking X Window

Starting X can be done in several ways. On your system it may be set up to start automatically, and you will not need to read this section. Most Linux systems, however, presently start at the command line upon login, and you have to decipher this for yourself.

The most basic way to start X is with **xinit**. This will put you at a blank desktop, by default, and with no window manager loaded. If no client program is specified on the command line, **xinit** will look for the *.xinitrc* file to run as a shell script, to start up client programs. If this file does not exist, **xinit** will use the following command as a default:

```
xterm -geometry +1+1 -n login -display :0
```

As you see, this is not very helpful. The most common way to start X is with the command **startx**. This is the most civilized fashion to start the windowing system, but requires that you log in from a text shell,

TOC

and start the windowing system yourself. For many Linux users this is the most common way to start it is also the most flexible. You can issue commands such as the following:

```
startx -- -bpp 8    #start x in 256 color mode
startx -- -bpp 32   #start x in true color mode
```

The double dashes pass arguments directly to xinit, and this way you can start up X in the resolution your work will require, and still have it use the configuration files we will cover later in this document.

2.4 The X Display Manager

The program **xdm** provides similar services to **getty** and **login**, which allow users to log into a system and start their basic shell. If you start X with **xdm**, however, users need only to type in their username and password at a friendly prompt, and they are dropped directly into the graphical environment. This is simple and easy to use, and is seen frequently in college campuses, cyber cafes, business environments, anywhere you have users not necessarily familiar with Unix to any great extent.

xdm can be configured with configuration files located in */usr/X11R6/lib/X11/xdm* on your Linux system. The file `xdm-config` is for configuring how the login screen appears to users, and `Xsetup_0` is used to tell xdm what programs should be launched when X is started. Some of the configuration a normal user would put in their `.xinitrc` file should go in here, if xdm is to be normally used.

Here is a sample `Xsetup_0` file to look at, which might help to configure your system. The `xfstt` program is the TrueType font server, and is discussed later in this document. Also, notice that we're using a shell script here, and it's trying to call **xv** to set the background to a nice picture (instead of the boring black and white background pattern), and if that call fails, **xsetroot** is called, to at least try to set the background to a nice blue color.

```
#!/bin/sh
xconsole -geometry 480x100-0-0 -daemon -notify -verbose -fn \
'-schumacher-clean-medium-r-*-*-10-*-*-*-*-*-*-*' -exitOnFail
/usr/X11R6/bin/xfstt &
/usr/X11R6/bin/xv -quit -root                                \
/usr/local/share/WindowMaker/Backgrounds/InDreams.jpg        \
|| xsetroot -solid darkblue
xset fp+ unix/:7100
```

Many distributions of Linux include this facility automatically. If you are able to look at the run levels for your Linux system, you can probably see that run level 3 is the normal startup (look under `/etc/rc.d`) unless you're booting into something like *xdm*. If you are, you're going to be starting in run level 5. This is something of a standard on Linux (and similar) systems. Programs such as *linuxconf* and its functional equivalents should be able to adjust this.

Also on many newer Linux systems are facilities such as *gdm* and *kdm*, which are GNOME and KDE

aware equivalents of this program. Primarily, this just changes the look and feel to suit your desktop preferance, but often these other versions contain more features, such as remembering which desktop environment you last logged in to, the ability to shutdown and reboot from the console, and so forth.

TOC

3. Choosing a Window Manager

This is a discussion of the window managers available to the X Window user, with a comparison of some finer ones to choose from. In order to prevent this document from becoming overwhelmingly large (and proportionally difficult to maintain, as well) I am limiting the discussion to the more popular and well known window managers out there. I believe these are most representative for our purposes, and once you grasp the basic concepts of a few, you more or less will have the hang of them all.

Your choice of window manager can dramatically influence how pleasant your computing experience will be. You spend much of your time dealing with windows when you're in X, and you don't want something that's too obtrusive, one that will get in your way. Some window managers are extremely customizable, to the point that you can pretty much feel like it's a new operating system. If you don't like icons, get rid of them! If you miss that toolbar, you can build a better one, and it can be a pretty painless and rewarding process besides.

3.1 FVWM And Its Ancestors

The most common window managers nowadays have their roots in Robert Nation's FVWM window manager. These include FVWM, FVWM2, FVWM95, and many more. FVWM itself is derived partly from TWM, which comes with an X Window installation, but which we will not cover here. The syntax and usage of TWM is similar to FVWM, but it actually uses more memory, and feels clumsy and awkward to most users.

FVWM is the original and old standby favorite Linux window manager. Now that the 2.0 of versions of FVWM have become stable and in more general usage, this newer version is preferred, for the syntax is much more direct and simplified, and much more flexible as well. However, many people still have the original FVWM on their systems, as it has been a long time favorite and standard, and most of the examples in this text are equally applicable to this older version. And if the older one is working fine for you, there may be no need to upgrade, since it may break your configuration files to some degree, for some of the syntax has indeed changed, and is not altogether backward compatible. Such is the nature of progress.

Nobody seems quite sure what F in FVWM stands for (not even the author, from what I can tell!), but the VWM would seem to (correctly) indicate Virtual Window Manager, and indeed the FVWM series are virtual window managers, and that is one of their strengths. You can bind keys to any function, including the switching between the virtual desktops, and do this with ease.

FVWM2 is the new standard, including many improvements and features not found in the previous version. Unlike many newer window managers, it works just fine on 8-bit, 256 color displays, which I am using at the moment, and still can be improved with little icons and gradients, to look as much like the Win98 interface as you may or may not want. This version is also much more extensible in general, and allows the use of themes and dynamic menus. Technically, FVWM2 is still in beta, but it works just great, and I have not yet had a single problem with it.

Resources:

The official FVWM and FVWM2 site is http://www.hpc.uh.edu/fvwm/. The latest version should always be available at ftp://ftp.hpc.uh.edu/pub/fvwm/.

3.2 The Wide World of Window Systems

For users more familiar with other window environments on other operating systems, there are plenty of window managers to emulate the desktop you're accustomed to.

The icewm window manager is aimed at a more consistent look and feel, and can emulate many other windowing system standards, hot key bindings, and such. The OS/2 emulations are particularly nice. AmiWM is an Amiga Workbench type window manager, mlvwm is a MacOS emulator. A nice addition to these window managers is dfm, a desktop and file manager that pleasantly resembles the OS/2 Workplace Shell.

Resources:

The Window Managers website can be found at http://www.PLiG.org/xwinman/, and is a consistently good resource for finding out about the variety of window managers available, particularly for finding good window managers to emulate other windowing environments. Many more still are listed at http://www.PLiG.org/xwinman/others.html, provided by the same author as the previous link, and this is quite a formidable and complete list indeed. You can download Linux versions of most popular window managers at the metalab archive.

3.3 The X Graphical Interfaces

Open Look and Motif were early attempts to standardize X Window, and make it usable in more mainstream environments, around a greater variety of workstations. Both were somewhat successful at those attempts (in look and feel, if not politically) and can be used under a modern Linux system. Both require specific libraries, which can be used to compile a variety of applications specifically designed for one or the other environment, to give all programs a similar look and feel.

Starting with X11 Release 5, Sun Microsystem's **OpenWindows** graphical interface was available to X users. This includes two versions of the Open Look Window Manager, *olwm* and *olvwm* (with a virtual desktop). This was developed by Sun in hopes of making a standard windowing environment, and indeed it was standard with Solaris for many years. If your distribution of Linux did not come with this windowing system installed, you must remember to install the *XView* libraries to get it to work, and you will also have to put the */usr/openwin/...* directories in your search path.

If you want the actual **OSF/Motif Toolkit** for Linux, you'll have to pay, the programs and toolkit are not free. However, the *Hungry Programmers* have written **LessTif**, which allows you to compile Motif programs more or less as if you owned to toolkit. LessTif is a clone of the Motif toolkit. Currently LessTif is partially implemented with most of the API in place. Many programs already work under this free version of the toolkit, and it even comes with a window manager, derived from FVWM code, which you'd swear was the Motif Window Manager.

TOC

The most useful feature of this toolkit, however, is compiling programs dependent on having a Motif library on your system. The window manager is nothing spectacular, and mostly useful if you're migrating from the original Motif, and want to keep your configuration file. For all intents and purposes, you will find FVWM much more feature-filled and useful, and it looks and behaves almost identically, even recognizing the window hints supplied by programs built with the Motif toolkit.

Resources:

The Hungry Programmers LessTif can be found at http://www.lesstif.org/.

3.4 The X Desktop Environments

The second generation of Linux window managers was brought about by KDE, and soon joined by GNOME. There are some striking similarities, and some great differences, between these two, and I will attempt to cover them here. The most important thing to remember at this point is that neither of them is in any way a complete product. Both are at the start of a long development cycle, and not completely stable yet, and as such are not suited to mission-critical work at this time.

The KDE Desktop Environment

(A quote from the home page:) *``KDE is a completely new desktop, incorporating a large suite of applications for Unix workstations. While KDE includes a window manager, file manager, panel, control center and many other components that one would expect to be part of a contemporary desktop environment, the true strength of this exceptional environment lies in the interoperability of its components."*

The KDE Desktop Environment is an attempt to make a desktop environment, not just a window manager. The tools of KDE work together so well, for instance, one might be fooled into thinking KDE was an entire operating system. All the tools to work in a windowing system are included, and many more have been ported to the KDE environment. KDE has achieved a surprising level of maturity already, but many are reluctant to install it on their desktop, because of the licensing stipulations of the QT toolkit, upon which KDE is based. This has changed a little lately, and the licence now qualifies as Open Source by definition, but is not the same as that of GNU software.

For those just looking to get down to business, KDE is often the way to go. This project has been around for some time now, and has let go of some flashiness for the ability to get lots done. In many ways you can become fooled into thinking you're using a product strikingly similar to the good parts of Windows. Which can be a good thing. But it can also be somewhat frustrating, and one longs after a while for something a little different from the paradigm upon which KDE is firmly based.

GNOME: The GNU Object Model Environment

(A quote from the home page:) *``GNOME stands for GNU Network Object Model Environment. The GNOME project intends to build a complete, user-friendly desktop based entirely on free software. GNOME is part of the GNU project, and GNOME is part of the OpenSource(tm) movement. The desktop will consist of small utilities and larger applications which share a consistent look and feel. GNOME uses GTK+ as the GUI toolkit for all GNOME-compliant applications."*

TOC

The GNOME project is an attempt to do much of the same work as KDE, but even a little more than GNOME is less tied to one window manager, for instance, and it is interoperable not just between applications, but computers and platforms, as it uses the Common Object Resource Broker Architecture (CORBA). Also, and to many most importantly, GNOME is based on the GTk+ toolkit, which is free and open source, unlike the underlying toolkit of KDE, thereby following in the philosophy of Linux itself.

Resources:

> The official KDE website is http://www.kde.org/. The official GNOME website is http://www.gnome.org/. More detailed information regarding the issues surrounding GNOME can be found at http://www.gnome.org/gnomefaq/FAQ.txt. The home page of CORBA is located at http://www.corba.org/ , and the GTk+ toolkit home is http://www.gtk.org/ .

3.5 The Flashy Window Managers

The latest generation of window manger is very very pretty indeed. Sporting every convenience you could think of, and emulating the most beautiful operating systems ever used on the most gorgeous workstations in the world, these are the window managers to run if you've got the memory and CPU cycles to burn.

Window Maker

(A quote from the home page:) ``*Window Maker is an X11 window manager designed to give additional integration support for GNUstep applications. It tries to emulate the elegant look and feel of the NEXTSTEP(tm) GUI. It is relatively fast, feature rich, and easy to configure and use."*

A big strength of this window manager is that it supports the GNU desktop, meaning that it makes a great and very pretty front-end to GNOME. This is also one of the most easily configurable window managers, and can be configured from a graphical interface, and supports the OffiX drag and drop protocol, easy switching of desktop themes, and it's now available within the popular Red Hat distribution, so it's easy and painless to switch from FVWM when the mood finally strikes. As of the 0.50 release, Window Maker supports KDE compliance as well.

AfterStep

(A quote from the home page:) ``*AfterStep is a Window Manager for X which started by emulating the NEXTSTEP look and feel, but which has been significantly altered according to the requests of various users. Many adepts will tell you that NEXTSTEP is not only the most visually pleasant interface, but also one of the most functional and intuitive out there. AfterStep aims to incorporate the advantages of the NEXTSTEP interface, and add additional useful features. The developers of AfterStep have also worked very hard to ensure stability and a small program footprint."*

Enlightenment

Enlightenment is more than just a window manager, it is an extreme, detailed, and configurable environment, and is particularly attractive in that it allows irregular and completely customizable window shapes. It is open in design, and instead of dictating a policy it allows the user to define their own policy right down to the minute and infinitesimal details; from its functionality right on through to its looks.

If you are using GNOME, you will find that Enlightenment is the default window manager, and in fa must be installed for GNOME to function. It is also basically the de facto implementation of the GNOME features for integration, making it the most practical choice for a desktop in that situation. Many other window managers will work alright with GNOME, but you will find that Enlightement excels. Unfortunately, it is still in development, and runs slowly and imperfectly from time to time.

Resources:

The official Window Maker website is http://www.windowmaker.org/. The official AfterStep website is http://www.afterstep.org/. Enlightenment can be found, somewhat predictably, at http://www.enlightenment.org/.

---

4. Working In X

In this section we will become familiar with running clients in X Window and the basic procedures involved in using the system. X is not an intuitive interface on its own, and without any window manager, it is easiest to use as a display for programs started from the command line. One of the most common uses for X is just to have several xterm windows open at the same time. Not exactly maxing out the graphics capabilities of the computer, but it's a nice feature to start with.

The flexibility and usefulness of the command line is so great, in fact, that you don't really need a window manager. It's nice, and you will want it very quickly, but the fact is that you can get by without one, just the command line and the mouse. If you give it a shot, you'll be surprised as to the exact distinctions between the clients and the window manager. If you start toying around with this, however, *remember* that the key combination Ctrl-Alt-Backspace gets you out of X Window in a pinch.

4.1 Command Line Options

Most X programs try to use the same basic names for command line options. All applications written using the MIT X Toolkit Intrinsics automatically accept the following options:

-display *display*

This option specifies the X server to use. See the section on Display Names for details.

-geometry *geometry*

The initial size and location of the window, in a format such as *widthxheight+hoffset+voffset* or *+hoffset-voffset*. Note that if you put in a negative horizontal or vertical offset, the window will be placed counting backward from the right or the bottom of the screen, respectively.

-font *font*

The font to use for displaying the text in your window.

-bg *color*

The color to use for the window background.

-fg *color*

The color to use for the window foreground.

-name *resource-name*

Useful for specifying the name under which the resources for this application will be found. This is useful to distinguish between invocations of the same application, for example, two **xterm**s can be named differently so that they may inherit different resources based upon those names in the resource database.

-title *string*

TOC

This is the title to be used for the window on your display, generally used by the window man to put a descriptive title at the top of the window. Not to be confused with the **-name** option.

-iconic

Invoke window as an icon.

-xrm *resource-string*

This option specifies a resource name and value to override any defaults that may already be set. Also useful for setting X resources that do not have explicit command line options. For instance, the command line `xterm -xrm "xterm*background: blue"` is functionally identical to typing `xterm -bg blue`.

4.2 Display Names

Every X Server has a display name of the form:

hostname:displaynumber.screennumber

The *hostname* specifies the name of the machine to which the display is actually, physically connected. The *hostname* can be omitted, and if so, the server on the same machine will be chosen. In fact, if you are the only one using X on your computer, you will want to just leave this off of your display specifications. The *displaynumber* should probably be zero, this is used if the X Server is controlling more than one keyboard and monitor unit, for instance, a network of X terminals. The *screennumber* specifies which monitor in a multiple monitor setup should be used. Following this specification, you would open an **xterm** window on your local machine with the option **-display :0.0**. You can see that we have omitted the *hostname* from the option, so the current machine is assumed.

On Linux systems, your *DISPLAY* variable holds your display name, which on my system is **:0.0**. This is usually set by **xterm**, or one of the scripts that starts X Window for you, although you can set it yourself, or as discussed above, use the **-display** command line option when invoking your application.

If you have opened an **xterm** or **rxvt** window, and then opted to do super-user work via the `su` command, you will find that if you try to launch an X application you will have no display to launch it on. Silly it seems, because you are sitting right in front of your display, but the trick to getting this to work is to pass an option on the command line to the program such as `-display :0.0`, and it will work fine.

4.3 XTerm Versus Rxvt, or, Know Thy Terminal Emulator

Your choice of **terminal emulator** can affect your experience in X almost as much as your window manager, so a little discussion of your best options is due here. A terminal emulator is a program such as **xterm**, discussed above, which lets you emulate the simple console of Linux in X. You will rely heavily on the program you choose, so it pays to choose it wisely to begin with.

TOC

If you have a slow video card, the first thing you should consider doing is switching to **rxvt**. Using **xterm** is a good starting point, mainly because it comes standard with all X distributions, but it's not always the best choice. It contains a lot of legacy code, and is quite bloated for a simple terminal emulator. This is partly due to the fact that it's not just a plain text terminal emulator, but also emulates graphics modes that you simply will never use. Because of this, you may wish to switch to **rxvt**, and also because **xterm** can be extremely slow. I'm not sure why it's so slow, but if you have a slow video card you'll know what I mean, and you'll be amazed at the difference when you dump **xterm**.

Some other perks of **rxvt** include pixmap backgrounds, and a much nicer scrollbar. If you specify the `-pixmap` option on the command line (and have support compiled in to your version!) you will get any `.xpm`-type picture in the background. A very cool feature, and it surprisingly doesn't slow down your output at all, it still redraws faster than your **xterm** window. Give it a shot, I haven't seen it around lately but I think it can be found on metalab.

---

5. X Startup

We will presume for the following examples that we have picked a fairly stable window manager, such as **FVWM2**, to try out some sample configurations. I would suggest giving that a shot for the purpose of learning these topics, as most of what you will learn here and in the following sections will apply to any window manager out there, but the topics seem most easily picked up using FVWM2.

5.1 A Sample Starting Configuration

Our first step is to write ourselves an initialization file for X itself. This file can be either a system-wide file, in which case it would likely be placed in */var/X11R6/lib/xinit/xinitrc*, or it can be overridden on a per-user basis by placing the file `.xinitrc` in your home directory. Generally, it is expected that there will be a basic, default file in the system-wide location, possibly enforced if necessary for security reasons, but otherwise users will probably wish to configure the file themselves.

First let's create a file in your home directory called `.xinitrc`. Open up your ``favorite'' text editor, and paste the following, or something like it, in that file:

```
 #!/bin/sh

 # if your backspace and delete are reversed, try this:
 xmodmap -e "keysym BackSpace=Delete" -e "keysym
Delete-BackSpace"

 xsetroot -solid darkslateblue

 # start some basic applications
 xclock -geometry 96x96+2+2 -bg grey40 -fg black -hl white &
 xload -geometry 120x96+2+147 -bg grey40 -fg white -hl darkred -update 4
&
 xterm -sb -ls -geom 80x25-2+2 -title "shell" &
 xterm -sb -ls -geom 80x25-2-2 &

 # start the window manager
 /usr/X11R6/bin/fvwm2
```

There are plenty of things to learn from this example. First of all, this file will be a shell script, as indicated by the first line. The **xsetroot** command on the second line turns the background of our desktop to a pleasant blue color, not a bad idea if we're going to be staring at that color predominantly all day.

The third and fourth lines are some programs that I like to leave running while I'm fast at work. You'll notice that some of the options make for a nicer setup, for example, specifying the colors and geometry (location on screen). I'll give you some tips for figuring this stuff out in a bit. The fifth and sixth lines follow similarly, opening up two handy **xterm** windows for us, which we will no doubt be needing soon.

The last line is very important--it is this line that starts up your window manager! Notice that the only comm we did not run as *background processes* (by putting the **&** at the end) were **xsetroot**, **xmodmap**, **fvwm2**. Wi these first two it doesn't matter, as the programs exit immediately. But all the rest of the programs have to be in the background, otherwise when you closed one, it would kill your X Window session. That would not be very pleasant, nor very expected. As shown above, when you close **fvwm2**, you exit X.

5.2 A More Intelligent Startup

We can add lots to our primitive example of a startup file. For instance, this is a good way to warn yourself when you may have carelessly stared X as the root user. Red Hat users seem to do this often, for many of the configuration programs which must be run as root, must also be run in X. You can avoid this by issuing an **su** command to become root during your normal X user session, and then calling the program you need to run as root with the option **-display :0.0** discussed above.

```
# change background color for root
if [ "$USER" = "root" ];
then
     xsetroot -solid darkred
else xsetroot -solid darkslateblue
fi
```

This will check to see if you are the user named root, and if you are, it will set the background to a harsh red, rather than the usual friendly blue, to warn you. This next bit of code, also intended for your `.xinitrc` file, will merge in your user-specific and system-wide resources, first checking to be sure the files exist.

```
userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
sysresources=/usr/X11R6/lib/X11/xinit/.Xresources
sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap
# Merge in defaults and keymaps
if [ -f $sysresources ]; then
   xrdb -merge $sysresources; fi
if [ -f $sysmodmap ]; then
   xmodmap $sysmodmap; fi
if [ -f $userresources ]; then
   xrdb -merge $userresources; fi
if [ -f $usermodmap ]; then
   xmodmap $usermodmap; fi
```

5.3 Getting The Windows Where You Want Them

Placing everything on your screen by guesswork with the *-geometry* option can get very tedious indeed. Particularly when you consider that you can specify the *-geometry* option to pretty much any program you can run in X. This allows a great precision in tuning the interface, but that's a heck of a lot of options to set, indeed.

This brings up one very nice feature of the FVWM window managers. By default, when you move a window

around the screen, you see the geometry specification appear on the screen. Go ahead, try moving a window around with the left button. Now try resizing it. As you can see, you can get some primitive specifications this way. However, even this method can be a little difficult, and it would be nice to have all the details about your window in one concise list.

It is at this point that we will introduce the program **xwininfo**. To use this program, go to an **xterm** window and type in that program name. It will ask you to click on another window that you want information about, and after you click it will dump out useful information that it knows about that window. This is useful for plugging information about windows once you have them set up how you want them on your screen - run this program, then click on the window, then put in those parameters in your startup file, and your window system will henceforth be frozen in a pristine state of immaculate precision.

---

6. Configuring the Window Manager

Now we take a look at the files to configure your window manager. These vary from manager to manager, but for our examples we will focus on the per-user files, and we are looking at FVWM2 in this example, so you will modify the file *.fvwm2rc* in your home directory.

6.1 Basic FVWM2 Configuration

Configuring your window manager resource file in earlier versions of FVWM was a rather arduous process, because the order of the items in the file needed to be very particular, but that has relaxed quite a bit in this version. To my knowledge, the only crucial part of the ordering is the bindings for the menus, but we'll cover that a bit later. Your FVWM2 installation should have come with an example resource file, and you should be able to find that in */var/X11R6/lib/fvwm2/system.fvwm2rc*. This is the default system-wide configuration file. A good idea would be to copy that file into your home directory and call it *.fvwm2rc*. From that point on, when FVWM2 starts up it will read the file in your home directory, and ignore the system-wide configuration file.

Now that you've got a working, personalized copy of the window manager resource file in your home directory, open it in your favorite text editor, and take a look at it. If you're in X at the moment, you can see the correlation between the file and what you see on your desktop. Very helpful indeed. Let's try changing something and see if we can make it look a little better. Probably the first entry in your file (that is not followed by the # comment character) is the `WindowFont` entry, followed by a very bizarre font name. If you want to figure out how to set up the fonts and colors to their fullest, skip ahead to the next section on X Fonts and Colors.

6.2 Advanced FVWM2 Configuration

Configuring FVWM2 can get incredibly subtle and complex. Take this code snippet from the **fvwm2gnome** setup for the *.fvwm2rc* file:

```
# import config files

Echo -Styles-
Read .fvwm2gnome/config/styles/app.styles
Read .fvwm2gnome/config/styles/window.styles

Echo -Buttons & Keys-
Read .fvwm2gnome/config/buttons.config
```

This is an example of a way to read in other configurations. In this fashion you can modularize your

.fvwm2rc -- not a bad idea, considering how jumbled one huge configuration file can become. This c make it easier to change and to debug, and it also makes it easier to configure so-called *themes*, whic will cover in a moment.

6.3 FVWM2 Configuration Shortcuts

A really neat way to configure your script for FVWM2 (and FVWM, and **bash**, and a whole lot of other programs besides!) is to use the **Dotfile Generator**, by Jesper K. Pedersen, available from http://www.imada.ou.dk/~blackie/dotfile/. You will need to have a recent version of **Tcl/Tk** installed (available with most Linux distributions). This program starts with a configuration preset, or takes the one you presently have, and allows you to tweak it via many structured menus of options.

Be warned, however, for I advise learning how to set up your configuration file yourself! Not only is this the *Unix Way* (for what that's worth) but it's much easier to make small and incremental changes to your `.fvwm2rc` file that way, and get things exactly how you like them. Also, being a **Tcl/Tk** program, it's a little slow. Nevertheless, if you want a painless way to start with a decent configuration, this is definitely worth a shot.

6.4 FVWM2 Themes

Among the many new possibilities of FVWM2 are something commonly called *themes*. This basically means that you can switch between a basic look for all of your windows on your desktop, on the fly. Note that if you've made the distinction between the functions of the window manager and the functions of the applications themselves, you will realize that a window manager theme is not going to affect the look and feel of the applications themselves. Integrated toolkits such as KDE and GNOME do have this capability, however, and the two might be used together very effectively.

In order to generate a theme, you must roll up your sleeves and modify your *.fvwm2rc* file a little bit. Here's something I added near the beginning of my file, try putting it right after your `Style` definitions:

```
# Blue Theme
DestroyDecor Blue
AddToDecor Blue
+ WindowFont -b&h-lucida-bold-r-*-*-*-140-*-*-*-*-*-*
+ TitleStyle ActiveDown (Solid DarkSteelBlue)\
    ActiveUp (Solid SteelBlue) Inactive (Solid Grey)
+ HilightColor white blue
+ ButtonStyle 1 -- UseBorderStyle
Style Blue UseDecor Blue, BorderWidth 5, HandleWidth 5,\
    MWMborder, MWMbuttons

# Function to change all windows to a new style.
DestroyFunc ChangeStyle
AddToFunc ChangeStyle
+ "I" Style $0 $1
```

```
+ "I" Recapture
```

That's a lot to swallow, I know. Basically we're first defining a theme called ``Blue,'' and you can use that first half as a model to design other themes. Themes can describe many more features than that, in fact, and can be quite remarkably different from one another. Then we're defining a function to change all the windows to a new style. Notice in both sections above that we destroy the object before creating it. This is a good idea since you may well be restarting FVWM2 a lot to try out your different styles, and this makes it work a bit more smoothly.

The function call is needed as a generic interface to call the definitions of the styles we have defined. Now we will make the menu items to call them.

```
DestroyMenu "Themes"
AddToMenu "Themes"
+ "Choose a theme..." Title
+ "" Nop
+ "Blue" ChangeStyle "*" "UseStyle Blue"
+ "Mwm" ChangeStyle "*" "UseStyle Mwm"
+ "Flat" ChangeStyle "*" "UseStyle Flat"
```

We're being a little terse with the menu definition here, but there should be a lot more in the sample file on your system. What we're doing is calling the `ChangeStyle` function that we defined above to change the style for all the windows on the screen to one of the presets we defined above that. Notice, again, our good practice in destroying the menu before creating it. Now if you restart FVWM2 (you should also have a menu option for that, hopefully!) you will see a new `Themes` menu selection, and you should be able to try out the different themes.

For more examples of FVWM2 Themes, visit http://www.vis.colostate.edu/~scriven/Linux/fvwm/index.html.

---

7. Fonts and Colors

There are a lot of tricks to the fonts and colors used in X. They are not quite as simple as in some other systems, for instance, the font is not just a one-word name. You specify these resources quite explicitly, and it seems rather complex at first, but with a little explanation you'll be a whiz in no time.

7.1 Fonts Demystified

The *X Logical Font Description* (``XLFD'') is the full name for a font. It consists of the following fields:

- `fndry` - font foundry, the company or individual which made the font
- `fmly` - font family, the popular nickname of the font
- `wght` - font weight (bold, medium, etc.)
- `slant` - font slant (italics, oblique, roman (normal), etc.)
- `sWdth` - font width (normal, condensed, extended, etc.)
- `adstyl` - additional style (sans serif, serif, etc.)
- `pxlsz` - pixel size, the number of pixels vertically in a character
- `ptSz` - approximate point size of the text (similar to pxlsz)
- `resx` - horizontal resolution, in dpi
- `resy` - vertical resolution, in dpi
- `spc` - spacing, only useful, apparently, in the Schumacher fonts
- `avgWidth` - average character width of the font
- `rgstry` - the recognized registry that lists the font
- `encdng` - nationality encoding

In light of this chaos, the program **xfontsel** (the default X Window font selection program) will come in enormously useful to you. Try launching it right now. You will see a strange nothing helpful in the main window, but try holding the left button down on the `fndry` button. If all your fonts are in order, you will see a menu of selections such as `adobe` and `b&h` and `bitstream` and so forth. Select one such as `b&h` and you will notice that the font in the lower window changes to something intelligible. This is generally the way you will select fonts with this program, starting from the left, which is the most general selection, and moving toward the right, to the more specific options. Selecting an option toward the rightmost end will not make much sense before the foundry, for instance, is selected, because the options are generally ordered by their dependence on each other.

When you go to select from the `fmly` selection, you will see most of the options greyed out, and only three remaining. That means that these three are the only families of font made by this foundry. Some families appear under more than one foundry, for instance, both *Adobe* and *Bitstream* make a variation of the Courier font. Now you can select the `wght`, and so forth. After you get far enough you will have

TOC

narrowed it down to the font that you want. You don't necessarily have to fill in all the options to ch a single font, there's not *that* many fonts on your system! The options that you do not select will be represented by a `*` indicating that any option will do in that spot.

When you are happy with your font selection, you can hit the select button, and your selection will be placed in the X clipboard, ready to be pasted into your document or whatever you are working on. For instance, go to your **xterm** window and type in something like `xterm -font` followed by an opening quotation mark. Then point to that spot on your screen, and click your middle mouse button (or click both the left and right, if you're middle-button impaired). This will paste the selection from the clipboard, which should be the font you just selected. Then enter the closing quote, and hit `Enter`. For instance, a nice big **xterm** with a Courier font specified would look like this: `xterm -font "-adobe-courier-medium-r-*-*-14-*-*-*-*-*-*-*"` A fresh **xterm** should the pop up using the font that you selected.

The utility **xfd** is very helpful for examining a font. If you launch it with a command line such as `xfd -fn fixed`, it will show you the character set for the font, much like the keycaps utility on a Macintosh. Note that you can also limit the number of fonts that you want **xfontsel** to display with the command line option *-pattern*, followed by a quoted font specification, as discussed above.

7.2 Font Aliases and Configuration

Sometimes it gets tiresome to remember all of the long font names, and very impractical too. Luckily, it is not necessary to type in a hundred keystrokes or so just to get the font name you want, for X provides something called *font aliases*.

If you look in the directory */usr/X11R6/lib/fonts/misc/fonts.alias*, you will find shortcut names for many of the fonts. For example, `8x16` is listed as a shortcut for `-sony-fixed-medium-r-normal--16-120-100-100-c-80-iso8859-1`, and anywhere you enter `8x16` as an X font resource or at a command line for a font name, the long version of the font will be substituted. The *75dpi* and *100dpi* directories have similar aliases to take advantage of, on most systems the Lucida Sans font has a nice shortcut.

If you add fonts or aliases to your system, you will have to issue a few commands (probably as root). If you add fonts, you will have to issue the following two commands (these are examples, you will have to put in the correct directory, or perhaps leave it off altogether, to have your fonts re-read correctly.

```
mkfontdir /usr/lib/X11/fonts/misc
xset fp rehash
```

If you change the alias file for a font, you may only have to issue the last command above, but it may be a good idea to issue them both, to be sure. With the **xset** command you can explicitly issue a font path you wish the server to use, you can delete a specific directory from your font path, see the man page for more information.

Another common problem is that some distributions (notably Red Hat 5.2, at the moment) come with the fonts configured in the wrong order. If you take a look at your */etc/XF86Config* (it may be somewhere

TOC

else in some distributions, unfortunately, and I'm not sure where it is in Red Hat at the moment, so maybe **locate** it...) Take a look at this file if your fonts are ugly in X (as in, very difficult to read even large sizes). You will see a bunch of entries that look something like this:

```
FontPath   "/usr/X11R6/lib/X11/fonts/misc/"
FontPath   "/usr/X11R6/lib/X11/fonts/75dpi/:unscaled"
FontPath   "/usr/X11R6/lib/X11/fonts/100dpi/:unscaled"
FontPath   "/usr/X11R6/lib/X11/fonts/Type1/"
FontPath   "/usr/X11R6/lib/X11/fonts/Speedo/"
```

The important thing to note here is that we have the unscaled bitmapped fonts (the directories misc, 75dpi, 100dpi) before the scaled fonts (such as Type1 and Speedo). Bitmapped fonts are preferred for the X Server, because scaled fonts don't look very pretty for common use, they're better for things like The Gimp or Netscape, perhaps. You should also do a sanity check to see that these directories all do, in fact exist, and if you have modified any lately, you might need to (as root) go in and issue an `mkfontdir *` command in the **fonts** directory.

Note that if you are running a more recent distribution, such as one based on Red Hat 6.0 or later, you won't have to worry about this, as the *xttfs* True Type font server is the default, and does not use the path system to find your fonts.

7.3 Using Type 1 Fonts in X

The font types X generally accepts are of limited use, considering that about the only place you'll find fonts of that kind used are in the X Window System itself, and unfortunately most media junkies and fontaholics work in operating systems that prefer other formats. **Type 1 fonts**, most commonly used in conjunction with PostScript document formats, can be found for free on the internet with considerable ease. Try ftp://ftp.cdrom.com/pub/os2/fonts/ for starters.

To make use of these fonts is not that difficult, and graphics programs like **GIMP** will take advantage of them tremendously, and best of all, the Linux X servers understand Type 1 fonts ``out of the box.'' To use them, first unpack the archive using the appropriate commands, and put the fonts with a `.pfb` extension in */usr/X11R6/lib/X11/fonts/type1/* on your system. Then, add mention of those fonts in the `fonts.scale` file in that directory, using the format for the other fonts, already there. Then save the file.

Now, you should be able to `mkfontdir` to tell the X server about the updates, and then run `xset fp rehash` to re-read the font path. If this doesn't work for you, you might have to restart X to see the changes.

If you find yourself using Type 1 fonts a lot, and for things other than just X (GhostScript, for instance, can handle this font format as well), you'll want to check out the **type1inst** utility by James Macnicol. This program can configure Type 1 fonts for GhostScript and X, and it can also generate sample sheets for your fonts, and does most of the work of configuration for you. This program can usually be found at ftp://metalab.unc.edu/pub/Linux/X11/xutils/.

TOC

7.4 Using TrueType Fonts in X

If you have been using an operating system such as Windows or the MacOS, you may have lots of fonts sitting on your computer already that are **TrueType** fonts. TrueType fonts are considered best on smaller, low-resolution displays, such as your computer monitor, and attempt to provide nice features like shading and antialiasing, making your display look smoother. It is also really easy to find cheap TrueType fonts in bulk, and CD's featuring upwards of 500 fonts in the TrueType format are common in computer stores.

X does not understand TrueType fonts, and has no innate capability (at present) to do so, and so it needs a separate program to do the *font rendering* for it. There is a FreeType library to do this, but if you just want to use them on your computer, all you will need is a program called **xfstt**, the X font server for TrueType. This program should be available at ftp://metalab.unc.edu/pub/Linux/X11/fonts/.

Installation is fairly straightforward. Extract the archive, and `make` the file, and then `make install`. You may get a few errors that don't make sense after the second command; ignore them. What you need to do next is create a writable directory called */usr/ttfonts* and put some of your `.ttf` fonts there, just to make sure it's going to work. Then synchronize the font server with the command `xfstt --sync`.

If you got no errors there, run the font server in the background with the command `xfstt &` and tell your X11 server about the font service with the command `xset fp+ unix/:7100`. If you got no errors, you should be able to run Netscape, the GIMP, even xfontsel, and have those fonts available to you. Your biggest problem may be finding the font you want, but that's not really a bad thing.

It has been brought to my attention that more recent versions of xset require a slightly modified version of this command, because of changes in the code of xset. If you are running a distribution based on Red Hat 6.0 or later, or something equivalent, you will first want to try `xset fp+ unix/:7101`.

If everything seems to be working fine, at this point you will want to configure your system so that the fonts will be working when you start X. If you are just starting X from the command line, this is easy. All you have to do is add the two commands from the previous paragraph to your `.xinitrc` file, in that order. When you next start X, it should work just fine. If you are starting X via **xdm**, you will need to add `/usr/X11R6/bin/xfstt &` to `/usr/X11R6/lib/X11/xdm/Xsetup_0`. Got that? Then add `xset fp+ unix/:7100` as well.

Bear in mind that the installation of a TrueType font server can be tricky business. Remember that the order that you issue the above commands is important. The font server must be started before X is started, otherwise you will run into problems. Be sure to read the FAQ file that came with the distribution, and the `xfstt` man page too. Many newer Linux versions come with this already working, so see if that's not already the case before worrying too much about it.

7.5 Colors

Let's go back to our terminal window and try some more things. Try opening an **xterm** with a command line like the following:

```
xterm -fg darkslateblue -bg red3 &
```


While that window may not be pretty, and you may not do much of your best work in it, it demonstrates one interesting theme of X, the names of the colors. While not very precise, this is a nice way to remember the colors more easily than remembering a series of hexadecimal numbers. Note that color names are never case-sensitive.

If you're interested in the gory details, or want to see the samples, or even want to replace those silly color names with your own for some strange reason, you can find the file listing all the colors with their hexadecimal representation in */usr/X11R6/lib/X11/rgb.txt* on your system. There are also some extremely useful utilities with names like **xcolorsel** and such. They can be found in the usual locations, on http://metalab.unc.edu/pub/Linux/ and elsewhere.

A more precise way of specifying colors is through the numerical definition. This consists of a color space name and a set of values in the following syntax:

<colorspace-name>:<value>/.../<value>

An RGB Device, which you will most likely be using, is specified with the prefix "rgb:" and has the following syntax:

rgb:<red>/<green>/<blue> , where <color> is a 1 to 4 digit hexadecimal number.

As an example, you might represent the color *red* with `rgb:ffff/0/0`. For backward compatibility, you can also specify red with a syntax such as `#ff0000` or `#ffff00000000`, and you will likely be seeing that (older) syntax a lot.

---

8. The X Resources

We will at this point consider some finer modifications to your system. Configuring the window manager only gets you so far. The window manager lets you customize everything that happens *outside* the windows, the frames, the borders, the buttons, the desktop... but what about making the *inside* of the windows pretty? The only way to do this is by modifying the *X Resources* for your system.

X Resources provide a mechanism for storing default values for program resources and tailoring your windowing environment to your favored look and feel. Resources are specified as text strings that are read in from various places when an application is run. Program components are named in a hierarchical fashion, with each object in the hierarchy identified by a class as well as an instance name. At the top level of the hierarchy is the class and instance name of the application itself. By convention, the class name of the application is the same as the program name, but with the first letter capitalized (e.g. Bitmap or Emacs) although some programs that begin with the letter ``X'' also capitalize the second letter for historical reasons (e.g. XTerm).

8.1 X Resources: The Basics

Let's try a few examples to ease you into the world of resources. Start by opening an xterm window. Most likely you have one open already, can open one with a menu selection, or may wish to backtrack a bit in this document to get your bearings! Type something like this into the xterm window:

```
xterm -background blue &
```

This command should have resulted in another xterm window popping up on your screen, but this time with a blue background. ``Wow,'' you say, ``that's amazing!'' Yes, indeed it is.

We'll need a little more background for our next example. Type exit in that xterm (which will close that window) and go back to the original window that you typed that command in. Try this: hold the Ctrl key and then hold the left mouse button. Now try that with the middle mouse button. Now the right. All xterm windows give you options to try while they are running. With the right button, for instance, you can change the font of the window. Notice the color of these menus. Now try this example:

```
xterm -xrm 'xterm*fontMenu*background: green' &
```

This time, it just looks like a normal xterm window. But if you try holding Ctrl and the right mouse button in that window, you will see that this menu (the ``fontMenu'' mentioned in the above example) is green. What just happened? If you look at the man page for **xterm**, you will see many command-line options, such as *-background* or *-font*, that you can set when you launch any given window in X, listed after the options just for **xterm** windows. These particular options are called *X Toolkit Options*, and they apply to just about anything in X.

8.2 Inside The X Resource Database With *editres*

To really get inside the resources of X, we need to run **editres**. Go ahead and launch it (and a sample program to play with) now:

```
xclock &
```

```
editres &
```

Probably you will see a simple clock, and the rather innocent looking, mostly-blank window of **editres**. The main window is showing us the resource tree of any given window on our desktop, and upon program launch, there is none. So let's give it one. Hold down the left button on the Commands menu, and select Get Tree. In the top of the window will appear `Click the mouse pointer on any toolkit client`. This message is more than it seems, and it is a hint to us that not all X applications are toolkit clients (though most of the basic ones are, and the program will usually tell us if it is not in the manual page). The mouse cursor will turn to a crosshair, and wait for you to click on another window.

For this example, let's first click on the **xclock**. You will see a couple things appear in the **editres** window. These are the configurable branches that the toolkit sees. Click on the bottom right one (clock). It should invert. Now select ``Show Resource Box'' from the ``Commands'' menu. You will see another window pop up, with the heading ".xclock.clock.unknown". Bingo!

>From here you can toy with the configuration options of the main **xclock** window. First, click on ``Set Save File'' at the bottom, and put in a filename such as */home/yourname/resources*, to indicate that you're testing some of the resources here. Now click on ``foreground'' and put in ``blue'' next to the ``Enter Resource Value:'' prompt. Then hit ``Apply'' at the bottom. The minute notches around the clock should turn blue. Go to ``background'' and enter ``navajowhite'' (or whatever color you prefer). Then apply that. As you can see, we can configure the whole window just fine this way. But if you were to try launching another **xclock**, it would appear just as this one appeared before any changes. We need to save these changes.

Click ``Save'' and then ``Popdown Resource Box'' (a fancy name indeed for the ``Close'' function!). If you view the file you just created (*cat /home/yourname/resources* perhaps) you will see all those resources spelled out, in correct X Toolkit format, for your convenience. But that does us little good, for if you try launching another **xclock**, it will still look Plain Jane. So here is the last and final step in loading in your resources:

```
xrdb -merge /home/yourname/resources
```

This command *merges* the resources we just wrote into the *X Resource Database* (**xrdb**) for your session of X Window. That means that for all future invocations of the X applications we modified, our changes will take place, and remain binding. So if you run another **xclock**, it will look as nice as you have just now set it up, every time you run it. Mind you, **xrdb** is a complex program, and you may want to have a look at the man page before moving on, or playing around with it some more. If you added the modification to `.xinitrc` listed earlier in this document, to load the resources automatically on X startup, you should only have to worry about the **xrdb** command when you make changes during your X session.

8.3 The Anatomy of X Resources

As you can see we have stumbled across a plethora of configuration options here. This method of configuring X, as has been stated, offers nearly limitless possibilities, and an equivalent amount of confusion. To get some picture of the scope of the resources in just the X Toolkit Intrinsics alone, run the program **viewres**, and play around with it a bit. This program graphically displays the tree of resources in the *Xt Library*.

If you read the manual page for **X**, you will find a rather obscure definition of the exact syntax for defining resources it understands. We can simplify this quite a bit and break it down into this essential syntax definition:

<program><binding><widget><binding><widget><...><resource>:<value>

That doesn't really seem that simple, actually. Well, let's define some things about what has just been said, and it will all start to make sense after all. You can peek ahead to the examples in the next section as you read along, if you wish.

TOC

The *program* in this definition is the invocation of an application in the resource database. This would be `XTerm` for an invocation of **xterm**, `emacs` for an invocation of the **emacs** text editor, or a user-defined name that was given when the applications was launched with the **-name** command line option. In this way you can define separate resources for **xterm**s which will be used in different ways. Which is pretty cool, really.

The *binding* can be one of two characters to separate the widgets and such. If you use a `.` (period), you get a *tight binding*, which means that one widget is directly above or below the other in the widget hierarchy. This also has the highest precedence of the specification methods. If you use a `*` (asterisk), you get a so-called *loose binding*, and you will skip any number of widgets in the hierarchy, and it will attempt to match the next possible widget defined.

The *widget* entries are items in the widget tree, in order of most-specific to least-specific, that they appear in the widget tree, visible with **editres**. Any single widget entry can also be replaced with a `?` (question mark) to skip a single widget definition, and match any possible widget item.

The *resource* item must be specified, and cannot be replaced with the `?` character. This is the most-specific item in the hierarchy, and usually contains items like the actual color to define, actual font to define, and so forth. In fact, everything else before the resource in this definition can be left out and replaced with a single asterisk, but the actual resource to define must be present. If you just put an asterisk and the most-specific resource name, such as `*background: blue`, X will attempt to define that resource globally, for all its clients, if possible.

Following the colon is the *value* entry. This entry defines what the resource will be set to, such as a font name or color value. The value can be specified (depending on context) as a boolean, numeric, or text data type. The value entry, also, cannot be omitted in a valid resource definition.

8.4 Making Your Changes Last With *.Xdefaults*

There is a magic file you can put in your home directory called *.Xdefaults*. If you copy the lines in the resources file from the last example into the *.Xdefaults* in your home directory, you will never have to configure **xclock** again! While this might not be the finest example of its utility, it makes its point. This file can be crammed full of every option you prefer for every type of program you run in X, and if you take proper care of it, you can still easily go back and make slight changes when you need to. But making lots of changes, and hunting down lots of subtle resources using **editres** can be an extremely tiring and painstaking procedure. Indeed, sometimes that's too much work, and most of these resources are already waiting for you, neat and orderly, grouped by program, on your system.

In the directory */var/X11R6/lib/app-defaults* you will find a great many files, all named after an X Toolkit program. If you examine these files you will find that they contain a great many configuration options for each one, and I do mean a great many! You would not want all of these options from all of these files in your *.Xdefaults* file, that would be quite tiresome to deal with. These are the defaults, and it is from these that you can decide what you would like to see changed for your particular configuration.

The following are some samples from my *.Xdefaults* file. Notice a few things we have not yet mentioned about the resource definition files. If a line begins with `!` (exclamation point), it is considered a comment, and the rest of the line is ignored. If the line begins with `#include` *`filename`*, that line is an *include directive*, and at that point in the resources another file will be merged, when it is loaded. This can help keep your resource files from becoming too bloated. And here are some examples:

```
! Default resources for me@localhost xterms
! start with the generic, move to the specific...
*Dialog*Text*font:     -b&h-lucida-medium-r-*-*-12-*-*-*-*-*-*-*
*dialog*value*background: white
*Dialog*Label*font:    -adobe-helvetica-bold-r-*-*-12-*-*-*-*-*-*-*
*MenuButton*font:      -adobe-helvetica-medium-r-*-*-12-*-*-*-*-*-*-*
*MenuButton*background:        grey80
```

TOC

```
*MenuButton*foreground:        black
*Label.font:           -adobe-helvetica-medium-r-*-*-10-*-*-*-*-*-*-*
*Label*shadowWidth:    1
*SmeBSB.font:          -adobe-helvetica-bold-r-*-*-12-*-*-*-*-*-*-*
*SimpleMenu*font:      -adobe-helvetica-medium-r-*-*-10-*-*-*-*-*-*-*
*OptionMenu*font:      -adobe-helvetica-medium-r-*-*-10-*-*-*-*-*-*-*
*Command.font:         -linotype-helvetica-bold-r-narrow-*-12-*-*-*-*-*-*-*
*commandBox*font:      -b&h-lucida-bold-r-*-*-12-*-*-*-*-*-*-*
*Toggle.font:          -adobe-helvetica-bold-o-*-*-12-*-*-*-*-*-*-*
*Form.background:      grey70
*TransientShell*Dialog.background: grey70
*Scrollbar.Foreground: grey80
*Scrollbar.Background: grey50
*Scrollbar*cursorName: top_left_arrow
*Scrollbar*width:      16
*shapeStyle:           Rectangle
*XlwMenu.shadowThickness: 1
*shadowWidth:          1

! xterm stuff
xterm*scrollbar.background: grey40
xterm*foreground: grey90
xterm*background: grey25
xterm*cursorColor: white
xterm*visualbell: on
! rxvt stuff (a quicker, better xterm)
rxvt*color12: steelblue
rxvt*color15: white
rxvt*color9: rgb:ff/7f/5f
rxvt*foreground: grey90
rxvt*background: grey10
rxvt*cursorColor: white
rxvt*font: lucidasanstypewriter-12
rxvt*loginShell: false
rxvt*saveLines: 1024
rxvt*title: shell
rxvt*geometry: 80x25
! Make Xman just a little bit more sane
xman*topBox: false
xman*background: lightsteelblue
xman*foreground: black
! xcalc is too bland by default...
xcalc*Command.font: -adobe-helvetica-bold-r-*-*-10-*-*-*-*-*-*-*
xcalc*customization: -color
! Disallow the <blink> tag in Netscape
Netscape*blinkingEnabled: False

! Merge other resources (example)
# include $HOME/.otherXresources
```

One word of warning with regard to X resources, for KDE users. There's a setting in the control panel which will cause

all your X Resources to be overwritten in your applications. This can be a really nice feature, but it can also be an annoyance if you have painstakingly configured your resources. Open the control panel and find the ``Fonts etc.'' section. There's an option to ``Apply style to non-KDE apps''. If you turn this on, all your programs will adopt the look and feel configured by KDE (you may need to restart X to notice this). If you leave this option turned off, your resources will remain as you have configured them.

8.5 Your Own User Resource Directory

You can also create a directory of resource files, just like the system-wide *app-defaults* directory mentioned above, with one file per program. Just create the directory (for our example we'll use app-defaults under your home directory,) and then set the environment variable `XAPPLRESDIR` to point to it. A good place to set this variable would be in the beginning of your `.xinitrc` file, for example, put in the line `export XAPPLRESDIR=$HOME/app-defaults` (if your files are going to be in an *app-defaults* directory under your home directory).

Now, whenever you start an X program, this directory will be searched for a file with the same name as the resource name of the program, just like the system-wide directory. This is the *client name* that you used in `.Xdefaults` files.

For example, a file called *XTerm* could contain the line `*background: gold`, and all your **xterm**s would, by default, come up with a gold background. This is a nice alternative to a single `.Xdefaults` file, and makes it more clear when trying to decide which settings to configure later on, and to find the ones for a certain program. There are still uses for the `.Xdefaults`, though. It's useful for setting resources not bound to a single program, like modifications that you would make to turn all of a certain kind of button blue, regardless of the application.

---

9. Clients and Application Tips

We have covered a few clients in X, and this section will cover some more. We will limit our discussion here to the most basic and important core items, those which come with X or you are likely to find yourself using with X. If you have installed an integrated desktop environment such as KDE or GNOME, you will have many others to choose from which likely perform many of the same functions. However, it is important to know about and understand many of the fundamental programs in X, because they can be very useful for working with your environment and such. Also X offers many new options that even your normal console applications can take advantage of.

9.1 Screen Savers for X

A common feature of many operating systems is the ability to blank the screen after a specified amount of time, and optionally display some sort of nifty graphics demo thing, a screen saver. There are a couple ways to do that in X, too.

The most basic way to use this feature is by putting a command in your `.xinitrc` startup file such as `xset dpms 2400 3600 4800`. The **xset** program can configure the screen saving features of the X server, not to be confused with the screen blanking that the kernel does when you are at the text console. With the `dpms` option, X can use the power saving features of your monitor as well. The first option configures how many seconds before the screen blanks, the second option is how many seconds before the power saving feature starts, and the third option is for the "off" mode. Turning on an option implicitly enables the feature, setting a feature to zero explicitly disables it.

Many Linux distributions come with **xlock** preinstalled, or as an option. This is a pretty basic and fairly nice screen saver. If you run it with the `-nolock` option, you can see some of the modes that it offers, and if you leave that option off, it will ask you for a password when you move the mouse or press a key, as a security feature. Note that this is no real security, for at a Linux console a user could restart the computer or just drop out of X with a combination of keystrokes. The last method can be disabled, however, and if you are using **xdm**, it will offer as much security as your login, so it may be that only rebooting will let someone in.

A newer and better program is **xscreensaver** by Jamie Zawinski. This program offers a great many niceties, for instance, it can run its processes at a nicer priority level, lessening the load to the system while it's running, and it automatically detects when the screen has been powered down by **xset** and doesn't waste processor time. Also, all of the graphics routines it calls are modular demos, and you can add routines without upgrading the whole package, and it can also call other programs, such as `xearth` or `xdaliclock`, as modules.

The latest version of **xscreensaver** can be found at http://www.jwz.org/xscreensaver/. Once you get it installed and ready to go, here are some nice additions you might wish to add to your `.Xdefaults` file:

```
!!! some XScreenSaver sample defaults
! Time out after 3 minutes, cycle mode after each 2
xscreensaver.timeout: 3
xscreensaver.cycle: 2
! Run very low priority, and fade between modes
xscreensaver.nice: 12
xscreensaver.fadeSeconds: 2
```

9.2 Emacs and XEmacs

If you are a fan of the text editor **EMACS**, or just someone who uses it a lot, you will find your work even easier in X Windows. If you have not tried XEmacs, you may want to get it for use in X. There are features in XEmacs that are nice even if you are not in X, for instance, your text can be colored to match the markup style you are editing automatically. You should give the following modification to your `.emacs` file a shot, and read the info pages for more options. Also look for an option to edit faces in the menus.

```
(global-font-lock-mode t)
(setq font-lock-maximum-decoration t)
```

9.3 Some Useful Programs and Tricks

appres

The appres program prints the resources seen by an application (or sub-hierarchy of an application) with the specified class and instance names. It can be used to determine which resources a particular program will load. Useful for debugging your X defaults and such.

rclock

Many distributions come with this nice replacement for xclock, which saves memory, alerts you when your mail comes, and can pop up reminder messages and launch programs. The **Battery-Powered Mini-HOWTO** contains instructions on patching this utility to show how much battery is left in your laptop, too.

rxvt

A nice replacement for xterm - uses less memory, works faster, lets you put in a background pixmap, and lets you switch fonts with keyboard hotkeys, rather than menus.

xcpustate

Displays CPU state (idle, nice, system, kernel) statistics, as well as Ethernet information.

xearth

Display the earth on your root window, many options for display available. **Xscreensaver** can use

this as a screensaver module, for maximum fun.

xfig

A vector drawing program, particularly useful for charts and documentation. Quite useful but hard to get the hang of at first.

xfontsel

Font selection utility for X Window. Try the command `xterm -fn `xfontsel -print` &` to pick a font and then open the xterm window using that font.

xload

Monitor your memory usage with a moving graph or the lights on your keyboard! If you use Window Maker, look for **wmmon** to do the same, but prettier.

xmag

A magnifying glass for X, with a couple other useful features.

xman

Manual page browser for X. If the little box it starts with gets annoying, launch it with the *-notopbox* option.

xmodmap

Edit and display the keyboard modifier map and keymap table that are used by client applications to convert event keycodes into keysyms, usually run from user's startup script. An example was given earlier in this document, see the man pages for more info.

xpaint

Basic bitmap painting program, for any real work you should grab GIMP.

xset

User preference utility for X. You can change all sorts of stuff with this. For instance, `xset s 600` sets the screen to blank after ten minutes.

xsetroot

Change the color of your desktop. If you have a color selector program like **xcolorsel** installed, try a command like `xsetroot -solid `xcolorsel` ` to pick a color and set your desktop to that color.

xwininfo

You can run this program and click on any window for lots of useful information about it.

The Intellimouse

You can use the Intellimouse in X with a great many applications. There is an excellent resource page located at http://www.inria.fr/koala/colas/mouse-wheel-scroll/.

10. Advanced X Usage

There's a lot to learn about X, a lot of info buried in the inscrutible land of man pages, a lot of things most folks don't bother reading. Some if it is rather important, and it's easy to make some mistakes, and get completely stuck. It can be much harder to build a program in X, or check the security of X, or many other things, because X and all the things that run on top of it are so vast. This section is an attempt to cover some of the more advanced and confusing situations that may arise.

10.1 Libraries and Compiling X Applications

Sooner or later you will have to deal with compiling applications of your own. Later, if you just installed a nice distribution of linux, and are happy with what you've got, sooner, if you're the kind of person who likes to tinker and install. Remember, this is a privilege, not a right, so have fun with it!

First, a few pointers on compiling programs with X. Many newer applications, GNU applictions in particular, come with a script in the root directory called **configure**. This assumes of course that you've extracted the file and are in the directory. This program should be run as `./configure`, and will automatically detect many things about your system. Afterwards, simply running `make` and perhaps becoming root and running `make install` will get the program up on your system.

You may have to do a little more tinkering if you do not have a **configure** script available. Many X programs require you to run a program to make the **Makefile**, called `xmkmf`. If you don't see a **Makefile** in your directory, this will sometimes work, and will generate a suitable configuration for you, and you'll be ready to `make` from there.

Sometimes you will have run the **configure** script, and have been warned that you lack a library that would be helpful or necessary to properly use that application. For instance, I recently installed the **xscreensaver** application, and found that it would support several 3D modes if the **Mesa** library was installed. If you run into this situation and want that library installed, the first thing you should do is check the CD or installation media for your Linux distribution to see if you have the library on there. That may save you lots of trouble trying to compile the library.

Once you get a library compiled and installed and ready to use, you can go back to the directory you were installing your X program from, remove the `config.cache` file, and run the **configure** script again. Hopefully it will find the library you have installed. Sometimes it does not, and you have to tinker to get it working. This happens occasionally with libraries you download off the net, that do not adequately support the Linux File System Standard (most do, however).

10.2 Basic X Security

TOC

It has often been said that X has a very simple security model: All or Nothing. This is not much of an exaggeration at all. X can be configured to use somewhat sophisticated security, via encryption, but that is beyond the scope of this HOWTO (for the present). It is assumed that the user is not using any encryption for this discussion.

First of all, you should try and follow some simple rules when you're compiling programs for X (or for any reason, really). Try not to become root any more than necessary. Configure your programs as a normal user with the `./configure && make` (or just `make` if there's no configure script), and then issue the single command to install it as root by typing `su -c "make install"`, and then typing in the root password at the prompt. That way you won't have to be logged in as root, you can just issue a single command, and you're back to your normal user prompt. This is a smart, safe way to install software.

The next thing to think about is running X software as root. Realize that X is more or less inherently insecure, and if your primary concern for a particular Linux box is security, you really don't want to install X at all! Having said that, some folks want to run nice fancy configuration programs or package management tools in X. I **do not** recommend starting X as root. It's just not a good idea. There's much, much better ways to do these things!

If you want to run an X application as root, just log in under your normal user account, and launch it from there. As I mentioned above, you don't want to be logged in as root any more than absolutely necessary. The root user has the ability to do just about anything on the system, including about a million ways to destroy it completely. Simply go to your **xterm** or such application and type in a command such as `su -c "glint -display :0.0"` to launch the window on your desktop. Now you have a single window running as root, and not the whole desktop. You might also want to consider using the complete pathname to the program you're running as root, for extra security.

Tomasz Motylewski also offers the following tip for **ssh** and **ssh-agent** lovers. If you put the following in your */usr/X11R6/lib/xinit/Xclients* file:

```
eval "exec ssh-agent fvwm${FVWMVER} ${FVWMOPTIONS}" >
"$HOME"/.FVWM${FVWMVER}-errors 2>&
```

That is, replace your standard `fvwm${FVWMVER} ${FVWMOPTIONS}` command with the preceeding. That is to start your window manager as a child of **ssh-agent**. Then you should try `ssh-add < /dev/null` from an **xterm** session (if you have your identity RSA key in the file */.ssh/identity*). If this works for you then you should find that you are authenticated in all your X windows.

It should be noted that this is a suggestion from the reader, and the author has no experience with **ssh** at the present, so proceed at your own risk! More recent Linux distributions allow for setting up **ssh** automatically, via `setup` or something similar, which means that you may not have to worry too much about setting it up, just understanding how it works.

10.3 More About X Authority

TOC

The mysteries of user authentication in X are a prime example of the security problems many users encounter with X. Tomasz Motylewski relates the following story, which sums up the situation quite well.

``...in the default XFree86 configuration X session started by ``startx'' accepts commands from everyone connecting from localhost. If you start X as user1 and you have user2 remotely logged in, user2 has full access to your keyboard and desktop (once my friend played a joke on me and put in his cron commands dumping my X desktop image and sending to him every hour). I did not notice it for 2 weeks!''

Well that's not exactly a good thing, but unfortunately many readers have this exact problem. Tomasz goes on to point out that if you use **xdm** to log in (as detailed earlier in this HOWTO), rather than just running **startx**, this won't be a problem because access is controlled by the **xauth** program. By issuing the **xhost** command, you can see who is allowed access to your X session. Any user from the hosts that are listed in the output of the **xhost** command is authorized full access to your screen and keyboard.

If you want to start your X server in a secure fashion from the command line, you will need to make the following modifications to your configuration. First, modify **startx** (probably at */usr/X11R6/bin/startx*) by changing the last line `xinit $clientargs -- $serverargs` to read `exec xinit $clientargs -- $serverargs -auth ~/.Xauthority`.

Then you will also need to edit your **.xinitrc** or your system-wide */usr/X11R6/lib/xinit/xinitrc* (whichever you are actually using, probably the one in your home directory), by adding this at the beginning of the file:

```
 # if Xserver has been run with "-auth ~/.Xauthority" option
this will
 # prevent other users on your machine to connect to your X server
 # unless you allow it explicitly using xhost +host or give them
 # your ~/.Xauthority file.
 xauth generate $DISPLAY . trusted
```

Always be sure and run **xhost** to check the security that you have configured, to make sure everything is working correctly.

---

11. Bibliography and Other Resources

There is lots of information on X Window to be found, on your computer right now, on the Internet, and in some excellent books available at your favorite bookseller. Give some of the following resources a try, they have been extremely helpful to the author, and can be extremely helpful to you, too.

- Check the Linux Documentation Project website for a lot more documentation on Linux, X, and related items.
- http://www.x11.org/is sort of a clearinghouse for all things X.
- http://www.themes.org/is a headquarters for themes for various window managers.
- The X Consortium's web site is http://www.x.org/... or perhaps it's moved to http://www.opengroup.com/.
- XFree86 can be found at http://www.xfree86.org/.
- The O'Reilly series on X Window! Visit http://www.ora.com/for the definitive books on X.
- Much more information on using TrueType with X is at http://www.freetype.org/.
- The man pages for **X**, **xterm**, **XFree86**, and for any other clients you find yourself using often, are very useful and quite information-packed, and highly recommended. As the oft-repeated saying goes, RTFM.
- There is a **Remote X Apps** MINI-HOWTO that is very helpful in figuring out how to run local and remote clients with X.
- Don't forget to visit http://www.gnome.org/ and http://www.kde.org/ for the latest on unified desktop environments in Linux, which are becoming more and more commonplace all the time.

The 3 Button Serial Mouse mini-HOWTO

Geoff Short, geoff@kipper.york.ac.uk

v1.33, 31 May 1998

How to get a 3 button serial mouse working properly under Linux.

1. Disclaimer

2. Introduction

3. Serial Ports

4. Switched Mice

5. Normal Mice

6. Switching a Mouse to 3-Button Mode

7. Wheeled mice

8. Using gpm to Switch Mouse Modes

9. Using two mice

10. XF86Config and Xconfig file examples

11. Cables, extensions and adaptors

12. Miscellaneous Problems and Setups

13. Models Tested

14. Further Information

15. Mouse Tail

1. Disclaimer

The following document is offered in good faith as comprising only safe programming and procedures. No responsibility is accepted by the author for any loss or damage caused in any way to any person or equipment, as a direct or indirect consequence of following these instructions.

2. Introduction

The most recent version of this document can always be found at http://kipper.york.ac.uk/mouse.html

There is a Japanese translation at http://jf.gee.kyoto-u.ac.jp/JF/JF-ftp/euc/3-Button-Mouse.euc; and a French one at http://www.freenix.fr/linux/HOWTO/mini/3-Button-Mouse.html. Other translations may be available - check your local LDP mirrors.

Most X applications are written with the assumption that the user will be working with a 3 button mouse. Serial mice are commonly used on computers and are cheap to buy. Many of these mice have 3 buttons and claim to use the Microsoft protocol, which in theory means they are ideal for the X windows setup. (The record for the cheapest working 3 button mouse currently stands at $1.14!)

Most dual-protocol mice will work in two modes:

- 2-button Microsoft mode.
- 3-button MouseSystems mode.

This document leads you through the different steps needed to configure your mouse in these two different modes, especially the steps needed to use the more useful 3-button mode.

As distributions become easier to set up, some of the problems ought to go away. For instance, RedHat have a `mouseconfig` program to set things up for you. However, some versions of RH5.0 had a bug in `mouseconfig`, so make sure you check for patches.

3. Serial Ports

The first thing to do is to make sure the software can find the mouse. Work out which serial port your mouse is connected to - usually this will be `/dev/ttyS0` (COM1 under DOS) or `/dev/ttyS1` (COM2). (`ttyS0` is usually the 9 pin socket, `ttyS1` the 25 pin socket, but of course there is no hard and fast rule about these things.) There are also an equivalent number of `/dev/cua` devices, which are almost the same as the `ttyS` ones, but their use is now discouraged. For convenience make a new link `/dev/mouse` pointing at this port. For instance, for `ttyS0`:

```
ln -s /dev/ttyS0 /dev/mouse
```

4. Switched Mice

Some mice, not usually the cheapest ones, have a switch on the bottom marked `2/3'. Sometimes this may be `PC/MS'. In this case the `2' setting is for 2 button Microsoft mode, and the `3' for 3 button MouseSystems mode. The `PC/MS' switch is a bit more complicated. You will probably find the `MS' setting is for Microsoft, and the `PC' is for MouseSystems. You may find the `PC' setting described as ps/2 mode, but it should do MouseSystems as well. If you have such a mouse, you can switch the switch to `3' or `PC', put the MouseSystems settings in your XConfigs (see below) and the mouse should work perfectly in 3-button mode.

5. Normal Mice

If you don't have any switches, and no instructions, then a little bit of experimentation is needed. The first thing to try is to assume the mouse maker is telling the truth, and the mouse is full Microsoft. Set up your Xconfigs to expect a Microsoft mouse (see the Xconfig section) and give it a try.

If the mouse didn't work at all, then you don't have a Microsoft mouse, or there is some other problem. Try the other protocols in the configs, the man page for the config file is the best place to start looking. Also look in the Miscellaneous Problems section below.

What you will probably find is that when you run X, the mouse works fine but only the outer two buttons do anything. You can of course accept this, and emulate the third button (press both buttons at once to click the middle one) like you do with a two button mouse. To do this, change your Xconfig file as shown in the Xconfig example section below. This may mean you have bought a 3 button mouse for no good reason, and you are certainly no further forward. So, now you need to look at your hardware.

6. Switching a Mouse to 3-Button Mode

Even cheap mice can also work under the Mouse Systems protocol, with all three buttons working. The trick is to get the mouse to think it's a Mouse Systems one, something you rarely see in your instructions.

- Before you power up your computer, hold down the left mouse button (and keep it held down until it has booted to be on the safe side).

When the mouse first gets power, if the left button is held down it switches into Mouse Systems mode. A simple fact, but not always publicised. Note that a soft reboot of your computer may not cut the mouse power and therefore may not work. There are a number of other ways of switching the mode, which may or may not work with your particular mouse. Some of these are less drastic than rebooting your computer, two are more so!

- If your computer is get-at-able you can unplug the mouse and plug it back in with the button held down (although you shouldn't normally plug things in to a live computer, the RS232 spec says it is OK).
- You may be able to reset the mouse by typing `echo "*n" > /dev/mouse`, which should have the same effect as unplugging it. Hold the left button down for Mouse Systems mode, not for Microsoft. You could put this in whatever script you use to start X up.
- Bob Nichols (rnichols@interaccess.com) has written a small c program to do the same thing, which may work if `echo "*n"` does not (and vice versa). You can find a copy of his source code at http://kipper.york.ac.uk/src/fix-mouse.c
- Someone has reported that the `ClearDTR' line in the Xconfig is enough to switch their mouse into Mouse Systems mode.
- If you are brave enough, open the mouse up (remember that this will invalidate your warranty) and have a look inside. In some cases, the mouse may have a switch inside, for some strange reason known only to the manufacturer. More likely on the cheap mice is a jumper which you can move. The switch or jumper may have the same effect as a `MS/PC' switch described in the Switched Mice section above. You may find that the circuit board is designed for a switch between 2 & 3 buttons, but it hasn't been fitted. It will look something like:

```
 -----------
| o | o | o |   SW1
 -----------
  1   2   3
```

  Try linking pins 1-2 or 2-3, and see if it changes the behaviour of the mouse. If it does, you can either fit a small switch, or solder across the contacts for a quick and permanent solution.
- Another soldering solution which might be a last-resort for mice which don't understand MouseSystems at all, from Peter Benie (pjb1008@chiark.chu.cam.ac.uk). If the middle button's switch is double-pole, connect one side of the switch to the left button's switch, and the other side to right button's switch. If it's not a double pole switch then use diodes rather than wire. Now, the middle button pushes the left and right buttons down together. Select `ChordMiddle` in the XF86Config and you have a working middle button.
- The ultimate recourse with the soldering iron was first described to me by Brian Craft (bcboy@pyramid.bio.brandeis.edu). Two common generic mouse chips are the 16 pin **Z8350**, and the 18 pin **HM8350A**. On each of these chips, one pin controls the mode of the chip, as follows.

```
Pin 3   Mode
-----   ----
Open    Default Microsoft. Mouse Systems if a button is held on power-up.
GND     Always Mouse Systems.
```

```
Vdd     Always Microsoft.
```

(Pins are numbered as follows:)

```
          ____
pin1  -| \/ |-
pin2  -|    |-
pin3  -|    |-
      -|    |-
      -|    |-
      -|    |-
      -|    |-
pin8  -|____|-
```

(This info comes courtesy of Hans-Christoph Wirth, and Juergen Exner, who posted it to de.comp.os.linux.hardware) You can solder a link between pin 3 and gnd, which will fix the mouse into MouseSystems mode.

- Peter Fredriksson (peterf@lysator.liu.se) has tried the SYSGRATION **SYS2005** chip, and found that linking Pin 3 to Gnd forced Mouse System mode.
- Uli Drescher (ud@digi.ruhr.de) confirms it works on an **HN8348A** chip; Ben Ketcham (bketcham@anvilite.murkworks.net) confirms the **HM8348A** (Pin 9 is Gnd).
- Urban Widmark (ubbe@ts.umu.se) says the same applies to the **EC3567A1** chip, where Pin 8 is ground. I've tried it as well and it works fine.
- Timo T Metsala (metsala@cc.helsinki.fi) has found that on the **HT6510A** chip pin 3 is mode select, pin 9 is Gnd. The same works for the **HT6513A** chip. Holtek also make **HT6513B** and **HT6513F** chips - on these, pin 8 is Gnd.
- Robert Romanowski (robin@cs.tu-berlin.de) says pin 3 - pin 8 (Gnd) works on an **EM83701BP** chip too.
- Robert Kaiser (rkaiser@sysgo.de) confirms that pin 3 - Gnd works on a **EC3576A1** chip too.
- Sean Cross (secross@whidbey.com) found it was pin 2 - pin 7 (Gnd) on a **HM8370GP** chip.
- Peter Fox (fox@roestock.demon.co.uk) used pin 3 - pin 8 on a **HM8348A** chip.
- Jon Klein (jbklein@mindspring.com) found pin 3 - pin 9 did the trick for a **UA5212S** chip.

- As an alternative to the above soldering methods, you can get the mouse to hold it's own button down when booting: this circuit from Mathias Katzer.

```
          -----
       ---  R  ---------O------ + Supply
      |   -----   |        |                     C = 100nF capacitor
      |           | E      |                     R = 100kOhm
      |       __ /         |                     T = BC557 transistor
      |      /  \          O
      |   B | #V | T         /
      |-----|-#  |          /   Left button switch of the mouse
      |     | #\ |         O
      |      \__/          |
     ---         \  C      |
     --- C        ------O----------> (to somewhere deep inside the mouse)
      |
     ###  Ground
```

The test mouse was a no-name model MUS2S - whether this works in other mice depends on the circuit of the

mouse; if the switch is connected to ground and not to +Supply, an npn-transistor like the BC547 should work and C have to be swapped then, too.

So there you have it, the choice is yours. Stick with the default Microsoft two buttons, or work out how to switch the mode and set X up to take advantage of this.

---

7. Wheeled mice

Mice with wheels have emerged in the last few years, starting with the Microsoft Intellimouse and spreading to other manufacturers. The wheel can be clicked like a button, or rolled up and down. Far and away the best reference for information is http://www.inria.fr/koala/colas/mouse-wheel-scroll/ which describes how to get lots of X applications to recognise the scrolling action.

In general, you'll need a fairly new Xserver to use the scrolling action, but some older servers will recognise the clicking actions. For instance, the Intellimouse is supported by XFree 3.3.1 and later.

8. Using gpm to Switch Mouse Modes

`gpm` is the program that lets you use the mouse in console mode. It is usually included in linux distributions, and can be started from the command line or in the startup script `/etc/rc.d/rc.local`. Note that distributions don't always have the most recent version (1.13 at time of writing) which can be found on mirrors of sunsite.unc.edu.

The main modes for serial mice under gpm are:

```
gpm -t ms
gpm -t msc
gpm -t help
```

for Microsoft or MouseSystems modes, or to probe the mouse for you and tell you what it found. To run gpm in MouseSystems mode, you may need a `-3` flag, and possibly a DTR option, using the `-o dtr` flag:

```
gpm -3 -o dtr -t msc
```

`gpm` is often able to recognise all three buttons of the mouse even in Microsoft mode. And newer versions (Version 1.0 and later (?)) can then make this information available to other programs. For this to work, you need to run gpm with the -R tag, like this:

```
gpm -R -t ms
```

This will make gpm re-export the mouse data to a new device, called `/dev/gpmdata`, which looks like a mouse to any other program. Note that this device **always** uses the MouseSystems protocol. You can then set your Xconfig to use this instead of `/dev/mouse` as shown below, but of course you must ensure gpm is always running when you use X. Some people have reported that some middle-button events are not correctly interpreted by X using this technique, this may be down to an individual mouse setup.

Changing button mapping for gpm and X (gustafso@math.utah.edu)

You may find that gpm uses different default button mappings to X, so using both systems on the same machine can be confusing. To make X use the same buttons for select and paste operations as gpm, use the X command

```
xmodmap -e "pointer = 1 3 2"
```

which causes the left button to select and the right button to paste, for either 2-button or 3-button mice. To force gpm to use the X standard button mapping, start it with a `-B` command, eg:

```
gpm -t msc -B 132
```

9. Using two mice

In some cases, for instance a laptop with a built-in pointing device, you may wish to use a serial mouse as a second device. In most cases the built-in device uses the PS/2 protocol, and can be ignored if you don't wish to use it. Simply configure gpm or X to use `/dev/ttyS0` (or whatever) as usual.

To use both at once, you can use `gpm -M` to re-export the devices. More details in the gpm man page. Also, XFree 3.3.1 and later support muliple input devices, using the XInput mechanism. Auto-generated XF86Config files should have the necessary comments in them.

10. XF86Config and Xconfig file examples

The location of your configuration file for X depends on the particular release and distribution you have. It will probably be either `/etc/Xconfig, /etc/XF86Config` or `/usr/X11/lib/X11/XF86Config.` You should see which one it is when you start X - it will be echoed to the screen before all the options are displayed. The syntax is slightly different between the XF86Config and Xconfig files, so both are given.

Microsoft Serial Mouse

- XF86config:

```
Section "Pointer"
    Protocol "microsoft"
    Device "/dev/mouse"
EndSection
```

- Xconfig:

```
#
# Mouse definition and related parameters
#
Microsoft      "/dev/mouse"
```

Microsoft Serial Mouse with Three Button Emulation

- XF86config:

```
Section "Pointer"
    Protocol "microsoft"
    Device "/dev/mouse"
    Emulate3Buttons
EndSection
```

- Xconfig:

```
#
# Mouse definition and related parameters
#
Microsoft      "/dev/mouse"
Emulate3Buttons
```

MouseSystems Three Button Serial Mouse

- XF86config:

```
Section "Pointer"
    Protocol "mousesystems"
    Device "/dev/mouse"
    ClearDTR            #  These two lines probably won't be needed,
```

```
        ClearRTS            #  try without first and then just the D
    EndSection
```

- Xconfig:

```
    #
    # Mouse definition and related parameters
    #
    MouseSystems    "/dev/mouse"
    ClearDTR                #  These two lines probably won't be needed,
    ClearRTS                #  try without first and then just the DTR
```

Microsoft Serial Mouse with gpm -R

- XF86config:

```
    Section "Pointer"
        Protocol "MouseSystems"
        Device "/dev/gpmdata"
    EndSection
```

- Xconfig:

```
    #
    # Mouse definition and related parameters
    #
    MouseSystems      "/dev/gpmdata"
```

11. Cables, extensions and adaptors

The only wires needed in a mouse cable are as follows: TxD and RxD for data transfer, RTS and/or DTR for power sources, and ground. Translated into pin numbers, they are:

	9-pin port	25-pin port
TxD	3	2
RxD	2	3
RTS	7	4
DTR	4	20
Gnd	5	7

The above table may be of use if you wish to make adaptors between 9- and 25-pin plugs, or extension cables.

12. Miscellaneous Problems and Setups

- If you have trouble with your mouse in X or console mode, check you are not running a getty on the serial line, or anything else such as a modem for that matter. Also check for IRQ conflicts.
- It is possible that you need to hold down the left button when booting X windows. Some systems may send some sort of signal or spike to the mouse when X starts.
- Problems with serial devices may be due to the serial port not being initialised correctly at boot. This is done by the `setserial` command, run from the start-up script `/etc/rc.d/rc.serial.` Check the man page for `setserial` and the Serial-HOWTO for more details. It may be worth a little experimentation with types, for instance try `setserial /dev/mouse uart 16550` or `16550a` regardless of what port you actually have. (For instance, mice don't like the 16c550AF).
- The `ClearDTR` flag may not work properly on some systems, unless you disable the RTS/CTS handshaking with the command:

  ```
  stty -crtscts < /dev/mouse
  ```

  (Tested on UART 16450/Pentium by Vladimir Geogjaev geogjaev@wave.sio.rssi.ru)
- Logitech mice may require the line `ChordMiddle` to enable the middle of the three buttons to work. This line replaces `Emulate3Buttons` or goes after the `/dev/mouse` line in the config file. You may well need the `ClearDTR` and `ClearRTS` lines in your Xconfig. Some Logitech mice positively do not need the `ChordMiddle` line - one symptom of this problem is that menus seem to move with the mouse instead of scrolling down. (From: chang@platform.com)
- Swapping buttons: use the `xmodmap` command to change which physical button registers as each mouse click. eg: `xmodmap -e "pointer = 3 2 1"` will turn round the buttons for use in the left hand. If you only have a two-button mouse then it's just numbers 1 & 2.
- Acceleration: use the `xset m` command to change the mouse settings. eg `xset m 2` will set the acceleration to 2. Look at the manpage for full details.
- Pointer offset: If the click action appears to be coming from the left or right of where the cursor is, it may be that your screen is not aligned. This is a problem with the S3 driver, which you may be able to fix using xvidtune. Try `Invert_VCLK/InvertVCLK, or EarlySC.` This info from Bill Lavender (lavender@MCS.COM) and Simon Hargrave. In the XF86Config, it might look like this:

  ```
  Subsection "Display"
      Modes       "1024x768" "800x600" "640x480" "1280x1024"
      Invert_VCLK "*" 1
      ...
  ```
- If you are getting `bouncing' of the mouse buttons, ie two clicks when you only wanted one, there may be something wrong with the mouse. This problem has been solved for Logitech mice by Bob

Nichols (rnichols@interaccess.com) and involves soldering some resistors and a chip to the mouse to debounce the microswitches.

- If some users cannot get the mouse to work but some (eg root) can, it is possible that the users are not running exactly the same thing - for instance a different version of X or a different Xconfig. Check the X start-up messages carefully to make sure.
- If you find the mouse pointer is erasing things from your screen, you have a server config problem. Try adding the option `linear`, or maybe `nolinear` to the graphics card section, or if it is a PCI board, the options `tgui_pci_write_off` and `tgui_pci_read_off`. (This seems to be a Trident Card problem.)
- If the mouse cursor doesn't show up on the screen, but otherwise seems to be working, try the option `"sw_cursor"` in the Device section of the config file.
- If your mouse stops working when its sunny or when you turn a light on, it may be that the sensors are being swamped by light getting through the case. You could try painting the inside of the case black, or putting some card in the top.
- Microsoft Brand mice are often a cause of problems. The newest ``Microsoft Serial Mouse 2.1A'' has been reported not to work on many systems, although unplugging it and plugging it in again may help. `gpm` version 1.13 and higher should also support 2.1A mice, using the `pnp` mouse type. (See the gpm section for how to re-export this.) The ``Microsoft Intellimouse'' also causes problems, although it should now be supported by XFree version 3.3 and later.

---

13. Models Tested

There are a lot of different mice out there, and I cannot honestly say that you should go out and buy one rather than the other. What I can do is give a list of what I think these mice do, based on experience and heresay. Even with this information you should be a little cautious - we had two identical mice in our office on two computers, some things worked on one and not t'other! Any additions to this list would be welcome.

Mouse Systems optical mouse, serial version
Works well (as you might expect from the name!) without ClearDTR or ClearRTS in the config.
WiN mouse, as sold by Office World for eight quid.
Standard dual-mode Microsoft/MouseSystems.
Agiler Mouse 2900
Standard dual-mode Microsoft/MouseSystems. SYSGRATION SYS2005 chip is solderable.
Sicos mouse,
Works ok, needs ClearDTR & Clear RTS in config.
Index sell a mouse for 10 quid,
Doesn't work in 3 button mode, but does have nice instructions :-)
Artec mouse
Usual dual-protocol mouse, needs `ClearDTR' set in config, NOT `ClearRTS'
DynaPoint 3 button serial mouse.
Usual dual-protocol mouse, needs `ClearDTR' AND `ClearRTS' in Xconfig.
Genius Easymouse 3 button mouse
Works fine with Mouseman protocol without the ChordMiddle parameter set. From Roderick Johnstone (rmj@ast.cam.ac.uk)
Truemouse, made in Taiwan
Works OK, needs `ClearDTR' in config. (From Tim MacEachern)
Champ brand mouse
Needs to have switch in PC mode, which enables MouseSystems protocol also. (From tnugent@gucis.cit.gu.edu.au)
MicroSpeed mouse
Usual dual-protocol mouse.
Venus brand ($7)
Has a jumper inside to switch between 2 and 3 button mode. (From mhoward@mth.com)
Saturn
Switched mouse, works OK as MouseSystems in 3-button position. (From grant@oj.rsmas.miami.edu .)
Manhattan mouse.
Switch for `MS AM' / `PC AT' modes, MS mode works fine with the `gpm -R` method. (From komanec@umel.fee.vutbr.cz).
Inland mouse.

TOC

Switch for `PC/MS' modes, works fine. (From http://ptsg.eecs.berkeley.edu/~venkates).

qMouse (3-button), FCC ID E6qmouse X31.
Sells in the USA for about $10. Works with `gpm -t msc -r 20'. No jumpers or switches for MouseSystems 3-button mode. Unreliable in X. Does not respond to `echo "*n" > /dev/mouse`.

Mitsumi Mouse (2-button), FCC ID EW4ECM-S3101.
Sells in the USA for about $12. Reliable in X and under gpm, smooth double-button. (These two from gustafso@math.utah.edu)

PC Accessories mouse that i got from CompUSA for under $10.
Has PC/MS switch on bottom. Works OK. (From steveb@communique.net)

First Mouse - seriously cheap at 7.79 pounds at Tempo.
Dual Microsoft/MouseSystems, mode set by button depress at power-up. No switches, no links. Four wire connection, echo '*n' doesn't work. `gpm -R' works a treat. (From peterk@henhouse.demon.co.uk)

Trust 3-button mouse.
Dual-mode with switch, works OK as MouseSystems in `PC' mode. gpm doesn't like the Microsoft mode.

Chic 410
Works perfectly when kept in ms mode and used with the gpm -R command. From Stephen M. Weiss (steve@esc.ie.lehigh.edu)

KeyMouse 3-button mouse.
Works OK with ClearDTR and ClearRTS in Xconfig; `-o dtr' needed with gpm. (From EZ4PHIL@aol.com)

Qtronix keyboard `Scorpio 60'
All three buttons work in MouseSystems protocol. (From hwe@uebemc.siemens.de)

Tecra 720 laptop
The glidepoint is on /dev/cua0; the stick is on /dev/psaux. (From apollo@anl.gov)

Anubis mouse
Works fine, need to hold down left button whenever switching to the X virtual console. (From Joel Crisp)

Yakumo No.1900 mouse
Works with `gpm -R -t ms` exporting to X. (From Oliver Schwank)

Genius `Easy Trak' Trackball
Is **not** Microsoft compatible, use `Mouseman` in the Xconfig and it will work fine. (From VTanger@aol.com.)

Highscreen Mouse Pro
`Works fine' says alfonso@univaq.it.

Logitech CA series
Works in X using MMseries protocol, at 2400 Baud, 150 SampleRate. (Should also apply to Logitech CC, CE, C7 & C9 mice). (From vkochend@nyx.net.)

A4-Tech mouse
Works OK, needs `DTR` line under both X and gpm. (From deane@gooroos.com)

Vertech mouse
Normal Microsoft/Mousesystems behaviour, can be soldered for a permenant fix. (From

TOC

duncan@fs3.ph.man.ac.uk.)

Boeder M-7 ``Bit Star'' (and other M series apart from M13)
Switches to Mousesystems protocol by holding any button down at power-on. (From mailto:sjt@tappin.force9.co.uk.)

Mouse Systems ``Scroll'' Mouse (four buttons and a roller/button)
Has a 2/3 switch - in mode 3 functions as a three button MouseSystems mouse, ignoring extra button & wheel. Doesn't need `ClearRTS/DTR.` (From parker1@airmail.net.)

Radio Shack 3-button Serial Mouse
Model 26-8432, available in Tandy for about 20 quid. Works as Mousesystems with ClearDTR. (From Sherilyn@sidaway.demon.co.uk.)

Dexxa serial mouse
Works fine using Microsoft protocol in Xconfig, no ChordMiddle or anything needed. (From mailto:slevy@ncsa.uiuc.edu.)

Belkin 3 button mouse
As purchased from Sears (\$10), needs `-o rts` under gpm (and probably `ClearRTS` under X) when in PC mode. (From mailto:mmicek@csz.com.)

---

14. Further Information

- Mouse Systems has a web site at http://www.mousesystems.com/. They have a Windows driver if you need one.
- The Linux Serial HOWTO is available from mirrors of sunsite around the world. If you don't know where your nearest mirror is, start at http://sunsite.unc.edu/mdw/linux.html
- There is a very good explanation of how mice work at http://www.4QD.co.uk/faq/meece.html.
- Fuller details of the Xconfig and XF86Config files are found on the relevant man pages, and in the documentation about installing X windows such as the Xfree86 HOWTO. Also, see the XFree86 FAQ at a mirror of http://www.XFree86.org/.
- Information about gpm can be found on the man page, also try the web page of Darin Ernst at http://www.castle.net/X-notebook/mouse.txt.
- Lots of information on mice hardware and software can be found at http://www.hut.fi/Misc/Electronics/pc/interface.html#mouse

15. Mouse Tail

Much of the information for this document has been trawled from the various linux newsgroups. I am sorry I did not keep a record of everyone who has indirectly contributed by this route, thank you all very much.

So, to sum up:

- Even cheap 3 button Microsoft mice can be made to work.
- Configure your copy of X to expect a Mouse Systems mouse.
- Hold down the left button at power-on to switch the mouse to MouseSystems mode.
- You might need to hold the left button down when starting X.
- Mice are more intelligent than you think.

Linux ADSM Mini-Howto

by Thomas König, Thomas.Koenig@ciw.uni-karlsruhe.de

v, 15 January 1997

This document describes how to install and use a client for the commercial ADSM backup system for Linux/i386.

1. Introduction

2. Installing the iBCS module

3. Installing the ADSM client

4. Running the client

5. Known Problems

1. Introduction

ADSM is a network-based backup system, sold by IBM, in use at many organizations. There are clients for a large variety of systems (different UNIX brands, Windows, Novell, Mac, Windows NT). Unfortunately, at the time of this writing, there is no native Linux version.

You will have to use the SCO binary, and install the iBCS2-emulator for running ADSM. This description is for ADSM v2r1.

At the time if this writing, I am only aware of a version which works with the i386 version of Linux.

2. Installing the iBCS module

The iBCS2 module is available from ftp://tsx-11.mit.edu/pub/linux/BETA/ibcs2. If you are running kernel version 1.2.13, get `ibcs-1.2-950721.tar.gz`, unpac it and apply the patches `ibcs-1.2-950808.patch1` and `ibcs-1.2-950828.patch2`. You can then type "`make`" and install the iBCS modlue with "`insmod`".

For a 2.0 kernel version, get `ibcs-2.0-960610.tar.gz`, unpack it in a suitable place, chdir into that directory, and apply the following patch:

```
--- iBCSemul/ipc.c.orig Wed Jan 15 21:32:15 1997
+++ iBCSemul/ipc.c      Wed Jan 15 21:32:31 1997
@@ -212,7 +212,7 @@
        switch (command) {
                case U_SEMCTL:
                        cmd = ibcs_sem_trans(arg3);
-                       arg4 = (union semun *)get_syscall_parameter (regs, 4);
+                       arg4 = (union semun *)(((unsigned long *) regs->esp) + (5));
                        is_p = (struct ibcs_semid_ds *)get_fs_long(arg4->buf);
 #ifdef IBCS_TRACE
                        if ((ibcs_trace & TRACE_API) || ibcs_func_p->trace)
```

Then, copy `CONFIG.i386` to `CONFIG`, and type `make`.

If you don't have them already, create the needed device files by executing

```
# cd /dev
# ln -s null XOR
# ln -s null X0R
# mknod socksys c 30 0
# mknod spx c 30 1
```

3. Installing the ADSM client

The SCO binary is supplied as three tar files, or disks. Change to the root directory, set your umask according to your policies, and unpack them from there (as root). In your Directory /tmp, you will find an installation script; execute that.

You will then have to hand-edit `/usr/adsm/dsm.sys` and `/usr/adsm/dsm.opt`. In `dsm.sys`, important lines to specify are:

Servername

The name of the server

TCPServeraddress

The fully qualified host name of the server

NODename

Your own hostname

In `dsm.opt`, you will have to specify

Server

As before

Followsymbolic

Wether or not to follow symbolic links (not a good idea, in general)

SUbdir

Wether to back up subdirectories (you usually want that)

domain

The file systems to back up

You will then have to create a SCO-compatible `/etc/mnttab` from your `/etc/fstab`. You can use the following Perl script, `fstab2mnttab`, for this.

```
#!/usr/bin/perl

$mnttab_struct = "a32 a32 I L";

open(MTAB, "/etc/mtab") || die "Cannot open /etc/mtab: $!\n";
open(MNTTAB, ">/etc/mnttab") || die "Cannot open /etc/mnttab: $!\n";

while(<MTAB>) {
    next if /pid/;
```

```
        chop;
        /^(\S*)\s(\S*)\s(\S*)\s.*$/;
        $device = $1;
        $mountpt = $2;
        $fstype = $3;
        if($fstype ne "nfs" && $fstype ne "proc") {
            $mnttab_rec =
                pack($mnttab_struct, $device, $mountpt, 0x9d2f, time());
            syswrite(MNTTAB, $mnttab_rec, 72);
            print "Made entry for: $device $mountpt $fstype\n";
        }
    }

    close(MNTTAB);
    exit 0;
```

You do not need to install any shared libraries for these clients; everything is linked statically.

---

4. Running the client

There are two clients, `dsm`, which is an X11 interface, and `dsmc`, a command-line interface. Your computer centre will tell you how to run it. Some startup script at boot, for example

```
dsmc schedule -quiet 2>&1 >/dev/null &
```

will probably be required.

5. Known Problems

Unfortunately, SCO can only deal with hostnames no longer than eight characters. If your hostname is longer, or fully qualified, you may need to specify your hostname on the `NODename` line in `/usr/adsm/dsm.sys`.

If you use the DISPLAY variable, you will have to supply the fully qualified host name (i.e. `DISPLAY=host.full.do.main:0` instead of `DISPLAY=host:0`).

ADSL HOWTO for Linux Systems

David Fannin, dfannin@sushisoft.com

v0.92, 10 April 1999

How to setup a Linux System to work with Asymmetric Digital Subscriber Loop (ADSL), a new high-speed digital access line technology available from Telcos. ADSL is one of a family of digital subscriber line (DSL) technologies available to residential and business customers using existing copper loops, providing speeds from 384kbps to 1.5Mbps. This document provides an introduction to ADSL, and information on how to install, configuration and turn up service.

1. Legal Stuff

2. Introduction

3. ADSL Overview

- 3.1 What is ADSL?
- 3.2 ADSL Applications
- 3.3 What is xDSL/DSL?
- 3.4 Why so many speeds?

4. How it works.

- 4.1 CPE: ADSL ANT and NIC
- 4.2 Splitter or Splitterless Design
- 4.3 DSLAM
- 4.4 ISP connection

TOC

5. Ordering Service

- 5.1 Home Requirements
- 5.2 Telco Options
- 5.3 ISP Options

6. Wiring it up

- 6.1 Wire the Splitter/NID (at the SNI)
- 6.2 Wire the DSL Jack (at the computer location)
- 6.3 Install the ANT (at the computer location)

7. Configuring Linux

- 7.1 Install and Connect the NIC card
- 7.2 Configure the Ethernet Interface
- 7.3 Setting up a Router
- 7.4 Setting up a Firewall/Masquerading

8. Appendix

- 8.1 FAQs
- 8.2 Links
- 8.3 Credits
- 8.4 Glossary

---

1. Legal Stuff

ADSL HOWTO for Linux Systems

Copyright (C)1998,1999 David Fannin.

This document is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You can get a copy of the GNU GPL at at http://www.gnu.org/copyleft/gpl.html.

2. Introduction

This document addresses the ordering, installation and configuration of ADSL service for use on Linux Systems.

ADSL or Asymmetric Digital Subscriber Loop is a high-speed Internet access technology that uses a commonly available telephone copper loop (same wire as your phone service). Designed to run on up to 80% of the telephone available in the United States, and utilizing line-adaptive modulation, ADSL provides data speeds from 384kbps to 1.5 Mbps, normally using different speeds for the upstream and downstream channel (hence the " Asymmetric"). ADSL provides a direct, dedicated pipe to an ISP.

ADSL was designed for SOHO applications, such as a small business with 5-30 systems, or the high-end Linux user that has wants bandwidth and has some money to spend. Applications range from low-bandwidth server interconnection to streaming video applications. ADSL is designed and priced somewhere in between ISDN and T1 service, providing near T1 speeds without the cost/complexity/availability issues of T1. Since ADSL is a dedicated service, it avoids the delays and use charges inherent in ISDN service.

This HOW-TO starts with a description of ADSL services and it many variants, and a block diagram of the components that make up the service. Next, a list of requirements for ordering ADSL are provided. Also included is guide on ADSL wiring and configuring your Linux System. I have also included an Appendix with a FAQ, a listing of interesting Links,and a Glossary.

I welcome any and all comments on this document; They can be sent to `dfannin@dnai.com` .

The latest version of this FAQ is available from http://www.sushisoft.com/adsl/.

3. ADSL Overview

3.1 What is ADSL?

Asymmetric Digital Subscriber Line (ADSL) is:

- A Telephone Loop Technology that uses existing phone lines
- Provides high-speed data and analog voice (Data over Voice)
- Dedicated digital line for an IP connection
- Data rates (North America) combinations of :
  Upstream/downstream
  256 kbps/256 kbps
  384 kbps/128 kbps
  384 kbps/384 kbps
  384 kbps/1.5 Mbps
  and many others
- Wide range of CPE options, including Ethernet 10baseT Interfaces.
- Dedicated ISP connection (static or dynamic addresses)
- Can support an IP subnet (from 1 to 254 IP addresses, depending on ISP)
- Priced lower than dedicated private line (T1) connections

3.2 ADSL Applications

ADSL was designed to provide a dedicated, high-speed data connection for Internet/Intranet Access, using existing copper phone lines. This allows ADSL to work on over 60-80% of the phone lines existing in the U.S. without modification. Additionally, ADSL provides speeds approaching T1 (1.5Mbps), which are much greater than analog modems (56kbps) or ISDN (128kbps) services provided over the same type of line. ADSL is usually priced to be much less other dedicated digital services, and is expected to priced somewhere between T1 and ISDN services (including the ISDN usage charges).

The Telcos see ADSL as a competitive offering to the Cable Company's Cable Modems, and as such, are expected to provide competitive pricing/configuration offerings. Although Cable Modems are advertised as having 10-30Mbps bandwidth, they use a shared transmission medium with many other users on the same line, and therefore performance varies, perhaps greatly, with the amount of traffic and other users.

ADSL is positioned for Home and Small Office (SOHO) applications that require high-speed Internet Access. Since it also provides dedicated access, It can be used for interconnecting low-bandwidth servers to the Internet, and would provide a great access solution for 5-20 PCs in an Office location. It is also a great solution for those Linux power users that just want high speed access from home:-).

TOC

3.3 What is xDSL/DSL?

Digital Subscriber Line (DSL) provides a dedicated digital circuit from your home to the Telcos central office, using analog telephone line. DSL also provides a separate channel for voice phone conversations, which means analog calls (voice, fax. etc.) can be carried at the same time high-speed data is flowing across the line. DSL uses the frequency spectrum between 0kHz-4kHz for Analog Voice, and 4kHz-2.2MHz for data. xDSL is a generic acronym for a family of dedicated services, where the " x"stands for:

- **ADSL** Asymmetric Digital Subscriber Line: 1.5 Mbps-384kbps/384-128kbps
- **HDSL** High-bit-rate Digital Subscriber Line: 1.5 Mbps/1.5 Mbps (4Wire)
- **SDSL** Single-line Digital Subscriber Line: 1.5 Mbps/1.5 Mbps (2Wire)
- **VDSL** Very high Digital Subscriber Line: 13 Mbps-52 Mbps/1.5 Mbps- 2.3 Mbps.
- **IDSL** ISDN Digital Subscriber Line: 128 Kbps/128 Kbps.
- **RADSL** Rate Adaptive Digital Subscriber Line: 384kbps/128kbps
- **UDSL** Universal Digital Subscriber Line: 1.0Mbps-384kbps/384kbps-128kbps
  also called " splitterless" DSL or DSL-Lite, as it doesn't require a splitter.

where Xbps/Ybps is X=Downstream Bit rate, Y=Upstream Bit rate

3.4 Why so many speeds?

ADSL has to work over existing phone lines, which were designed 100 years ago, and were never designed for digital services (See the FAQ answers for more information). Also, ADSL is a new service, and all the providers are trying to find the right price/feature combinations that will make it in the market.

For the average user, the basic way of thinking about it is to segment the options into three categories:

- **Low End Residential**

  Speed ranges from 384kbps-128kbps, Asymmetric
- **High End Residential or Business End User**

  Speed ranges from 1.5Mbps-384kbps, Asymmetric
- **High End Server**

  Speed ranges from +2.0Mbps-1.1kbps, Symmetric

---

4. How it works.

ADSL is made up of several parts (shown by Figures 1 and 2):

- ADSL Network Termination (ANT) and Network Interface Card (NIC)
- Splitter or Splitterless Design
- DSLAM and Telco Loop
- ISP connection

```
Figure 1: ADSL Block Diagram (POTS Splitter)


<-------Home/Office--------------->              <-----Telco Central Office---->
                                NID
                               -----                -----
2 wire     X-----------Voice-=| S |               | D |
phone                         | P |               | S |=----- Voice Switch
line                          | L |    2 wire     | L |
                              | I |=-------------=| A |
                              | T |   Local Loop  | M |=----- ISP Connection
10baseT       ---------- Data | T |               |   |
Ethernet  X--=|        |=----=| E |               -----
  or          ----------      | R |
ATMF            ADSL          -----
NIC             ANT
```

4.1 CPE: ADSL ANT and NIC

The Customer Premises Equipment (CPE) for ADSL consists of the ANT and/or NIC card. The ADSL Network Termination (ANT), shown in Figure 1, is located at your home or office, provides an IP connection. ANTs come in several types:

- Router ANT with 10/100baseT Interface
- Bridge ANT with 10/100baseT Interface
- ANT with ATMF Interface
- ANT with USB Interface
- Integrated ANT/NIC Card

In each case, the ANT/NIC provides the a router address to an ISP. Each Telco will specify the configurations that they will allow. The most desirable configuration for the Linux user is the ANT with a 10baseT Interface, since the cost/setup is the easiest. The other options require special drivers, which have, to date, not been made available for Linux. The bad news is that some providers allow only integrated ANT/NIC PCI cards that do not have Linux Drivers.

Warning! Make sure any third party ANT/NIC you may purchase are compatible with your Telco provider. There are two major line encodings for ADSL (CAP, DMT), and several options for IP encapsulation. Your Telco should provide

you a list of allowable options.

The ANT is connected to your house's inside wire (2 wire phone line). This inside wire is connected to the data side of a POTS splitter, or, in the case of the splitterless version, directly connected to the local loop. Figure 1 shows the POTS splitter wiring, and Figure 2 shows a splitterless type.

In my case, I was provided with an Alcatel ANT, which supports a 10baseT (wired as a Crossover) RJ45 jack. I understand that a NIC card that fits directly in the PC will be available sometime in the future.

Figure 2: ADSL Block Diagram (Splitterless Design)

```
<-------Home/Office--------------->          <-----Telco Central Office---->
                             SNI
                              -               -----
2 wire    X-[RJ11]---Voice----|               | D |
phone       Filter            |               | S |=----- Voice Switch
line                          |    2 wire     | L |
                              |=-------------=| A |
                              |   Local Loop  | M |=----- ISP Connection
10baseT       ---------- Data |               |   |
Ethernet  X--=|        |=-----|               -----
  or          ----------
ATMF            ADSL
NIC             ANT
```

4.2 Splitter or Splitterless Design

Somehow, the digital and analog signals need to be separated for all of this to work. Thus, a filter needs to be placed in the signal path at some point. There are two methods for doing this: Using a POTS Splitter or using RJ11 phone jack filters.

First, in the POTS splitter method, device is located on the "side of the house" where the Telco line is connected. The splitter provides two functions. First, it is the "demarcation point" that separates the Telco wiring from the inside wiring. Second, it "splits" the DSL signal from Telco into a separated data channel and a voice channel. The voice channel is a normal analog phone line (2 wire), and the data channel is sent to the ANT. The splitter is a passive, non-powered device, which will allow the voice channel to operate even if the power fails at the home location. The Telco signal is sent to the splitter using an existing 2 wire line to the home. The Splitter is housed in the Network Interface Device (NID) on the outside of your house.

Second, in the splitterless design, shown in Figure 2, the outside local loop is connected directly to the inside wire at the Subscriber Network Interface (SNI), the same box that is used today at your house. At each extension jack where you wish to plug in an analog phone, you place a special jack that contains a filter that removes the digital signal. This is called an RJ11 filter (RJ11 is the official Telco term used for your 4/6 pin phone jack). The extension used for your ANT does NOT use a filter (otherwise it won't work). That's all there is to it! It should also be noted that some low speed ADSLs will not require RJ-11 filters.

The splitterless design is very desirable from the Telco point of view, as they won't have to roll any trucks to do the install work, and allows them to offer ADSL at a lower price. For most users, it doesn't really matter, in fact, the analog phones will still work without the RJ11 filter in place. The only thing is that you will hear a bit of a high pitched whine when you use the phone. However, this is not recommended, as later version may damage the phone or have some other nasty effect.

TOC

4.3 DSLAM

The DSLAM is the equipment at the Telco Central Office that that splits the signal and connects the user to the Voice Switch and ISP. That's all you need to know from a user perspective.

4.4 ISP connection

An ISP connects to the DSLAM via a high-speed data connection, usually ATM over a T3 (45Mbps) or OC-3 (155Mbps). The important thing here is that an ISP must "subscribe" with your Telco to provide this connection.

---

5. Ordering Service

The basic procedure is as follows:

- Check to see if you meet the following requirements for ADSL.
- Select a Telco and ISP, and see what configurations they offer.
- Call the Telco and order the service.

Once you have ordered the service, the Telco should provide you with a Due Date. I was able to do the entire order in one phone call through my Telco, and they coordinated with the order with my ISP. I was given a due date that was one week later, and everything was completed on time, with no problems.

5.1 Home Requirements

- An appropriate NIC card

  The system (PC, workstation, router or hub) that will be directly connected to the ADSL ANT .
- Inside Wiring

  You need inside wire (2 or 4 wire) from the Telco Demarcation Point (the side of your house) to your PC location. Your existing phone line can be used. However, this inside wire can only be used for the ANT, and no other phones or extensions can be connected to this line. The Telcos or other companies can be contracted to install a new wire for you as well.
- Installation

  Who will wire it up in your house? You can do the splitter wiring yourself (My local Telco provided a wiring kit and instructions for performing the inside wiring) or you can contract them do it for an additional fee. You can save around $ 150-250 dollars by doing it yourself, which I did, or you can save yourself the hassle and have them do it. You can decide what you want to do after reading the procedure that I have included in this document. The bottom line is if you are used to dealing with phone and/or network LAN wiring procedures, and feel comfortable with doing this, then you should consider going for it. If you haven't, and your company is going to pay for it, then have them do it.

5.2 Telco Options

- Qualified Copper Loop

  The Telco will normally test your line before you order service to see if it is can pass the ADSL signal. You must be within 2-3 miles (11-16 kilofeet) of the Central Office, and have a loop that has no loading coils, bridge taps, DAMLs, or other impediments. Most of the Loops that support ISDN can also support ADSL, but since ADSL is more restrictive, this is no guarantee of success.

TOC

Your Telco will help you here.

- ADSL Available from your Telco

  Is the ADSL/DSLAM equipment is available at your Central Office? Again, the Telco will check to see if it available in your area. Every major Telco in the US, and many new dedicated ADSL providers have announced plans for ADSL. I expect wide scale deployment by 1Q99.

- Bit Rate

  ADSL is priced according to the Bit Rate - Check with your Telco. You can usually order some combination of 128k,256k,384k, or 1.5M speeds. Your loop may restrict the bit rate you can be offered. My Telco also prices the service differently if you're a Residence or Business Customer.

- POTS Service

  Some providers (usually the big Telcos) offer POTS with ASDL. If you choose this option, you will need to specify which phone number (new or existing) you want assigned to it. You will need to designation whether you want the ADSL to go over an existing POTS service, or whether you want a new phone number for the analog phone line. If you have ISDN service, and want to use the same line, the ISDN service must first be disconnected. Note that some Telcos (usually dedicated DSL providers) do not offer this option.

5.3 ISP Options

- ISP Presubscription

  Your ISP of choice must be presubscribed with the ADSL Telco provider. Check with your ISP to see if they are connected. If they are not connected then you must either move to another ISP, convince them to connect up, or decide not to use ADSL.

- IP Subnet and Address.

  First, you'll need to know whether your IP addressing will be static or dynamic. Dynamic addressing is set through the use of DHCP.

  Additionally, for static addressing, ISPs will allow anywhere from 1 to 254 IP addresses to be assigned to an ADSL line, with the corresponding Variable Length Subnet Mask (VLSM).

  My ISP charges more money as you get more hosts. I suspect that most applications will use either 1 host (32 bit mask) or 6 hosts (29 bit mask). If your confused, then read the *IP-Subnetting HOWTO*. Also remember than multi-host subnets require 1 of the hosts to be the ISP router address (meaning that the 6 host subnet is really 5 hosts to you) and you can use IP Masquerading/ (see the HOWTO) or Network Address Translation (NAT) to support a larger number of addresses behind your LAN. If you request more than 30 hosts, be prepared to fill out a justification form for your ISP.

- Gateway Address

  The default gateway address. Normally, this is the first host address in your subnet. (e.g. if your subnet range and mask is 192.168.1.240/29, then your host range is .241 through .246, with the default gateway assigned to the .241 address.

- Number of MAC addresses

  The ANTs can support a set number of MAC addresses for bridging. For example, the ADSL ANT that I use is limited to 16 MAC addresses. This should be sufficient for most applications. However, if you plan to support a large number of machines directly off the hub of the ANT, you should check first to see what your limit is.

- DNS Servers

  The ISP should provide you with one or more DNS server addresses to support Domain Name Lookups. Some ISPs also provide Domain Name Server hosting (either primary or secondary server) for you as part of the dedicated service packages. See your ISP for more details.

- User ID and Password.

  The ISP will normally assign you a login/password for shell, mail, etc. access. My ISP also provides web space, mail server, and some other goodies.

---

6. Wiring it up

If you have ordered the installation option from the Telco, then you can skip this section and move to " Configuring Linux" . The following procedures are meant to illustrate the wiring process. Please note that your procedures may be different at your location. Make sure you follow any warnings or safety instructions provided, and that you are familiar with Telco wiring procedures.

The first step will be to wire up the connections from your Telco. Identify the line on which service will be installed, and the locations of your splitter and DSL jacks.

6.1 Wire the Splitter/NID (at the SNI)

If you have the splitterless design, you can skip this part.

The splitter will consist of two parts, the splitter and a small outdoor housing, called the Network Interface Device. Mount the splitter and NID housing per the Telcos instructions at the Subscriber Network Interface (SNI) point, usually the side of your house where the phone line is located. The phone company will need to access the splitter for maintenance, so its advisable to locate it on the outside where they can get at it. Connect the incoming 2W Telco line to the " LINE"side of the splitter. Then wire the inside pair for your telephone to the "VOICE" , and your inside wire pair for the ANT to "DATA".

Checkstep At this point, you should be able to pull dial tone off the voice side of the splitter. If this doesn't work, then either you've wired it wrong, or the ADSL service is not yet connected on the Telco side.

6.2 Wire the DSL Jack (at the computer location)

Wire up the DSL jack (RJ11) at your computer location (connected to the DATA side of the splitter). The specifics differ greatly for each situation, but basically you will have a 2 wire pair that you will connect to the DSL jack. Make sure you read the directions, as the DSL-RJ11 wiring is different for phones and ADSL jacks. In my case, you wire the phone jack on the red/green pair (the two inside prongs of the RJ11 jack) and the ADSL jack is wired on the yellow/black pair (the two outside prongs of the RJ11 jack).

6.3 Install the ANT (at the computer location)

Connect up the ANT's power cord, and connect the phone line (category 3 cable) between the DSL jack and the ANT. This cable is usually provided.

Checkstep At this point, verify the ANT syncs up to the Telco signal. Most ANTs have an green LED

that lights up when the signal is good. If it doesn't sync, then check your wiring, or make sure that th Telco signal is being sent (do this by calling your Telco and verifying they have activated the service Note that having dial tone on the line does NOT confirm the presence of the ADSL data signal.)

If you have completed the previous steps, you are now ready to move on to connecting your Linux System.

---

7. Configuring Linux

After you have wired up the ANT and you're getting the sync signal, then you're ready to configure your Linux System and verify your connection to your ISP. Although I will refer to a Linux System, you can connect any type of 10baseT device to the ANT. This includes a router, hub, PC, or any other system that you wish to use.

Caution! *Before you connect to your ISP*, make sure you understand all security issues of having a direct connection to the Internet via ADSL. Depending on your ISP, most outside uses can access your systems, and you should setup any firewalls, deactivate ports/services, and setup any passwords prior to connecting your machine to the world. Read the *Security-HOWTO* if you need a good overview on this subject.

7.1 Install and Connect the NIC card

Install your NIC card in your Linux machine, configure the kernel, etc., etc. See the various Linux references for doing this. See the *Ethernet-HOWTO* for more information.

Connect the RJ45 cable between the NIC and the ANT. **Note** A gotcha here is that some ANTs are already wired as a 10baseT crossover, and require a direct Category 5 cable for a direct connection to a NIC, rather than a crossover cable. I lost around 12 hours figuring this one out, so don't make the same mistake - make sure you read the instructions first.

7.2 Configure the Ethernet Interface

Configure the IP address, Subnet Mask, Default Gateway, and DNS server information. Each Linux Distribution (RH, debian, Slackware, S.U.S.E.) has a different way of doing this, so check on your particulars. You can also do this manually using the `ifconfig` and `route` commands. See the *NET3-HOWTO* for information for more information.

Once your system is configured, see if you can ping to the your default gateway address provided by the ISP. If the ping is successful, the you should see around 20 ms roundtrip delay for this connection. Congratulations, you're connected to the Net!

7.3 Setting up a Router

Depending on your local setup, you should consider some other issues. These include a firewall setup, and any associated configurations. For my setup, shown in Figure 3, I use an old i486 machine configured as a firewall/router between the ADSL connection and the rest of my machines. I use private IP addresses on my Private LAN subnet, and have configured my router to provide IP Masquerading and Firewalling between the LAN and Internet connection. See the *IP_Masquerading-HOWTO*, and

TOC

Firewall-HOWTO for more information. My experience is that Linux provides superior routing/firewalling performance , and is much cheap than a commercial router, if you find an old 386 machine that you may be using as a doorstop somewhere.

```
Figure 3: My  SOHO Network Setup


<-Private Subnet-->         <-Public Subnet->    <-ADSL Line--------->
                                     |
                                 X----|
                                     |
     X------|                    X----|     |----|
            |      |--------|         |     |ADSL|            Internet
            |      | Linux  |         |-----|ANT |----------> Service
     X------|------| System |--------|     |    |            Provider
            |    E1|(Router)|E0       |     |----|            Router
            |      |--------|         |
     X------|        IP_Masq      10baseT
                  IP_Firewall     Hub
```

What I did is setup a router (Linux RH 5.0 on a i486) with two Ethernet interfaces. One interface routes to the ISP subnet/gateway, and the other interface supports a class private network address (i.e. 192.168.2.x). Using the private network address behind your router allows some additional security because it is not directly addressable from outside your ISP. You have to explicitly masquerade your private addresses in order to connect to the Internet.

Caution Make sure your kernel is complied with IP forwarding and the IP forwarding is turned on. You can check this by

```
cat /proc/sys/net/ipv4/ip_forward
```

The value is "1" for on, and "0" for off. You can change this value by echoing the desired value into this file.

```
(e.g.) echo 1 > /proc/sys/net/ipv4/ip_forward
```

Will turn forwarding on.

7.4 Setting up a Firewall/Masquerading

If you have a direct connection to the Internet, then you want to also turn on Firewall Administration and Masquerading. Figure 4 shows a picture of this.

Caution! I also need to stress that this configuration is only part of the things that need to be performed to create a secure environment. Other considerations include turning off ftp, telnet, and other services on the Router, and ensuring all password, login, etc. configurations are correctly setup for your environment. Make sure you read the *Security-HOWTO*.

TOC

Figure 4: Firewall/Masquerading for ADSL

```
       |-------|       |-------|     |-X
======X| ADSL  |=------| Linux |-----|
ADSL   |  ANT  |     E0|       |E1   |-X   Private Network
Line   |-------|       |-------|     |     (e.g. 192.168.2.x)
               <------->             |...
             ISP Subnet or host
            (Public Net Address)
```

The kernel for the Linux router is complied for IP forwarding/masquerading, and has the "ipfwadm" (IP firewall software) installed with the following options:

file: /etc/rc.d/rc.firewall (called by rc.sysinit in RH5.0)

```
echo "Setting up the firewall"
#
# From the "Firewall-HOWTO"
#
# flushes all setting
#
ipfwadm -F -f
#
# set the firewall
#
ipfwadm -F -p deny
#
# allow any machine with address 192.168.2.x to masquerade.
#
ipfwadm -F -a accept -m -S 192.168.2.0/24 -D 0.0.0.0/0
#
# allow the domain name server to work (udp 53)
#
ipfwadm -F -a accept -b -P udp -S 0.0.0.0/0 53 -D 192.168.2.0/24
ipfwadm -F -p masquerade
#
# the rest just list out the options for your enjoyment
#
ipfwadm -F -l
ipfwadm -O -l
ipfwadm -I -l
```

You need to be careful, as some application will still not work without special modules (namely ftp, real audio, and some others). Check the ipfwadm documentation for more information. I found this pretty easy to set up.

Additionally, using the private network addressing scheme is cheap, and gives an administrator com flexibility in setting up their local LAN. The drawback is that Masquerading has a limit on the numb private hosts that it can reasonably support, and that some IP applications that pass the host address in their data fields will not work, but this tends to be a limited number.

---

8. Appendix

8.1 FAQs

This is the Frequently Asked Questions (FAQ) section for ADSL.

1. Q: Are there ADSL Standards.

   A: Sort of. The U.S. Bell Operating Companies have standardized on Discrete Multi-Tone (DMT) ANTs (ANSI T1.413) in their current rollout. Most others should follow their lead in the states. There are other types of ANTs, most notably Carrier-less Amplitude Phase Modulation (CAP), which, of course, are incompatible with each other.

   A biased comparison from an DMT-based vendor on this subject can be found at the Aware. Still, it provides the best detail on this issue I have seen so far.

   A rather expensive copy of the ANSI standard can be ordered at: American National Standards Institute ANSI Home Page
   Asymmetric Digital Subscriber Line (ADSL) Metallic Interface
   ANSI TI.413-1995
   Note: ANSI TI.413 Issue 2 was released September 26, 1997

2. Q: Can I use ATM to connect to ADSL ANT?

   A: Yes, you can! Some ADSL ANT (at least the Alcatel version) has a ATM Forum 25Mbps interface, which connects to a PCI NIC card. However, I have not yet heard of any Linux drivers for such cards.

3. Q: Why the heck does ADSL have all these bit rates (384/1.5/8M/20M/etc) options?

   A: The basic problem is the 100 year old design of the copper loop. It works great for analog phone, but it presents a real challenge for a digital signal. Remember that the distance of a loop is inversely proportional to the data rate that it can carry. Rate-Adaptive technologies are great for making a digital signal work in many situations, but it can't provide a consistent bandwidth for all applications, especially for very long (over 18 kilofeet) loops. The different bandwidth that you see advertised reflect various marketing wars of vendors equipment, and the Telco struggle to finalize on a "standard" set of data rates. I think that the 384k/1.5Mbps will become the standard for now. The high bit rates will only be available for special application and/or situations, since they can only be provided on a small percentage of the available loops.

   Also, check out the next question on the loop imparments that cause this to happen.

4. Q: What are all these loop imparments (bridge taps, loading coils) that could disqualify my line from using ADSL? (thanks to Bruce Ediger)

   Load coils: in-line inductances that improve voice-frequency transmission characteristics of a

TOC

telephone circuit. Essentially, a "load" steals energy from high frequencies and gives it to low frequencies. Typically only used in very long (>9,000 ft) phone lines.

By "bridges" I assume you mean "bridged taps". In older neighborhoods, the phone wiring will have been used by more than one customer. Perhaps these customers lived at different (though near-by) addresses. The unconnected "spur" of wiring is a "bridged tab" on the currently connected circuit.

Digital loop carriers: there's a bunch of systems for carrying more than one voice transmission on a single pair of wires. You can shift the frequencies up or down, or you can digitize the voice transmissions and divide the telephone circuit by time or code or something. The more general term is "pair gain".

These things cause different problems for high-frequency communication.

Loads will completely mess up things by filtering high frequencies and passing low frequencies. They probably also change the "delay envelope", allowing some frequencies to arrive before others. One byte's tones will interfere with the next byte's.

Bridged taps act as shunt capacitances if they're long in relation to the signals wavelength, and they'll actually act as band pass filters if they're about 1/4 wavelength of the signal. That is, they'll pass particular frequencies freely. Particular tones of a DMT modem might get shunted back, rather than passed along to the receiving modem, reducing bandwidth for that telephone line.

Pair gain, digital or analog, limit the bandwidth available to one transmission in order to multiplex several on one wire. High and low tones of a DMT transmission get filtered out by the apparatus.

The book "Subscriber Loop Signaling and Transmission Handbook", by Whitham D. Reeve, , IEEE Press 1992, ISBN 0-87942-274-2 covers the math of how to calculate the effect of line length, bridged tap, etc on the transmission characteristics of a telephone line. It's pretty expensive, however.

5. Q: Do you have examples of ADSL ANTs?

   A: Short Answer: Yes. Real Answer: The evolution of this technology is moving too rapidly for anyone to keep up to date in a HOWTO. A good source of ADSL ANTs is the ADSL Forum Home Page . Go to the Vendors pages to see what's happening.

   However, I will provide a list of some of the current technology as of June 1998.

   - Router ANT with 10/100baseT Interface

     Examples: Flowpoint 2000 DSL (CAP), Netspeed Speedrunner 202 (CAP), Speedrunner 204 (CAP), 3COM Viper-DSL (CAP), StarNet Ezlink 500/100 (DMT), Westell ATU-R-Flexcap (CAP), Aware x200
   - Bridge ANT with 10/100baseT Interface

     Examples: Alcatel A1000 (DMT), Westell ATU-R-Flexcap2 (CAP)
   - ANT with ATMF Interface

     Examples: Alcatel A1000 (DMT), Netspeed Speedrunner 203 (CAP), Ariel Horizon II

TOC

- Bridge ANT with V.35 Serial Interface (T1, Serial Router)

  Examples: Westell ATU-R
- ANT with USB Interface

  Rumored to being pushed by Intel.
- Integrated ANT/NIC Card

  Examples: Netspeed PCI Runner (CAP), Efficient Networks Speedstream 3020 (DMT)

  These are NOT endorsements of the products listed, just provided for illustration.;-).

8.2 Links

- ADSL Forum Home Page A comprehensive web site created by the adsl vendors. Fairly complete for reference information on ADSL.
- Dan Kegels ADSL Page A good general reference on xDSL - includes vendor, service provider, and other links. This page was getting a little long in the tooth as of 2Q98. Dan also maintains a super page on ISDN.
- PacBell's ADSL Page Pacific Bell is the local Telco and my provider of ADSL service.
-
- Jeremie's Unofficial Ameritech ADSL FAQ
- Telechoice xDSL News Page
- ADSL Deployment 'round the World Claims to have a complete list - looked accurate for my area - gives providers, prices, speeds, etc.
- Bell Atlantic ADSL Home Page
- comp.dcom.xdsl FAQ

8.3 Credits

Thanks to all those that contributed information to this HOWTO. I have anti-spammed their email addresses for their safety (and mine!). Remove the X's from their names.

- **B Ediger** (Xbediger@csn.net) Great Description of loop impairment.
- **C Wiesner** (Xcraig@wkmn.com) List of many ADSL URLs.
- **J Leeuw** (Xjacco2@dds.nl) Many tips on ADSL, especially in Europe
- **J Kass** (Xjeremie@umich.edu) Unoffical Ameritech ASDL FAQ
- **N Silberstein** (Xnick@tpdinc.com) Info on Netrunner and his experience with US Worst.

8.4 Glossary

TOC

A dictionary of some of the jargon I use in this FAQ.

2 wire Copper Loop

The two wire twisted pair from the Telco Central Office that terminates at a customer location.

ADSL

Asymmetric Digital Subscriber Line

ANT

ADSL Network Termination (a.k.a. the ADSL modem)

ATM

Asynchronous Transfer Mode - provides high-speed packet switching from 155 Mbps to (currently) 2Gbps. Used to provide backbone switching for the Internet.

ATMF-25Mbps

ATM Forum Interface - 25Mbps speed, provided by a PCI NIC card.. One of the interfaces used between the ANT and PC.

Central Office

Usually refers to one of two meanings -1) The Telco Building that houses Telephone equipment 2) The Telco Voice Switch that provides dial tone.

CPE

Customer Premises Equipment - The Telco term for customer equipment (i.e. the stuff you are responsible for fixing). Examples are CSU/DSU, modems, ANTs, and your phone.

DHCP

Dynamic Host Configuration Protocol - The IP protocol used to set up dynamically assigned IP addresses.

DS0

The basic digital circuit for Telcos - offered at 56 kbps or 64kbps. Can support one analog voice channel.

DSLAM

Digital Subscriber Line Access Multiplexer - The Telco equipment that concentrates and multiplexes the DSL lines.

xDSL

Digital Subscriber Line - A term describing a family of DSL services, including ADSL, SDSL, VDSL, etc.

HDC

TOC

See Section 2

ISDN

Innovations Subscribers Don't Need; I Still Don't kNow or maybe Integrated Services Digital Network, a digital phone service that uses a single copper pair to run 2B (64k) + 1D(16k) channels that can be used for switched voice or data.

ISP

Internet Service Provider

NID

Network Interface Device - The housing used to protect the ADSL splitter from the elements.

NIC

Network Interface Card - A PC card (PCI/ISA) that supports the required network interface. Usually an Ethernet 10baseT or an ATMF-25Mbps Card..

POTS

Plain Old Telephone Service - The service that provides a single analog voice line. (i.e. your phone line)

Recursion

See "Recursion"

SNI

Subscriber Network Interface - The Telco term for the phone wiring housing on the side of your house. It designates the point between the Telco side and the Inside Wire. This is also called the Demarcation Point.

Splitter

The passive device (low-bandpass filter) at the SNI that splits the ADSL signal into separate voice and data channels.

Splitterless

An ADSL installation that does not require the Splitter. For higher speeds, a RJ11 filter is placed on every extension phone jack where an analog phone is used, thus providing the filtering at the jack, rather than at the NID. For lower speeds, no filter is required.

SOHO

Small Office HOme

T1

a.k.a DS1 - A digital dedicated line at 1.544 Mbps, used for both Voice (24 DS0s) or Data.

T3

a.k.a DS3 - A digital dedicated line at 44.736 Mbps, provides for both Voice (672 DS0s or 28 DS1s) or Data

Linux & AI/Alife mini-HOWTO

TOC

Version 2.0

This document is maintained by John A. Eikenberry

The master page for this document is http://www.ai.uga.edu/~jae/ai.html

Last modified: Tue Jan 13 17:23:21 EST 1998

Table of Contents

- What's New
- Introduction
  - Purpose
  - Finding the Software
  - Updates and Comments
- Programming Languages
- Traditional Artificial Intelligence
  - AI class/code libraries
  - AI software kits, applications, etc.
- Connectionism (neural nets)
  - Connectionist Class/Code Libraries
  - Connectionist Applications
- Evolutionary Computing (EC) (genetic algorithms/programming)
  - EC Class/Code Libraries
  - EC Applications
- Artificial Life
  - Alife class/code libraries
  - Alife software kits, applications, etc.
- Autonomous Agents and Bots
- AI & Alife related newsgroups
- AI & Alife resource links

TOC

What's New

Note: For a list of new additions to this mini-howto, please visit the master page listed above.

(1.11.98)Thanks to Dirk Heise for his submission of Gas. I've also removed a couple more of the non-AI oriented scheme implementations. I just felt having 10 scheme implementations listed was a bit silly, especially since some of them weren't really meant for AI stuff. If you want a listing of more scheme's look here or here.

(12.29.97) Added a couple more entries and web links, including a patched lil-gp.

(10.1.97) Ok. I struck the motherload of Baysean Networks, see the Connectionism section for more. Plus I cleaned off the 'New' flags from anything before August.

(9.26.97) The reformat is done! :) If you are reading this somewhere other than the master page, please take a look and let me know what you think. Now I can get back to work on my thesis...

Purpose

The Linux OS has evolved from its origins in hackerdom to a full blown UNIX, capable of rivaling any commercial UNIX. It now provides an inexpensive base to build a great workstation. It has shed its hardware dependencies, having been ported to DEC Alphas, Sparcs, PowerPCs, with others on the way. This potential speed boost along with its networking support will make it great for workstation clusters. As a workstation it allows for all sorts of research and development, including artificial intelligence and artificial life.

The purpose of this Mini-Howto is to provide a source to find out about various software packages, code libraries, and anything else that will help someone get started working with (and find resources for) artificial intelligence and artificial life. All done with Linux specifically in mind.

Where to find this software

All this software should be available via the net (ftp || http). The links to where to find it will be provided in the description of each package. There will also be plenty of software not covered on these pages (which is usually platform independent) located on one of the resources listed under the Web Links section.

Updates and comments

If you find any mistakes, know of updates to one of the items below, or have problems compiling and of the applications, please mail me at: jae@ai.uga.edu and I'll see what I can do.

If you know of any AI/Alife applications, class libraries, etc. **Please** email me about them. Include your name, ftp and/or http sites where they can be found, plus a brief overview/commentary on the software (this info would make things a lot easier on me... but don't feel obligated ;).

I know that keeping this list up to date and expanding it will take quite a bit of work. So please be patient

(I do have other projects). I hope you will find this document helpful.

TOC

Programming languages

While any programming language can be used for artificial intelligence/life research, these are programming languages which are used extensively for, if not specifically made for, artificial intelligence programming.

ECoLisp [New]
Web site: www.di.unipi.it/~attardi/software.html

> ECoLisp (Embeddable Common Lisp) is an implementation of Common Lisp designed for being embeddable into C based applications. ECL uses standard C calling conventions for Lisp compiled functions, which allows C programs to easily call Lisp functions and viceversa. No foreign function interface is required: data can be exchanged between C and Lisp with no need for conversion. ECL is based on a Common Runtime Support (CRS) which provides basic facilities for memory managment, dynamic loading and dumping of binary images, support for multiple threads of execution. The CRS is built into a library that can be linked with the code of the application. ECL is modular: main modules are the program development tools (top level, debugger, trace, stepper), the compiler, and CLOS. A native implementation of CLOS is available in ECL: one can configure ECL with or without CLOS. A runtime version of ECL can be built with just the modules which are required by the application. The ECL compiler compiles from Lisp to C, and then invokes the GCC compiler to produce binaries.

Gödel
Web page: www.cs.bris.ac.uk/~bowers/goedel.html

> Gödel is a declarative, general-purpose programming language in the family of logic programming languages. It is a strongly typed language, the type system being based on many-sorted logic with parametric polymorphism. It has a module system. Gödel supports infinite precision integers, infinite precision rationals, and also floating-point numbers. It can solve constraints over finite domains of integers and also linear rational constraints. It supports processing of finite sets. It also has a flexible computation rule and a pruning operator which generalizes the commit of the concurrent logic programming languages. Considerable emphasis is placed on Gödel's meta-logical facilities which provide significant support for meta-programs that do analysis, transformation, compilation, verification, debugging, and so on.

LIFE
Web page: www.isg.sfu.ca/life

> LIFE (Logic, Inheritance, Functions, and Equations) is an experimental programming language proposing to integrate three orthogonal programming paradigms proven useful for symbolic computation. From the programmer's standpoint, it may be perceived as a language taking after logic programming, functional programming, and object-oriented programming. From a formal perspective, it may be seen as an instance (or rather, a composition of three instances) of a Constraint Logic Programming scheme due to Hoehfeld and Smolka refining that of Jaffar and Lassez.

CLisp (**Lisp**)

FTP site: sunsite.unc.edu/pub/Linux/devel/lang/lisp/

CLISP is a Common Lisp implementation by Bruno Haible and Michael Stoll. It mostly supports the Lisp described by 'Common LISP: The Language (2nd edition)' and the ANSI Common Lisp standard. CLISP includes an interpreter, a byte-compiler, a large subset of CLOS (Object-Oriented Lisp) , a foreign language interface and, for some machines, a screen editor.

The user interface language (English, German, French) is chosen at run time. Major packages that run in CLISP include CLX & Garnet. CLISP needs only 2 MB of memory.

CMU Common **Lisp**
Web page: www.mv.com/users/pw/lisp/index.html
FTP site: sunsite.unc.edu/pub/Linux/devel/lang/lisp/

CMU Common Lisp is a public domain "industrial strength" Common Lisp programming environment. Many of the X3j13 changes have been incorporated into CMU CL. Wherever possible, this has been done so as to transparently allow the use of either CLtL1 or proposed ANSI CL. Probably the new features most interesting to users are SETF functions, LOOP and the WITH-COMPILATION-UNIT macro.

GCL (**Lisp**)
FTP site: sunsite.unc.edu/pub/Linux/devel/lang/lisp/

GNU Common Lisp (GCL) has a compiler and interpreter for Common Lisp. It used to be known as Kyoto Common Lisp. It is very portable and extremely efficient on a wide class of applications. It compares favorably in performance with commercial Lisps on several large theorem-prover and symbolic algebra systems. It supports the CLtL1 specification but is moving towards the proposed ANSI definition. GCL compiles to C and then uses the native optimizing C compilers (e.g., GCC). A function with a fixed number of args and one value turns into a C function of the same number of args, returning one value, so GCL is maximally efficient on such calls. It has a conservative garbage collector which allows great freedom for the C compiler to put Lisp values in arbitrary registers.

It has a source level Lisp debugger for interpreted code, with display of source code in an Emacs window. Its profiling tools (based on the C profiling tools) count function calls and the time spent in each function.

Mercury
Web page: www.cs.mu.oz.au/research/mercury/

Mercury is a new, purely declarative logic programming language. Like Prolog and other existing logic programming languages, it is a very high-level language that allows programmers to concentrate on the problem rather than the low-level details such as memory management. Unlike Prolog, which is oriented towards exploratory programming, Mercury is designed for the construction of large, reliable, efficient software systems by teams of programmers. As a consequence, programming in Mercury has a different flavor than programming in Prolog.

DFKI **OZ**
Web page: www.ps.uni-sb.de/oz/
FTP site: ps-ftp.dfki.uni-sb.de/pub/oz2/

Oz is a high-level programming language designed for concurrent symbolic computation. It is based on a new computation model providing a uniform and simple foundation for several

TOC

programming paradigms, including higher-order functional, constraint logic, and concurrent object-oriented programming. Oz is designed as a successor to languages such as Lisp, Prolog, Smalltalk, which fail to support applications that require concurrency, reactivity, and real-time control.

DFKI Oz is an interactive implementation of Oz featuring a programming interface based on GNU Emacs, a concurrent browser, an object-oriented interface to Tcl/Tk, powerful interoperability features (sockets, C, C++), an incremental compiler, a garbage collector, and support for stand-alone applications. Performance is competitive with commercial Prolog and Lisp systems.

BinProlog

Web site(documentation): clement.info.umoncton.ca/BinProlog/UNCOMPRESSED/doc/html/art.html

FTP site(source): clement.info.umoncton.ca/BinProlog

FTP site(binary): clement.info.umoncton.ca/BinProlog/UNCOMPRESSED/bin

BinProlog is a fast and compact Prolog compiler, based on the transformation of Prolog to binary clauses. The compilation technique is similar to the Continuation Passing Style transformation used in some ML implementations. BinProlog 5.00 is also probably the first Prolog system featuring dynamic recompilation of asserted predicates (a technique similar to the one used in some object oriented languages like SELF 4.0), and a very efficient segment preserving copying heap garbage collector.

Although it (used to) incorporate some last minute research experiments, which might look adventurous at the first sight, BinProlog is a fairly robust and complete Prolog implementation featuring both C-emulated execution and generation of stand-alone applications by compilation to C.

SWI **Prolog**

Web page: swi.psy.uva.nl/projects/xpce/SWI-Prolog.html

FTP site: swi.psy.uva.nl/pub/SWI-Prolog/

SWI is a free version of prolog in the Edinburgh Prolog family (thus making it very similar to Quintus and many other versions). With: a large library of built in predicates, a module system, garbage collection, a two-way interface with the C language, plus many other features. It is meant as a educational language, so it's compiled code isn't the fastest. Although it similarity to Quintus allows for easy porting.

XPCE is freely available in binary form for the Linux version of SWI-prolog. XPCE is an object oriented X-windows GUI development package/environment.

Kali Scheme

Web site: www.neci.nj.nec.com/PLS/Kali.html

Kali Scheme is a distributed implementation of Scheme that permits efficient transmission of higher-order objects such as closures and continuations. The integration of distributed communication facilities within a higher-order programming language engenders a number of new abstractions and paradigms for distributed computing. Among these are user-specified load-balancing and migration policies for threads, incrementally-linked distributed computations, agents, and parameterized client-server applications. Kali Scheme supports concurrency and communication using first-class procedures and continuations. It integrates procedures and continuations into a message-based distributed framework that allows any Scheme object (including code vectors) to be sent and received in a message.

TOC

RScheme
Web site:www.rosette.com/~donovan/rs/rscheme.html
FTP site: ftp.rosette.com/pub/rscheme

RScheme is an object-oriented, extended version of the Scheme dialect of Lisp. RScheme is freely redistributable, and offers reasonable performance despite being extraordinarily portable. RScheme can be compiled to C, and the C can then compiled with a normal C compiler to generate machine code. By default, however, RScheme compiles to bytecodes which are interpreted by a (runtime) virtual machine. This ensures that compilation is fast and keeps code size down. In general, we recommend using the (default) bytecode code generation system, and only compiling your time-critical code to machine code. This allows a nice adjustment of space/time tradeoffs. (see web site for details)

Scheme48
Web site: www-swiss.ai.mit.edu/~jar/s48.html (download from ftp site)
FTP site: swiss-ftp.ai.mit.edu:/archive/s48/

Scheme 48 is a Scheme implementation based on a virtual machine architecture. Scheme 48 is designed to be straightforward, flexible, reliable, and fast. It should be easily portable to 32-bit byte-addressed machines that have POSIX and ANSI C support. In addition to the usual Scheme built-in procedures and a development environment, library software includes support for hygienic macros (as described in the Revised^4 Scheme report), multitasking, records, exception handling, hash tables, arrays, weak pointers, and FORMAT. Scheme 48 implements and exploits an experimental module system loosely derived from Standard ML and Scheme Xerox. The development environment supports interactive changes to modules and interfaces.

SCM (**Scheme**)
Web site: www-swiss.ai.mit.edu/~jaffer/SCM.html
FTP site: swiss-ftp.ai.mit.edu:/archive/scm/

SCM conforms to the Revised^4 Report on the Algorithmic Language Scheme and the IEEE P1178 specification. Scm is written in C. It uses the following utilities (all available at the ftp site).

- SLIB (Standard Scheme Library) is a portable Scheme library which is intended to provide compatibility and utility functions for all standard Scheme implementations, including SCM, Chez, Elk, Gambit, MacScheme, MITScheme, scheme->C, Scheme48, T3.1, and VSCM, and is available as the file slib2c0.tar.gz. Written by Aubrey Jaffer.
- JACAL is a symbolic math system written in Scheme, and is available as the file jacal1a7.tar.gz.
- Interfaces to standard libraries including REGEX string regular expression matching and the CURSES screen management package.
- Available add-on packages including an interactive debugger, database, X-window graphics, BGI graphics, Motif, and Open-Windows packages.
- A compiler (HOBBIT, available separately) and dynamic linking of compiled modules.

Shift [New]
Web site: www.path.berkeley.edu/shift/

Shift is a programming language for describing dynamic networks of hybrid automata. Such systems consist of components which can be created, interconnected and destroyed as the system

TOC

evolves. Components exhibit hybrid behavior, consisting of continuous-time phases separated discrete-event transitions. Components may evolve independently, or they may interact throug their inputs, outputs and exported events. The interaction network itself may evolve.

Traditional Artificial Intelligence

Traditional AI is based around the ideas of logic, rule systems, linguistics, and the concept of rationality. At its roots are programming languages such as Lisp and Prolog. Expert systems are the largest successful example of this paradigm. An expert system consists of a detailed knowledge base and a complex rule system to utilize it. Such systems have been used for such things as medical diagnosis support and credit checking systems.

AI class/code libraries

These are libraries of code or classes for use in programming within the artificial intelligence field. They are not meant as stand alone applications, but rather as tools for building your own applications.

AI Search
FTP site: ftp.icce.rug.nl/pub/peter/
Submitted by: Peter M. Bouthoorn

Basically, the library offers the programmer a set of search algorithms that may be used to solve all kind of different problems. The idea is that when developing problem solving software the programmer should be able to concentrate on the representation of the problem to be solved and should not need to bother with the implementation of the search algorithm that will be used to actually conduct the search. This idea has been realized by the implementation of a set of search classes that may be incorporated in other software through **C++**'s features of derivation and inheritance. The following search algorithms have been implemented:

- depth-first tree and graph search.
- breadth-first tree and graph search.
- uniform-cost tree and graph search.
- best-first search.
- bidirectional depth-first tree and graph search.
- bidirectional breadth-first tree and graph search.
- AND/OR depth tree search.
- AND/OR breadth tree search.

Peter plans to release a new version of the library soon, which will also be featured in a book about C++ and AI to appear this year.

Chess In Lisp (CIL)
FTP site: chess.onenet.net/pub/chess/uploads/projects/

The CIL (Chess In Lisp) foundation is a Common Lisp implementaion of all the core functions needed for development of chess applications. The main purpose of the CIL project is to get AI researchers interested in using Lisp to work in the chess domain.

DAI [New]
Web site: starship.skyport.net/crew/gandalf/DNET/AI

A library for the Python programming language that provides an object oriented interface to the CLIPS expert system tool. It includes an interface to COOL (CLIPS Object Oriented Language) that allows:

- Investigate COOL classes
- Create and manipulate with COOL instances
- Manipulate with COOL message-handler's
- Manipulate with Modules

Nyquist [New]
Web site: www.cs.cmu.edu/afs/cs.cmu.edu/project/music/web/music.html

The Computer Music Project at CMU is developing computer music and interactive performance technology to enhance human musical experience and creativity. This interdisciplinary effort draws on Music Theory, Cognitive Science, Artificial Intelligence and Machine Learning, Human Computer Interaction, Real-Time Systems, Computer Graphics and Animation, Multimedia, Programming Languages, and Signal Processing. A paradigmatic example of these interdisciplinary efforts is the creation of interactive performances that couple human musical improvisation with intelligent computer agents in real-time.

Screamer
Web site: www.cis.upenn.edu/~screamer-tools/home.html

Screamer is an extension of Common Lisp that adds support for nondeterministic programming. Screamer consists of two levels. The basic nondeterministic level adds support for backtracking and undoable side effects. On top of this nondeterministic substrate, Screamer provides a comprehensive constraint programming language in which one can formulate and solve mixed systems of numeric and symbolic constraints. Together, these two levels augment Common Lisp with practically all of the functionality of both Prolog and constraint logic programming languages such as CHiP and CLP(R). Furthermore, Screamer is fully integrated with Common Lisp. Screamer programs can coexist and interoperate with other extensions to Common Lisp such as CLOS, CLIM and Iterate.

AI software kits, applications, etc.

These are various applications, software kits, etc. meant for research in the field of artificial intelligence. Their ease of use will vary, as they were designed to meet some particular research interest more than as an easy to use commercial package.

ASA - Adaptive Simulated Annealing
Web site: www.ingber.com/#ASA-CODE
FTP site: ftp.ingber.com/

ASA (Adaptive Simulated Annealing) is a powerful global optimization C-code algorithm especially useful for nonlinear and/or stochastic systems.

ASA is developed to statistically find the best global fit of a nonlinear non-convex cost-function over a D-dimensional space. This algorithm permits an annealing schedule for 'temperature' T decreasing exponentially in annealing-time k, T = T_0 exp(-c k^1/D). The introduction of re-annealing also permits adaptation to changing sensitivities in the multi-dimensional

TOC

parameter-space. This annealing schedule is faster than fast Cauchy annealing, where T = T_0/k and much faster than Boltzmann annealing, where T = T_0/ln k.

Babylon

FTP site: ftp.gmd.de/gmd/ai-research/Software/Babylon/

BABYLON is a modular, configurable, hybrid environment for developing expert systems. Its features include objects, rules with forward and backward chaining, logic (Prolog) and constraints. BABYLON is implemented and embedded in Common Lisp.

CLEARS

Web site: www.coli.uni-sb.de/~clears/

The CLEARS system is an interactive graphical environment for computational semantics. The tool allows exploration and comparison of different semantic formalisms, and their interaction with syntax. This enables the user to get an idea of the range of possibilities of semantic construction, and also where there is real convergence between theories.

CLIPS

Web site: www.jsc.nasa.gov/~clips/CLIPS.html

FTP site: cs.cmu.edu/afs/cs.cmu.edu/project/ai-repository/ai/areas/expert/systems/clips

CLIPS is a productive development and delivery expert system tool which provides a complete environment for the construction of rule and/or object based expert systems.

CLIPS provides a cohesive tool for handling a wide variety of knowledge with support for three different programming paradigms: rule-based, object-oriented and procedural. Rule-based programming allows knowledge to be represented as heuristics, or "rules of thumb," which specify a set of actions to be performed for a given situation. Object-oriented programming allows complex systems to be modeled as modular components (which can be easily reused to model other systems or to create new components). The procedural programming capabilities provided by CLIPS are similar to capabilities found in languages such as C, Pascal, Ada, and LISP.

EMA-XPS - A Hybrid Graphic Expert System Shell

Web site: wmwap1.math.uni-wuppertal.de:80/EMA-XPS/

EMA-XPS is a hybrid graphic expert system shell based on the ASCII-oriented shell Babylon 2.3 of the German National Research Center for Computer Sciences (GMD). In addition to Babylon's AI-power (object oriented data representation, forward and backward chained rules - collectible into sets, horn clauses, and constraint networks) a graphic interface based on the X11 Window System and the OSF/Motif Widget Library has been provided.

FOOL & FOX

FTP site: ntia.its.bldrdoc.gov/pub/fuzzy/prog/

FOOL stands for the Fuzzy Organizer OLdenburg. It is a result from a project at the University of Oldenburg. FOOL is a graphical user interface to develop fuzzy rulebases. FOOL will help you to invent and maintain a database that specifies the behavior of a fuzzy-controller or something like that.

FOX is a small but powerful fuzzy engine which reads this database, reads some input values and calculates the new control value.

FUF and SURGE

Web site: www.dfki.de/lt/registry/generation/fuf.html
FTP site: ftp.cs.columbia.edu/pub/fuf/

FUF is an extended implementation of the formalism of functional unification grammars (FUGs) introduced by Martin Kay specialized to the task of natural language generation. It adds the following features to the base formalism:

- Types and inheritance.
- Extended control facilities (goal freezing, intelligent backtracking).
- Modular syntax.

These extensions allow the development of large grammars which can be processed efficiently and can be maintained and understood more easily. SURGE is a large syntactic realization grammar of English written in FUF. SURGE is developed to serve as a black box syntactic generation component in a larger generation system that encapsulates a rich knowledge of English syntax. SURGE can also be used as a platform for exploration of grammar writing with a generation perspective.

The Grammar Workbench
Web site: www.cs.kun.nl/agfl/GWB.html

The Grammar Workbench, or GWB for short, is an environment for the comfortable development of Affix Grammars in the AGFL-formalism. Its purposes are:

- to allow the user to input, inspect and modify a grammar;
- to perform consistency checks on the grammar;
- to compute grammar properties;
- to generate example sentences;
- to assist in performing grammar transformations.

GSM Suite [New]
Web site: www.slip.net/~andrewm/gsm/

The GSM Suite is a set of programs for using Finite State Machines in a graphical fashion. The suite consists of programs that edit, compile, and print state machines. Included in the suite is an editor program, gsmedit, a compiler, gsm2cc, that produces a C++ implementation of a state machine, a PostScript generator, gsm2ps, and two other minor programs. GSM is licensed under the GNU Public License and so is free for your use under the terms of that license.

Illuminator
Web site: documents.cfar.umd.edu/resources/source/illuminator.html

Illuminator is a toolset for developing OCR and Image Understanding applications. Illuminator has two major parts: a library for representing, storing and retrieving OCR information, heretofore called dafslib, and an X-Windows "DAFS" file viewer, called illum. Illuminator and DAFS lib were designed to supplant existing OCR formats and become a standard in the industry. They particularly are extensible to handle more than just English. The features of this release:

- 5 magnification levels for images
- flagged characters and words
- unicode support -- American, British, French, German, Greek, Italian, MICR, Norwegian,

TOC

Russian, Spanish, Swedish, keyboards
- reads DAFS, TIFF's, PDA's (image only)
- save to DAFS, ASCII/UTF or Unicode
- Entity Viewer - shows properties, character choices, bounding boxes image fragment for a selected entity, change type, change content, hierarchy mode

Jess, the Java Expert System Shell
Web site: herzberg.ca.sandia.gov/jess/

Jess is a clone of the popular CLIPS expert system shell written entirely in Java. With Jess, you can conveniently give your applets the ability to 'reason'. Jess is compatible with all versions of Java starting with version 1.0.2. Jess implements the following constructs from CLIPS: defrules, deffunctions, defglobals, deffacts, and deftemplates.

learn
FTP site: sunsite.unc.edu/pub/Linux/apps/cai/

Learn is a vocable learning program with memory model.

Otter: An Automated Deduction System
Web site: www.mcs.anl.gov/home/mccune/ar/otter

Our current automated deduction system Otter is designed to prove theorems stated in first-order logic with equality. Otter's inference rules are based on resolution and paramodulation, and it includes facilities for term rewriting, term orderings, Knuth-Bendix completion, weighting, and strategies for directing and restricting searches for proofs. Otter can also be used as a symbolic calculator and has an embedded equational programming system.

RIPPER
Web site: www.research.att.com/~wcohen/ripperd.html

Ripper is a system for fast effective rule induction. Given a set of data, Ripper will learn a set of rules that will predict the patterns in the data. Ripper is written in ASCI C and comes with documentation and some sample problems.

SNePS
Web site: www.cs.buffalo.edu/pub/sneps/WWW/
FTP site: ftp.cs.buffalo.edu/pub/sneps/

The long-term goal of The SNePS Research Group is the design and construction of a natural-language-using computerized cognitive agent, and carrying out the research in artificial intelligence, computational linguistics, and cognitive science necessary for that endeavor. The three-part focus of the group is on knowledge representation, reasoning, and natural-language understanding and generation. The group is widely known for its development of the SNePS knowledge representation/reasoning system, and Cassie, its computerized cognitive agent.

Soar
Web site: www.isi.edu/soar/soar.html
FTP site: cs.cmu.edu/afs/cs/project/soar/public/Soar6/

Soar has been developed to be a general cognitive architecture. We intend ultimately to enable the Soar architecture to:

- work on the full range of tasks expected of an intelligent agent, from highly routine to

extremely difficult, open-ended problems
- represent and use appropriate forms of knowledge, such as procedural, declarative, episodic, and possibly iconic
- employ the full range of problem solving methods
- interact with the outside world and
- learn about all aspects of the tasks and its performance on them.

In other words, our intention is for Soar to support all the capabilities required of a general intelligent agent.

TCM (Toolkit for Conceptual Modeling)
Web site: www.cs.vu.nl/~tcm/
FTP site: ftp.cs.vu.nl/pub/tcm/

TCM (Toolkit for Conceptual Modeling) is our suite of graphical editors. TCM contains graphical editors for Entity-Relationship diagrams, Class-Relationship diagrams, Data and Event Flow diagrams, State Transition diagrams, Jackson Process Structure diagrams and System Network diagrams, Function Refinement trees and various table editors, such as a Function-Entity table editor and a Function Decomposition table editor. TCM is easy to use and performs numerous consistency checks, some of them immediately, some of them upon request.

WEKA [New]
Web site: lucy.cs.waikato.ac.nz/~ml/

WEKA (Waikato Environment for Knowledge Analysis) is an state-of-the-art facility for applying machine learning techniques to practical problems. It is a comprehensive software "workbench" that allows people to analyse real-world data. It integrates different machine learning tools within a common framework and a uniform user interface. It is designed to support a "simplicity-first" methodology, which allows users to experiment interactively with simple machine learning tools before looking for more complex solutions.

Connectionism

Connectionism is a technical term for a group of related techniques. These techniques include areas such as Artificial Neural Networks, Semantic Networks and a few other similar ideas. My present focus is on neural networks (though I am looking for resources on the other techniques). Neural networks are programs designed to simulate the workings of the brain. They consist of a network of small mathematical-based nodes, which work together to form patterns of information. They have tremendous potential and currently seem to be having a great deal of success with image processing and robot control.

Connectionist class/code libraries

These are libraries of code or classes for use in programming within the Connectionist field. They are not meant as stand alone applications, but rather as tools for building your own applications.

ANSI-C Neural Networks
Web site: www.geocities.com/CapeCanaveral/1624/

TOC

This site contains ANSC-C source code for 8 types of neural nets, including:

- Adaline Network
- Backpropagation
- Hopfield Model
- (BAM) Bidirectional Associative Memory
- Boltzmann Machine
- Counterpropagation
- (SOM) Self-Organizing Map
- (ART1) Adaptive Resonance Theory

They were designed to help turn the theory of a particular network model into the design for a simulator implementation , and to help with embeding an actual application into a particular network model.

BELIEF [New]
Web site: www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/reasonng/probabl/belief/0.html

BELIEF is a Common Lisp implementation of the Dempster and Kong fusion and propagation algorithm for Graphical Belief Function Models and the Lauritzen and Spiegelhalter algorithm for Graphical Probabilistic Models. It includes code for manipulating graphical belief models such as Bayes Nets and Relevance Diagrams (a subset of Influence Diagrams) using both belief functions and probabilities as basic representations of uncertainty. It uses the Shenoy and Shafer version of the algorithm, so one of its unique features is that it supports both probability distributions and belief functions. It also has limited support for second order models (probability distributions on parameters).

IDEAL [New]
Web site: www.rpal.rockwell.com/ideal.html

IDEAL is a test bed for work in influence diagrams and Bayesian networks. It contains various inference algorithms for belief networks and evaluation algorithms for influence diagrams. It contains facilities for creating and editing influence diagrams and belief networks.

IDEAL is written in pure Common Lisp and so it will run in Common Lisp on any platform. The emphasis in writing IDEAL has been on code clarity and providing high level programming abstractions. It thus is very suitable for experimental implementations which need or extend belief network technology.

At the highest level, IDEAL can be used as a subroutine library which provides belief network inference and influence diagram evaluation as a package. The code is documented in a detailed manual and so it is also possible to work at a lower level on extensions of belief network methods.

IDEAL comes with an optional graphic interface written in CLIM. If your Common Lisp also has CLIM, you can run the graphic interface.

Matrix Class
FTP site: pink.cs.ucla.edu/pub/

A simple, fast, efficient C++ Matrix class designed for scientists and engineers. The Matrix class is

TOC

well suited for applications with complex math algorithms. As an demonstration of the Matrix class, it was used to implement the backward error propagation algorithm for a multi-layer feed-forward artificial neural network.

Pulcinella
Web site: iridia.ulb.ac.be/pulcinella/Welcome.html

Pulcinella is written in CommonLisp, and appears as a library of Lisp functions for creating, modifying and evaluating valuation systems. Alternatively, the user can choose to interact with Pulcinella via a graphical interface (only available in Allegro CL). Pulcinella provides primitives to build and evaluate uncertainty models according to several uncertainty calculi, including probability theory, possibility theory, and Dempster-Shafer's theory of belief functions; and the possibility theory by Zadeh, Dubois and Prade's. A User's Manual is available on request.

S-ElimBel [New]
Web site: www.spaces.uci.edu/thiery/elimbel/

S-ElimBel is an algorithm that computes the belief in a Bayesian network, implemented in MIT-Scheme. This algorithm has the particularity of being rather easy to understand. Moreover, one can apply it to any kind of Bayesian network - it being singly connected or muliply connected. It is, however, less powerful than the standard algorithm of belief propagation. Indeed, the computation has to be reconducted entirely for each new evidence added to the network. Also, one needs to run the algorithm as many times as one has nodes for which the belief is wanted.

Software for Flexible Bayesian Modeling [New]
Web site: www.cs.utoronto.ca/~radford/fbm.software.html

This software implements flexible Bayesian models for regression and classification applications that are based on multilayer perceptron neural networks or on Gaussian processes. The implementation uses Markov chain Monte Carlo methods. Software modules that support Markov chain sampling are included in the distribution, and may be useful in other applications.

Spiderweb2 [New]
Web site:

A C++ artificial neual net library. Spiderweb2 is a complete rewrite of the original Spiderweb library, it has grown into a much more flexible and object-oriented system. The biggest change is that each neuron object is responsible for its own activations and updates, with the network providing only the scheduling aspect. This is a very powerful change, and it allows easy modification and experimentation with various network architectures and neuron types.

There is still a lack of documentation and support for file I/O, but that will eventually get done.

Symbolic Probabilistic Inference (SPI)
FTP site: ftp.engr.orst.edu/pub/dambrosi/spi/
Paper (ijar-94.ps): ftp.engr.orst.edu/pub/dambrosi/

Contains Common Lisp function libraries to implement SPI type baysean nets. Documentation is very limited.
Features:

Probabilities, Local Expression Language Utilities, Explanation, Dynamic Models, and a TCL/TK based GUI.

TresBel

FTP site: iridia.ulb.ac.be/pub/hongxu/software/

Libraries containing (Allegro) Common Lisp code for Belief Functions (aka. Dempster-Shafer evidential reasoning) as a representation of uncertainty. Very little documentation. Has a limited GUI.

Various (C++) Neural Networks
Web site: www.mcs.com/~drt/svbp.html
Submitted by: Don Tveter

Example neural net codes from the book, The Basis of Artificial Intelligence. These are simple example codes of these various neural nets. They work well as a good starting point for simple experimentation and for learning what the code is like behind the simulators. The types of networks available on this site are: (implemented in C++)

- The Backprop Package
- The Nearest Neighbor Algorithms
- The Interactive Activation Algorithm
- The Hopfield and Boltzman machine Algorithms
- The Linear Pattern Classifier
- ART I
- Bi-Directional Associative Memory
- The Feedforward Counter-Propagation Network

Connectionist software kits/applications

These are various applications, software kits, etc. meant for research in the field of Connectionism. Their ease of use will vary, as they were designed to meet some particular research interest more than as an easy to use commercial package.

Aspirin/MIGRAINES (am6.tar.Z on ftp site)
FTP site: sunsite.unc.edu/pub/academic/computer-science/neural-networks/programs/Aspirin/

The software that we are releasing now is for creating, and evaluating, feed-forward networks such as those used with the backpropagation learning algorithm. The software is aimed both at the expert programmer/neural network researcher who may wish to tailor significant portions of the system to his/her precise needs, as well as at casual users who will wish to use the system with an absolute minimum of effort.

JavaBayes [New]
Web site: www.cs.cmu.edu/People/javabayes/index.html/

The JavaBayes system is a set of tools, containing a graphical editor, a core inference engine and a parser. JavaBayes can produce:

- the marginal distribution for any variable in a network.
- the expectations for univariate functions (for example, expected value for variables).
- configurations with maximum a posteriori probability.
- configurations with maximum a posteriori expectation for univariate functions.

TOC

Neureka ANS (nn/xnn)
Web site: www.bgif.no/neureka/
FTP site: ftp.ii.uib.no/pub/neureka/linux/

nn is a high-level neural network specification language. The current version is best suited for feed-forward nets, but recurrent models can and have been implemented, e.g. Hopfield nets, Jordan/Elman nets, etc. In nn, it is easy to change network dynamics. The nn compiler can generate C code or executable programs (so there must be a C compiler available), with a powerful command line interface (but everything may also be controlled via the graphical interface, xnn). It is possible for the user to write C routines that can be called from inside the nn specification, and to use the nn specification as a function that is called from a C program. Please note that no programming is necessary in order to use the network models that come with the system (`netpack').

xnn is a graphical front end to networks generated by the nn compiler, and to the compiler itself. The xnn graphical interface is intuitive and easy to use for beginners, yet powerful, with many possibilities for visualizing network data.

NOTE: You have to run the install program that comes with this to get the license key installed. It gets put (by default) in /usr/lib. If you (like myself) want to install the package somewhere other than in the /usr directory structure (the install program gives you this option) you will have to set up some environmental variables (NNLIBDIR & NNINCLUDEDIR are required). You can read about these (and a few other optional variables) in appendix A of the documentation (pg 113).

PDP++
Web site: www.cnbc.cmu.edu/PDP++/
FTP site (US): cnbc.cmu.edu/pub/pdp++/
FTP site (Europe): unix.hensa.ac.uk/mirrors/pdp++/

As the field of Connectionist modeling has grown, so has the need for a comprehensive simulation environment for the development and testing of Connectionist models. Our goal in developing PDP++ has been to integrate several powerful software development and user interface tools into a general purpose simulation environment that is both user friendly and user extensible. The simulator is built in the C++ programming language, and incorporates a state of the art script interpreter with the full expressive power of C++. The graphical user interface is built with the Interviews toolkit, and allows full access to the data structures and processing modules out of which the simulator is built. We have constructed several useful graphical modules for easy interaction with the structure and the contents of neural networks, and we've made it possible to change and adapt many things. At the programming level, we have set things up in such a way as to make user extensions as painless as possible. The programmer creates new C++ objects, which might be new kinds of units or new kinds of processes; once compiled and linked into the simulator, these new objects can then be accessed and used like any other.

Simple Neural Net (in Python)
Web site: starship.skyport.net/crew/amk/unmaintained/

Simple neural network code, which implements a class for 3-level networks (input, hidden, and output layers). The only learning rule implemented is simple backpropagation. No documentation (or even comments) at all, because this is simply code that I use to experiment with. Includes modules containing sample datasets from Carl G. Looney's NN book. Requires the Numeric

TOC

extensions.

SCNN

Web site: apx00.physik.uni-frankfurt.de/e_ag_rt/cnn/SCNN/homepage.html

SCNN is an universal simulating system for Cellular Neural Networks (CNN). CNN are analog processing neural networks with regular and local interconnections, governed by a set of nonlinear ordinary differential equations. Due to their local connectivity, CNN are realized as VLSI chips, which operates at very high speed.

Semantic Networks in Python

Web site: www-acs.ucsd.edu/~jstrout/python/ai/

The semnet.py module defines several simple classes for building and using semantic networks. A semantic network is a way of representing knowledge, and it enables the program to do simple reasoning with very little effort on the part of the programmer.

The following classes are defined:

- **Entity**: This class represents a noun; it is something which can be related to other things, and about which you can store facts.
- **Relation**: A Relation is a type of relationship which may exist between two entities. One special relation, "IS_A", is predefined because it has special meaning (a sort of logical inheritance).
- **Fact**: A Fact is an assertion that a relationship exists between two entities.

With these three object types, you can very quickly define knowledge about a set of objects, and query them for logical conclusions.

SNNS

Web site: www.informatik.uni-stuttgart.de/ipvr/bv/projekte/snns/snns.html

FTP site: ftp.informatik.uni-stuttgart.de/pub/SNNS/

Stuttgart Neural Net Simulator (version 4.1). An awesome neural net simulator. Better than any commercial simulator I've seen. The simulator kernel is written in C (it's fast!). It supports over 20 different network architectures, has 2D and 3D X-based graphical representations, the 2D GUI has an integrated network editor, and can generate a separate NN program in C.
SNNS is very powerful, though a bit difficult to learn at first. To help with this it comes with example networks and tutorials for many of the architectures.
ENZO, a supplementary system allows you to evolve your networks with genetic algorithms.

The Readme.linux file that comes with this package must be old. It's instructions for building the package are wrong. I've edited it to reflect what I had to do to get the package to compile. Please download SNNS.Readme.linux and use it instead of the Readme.linux file that comes with the distribution.

SPRLIB/ANNLIB

Web site: www.ph.tn.tudelft.nl/~sprlib/

SPRLIB (Statistical Pattern Recognition Library) was developed to support the easy construction and simulation of pattern classifiers. It consist of a library of functions (written in C) that can be called from your own program. Most of the well-known classifiers are present (k-nn, Fisher, Parzen, ...), as well as error estimation and dataset generation routines.

TOC

ANNLIB (Artificial Neural Networks Library) is a neural network simulation library based on data architecture laid down by SPRLIB. The library contains numerous functions for creating, training and testing feed-forward networks. Training algorithms include back-propagation, pseudo-Newton, Levenberg-Marquardt, conjugate gradient descent, BFGS.... Furthermore, it is possible - due to the datastructures' general applicability - to build Kohonen maps and other more exotic network architectures using the same data types.

Evolutionary Computing

Evolutionary computing is actually a broad term for a vast array of programming techniques, including genetic algorithms, complex adaptive systems, evolutionary programming, etc. The main thrust of all these techniques is the idea of evolution. The idea that a program can be written that will *evolve* toward a certain goal. This goal can be anything from solving some engineering problem to winning a game.

EC class/code libraries

These are libraries of code or classes for use in programming within the evolutionary computation field. They are not meant as stand alone applications, but rather as tools for building your own applications.

FORTRAN GA
Web site: www.staff.uiuc.edu/~carroll/ga.html

This program is a FORTRAN version of a genetic algorithm driver. This code initializes a random sample of individuals with different parameters to be optimized using the genetic algorithm approach, i.e. evolution via survival of the fittest. The selection scheme used is tournament selection with a shuffling technique for choosing random pairs for mating. The routine includes binary coding for the individuals, jump mutation, creep mutation, and the option for single-point or uniform crossover. Niching (sharing) and an option for the number of children per pair of parents has been added. More recently, an option for the use of a micro-GA has been added.

GAGS
Web site: kal-el.ugr.es/gags.html
FTP site: kal-el.ugr.es/GAGS/

Genetic Algorithm application generator and class library written mainly in C++.
As a class library, and among other thing, GAGS includes:

- A *chromosome hierarchy* with variable length chromosomes. *Genetic operators*: 2-point crossover, uniform crossover, bit-flip mutation, transposition (gene interchange between 2 parts of the chromosome), and variable-length operators: duplication, elimination, and random addition.
- *Population level operators* include steady state, roulette wheel and tournament selection.
- *Gnuplot wrapper*: turns gnuplot into a `iostreams`-like class.
- Easy sample file loading and configuration file parsing.

As an application generator (written in `PERL`), you only need to supply it with an ANSI-C or C++ fitness function, and it creates a C++ program that uses the above library to 90% capacity, compiles it, and runs it, saving results and presenting fitness thru `gnuplot`.

TOC

GAlib: Matthew's Genetic Algorithms Library
Web Site: lancet.mit.edu/ga/
FTP site: lancet.mit.edu/pub/ga/
Register GAlib at: http://lancet.mit.edu/ga/Register.html

GAlib contains a set of C++ genetic algorithm objects. The library includes tools for using genetic algorithms to do optimization in any C++ program using any representation and genetic operators. The documentation includes an extensive overview of how to implement a genetic algorithm as well as examples illustrating customizations to the GAlib classes.

GALOPPS
Web site: GARAGe.cps.msu.edu/software/software-index.html
FTP site: garage.cps.msu.edu/pub/GA/galopps/

GALOPPS is a flexible, generic GA, in 'C'. It was based upon Goldberg's Simple Genetic Algorithm (SGA) architecture, in order to make it easier for users to learn to use and extend.

GALOPPS extends the SGA capabilities several fold:

- (optional) A new Graphical User Interface, based on TCL/TK, for Unix users, allowing easy running of GALOPPS 3.2 (single or multiple subpopulations) on one or more processors. GUI writes/reads "standard" GALOPPS input and master files, and displays graphical output (during or after run) of user-selected variables.
- 5 selection methods: roulette wheel, stochastic remainder sampling, tournament selection, stochastic universal sampling, linear-ranking-then-SUS.
- Random or superuniform initialization of "ordinary" (non-permutation) binary or non-binary chromosomes; random initialization of permutation-based chromosomes; or user-supplied initialization of arbitrary types of chromosomes.
- Binary or non-binary alphabetic fields on value-based chromosomes, including different user-definable field sizes.
- 3 crossovers for value-based representations: 1-pt, 2-pt, and uniform, all of which operate at field boundaries if a non-binary alphabet is used.
- 4 crossovers for order-based reps: PMX, order-based, uniform order-based, and cycle.
- 4 mutations: fast bitwise, multiple-field, swap and random sublist scramble.
- Fitness scaling: linear scaling, Boltzmann scaling, sigma truncation, window scaling, ranking.
- **Plus** a whole lot more....

GAS [New]
Web site: starship.skyport.net/crew/gandalf
FTP site: ftp.coe.uga.edu/users/jae/ai

GAS means "Genetic Algorithms Stuff". GAS is freeware.
Purpose of GAS is to explore and exploit artificial evolutions. Primary implementation language of GAS is Python. The GAS software package is meant to be a Python framework for applying genetic algorithms. It contains an example application where it is tried to breed Python program strings. This special problem falls into the category of Genetic Programming (GP), and/or Automatic Programming. Nevertheless, GAS tries to be useful for other applications of Genetic

TOC

Algorithms as well.

GECO
FTP site: ftp://ftp.aic.nrl.navy.mil/pub/galist/src/

GECO (Genetic Evolution through Combination of Objects), an extendible object-oriented tool-box for constructing genetic algorithms (in Lisp). It provides a set of extensible classes and methods designed for generality. Some simple examples are also provided to illustrate the intended use.

GPdata
FTP site: ftp.cs.bham.ac.uk/pub/authors/W.B.Langdon/gp-code/
Documentation (GPdata-icga-95.ps): cs.ucl.ac.uk/genetic/papers/

GPdata-3.0.tar.gz (C++) contains a version of Andy Singleton's GP-Quick version 2.1 which has been extensively altered to support:

- Indexed memory operation (cf. teller)
- multi tree programs
- Adfs
- parameter changes without recompilation
- populations partitioned into demes
- (A version of) pareto fitness

This ftp site also contains a small C++ program (ntrees.cc) to calculate the number of different there are of a given length and given function and terminal set.

gpjpp Genetic Programming in Java
Web site: www.turbopower.com/~kimk/gpjpp.asp

gpjpp is a Java package I wrote for doing research in genetic programming. It is a port of the gpc++ kernel written by Adam Fraser and Thomas Weinbrenner. Included in the package are four of Koza's standard examples: the artificial ant, the hopping lawnmower, symbolic regression, and the boolean multiplexer. Here is a partial list of its features:

- graphic output of expression trees
- efficient diversity checking
- Koza's greedy over-selection option for large populations
- extensible GPRun class that encapsulates most details of a genetic programming test
- more robust and efficient streaming code, with automatic checkpoint and restart built into the GPRun class
- an explicit complexity limit that can be set on each GP
- additional configuration variables to allow more testing without recompilation
- support for automatically defined functions (ADFs)
- tournament and fitness proportionate selection
- demetic grouping
- optional steady state population
- subtree crossover

TOC

- swap and shrink mutation

GP Kernel
Web site: www.emk.e-technik.th-darmstadt.de/~thomasw/gp.html

The GP kernel is a C++ class library that can be used to apply genetic programming techniques to all kinds of problems. The library defines a class hierarchy. An integral component is the ability to produce automatically defined functions as found in Koza's "Genetic Programming II".Technical documentation (postscript format) is included. There is also a short introduction into genetic programming.

Functionality includes; Automatically defined functions (ADFs), tournament and fitness proportionate selection, demetic grouping, optional steady state genetic programming kernel, subtree crossover, swap and shrink mutation, a way of changing every parameter of the system without recompilation, capacity for multiple populations, loading and saving of populations and genetic programs, standard random number generator, internal parameter checks.

lil-gp
Web site: isl.msu.edu/GA/software/lil-gp/index.html
FTP site: isl.cps.msu.edu/pub/GA/lilgp/

patched lil-gp * [New]
Web site: www.cs.umd.edu/users/seanl/patched-gp

lil-gp is a generic 'C' genetic programming tool. It was written with a number of goals in mind: speed, ease of use and support for a number of options including:

- Generic 'C' program that runs on UNIX workstations
- Support for multiple population experiments, using arbitrary and user settable topologies for exchange, for a single processor (i.e., you can do multiple population gp experiments on your PC).
- lil-gp manipulates trees of function pointers which are allocated in single, large memory blocks for speed and to avoid swapping.

* The patched lil-gp kernel is strongly-typed, with modifications on multithreading, fixes to some *serious* bugs in lilgp, coevolution, and other tweaks and features.

PGAPack Parallel Genetic Algorithm Library
Web site: www.mcs.anl.gov/home/levine/PGAPACK/index.html
FTP site: ftp.mcs.anl.gov/pub/pgapack/

PGAPack is a general-purpose, data-structure-neutral, parallel genetic algorithm library. It is intended to provide most capabilities desired in a genetic algorithm library, in an integrated, seamless, and portable manner. Key features are in PGAPack V1.0 include:

- Callable from Fortran or C.
- Runs on uniprocessors, parallel computers, and workstation networks.
- Binary-, integer-, real-, and character-valued native data types.
- Full extensibility to support custom operators and new data types.
- Easy-to-use interface for novice and application users.
- Multiple levels of access for expert users.

TOC

- Parameterized population replacement.
- Multiple crossover, mutation, and selection operators.
- Easy integration of hill-climbing heuristics.
- Extensive debugging facilities.
- Large set of example problems.
- Detailed users guide.

Sugal
Web site: www.trajan-software.demon.co.uk/sugal.htm

Sugal [soo-gall] is the SUnderland Genetic ALgorithm system. The aim of Sugal is to support research and implementation in Genetic Algorithms on a common software platform. As such, Sugal supports a large number of variants of Genetic Algorithms, and has extensive features to support customization and extension.

EC software kits/applications

These are various applications, software kits, etc. meant for research in the field of evolutionary computing. Their ease of use will vary, as they were designed to meet some particular research interest more than as an easy to use commercial package.

ADATE(Automatic Design of Algorithms Through Evolution)
Web site: www-ia.hiof.no/~rolando/adate_intro.html

ADATE is a system for automatic programming i.e., inductive inference of algorithms, which may be the best way to develop artificial and general intelligence.

The ADATE system can automatically generate non-trivial and novel algorithms. Algorithms are generated through large scale combinatorial search that employs sophisticated program transformations and heuristics. The ADATE system is particularly good at synthesizing symbolic, functional programs and has several unique qualities.

esep & xesep [New]
Web site(esep): www.iit.edu/~linjinl/esep.html Web site(xesep): www.iit.edu/~linjinl/xesep.html

This is a new scheduler, called Evolution Scheduler, based on Genetic Algorithms and Evolutionary Programming. It lives with original Linux priority scheduler.This means you don't have to reboot to change the scheduling policy. You may simply use the manager program esep to switch between them at any time, and esep itself is an all-in-one for scheduling status, commands, and administration. We didn't intend to remove the original priority scheduler; instead, at least, esep provides you with another choice to use a more intelligent scheduler, which carries out natural competition in an easy and effective way.

Xesep is a graphical user interface to the esep (Evolution Scheduling and Evolving Processes). It's intended to show users how to start, play, and feel the Evolution Scheduling and Evolving Processes, including sub-programs to display system status, evolving process status, queue status, and evolution scheduling status periodically in as small as one mini-second.

FSM-Evolver [New]
Web site: pages.prodigy.net/czarneckid

TOC

A Java (jdk-v1.0.2+) code library that is used to evolve finite state machines. The problem included in the package is the Artificial Ant problem. You should be able to compile the .java and then run: java ArtificialAnt.

GPsys [New]
Web site: www.cs.ucl.ac.uk/staff/A.Qureshi/gpsys.html

GPsys (pronounced gipsys) is a Java (requires Java 1.1 or later) based Genetic Programming system developed by Adil Qureshi. The software includes documentation, source and executables.

Feature Summary:

- Steady State engine
- ADF support
- Strongly Typed
  - supports generic functions and terminals
  - has many built-in primitives
  - includes indexed memory
- Save/Load feature
  - can save/load current generation to/from a file
  - data stored in GZIP compression format to minimise disk requirements
  - uses serialisable objects for efficiency
- Fully Documented
- Example Problems
  - Lawnmower (including GUI viewer)
  - Symbolic Regression
- Totally Parameterised
- Fully Object Oriented and Extensible
- High Performance
- Memory Efficient

Alife

Alife takes yet another approach to exploring the mysteries of intelligence. It has many aspects similar to EC and Connectionism, but takes these ideas and gives them a meta-level twist. Alife emphasizes the development of intelligence through *emergent* behavior of *complex adaptive systems*. Alife stresses the social or group based aspects of intelligence. It seeks to understand life and survival. By studying the behaviors of groups of 'beings' Alife seeks to discover the way intelligence or higher order activity emerges from seemingly simple individuals. Cellular Automata and Conway's Game of Life are probably the most commonly known applications of this field.

Alife class/code libraries

These are libraries of code or classes for use in programming within the artificial life field. They are not meant as stand alone applications, but rather as tools for building your own applications.

John von Neumann Universal Constructor
Web site: alife.santafe.edu/alife/software/jvn.html
FTP site: alife.santafe.edu/pub/SOFTWARE/jvn/

> The universal constructor of John von Neumann is an extension of the logical concept of universal computing machine. In the cellular environment proposed by von Neumann both computing and constructive universality can be achieved. Von Neumann proved that in his cellular lattice both a Turing machine and a machine capable of producing any other cell assembly, when fed with a suitable program, can be embedded. He called the latter machine a "universal constructor" and showed that, when provided with a program containing its own description, this is capable of self-reproducing.

Swarm
Web site: www.santafe.edu/projects/swarm
FTP site: ftp.santafe.edu/pub/swarm

> The swarm Alife simulation kit. Swarm is a simulation environment which facilitates development and experimentation with simulations involving a large number of agents behaving and interacting within a dynamic environment. It consists of a collection of classes and libraries written in Objective-C and allows great flexibility in creating simulations and analyzing their results. It comes with three demos and good documentation.
>
> Swarm 1.0 is out. It requires *libtclobjc* and *BLT 2.1* (both available at the swarm site).

Alife software kits, applications, etc.

These are various applications, software kits, etc. meant for research in the field of artificial life. Their ease of use will vary, as they were designed to meet some particular research interest more than as an easy to use commercial package.

BugsX
FTP site: ftp.Germany.EU.net/pub/research/softcomp/Alife/packages/bugsx/

> Display and evolve biomorphs. It is a program which draws the biomorphs based on parametric plots of Fourier sine and cosine series and let's you play with them using the genetic algorithm.

The Cellular Automata Simulation System
Web site: www.cs.runet.edu/~dana/ca/cellular.html

> The system consists of a compiler for the Cellang cellular automata programming language, along with the corresponding documentation, viewer, and various tools. Cellang has been undergoing refinement for the last several years (1991-1995), with corresponding upgrades to the compiler. Postscript versions of the tutorial and language reference manual are available for those wanting more detailed information. The most important distinguishing features of Cellang, include support for:
>
> - any number of dimensions;

TOC

- compile time specification of each dimensions size; cell neighborhoods of any size (tho bounded at compile time) and shape;
- positional and time dependent neighborhoods;
- associating multiple values (fields), including arrays, with each cell;
- associating a potentially unbounded number of mobile agents [Agents are mobile entities based on a mechanism of the same name in the Creatures system, developed by Ian Stephenson (ian@ohm.york.ac.uk).] with each cell; and
- local interactions only, since it is impossible to construct automata that contain any global control or references to global variables.

dblife & dblifelib

FTP site: ftp.cc.gatech.edu/ac121/linux/games/amusements/life/

dblife: Sources for a fancy Game of Life program for X11 (and curses). It is not meant to be incredibly fast (use xlife for that:-). But it IS meant to allow the easy editing and viewing of Life objects and has some powerful features. The related dblifelib package is a library of Life objects to use with the program.

dblifelib: This is a library of interesting Life objects, including oscillators, spaceships, puffers, and other weird things. The related dblife package contains a Life program which can read the objects in the Library.

Drone

Web site: pscs.physics.lsa.umich.edu/Software/Drone/

Drone is a tool for automatically running batch jobs of a simulation program. It allows sweeps over arbitrary sets of parameters, as well as multiple runs for each parameter set, with a separate random seed for each run. The runs may be executed either on a single computer or over the Internet on a set of remote hosts. Drone is written in Expect (an extension to the Tcl scripting language) and runs under Unix. It was originally designed for use with the Swarm agent-based simulation framework, but Drone can be used with any simulation program that reads parameters from the command line or from an input file.

EcoLab [New]

Web site: parallel.acsu.unsw.edu.au/rks/ecolab.html

EcoLab is a system that implements an abstract ecology model. It is written as a set of Tcl/Tk commands so that the model parameters can easily be changed on the fly by means of editing a script. The model itself is written in C++.

Game Of Life (GOL) [New]

Web site: www.arrakeen.demon.co.uk/downloads.html

FTP site: sunsite.unc.edu/pub/Linux/science/ai/life

GOL is a simulator for conway's game of life (a simple cellular automata), and other simple rule sets. The emphasis here is on speed and scale, in other words you can setup large and fast simulations.

LEE

Web site: www-cse.ucsd.edu/users/fil/lee/lee.html

FTP site: cs.ucsd.edu/pub/LEE/

TOC

LEE (Latent Energy Environments) is both an Alife model and a software tool to be used for simulations within the framework of that model. We hope that LEE will help understand a broad range of issues in theoretical, behavioral, and evolutionary biology. The LEE tool described here consists of approximately 7,000 lines of C code and runs in both Unix and Macintosh platforms.

Net-Life & ZooLife

Web site:www.geocities.com/SiliconValley/Heights/1051**

FTP site: ftp.coe.uga.edu/users/jae/alife/

*(netlife-2.0.tar.gz contains both Net-Life and ZooLife)

Net-Life is a simulation of artificial-life, with neural "brains" generated via slightly random techniques. Net-Life uses artificial neural nets and evolutionary algorithms to breed artificial organisms that are similar to single cell organisms. Net-life uses asexual reproduction of its fittest individuals with a chance of mutation after each round to eventually evolve successful life-forms.

ZooLife is a simulation of artificial-life. ZooLife uses probabilistic methods and evolutionary algorithms to breed artificial organisms that are similar to plant/animal zoo organisms. ZooLife uses asexual reproduction with a chance of mutation.

Primordial Soup

Web site: alife.santafe.edu/alife/software/psoup.html

Primordial Soup is an artificial life program. Organisms in the form of computer software loops live in a shared memory space (the "soup") and self-reproduce. The organisms mutate and evolve, behaving in accordance with the principles of Darwinian evolution.

The program may be started with one or more organisms seeding the soup. Alternatively, the system may be started "sterile", with no organisms present. Spontaneous generation of self-reproducing organisms has been observed after runs as short as 15 minutes.

Tierra

Web site: www.hip.atr.co.jp/~ray/tierra/tierra.html

FTP site: alife.santafe.edu/pub/SOFTWARE/Tierra/

Alternate FTP site: ftp.cc.gatech.edu/ac121/linux/science/biology/

Tierra's written in the C programming language. This source code creates a virtual computer and its operating system, whose architecture has been designed in such a way that the executable machine codes are evolvable. This means that the machine code can be mutated (by flipping bits at random) or recombined (by swapping segments of code between algorithms), and the resulting code remains functional enough of the time for natural (or presumably artificial) selection to be able to improve the code over time.

TIN

FTP site: ftp.coe.uga.edu/users/jae/alife/

This program simulates primitive life-forms, equipped with some basic instincts and abilities, in a 2D environment consisting of cells. By mutation new generations can prove their success, and thus passing on "good family values".

The brain of a TIN can be seen as a collection of processes, each representing drives or impulses to behave a certain way, depending on the state/perception of the environment (e.g. presence of food, walls, neighbors, scent traces) These behavior process currently are : eating, moving, mating, relaxing, tracing others, gathering food and killing. The process with the highest impulse value

TOC

takes control, or in other words: the tin will act according to its most urgent need.

XLIFE
FTP site: ftp.cc.gatech.edu/ac121/linux/games/amusements/life/

This program will evolve patterns for John Horton Conway's game of Life. It will also handle general cellular automata with the orthogonal neighborhood and up to 8 states (it's possible to recompile for more states, but very expensive in memory). Transition rules and sample patterns are provided for the 8-state automaton of E. F. Codd, the Wireworld automaton, and a whole class of `Prisoner's Dilemma' games.

Autonomous Agents

Also known as intelligent software agents or just agents, this area of AI research deals with simple applications of small programs that aid the user in his/her work. They can be mobile (able to stop their execution on one machine and resume it on another) or static (live in one machine). They are usually specific to the task (and therefore fairly simple) and meant to help the user much as an assistant would. The most popular (ie. widely known) use of this type of application to date are the web robots that many of the indexing engines (eg. webcrawler) use.

AgentK
FTP site: ftp.csd.abdn.ac.uk/pub/wdavies/agentk

This package synthesizes two well-known agent paradigms: Agent-Oriented Programming, Shoham (1990), and the Knowledge Query & Manipulation Language, Finin (1993). The initial implementation of AOP, Agent-0, is a simple language for specifying agent behaviour. KQML provides a standard language for inter-agent communication. Our integration (which we have called Agent-K) demonstrates that Agent-0 and KQML are highly compatible. Agent-K provides the possibility of inter-operable (or open) software agents, that can communicate via KQML and which are programmed using the AOP approach.

Agent, the Perl5 Module
FTP site: ftp.hawk.igs.net/pub/users/jduncan/modules/Agent/

The Agent is a prototype for an Information Agent system. It is both platform and language independent, as it stores contained information in simple packed strings. It can be packed and shipped across any network with any format, as it freezes itself in its current state.

AGENT TCL
Web site: www.cs.dartmouth.edu/~agent/
FTP site: ftp.cs.dartmouth.edu/pub/agents/

A transportable agent is a program that can migrate from machine to machine in a heterogeneous network. The program chooses when and where to migrate. It can suspend its execution at an arbitrary point, transport to another machine and resume execution on the new machine. For example, an agent carrying a mail message migrates first to a router and then to the recipient's mailbox. The agent can perform arbitrarily complex processing at each machine in order to ensure that the message reaches the intended recipient.

Aglets Workbench
Web site: www.trl.ibm.co.jp/aglets/

TOC

An aglet is a Java object that can move from one host on the Internet to another. That is, an aglet that executes on one host can suddenly halt execution, dispatch to a remote host, and resume execution there. When the aglet moves, it takes along its program code as well as its state (data). A built-in security mechanism makes it safe for a computer to host untrusted aglets. The Java Aglet API (J-AAPI) is a proposed public standard for interfacing aglets and their environment. J-AAPI contains methods for initializing an aglet, message handling, and dispatching, retracting, deactivating/activating, cloning, and disposing of the aglet. J-AAPI is simple, flexible, and stable. Application developers can write platform-independent aglets and expect them to run on any host that supports J-AAPI.

Ara
Web site: www.uni-kl.de/AG-Nehmer/Ara/

Ara is a platform for the portable and secure execution of mobile agents in heterogeneous networks. Mobile agents in this sense are programs with the ability to change their host machine during execution while preserving their internal state. This enables them to handle interactions locally which otherwise had to be performed remotely. Ara's specific aim in comparison to similar platforms is to provide full mobile agent functionality while retaining as much as possible of established programming models and languages.

JAFMAS [New]
Web site: www.ececs.uc.edu/~abaker/JAFMAS

JAFMAS provides a framework to guide the coherent development of multiagent systems along with a set of classes for agent deployment in Java. The framework is intended to help beginning and expert developers structure their ideas into concrete agent applications. It directs development from a speech-act perspective and supports multicast and directed communication, KQML or other speech-act performatives and analysis of multiagent system coherency and consistency.

Only four of the provided Java classes must be extended for any application. Provided examples of the N-Queens and Supply Chain Integration use only 567 and 1276 lines of additional code respectively for implementation.

JATLite
Web site: java.stanford.edu/java_agent/html/

JATLite is providing a set of java packages which makes easy to build multi-agent systems using Java. JATLite provides only light-weight, small set of packages so that the developers can handle all the packages with little efforts. For flexibility JATLite provides four different layers from abstract to Router implementation. A user can access any layer we are providing. Each layer has a different set of assumptions. The user can choose an appropriate layer according to the assumptions on the layer and user's application. The introduction page contains JATLite features and the set of assumptions for each layer.

Java(tm) Agent Template
Web site: cdr.stanford.edu/ABE/JavaAgent.html

The JAT provides a fully functional template, written entirely in the Java language, for constructing software agents which communicate peer-to-peer with a community of other agents distributed over the Internet. Although portions of the code which define each agent are portable, JAT agents are not migratory but rather have a static existence on a single host. This behavior is in contrast to many other "agent" technologies. (However, using the Java RMI, JAT agents could

dynamically migrate to a foreign host via an agent resident on that host). Currently, all agent messages use KQML as a top-level protocol or message wrapper. The JAT includes functionality for dynamically exchanging "Resources", which can include Java classes (e.g. new languages and interpreters, remote services, etc.), data files and information inlined into the KQML messages.

Java-To-Go
Web site: ptolemy.eecs.berkeley.edu/dgm/javatools/java-to-go/

Java-To-Go is an experimental infrastructure that assists in the development and experimentation of mobile agents and agent-based applications for itinerative computing (itinerative computing: the set of applications that requires site-to-site computations. The main emphasis here is on a easy-to-setup environment that promotes quick experimentation on mobile agents.

Kafka [New]
Web site: www.fujitsu.co.jp/hypertext/free/kafka/

Kafka is yet another agent library designed for constructing multi-agent based distributed applications. Kafka is a flexible, extendable, and easy-to-use java class library for programmers who are familiar with distributed programming. It is based on Java's RMI and has the following added features:

Runtime Reflection:

Agents can modify their behaviour (program codes) at runtime. The behaviour of the agent is represented by an abstract class Action. It is useful for remote maintenance or installation services.

Remote Evaluation:

Agents can receive and evaluate program codes (classes) with or without the serialized object. Remote evaluation is a fundamental function of a mobile agent and is thought to be a push model of service delivery.

Distributed Name Service:

Agents have any number of logical names that don't contain the host name. These names can be managed by the distributed directories.

Customizable security policy

a very flexible, customizable, 3-layered security model is implemented in Kafka.

100% Java and RMI compatible:

Kafka is written completely in Java. Agent is a Java RMI server object itself. So, agents can directly communicate with other RMI objects.

Khepera Simulator
Web site: diwww.epfl.ch/lami/team/michel/khep-sim/

Khepera Simulator is a public domain software package written by Olivier MICHEL during the preparation of his Ph.D. thesis, at the Laboratoire I3S, URA 1376 of CNRS and University of Nice-Sophia Antipolis, France. It allows to write your own controller for the mobile robot Khepera using C or C++ languages, to test them in a simulated environment and features a nice colorful X11 graphical interface. Moreover, if you own a Khepera robot, it can drive the real robot using the same control algorithm. It is mainly oriented toward to researchers studying autonomous agents.

TOC

Mole

Web site: www.informatik.uni-stuttgart.de/ipvr/vs/projekte/mole.html

Mole is an agent system supporting mobile agents programmed in Java. Mole's agents consist of a cluster of objects, which have no references to the outside, and as a whole work on tasks given by the user or another agent. They have the ability to roam a network of "locations" autonomously. These "locations" are an abstraction of real, existing nodes in the underlying network. They can use location-specific resources by communicating with dedicated agents representing these services. Agents are able to use services provided by other agents and to provide services as well.

Odyssey

Web site: www.genmagic.com/agents/

Odyssey is General Magic's initial implementation of mobile agents in 100% pure Java. The Odyssey class libraries enable you to develop your own mobile agent applications. Use mobile agents to access data, make decisions and notify users. Your agent-enabled applications may also take full advantage of the Java platform and use other third party libraries, for example, to access remote CORBA objects or to access relational databases using JDBC. To see how it's done, take a look at the sample applications included as part of the Odyssey download.

Penguin!

FTP site: www.perl.org/CPAN/modules/by-category/23_Miscellaneous_Modules/Penguin/FSG/

Penguin is a Perl 5 module. It provides you with a set of functions which allow you to:

- send encrypted, digitally signed Perl code to a remote machine to be executed.
- receive code and, depending on who signed it, execute it in an arbitrarily secure, limited compartment.

The combination of these functions enable direct Perl coding of algorithms to handle safe internet commerce, mobile information-gathering agents, "live content" web browser helper apps, distributed load-balanced computation, remote software update, distance machine administration, content-based information propagation, Internet-wide shared-data applications, network application builders, and so on.

SimRobot

Web site: www.informatik.uni-bremen.de/~simrobot/

FTP site: ftp.uni-bremen.de/pub/ZKW/INFORM/simrobot/

SimRobot is a program for simulation of sensor based robots in a 3D environment. It is written in C++, runs under UNIX and X11 and needs the graphics toolkit XView.

- Simulation of robot kinematics
- Hierarchically built scene definition via a simple definition language
- Various sensors built in: camera, facette eye, distance measurement, light sensor, etc.
- Objects defined as polyeders
- Emitter abstractly defined; can be interpreted e.g. as light or sound
- Camera images computed according to the raytracing or Z-buffer algorithms known from computer graphics
- Specific sensor/motor software interface for communicating with the simulation

TOC

- Texture mapping onto the object surfaces: bitmaps in various formats
- Comprehensive visualization of the scene: wire frame w/o hidden lines, sensor and actor values
- Interactive as well as batch driven control of the agents and operation in the environment
- Collision detection
- Extendability with user defined object types
- Possible socket communication to e.g. the Khoros image processing software

TclRobots

FTP site: ftp.neosoft.com/pub/tcl/sorted/games/tclrobots-2.0/

Redhat Patch: ftp.coe.uga.edu/users/jae/ai/tclrobots-redhat.patch

RPMs: ftp://ftp.redhat.com/contrib/

TclRobots is a programming game, similar to 'Core War'. To play TclRobots, you must write a Tcl program that controls a robot. The robot's mission is to survive a battle with other robots. Two, three, or four robots compete during a battle, each running different programs (or possibly the same program in different robots.) Each robot is equipped with a scanner, cannon, drive mechanism. A single match continues until one robot is left running. Robots may compete individually, or combine in a team oriented battle. A tournament can be run with any number of robot programs, each robot playing every other in a round-robin fashion, one-on-one. A battle simulator is available to help debug robot programs.

The TclRobots program provides a physical environment, imposing certain game parameters to which all robots must adhere. TclRobots also provides a view on a battle, and a controlling user interface. TclRobots requirements: a wish interpreter built from Tcl 7.4 and Tk 4.0.

The Tocoma Project

Web site: www.cs.uit.no/DOS/Tacoma/index.html

An agent is a process that may migrate through a computer network in order to satisfy requests made by clients. Agents are an attractive way to describe network-wide computations.

The TACOMA project focuses on operating system support for agents and how agents can be used to solve problems traditionally addressed by operating systems. We have implemented a series of prototype systems to support agents.

TACOMA Version 1.2 is based on UNIX and TCP. The system supports agents written in C, Tcl/Tk, Perl, Python, and Scheme (Elk). It is implemented in C. This TACOMA version has been in public domain since April 1996.

We are currently focusing on heterogeneity, fault-tolerance, security and management issues. Also, several TACOMA applications are under construction. We implemented StormCast 4.0, a wide-area network weather monitoring system accessible over the internet, using TACOMA and Java. We are now in the process of evaluating this application, and plan to build a new StormCast version to be completed by June 1997.

Virtual Secretary Project (ViSe) (Tcl/Tk) [New]

Web site: www.cs.uit.no/DOS/Virt_Sec

The motivation of the Virtual Secretary project is to construct user-model-based intelligent software agents, which could in most cases replace human for secretarial tasks, based on modern

TOC

mobile computing and computer network. The project includes two different phases: the first phase (ViSe1) focuses on information filtering and process migration, its goal is to create a se environment for software agents using the concept of user models; the second phase (ViSe2) concentrates on agents' intelligent and efficient cooperation in a distributed environment, its goal is to construct cooperative agents for achieving high intelligence. (Implemented in Tcl/TclX/Tix/Tk)

VWORLD

Web site: www.ai.uga.edu/~jae/projects.html#vworld

Vworld is a simulated environment for research with autonomous agents written in prolog. It is currently in something of an beta stage. It works well with SWI-prolog, but should work with Quitnus-prolog with only a few changes. It is being designed to serve as an educational tool for class projects dealing with prolog and autonomous agents. It comes with three demo worlds or environments, along with sample agents for them.
There are two versions now. One written for SWI-prolog and one written for LPA-prolog. Documentation is roughly done (with a student/professor framework in mind), and a graphical interface is planned.

WebMate

Web site: www.cs.cmu.edu/~softagents/webmate/

WebMate is a personal agent for World-Wide Web browsing and searching. It accompanies you when you travel on the internet and provides you what you want.
Features include:

- Searching enhancement, including parallel search, searching keywords refinement using our relevant keywords extraction technology, relevant feedback, etc.
- Browsing assistant, including learning your current interesting, recommending you new URLs according to your profile and selected resources, monitoring bookmarks of Netscape or IE, sending the current browsing page to your friends, etc.
- Offline browsing, including downloading the following pages from the current page for offline browsing.
- Filtering HTTP header, including recording http header and all the transactions between your browser and WWW servers, etc.
- Checking the HTML page to find the errors or dead links, etc.
- Programming in Java, independent of operating system, runing in multi-thread.

AI & Alife related newsgroups

These newsgroups are not Linux specific. But they are good resources for anyone working in artificial intelligence or artificial life. If you can't access these newsgroups, many of their FAQs are available at: http://www.cis.ohio-state.edu/hypertext/faq/bngusenet/comp/ai/top.html

- comp.ai
- comp.ai.edu
- comp.ai.genetic
- comp.ai.neural-nets

TOC

- comp.ai.vision
- comp.ai.fuzzy
- comp.ai.games
- comp.ai.jair.announce
- comp.ai.jair.papers
- comp.ai.nat-lang
- comp.ai.nlang-know-rep
- comp.ai.philosophy
- comp.ai.shells
- comp.ai.alife
- comp.ai.doc-analysis.misc
- comp.ai.doc-analysis.ocr
- comp.lang.prolog
- comp.lang.lisp
- alt.irc.bots

AI & Alife resource links

These are a few of the many AI and Alife sites out there. These sites are good places to start hunting for more information or for finding software. I'll be adding more links to this list soon, as well as organizing it better. These links are not Linux specific, but I wanted to include them to provide a jump off point to the huge amount of info related to these topics located on the web.

AI/Alife Archives, Research, Bibliographies, etc.

All Catagories

- SAL.KachinaTech.COM/Z/3/-Scientific Applications for Linux's AI page
- www.cs.cmu.edu/Web/Groups/AI/html/repository.html-CMU Artificial Intelligence Repository
- liinwww.ira.uka.de/bibliography/Ai/index.html -Bibliographies on Artificial Intelligence [New]

Traditional AI

- www.mcs.net/~jorn/html/ai.html-Outsider's Guide to AI
- www.cs.cmu.edu/afs/cs.cmu.edu/project/ai-repository/ai/html/cltl/clm/clm.html-Common Lisp Book
- www.elwoodcorp.com/alu/table/contents.htm-The Association of Lisp Users
- www.cs.indiana.edu/scheme-repository/home.html-Scheme repository

TOC

- www.aic.nrl.navy.mil/-Navy Center for Applied Research in Artificial Intelligence
- intranet.ca/~sshah/waste/waste.html-WASTE (AI Contest)
- www.cs.washington.edu/research/jair/home.html-Journal of Artificial Intelligence Research
- www.cs.ucl.ac.uk/misc/ai/-University of London's AI Resource Page
- www.cris.com/~swoodcoc/ai.html -Artificial Intelligence in Games
- strips.lboro.ac.uk/bib/ -An Online Bibliography on Planning and Scheduling

Connectionism

- www.neuronet.ph.kcl.ac.uk/neuronet/software/software.html-NEuroNet - ANN software

Evolutionary Computing

- ftp://ftp.mad-scientist.com/pub/genetic-programming/-John Koza's Genetic Programming archive.
- www.cpsc.ucalgary.ca/~jacob/Evolvica/ -Tutorial: Evolutionary Algorithms in Action
- alife.santafe.edu/~joke/encore/-ENCORE-The Hitch-Hikers Guide to Evolutionary Computation
- gal4.ge.uiuc.edu/illigal.home.html-IlliGAL Home Page (GA's)
- isl.msu.edu/GA/-MSU GARAGe Home Page (GA's)
- www.aracnet.com/~wwir/NovaGenetica/-Nova Genetica (GA's)

Artifical Life

- alife.santafe.edu/~joke/zooland/-ZooLand Artificial Life Resources
- alife.santafe.edu-Santafe's Alife page
- www.krl.caltech.edu/~brown/alife/-Alife FAQ
- www.cogs.susx.ac.uk/users/ezequiel/alife-page/alife.html-ALife Bibliography
- complex.csu.edu.au/complex/-Complex Systems Information Network
- www.geneticprogramming.com/-The Genetic Programming Notebook
- gracco.irmkant.rm.cnr.it/luigi/lupa_algames.html-The Artificial Life Games Homepage
- www.krl.caltech.edu/~charles/alife-game/-Project: Von Neumann

Agents & Bots

- agents.www.media.mit.edu/groups/agents/-MIT Media Lab, Autonomous Agents Group
- www.cs.umbc.edu/agents/agentnews/-AgentNews Webletter
- www.cs.umbc.edu/agents-UMBC AgentWeb (includes KQML info) [New]
- www.agent.org-Agent Society Home Page
- www.botspot.com/main.html-The BotSpot (a software agent resource page) [New]
- www.cselt.it/fipa/-FIPA Foundation for Intelligent Physical Agents
- www.robotmag.com/-Robot Magazine

TOC

- www.geocities.com/SiliconValley/3086/robots/index2.htm-Intro to Indexing Bots (aka spiders) [New]
- luz.cs.nmt.edu/~rtlinux-Real time linux (for robotics, etc)

Linux Advocacy mini-HOWTO

Paul L. Rogers, Paul.L.Rogers@li.org

v0.5, 7 May 1998

This document provides suggestions for how the Linux community can effectively advocate the use of Linux.

1. About this document

2. Copyright Information

3. Introduction

4. Related Information

5. Advocating Linux

6. Canons of Conduct

7. User Groups

8. Vendor Relations

9. Media Relations

10. Acknowledgements

1. About this document

This is the Linux Advocacy mini-HOWTO and is intended to provide guidelines and ideas to assist with *your* Linux advocacy efforts.

This mini-HOWTO was inspired by Jon ``maddog'' Hall when he responded to a request for feedback on guidelines for advocating Linux during NetDay activities. He responded positively to the guidelines and observed that they were the basis of a list of ``canons of conduct'' that would benefit the Linux community.

This document is available in HTML form at http://www.datasync.com/~rogerspl/Advocacy-HOWTO.html.

Nat Makarevitch <nat@nataa.fr.eu.org> has translated this document into French.

Chie Nakatani <jeanne@mbox.kyoto-inet.or.jp> has translated this document into Japanese.

Janusz Batko <janus@krakow.linux.org.pl> has translated this document into Polish.

Bruno H. Collovini <buick@microlink.com.br> has translated this document into Portuguese.

Mauricio Rivera Pineda <rmrivera@hotmail.com> has translated this document into Spanish.

The author and maintainer of the Linux Advocacy mini-HOWTO is Paul L. Rogers <Paul.L.Rogers@li.org>.

Comments and proposed additions are welcome.

If you need to know more about the Linux Documentation Project or about Linux HOWTO's, feel free to contact the supervisor Tim Bynum <linux-howto@sunsite.unc.edu>. Tim Bynum will post this document to several national and international newsgroups on a monthly basis.

A personal note: Due to various circumstances, I have not been able to dedicate as much time to maintaining this mini-HOWTO and interacting with the Linux community as I would have desired. I apologize for this and if you have attempted to contact me and I was slow in responding, please forgive me being so inconsiderate. While I still have many other commitments, I am anticipating that they will start requiring less time to meet and allow me to catch up on other parts of my life. I appreciate your patience and would like to extend a special thanks to all who have taken the time to suggest additions and corrections.

2. Copyright Information

This mini-HOWTO is Copyright © 1996-1998 by Paul L. Rogers. All rights reserved.

A verbatim copy may be reproduced or distributed in any medium physical or electronic without permission of the author. Translations are similarly permitted without express permission if it includes a notice on who translated it.

Short quotes may be used without prior consent by the author. Derivative work and partial distributions of the Advocacy mini-HOWTO must be accompanied with either a verbatim copy of this file or a pointer to the verbatim copy.

Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

We further want that *all* information provided in the HOWTOs is disseminated. If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu.

3. Introduction

The Linux community has known for some time that for many applications, Linux is a stable, reliable, robust (although not perfect) product. Unfortunately, there are still many people, including key decision-makers, that are not aware of the existence of Linux and its capabilities.

If Linux and the many other components that make up a Linux distribution are to reach their full potential, it is critical that we reach out to prospective ``customers'' and advocate (being careful not to promise too much) the use of Linux for appropriate applications. The reason that many company's products have done well in the marketplace is not so much due to the product's technical superiority but the company's marketing abilities.

If you enjoy using Linux and would like to contribute something to the Linux community, please consider acting on one or more of the ideas in this mini-HOWTO and help others learn more about Linux.

4. Related Information

Lars Wirzenius, former comp.os.linux.announce moderator and long-time Linux activist, also has some thoughts about Linux advocacy.

Eric S. Raymond provides an analysis of why the development model used by the Linux community has been so successful.

The free software community has recognized that the terms "free software" and "freely available software" are not appropriate in all contexts. For more information about using the term "open-source software" when marketing "free software", please visit the Open Source site.

If you need to brush up on your Linux sales techniques, take a look at the Linuxmanship essay by Donald B. Marti, Jr.

The Linux PR site discusses the importance of press releases to the Linux community. Another way to gain valuable experience in this area is to organize a NetDay at a local school using the guidelines presented in the NetDay How-To Guide.

Linux International's goal is to promote the development and use of Linux.

The Linux Documentation Project is an invaluable resource for Linux advocates.

The Linux Center Project provides a thematical index of resources about Linux and free software.

The Linux Business Applications site provides a forum for organizations that depend on Linux for day-to-day business operations to share their experiences.

Linux Enterprise Computing and Freely Redistributable Software in Business cover resources and topics of interest to those deploying Linux in a business/commercial/enterprise setting.

The Linux Advocacy Project's goal is to encourage commercial application developers to provide native Linux versions of their software.

The Linux CD and Support Giveaway program is helping make Linux more widely available by encouraging the reuse of Linux CD-ROMs.

Specialized Systems Consultants, Inc. (SSC) hosts the Linux Resources site and publishes the *Linux Journal*.

The linux-biz mailing list is a forum created to discuss the use of Linux in a business environment.

The Linux Mission Critical Systems survey documents successful existing systems which have a large load and are up 24 hours per day.

A number of online publications are now devoted to covering Linux. These include:

- *LinuxFocus*
- *Linuxove noviny*
- *Linux Gazette*
- *PLUTO Journal*.

Additional links to online publications can be found at the Linux Documentation Project and the Linux Center Project.

---

5. Advocating Linux

- Share your personal experiences (good and bad) with Linux. Everyone knows that software has bugs and limitations and if we only have glowing comments about Linux, we aren't being honest. I *love* to tell people about having to reboot four times (three scheduled) in three years.
- If someone has a problem that Linux may be able to solve, offer to provide pointers to appropriate information (Web pages, magazine articles, books, consultants, ...). If you haven't actually used the proposed solution, say so.
- If you are available for making presentations about Linux, register with the Linux Speakers Bureau.
- Offer to help someone start using Linux. Follow up to make sure that they are able to use their system effectively.
- Some people still believe that Linux and similar systems operate only in text-mode. Make sure that they are aware of the availability of graphical applications, such as the Gimp.
- Try to respond to one ``newbie'' posting each week. Seek out the tough questions, you may be the only one to respond and you may learn something in the process. However, if you aren't confident that you can respond with the correct answer, find someone that can.
- Seek out small software development firms and offer to make a presentation about Linux.
- If the opportunity arises, make a presentation to your employer's Information Technology group.
- Participate in community events such as NetDay. While your first priority must be to contribute to the success of the event, use the opportunity to let others know what Linux can do for them.
- Always consider the viewpoints of the person to whom you are ``selling'' Linux. Support, reliability, interoperability and cost are all factors that a decision-maker must consider. Of the above, cost is often the least important portion of the equation.
- Availability of support is often mentioned as a concern when considering the adoption of Linux. Companies such as Caldera, Cygnus Solutions, Red Hat, and S.u.S.E. offer support for some or all components of a typical Linux distribution. In addition, the Linux Consultants HOWTO provides a listing of companies providing commercial Linux related support. Of course, some of the best support is found in the *comp.os.linux* and *linux* newsgroup hierarchies.
- Point out that the production of open-source software takes place in an environment of open collaboration between system architects, programmers, writers, alpha/beta testers and end users which often results in well documented, robust products such as Apache, GNU Emacs, Perl and the Linux kernel.
- Stand up and be counted! Register with the Linux Counter.
- Report successful efforts of promoting Linux to Linux International (li@li.org) and similar organizations.
- Find a new home for Linux CD-ROMs and books that you no longer need. Give them to someone

interested in Linux, a public library or a school computer club. A book and its CD-ROM woul most appropriate for a library. However, please be sure that making the CD-ROM publicly available does not violate a licensing agreement or copyright. Also, inform the library staff that the material on the CD-ROM is freely distributable. Follow up to make sure it is available on the shelves.

- When purchasing books about software distributed with Linux, give preference to books written by the author of the software. The royalties that authors receive from book sales may be the only monetary compensation received for their efforts.
- Encourage Linux-based sites to submit their entry for the Powered by Linux page and suggest that banners promoting Linux, Apache, GNU, Perl ... be displayed on their site.
- Participate! If you have benefited from open-source software, please consider assisting the free software community by:
  - submitting detailed bug reports
  - writing documentation
  - creating artwork
  - supplying management skills
  - suggesting enhancements
  - providing technical support
  - contributing software
  - donating equipment
  - furnishing financial support.

  The Linux Documentation Project provides a list of Linux and Linux-related projects.
- Finally, keep in mind that we all have infinitely more important issues to deal with than the selection of a computing environment.

---

6. Canons of Conduct

- As a representative of the Linux community, participate in mailing list and newsgroup discussions in a professional manner. Refrain from name-calling and use of vulgar language. Consider yourself a member of a virtual corporation with Mr. Torvalds as your Chief Executive Officer. Your words will either enhance or degrade the image the reader has of the Linux community.
- Avoid hyperbole and unsubstantiated claims at all costs. It's unprofessional and will result in unproductive discussions.
- A thoughtful, well-reasoned response to a posting will not only provide insight for your readers, but will also increase their respect for your knowledge and abilities.
- Don't bite if offered flame-bait. Too many threads degenerate into a ``My O/S is better than your O/S'' argument. Let's accurately describe the capabilities of Linux and leave it at that.
- Always remember that if you insult or are disrespectful to someone, their negative experience may be shared with many others. If you do offend someone, please try to make amends.
- Focus on what Linux has to offer. There is no need to bash the competition. Linux is a good, solid product that stands on its own.
- Respect the use of other operating systems. While Linux is a wonderful platform, it does not meet everyone's needs.
- Refer to another product by its proper name. There's nothing to be gained by attempting to ridicule a company or its products by using ``creative spelling''. If we expect respect for Linux, we must respect other products.
- Give credit where credit is due. Linux is just the kernel. Without the efforts of people involved with the GNU project, MIT, Berkeley and others too numerous to mention, the Linux kernel would not be very useful to most people.
- Don't insist that Linux is the only answer for a particular application. Just as the Linux community cherishes the freedom that Linux provides them, Linux only solutions would deprive others of their freedom.
- There will be cases where Linux is not the answer. Be the first to recognize this and offer another solution.

7. User Groups

- Participate in a local user group. An index of Linux User Group registries is part of the Linux Documentation Project. If a user group does not exist in your area, start one.
- The Linux User Group HOWTO covers many of the issues involved with starting an user group and discusses the importance of Linux advocacy as one of the goals of a user group.
- Make speakers available to organizations interested in Linux.
- Issue press releases about your activities to your local media.
- Volunteer to configure a Linux system to meet the needs of local community organizations. Of course, the installation process must include training the user community to use the system and adequate documentation for ongoing maintenance.
- Discus the Linux Advocacy mini-HOWTO at a meeting. Brainstorm and submit new ideas.

8. Vendor Relations

- When contemplating a hardware purchase, ask the vendor about Linux support and other user's experiences with the product in a Linux environment.
- Consider supporting vendors that sell Linux based products and services. Encourage them to have their product listed in the Linux Commercial HOWTO.
- Support vendors that donate a portion of their income to organizations such as the Free Software Foundation, the Linux Development Grant Fund, the XFree86 Project or Software in the Public Interest. If possible, make a personal donation to these or other organizations that support open-source software. Don't forget that some employers offer a matching gift program program.
- If you need an application that is not supported on Linux, contact the vendor and request a native Linux version.

9. Media Relations

- Linux International is collecting press clippings that mention Linux, GNU and other freely redistributable software. When you see such an article, please send the following information to clippings@li.org:
  - Name of publication
  - Publisher's contact address
  - Name of author
  - Author's contact address
  - Title of article
  - Page number where the article starts
  - The URL if available online
  - A summary of the article, including your opinion
- If you believe that Linux was not given fair treatment in an article, review or news story, send the details, including the above information, to li@li.org so that an appropriate response can be sent to the publisher. If you contact the publisher directly, be professional and sure of your facts.
- If you involved with a Linux related project, issue press releases to appropriate news services on a regular basis.

10. Acknowledgements

Grateful acknowledgement is made to all contributors, including:

```
Kendall G. Clark           <kclark@cmpu.net>
Bruno H. Collovini         <buick@microlink.com.br>
Allan "Norm" Crain         <allanc@idea-inc.com>
Jon "maddog" Hall          <maddog@zk3.digital.com>
Greg Hankins               <gregh@cc.gatech.edu>
Eric Ladner                <eladner@goldinc.com>
Chie Nakatani              <jeanne@mbox.kyoto-inet.or.jp>
Daniel P. Kionka           <dkionka@worldtalk.com>
Nat Makarevitch            <nat@nataa.fr.eu.org>
Martin Michlmayr           <tbm@cyrius.com>
Rafael Caetano dos Santos <rcaetano@dijkstra.ime.usp.br>
Idan Shoham                <idan@m-tech.ab.ca>
Adam Spiers                <adam.spiers@new.ox.ac.uk>
C. J. Suire                <suire@datasync.com>
Juhapekka Tolvanen         <juhtolv@silmu.cc.jyu.fi>
Lars Wirzenius             <liw@iki.fi>
Sean Woolcock              <sbwoolco@eos.ncsu.edu>
```

Alsa-sound-mini-HOWTO

Valentijn Sessink `valentyn@alsa-project.org`

v0.2, 18 May 1999

Describes the installation of the ALSA sound drivers for Linux. These sound drivers can be used as a replacement for the regular sound drivers, as they are fully compatible.

1. Introduction

- 1.1 Acknowledgments
- 1.2 Revision History
- 1.3 New versions of this document
- 1.4 Feedback
- 1.5 Distribution Policy

2. Before you start

- 2.1 Introduction
- 2.2 General information about the ALSA drivers
- 2.3 Supported hardware
- 2.4 Other HOWTO's

3. How to install ALSA sound drivers

- 3.1 What you need
- 3.2 Getting the drivers
- 3.3 Extracting
- 3.4 Compiling
- 3.5 Preparing the devices

4. Loading the driver

- 4.1 Inserting with modprobe
- 4.2 Which module for which card?
- 4.3 modprobe for drivers without auto-probing
- 4.4 The kerneld approach
- 4.5 Backwards compatibility

5. Testing and using

- 5.1 The /proc filesystem
- 5.2 The mixer
- 5.3 The /dev/snd/ devices
- 5.4 Additional information

6. Tips and Troubleshooting

- 6.1 Compiling the driver
- 6.2 Loading the driver
- 6.3 Driver loaded... but no sound
- 6.4 General suggestions
- 6.5 Bug reports
- 6.6 Tip: playing CD's
- 6.7 Tip: installing the MIDI serial driver
- 6.8 Tip: new kernel? New modules!
- 6.9 Tip: KDE and ALSA drivers
- 6.10 Tip: use the ALSA devices
- 6.11 Tip: removing all modules

1. Introduction

This is the ALSA Sound drivers mini-HOWTO. It gives you information about installing and using the ALSA sound drivers for your soundcard. The ALSA drivers are fully modularized sound drivers that support kerneld and kmod. They are compatible with, but surpass the possibilities of, the current OSS API. In other words: compatible, but better.

1.1 Acknowledgments

This documents contains information I got from the ALSA driver page. The structure was ripped off the SB-mini-HOWTO, mainly because it had about the structure I was looking for. Thanks to the SGML Tools package, this HOWTO is available in several formats, all generated from a common source file. Thanks to Erik Warmelink for proof reading, thanks to Alfred Munnikes for a couple of questions and helpful suggestions. Yamahata Isaku thanks for the Japanese translation, Miodrag Vallat for the translation in French. Later on, Steve Crowder did a great job by reading and editing the whole text. Thanks to Marc-Aur`ele Darche, Piotr Ingling, Juergen Kahrs and Tim Pearce for useful tips and additions.

1.2 Revision History

Version 1.4 - May 18, 1999. Included the URL to the French version, changed more URLs.

Version 1.3 - May 16, 1999. Thanks to Jaroslav this HOWTO has found a home at the ALSA-project website. As a result of that, some updates in mail and web addresses.

Version 1.2 - May 11, 1999. Several updates.

Version 1.1 - March 11, 1999. Added a couple of sound cards from the new 0.3 series drivers, wrote a bit about the 2.2 series kernel.

Version 1.0 - February 8, 1999. Added a few things to the troubleshooting section, but we seem fairly complete.

Version 0.3 beta - January 20, 1999. A link on the ALSA-homepage. Ha, we're official!

Version 0.2 alpha - Mid January 1999, first .sgml-version.

Version 0.1 alpha - January 1999, first version, mostly HTML.

Still: please submit any patches in plain English, you native speakers!

1.3 New versions of this document

TOC

The latest version can be found at http://www.alsa-project.org./~valentyn

Other formats (full size html, sgml, txt) are in the directory other-formats.

Yamahata Isaku has translated a Japanese version, which will be available at the Japanese ALSA site, http://plaza21.mbn.or.jp/~momokuri/alsa/index.html

Miodrag Vallat translated a French version, which is available at http://www.freenix.fr/unix/linux/HOWTO/mini/Alsa.html.

If you make a translation of this document into another language, let me know and I'll include a reference to it here.

1.4 Feedback

I rely on you, the reader, to make this HOWTO useful. If you have any suggestions, corrections or comments, please send them to me (alsa-howto@alsa-project.org), and I will try to incorporate them in the next revision.

Please note: I do not get a lot of mail about the ALSA drivers and any addition is welcome. Even a ``thank you for'' is appreciated - maybe it's not too much work to add a ``I appreciated most'' or ``this-or-that was not immediately clear to me''-section.

If you publish this document on a CD-ROM or in hardcopy form, a complimentary copy would be appreciated. Mail me for my postal address. Also consider making a donation to the Linux Documentation Project to help support free documentation for Linux. Contact the Linux HOWTO co-ordinator, Tim Bynum linux-howto@metalab.unc.edu, for more information.

1.5 Distribution Policy

Copyright 1998/1999 Valentijn Sessink

This HOWTO is free documentation; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but **without any warranty**; without even the implied warranty of **merchantability** or **fitness for a particular purpose**. See the GNU General Public License for more details.

You can obtain a copy of the GNU General Public License by writing to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

---

2. Before you start

2.1 Introduction

This document tries to help you install and use the ALSA sound drivers in your Linux system. The reference system is a Slackware 3.5 distribution of Linux on an AMD/K6 computer (x86 compatible), but it should work with any other Linux distribution. I do not know if the ALSA drivers work on other platforms, according to the documentation, Alpha has been tested and proven to work. I have only x86 PC's here, so any additional information you may have would be appreciated.

It might be handy to read the Linux Sound HOWTO (see section Other HOWTO's), but that HOWTO focuses on the built-in kernel drivers.

2.2 General information about the ALSA drivers

The ALSA sound driver was originally written as a replacement for the Linux kernel sound for Gravis UltraSound (GUS) cards. As this GUS replacement proved to be a success, the author started the ALSA project for a generic driver for several sound chips, with fully modularized design.

It is compatible with the OSS/Free and OSS/Linux sound drivers (the drivers in the kernel), but has its own interface that is even better than the OSS drivers. A list of features can be found at http://www.alsa-project.org/intro.html

The main page of the ALSA project is http://www.alsa-project.org/

2.3 Supported hardware

The ALSA drivers support only a subset of all sound cards available. As the time of writing, the following cards are supported:

- Cards with a Trident 4D Wave DX/NX chipset, thanks to Trident Microsystems who offered ALSA ``first cut'' GPL'd drivers (MIXER and PCM devices only) and documentation for their 4D Wave PCI audio chipsets. See http://www.tridentmicro.com/HTML/products%20folder/audio.htm for more information. The drivers are now in cvs, and will be included with the ALSA 0.3.0-pre5 release.
- Gravis Ultrasound (GUS): "PnP", Extreme, Classic/ACE, MAX
- Cards with a GUS chipset: Dynasonic 3-D, STB Sound Rage 32, UltraSound 32-Pro (STB)
- Soundblaster: 1.0, 2.0, Pro, 16, AWE32/64, PCI64
- ESS AudioDrive ESx688
- ESS ES18xx (chipsets). Please note that I personally experienced a lot of trouble with the

TOC

ESS1888. The developer of the driver for this card did his best, but to no avail.

- ESS Solo-1 ES1938
- Yamaha: OPL3-SA2, OPL3-SA3 (chipsets)
- OAK Mozart
- Schubert 32 PCI (PINE, S3 SonicVibes PCI chipset)
- Ensoniq AudioPCI ES1370/1371 PCI soundcards (Soundblaster PCI64)
- SonicVibes PCI soundcards (PINE Schubert 32 PCI)

Then a whole lot of Crystal Semiconductors-based sound boards are supported. These chips can be found in a lot of hardware, in separate cards (some Philips PCA series)and on motherboards (e.g. IBM Aptiva, Dell computers). Boards based on the following chipsets are supported:

- 4232
- 4232A
- 4235
- 4236B
- 4237B
- 4238B
- 4239

A more recent list may be found inside the driver package itself, that is in doc/SOUNDCARDS

2.4 Other HOWTO's

This ALSA-sound-mini-HOWTO is just mini. Other HOWTO's may help you out in case this one is too terse. I will name a few things you may come across while trying to install the ALSA drivers. HOWTO's can generally, be found at mirrors of Metalab (the former Sunsite). So take a look at http://metalab.unc.edu/LDP/mirrors.html and pick out your closest mirror site. You can find HOWTO's in the directory LDP/HOWTO/. Please note: the links in this document will all be relative to /LDP/HOWTO/mini. If you look at this document from a reasonably good mirror site, you will find the HOWTO's.

Then a note for the 2.2.x kernel series. For the 2.2.x kernel series, sound support is like any other support: it works, but it is different from what you used to do. This HOWTO (like any other HOWTO) will concentrate on the 2.0 series kernel, although I'll try to point out the differences. I will gladly add anything that would help bring this HOWTO upto date with the 2.2 series kernel.

Sound cards

Perhaps you bought a sound card already, or maybe it has been installed in your computer for ages. And now you are going to use it ! Have a look at the Sound-HOWTO to see if this is all worth the trouble. (You might want to buy this new Mega-Rumble-Blaster first, then try the ALSA drivers.)

TOC

Plug and Play cards

Most modern sound cards for the Intel platform are ISA PnP cards, which is an abbreviation for "Plug and Play". This means, that the card has to be configured by the operation system. This has to be done through an initialization routine at boot time. You probably need to configure your card with the PnP-utils-package. Every recent Linux distribution includes these tools. For usage have a look at the Plug-and-Play-HOWTO

The ALSA-drivers seem to have built in their own ISA-PnP-support for a couple of sound cards. Unfortunately, as I cannot find documentation about this, I cannot tell you how it works. If anyone out there wants to try ALSA sound support while deliberately not using the ISA-PnP-tools, please drop me a line.

Loadable modules

The ALSA sound drivers are built as modules. You can find more information about modules in the Kernel-HOWTO. There is also a module-HOWTO, but that is unmaintained at the moment; take a look at the umaintained section of the Howto-HOWTO. There is a Modules-mini-HOWTO though that may be useful.

Kerneld

Another HOWTO that will be useful for some, is the Kerneld-mini-HOWTO. Kerneld is a daemon that installs and removes kernel modules as needed. (I have zero experience with it, so additional information on the topic is welcome. The ALSA driver documentation contains some information about configuration of the kerneld, this has been included in this mini-HOWTO.)

---

3. How to install ALSA sound drivers

3.1 What you need

- a functional Linux system (e.g. the Slackware distribution), with the "Development" packages installed (i.e. gcc, make etc.)
- a supported sound card
- some knowledge about Linux (meaning you know how to use "ls", "cd", "tar" etc.)
- a root-account

If you have a PnP card, you will also need:

- the isapnptools software package.

The INSTALL text in the driver directory suggests that for some cards, PnP support is native. I also received a suggestion from Jaroslav about this. When I get further information about this topic I will add it to this mini-HOWTO.

Please note that you should not have any sound drivers active when you want to use the ALSA drivers. If you have a kernel with sound drivers compiled in, you'll need a kernel recompilation. If you have the old "sound.o" module active, you need to deactivate it. If you use kerneld, this probably means deleting sound.o from the /lib/modules/<kernel_version>/misc directory. Newer RedHat systems have a different sound approach, with several sound modules active. You need to deactivate them all.

The 2.2 series kernel has a new approach to sound. You should include sound support here ! Yep, that's right: you add sound support to the kernel, but do not include any sound card. Then compile and install the kernel and after that, compile the ALSA-drivers.

3.2 Getting the drivers

The ALSA drivers are available from ftp://ftp.alsa-project.org/pub/ and there is a mirror at ftp://linux.a2000.nl/alsa For a fully functional ALSA-installation, you will need the driver, the libs and the utilities; e.g if you chose the mirror you would get ftp://linux.a2000.nl/alsa/driver/alsa-driver-0.3.0-pre4.tar.gz, ftp://linux.a2000.nl/alsa/lib/alsa-lib-0.3.0pre4.tar.gz and ftp://linux.a2000.nl/alsa/utils/alsa-utils-0.3.0-pre4.tar.gz (and NO these are not full links, you'd better look at the original place if there are newer versions available, simply get the last one).

TOC

3.3 Extracting

You extract the drivers by some reasonable command, like the all-time tar -zxf <file>. For newbies and for those with amnesia:

```
cd /usr/src

tar -zxf ~/alsa-driver-0.3.0-pre4.tar.gz
```

(If this worked you are either logged in as root always, which is the wrong way, stop here and read the DOS/Win95-to-Linux-HOWTO, or you already thought this would be the administrative thing you needed superuser/rootprivs for. So: for this you need rootprivs, type a "su" and the password).

```
tar -zxf ~/alsa-lib-0.3.0pre4.tar.gz

tar -zxf ~/alsa-utils-0.3.0-pre4.tar.gz
```

Also working and more fun: find ~ -name alsa* -exec tar -zxf {} \; (Don't try this at home kids, it's just an example). Note that when downloading the drivers with Netscape, you may accidentally get unpacked drivers with a ".tgz" extension. If tar complains about the file format, you may get better results by leaving off the "z" in the tar options.

3.4 Compiling

You need the drivers before you can compile and use the libs. You need the libs before you can compile or use the utils. So let's begin:

```
cd alsa-driver-0.3.0-pre4
```

(and for those not so experienced: try typing a <tab> (the "tab"-key) after "alsa-d". That's called *command line completion*.)

```
./configure

make
```

Now you need to be 'root' to install the stuff (you probably were "root" already)

```
make install
```

If this tells you that something like ``version.h" cannot be found, then you probably do not have a proper kernel source tree. You need a couple of files of your kernel source to be able to compile the ALSA-drivers. Unpack your favorite linux-2.x.y.tar.gz in /usr/src, and issue a `make menuconfig`. (Actually, `make symlinks` may be enough). Now compile the libraries:

```
cd ../alsa-lib-0.3.0-pre4
```

```
./configure

make

make install
```

OK, you're getting it, the utilities:

```
cd ../alsa-utils-0.3.0-pre4

./configure

make

make install
```

Note: you can leave out the "make install" for the utilities at first. You could even leave out the whole library-making and utility-making, just to check if the driver works.

3.5 Preparing the devices

There is a script in the driver-directory that will install the ALSA-sound-devices in your /dev directory. Type

```
./snddevices
```

from the driver-directory. There should be a /dev/snd subdirectory now (test if it is there. If you are not familiar with even the "ls" command, please consider reading other HOWTO's first. You should have some basic Linux knowledge to install these drivers).

Now you're ready to insert the driver, so please turn over to the next paragraph.

---

4. Loading the driver

There are two ways to use the ALSA-sound-modules. I personally prefer using the manual method, meaning that I insert the driver at startup. The ALSA-drivers were designed as loadable/unloadable modules - for instance they do not reset the mixer after loading - so you can easily use the kerneld approach.

Please do read the section Backwards Compatibility. You need it to have sound support ``the old way''.

4.1 Inserting with modprobe

Note: If you have a PnP audio-card, you first need to set it to the right (or at least some known) IO/IRQ/DMA. See the Plug-and-Play-HOWTO. Did you configure your Plug-and-Play-soundcard ? Ok, then read on please. The main part is: do a "modprobe snd-<soundcard>". This should do the trick. Please note that not all distributions do include /sbin in your path. If you get a "bash: modprobe: command not found", this will most likely mean that modprobe is not in your path. Try "/sbin/modprobe snd-sb16", or try to find the modprobe utility elsewhere.

The most important difficulty is with the Crystal chipsets, for these the ALSA-drivers are not auto probing. More recent information may be acquired from the INSTALL file in the driver-directory. Two examples, then a list: Gravis UltraSound (GUS) and compatibles:

```
/sbin/modprobe snd-gusclassic
```

For all 16-bit Soundblaster-cards (SoundBlaster 16 (PnP), SoundBlaster AWE 32 (PnP), SoundBlaster AWE 64 (PnP):

```
/sbin/modprobe snd-sb16
```

4.2 Which module for which card?

Gravis UltraSound Extreme

```
modprobe snd-gusextreme
```

Gravis UltraSound MAX

```
modprobe snd-gusmax
```

ESS AudioDrive

ESS AudioDrive ES-1688 and ES-688 soundcards

```
modprobe snd-audiodrive1688
```

Gravis UltraSound PnP

Gravis UltraSound PnP, Dynasonic 3-D/Pro, STB Sound Rage 32 and other soundcards based on AMD InterWave (tm) chip.

```
modprobe snd-interwave
```

TOC

UltraSound 32-Pro

UltraSound 32-Pro (soundcard from STB used by Compaq) and other soundcards based on AMD InterWave (tm) chip with TEA6330T circuit for extended control of bass, treble and master volume

```
modprobe snd-interwave-stb
```

Soundblaster

8-bit Soundblaster cards (SoundBlaster 1.0, SoundBlaster 2.0, SoundBlaster Pro)

```
modprobe snd-sb8
```

Soundblaster 16

16-bit SoundBlaster cards (SoundBlaster 16 (PnP), SoundBlaster AWE 32 (PnP), SoundBlaster AWE 64 (PnP). Please note: this module does not support the SoundBlaster VibraX16 soundcard.

```
modprobe snd-sb16
```

Yamaha OPL3-SA2/SA3 soundcards

Just "modprobe snd-opl3sa" will not work, this driver does not do autoprobing. See below.

S3 SonicVibes

S3 SonicVibes PCI soundcards. (PINE Schubert 32 PCI)

```
modprobe snd-sonicvibes
```

Ensoniq/Soundblaster PCI64

Ensoniq AudioPCI ES1370/1371 PCI soundcards. (SoundBlaster PCI 64)

```
modprobe snd-audiopci
```

CS4232/4232A

All soundcards based on CS4232/CS4232A chips. Just "modprobe snd-card-cs4232" will not work, no auto-probing. See below.

4235 and higher

All soundcards based on CS4235/CS4236/CS4236B/CS4237B/CS4238B/CS4239 chips. Just "modprobe snd-card-cs4236" will not work, no auto-probing. See below.

4.3 modprobe for drivers without auto-probing

If you have a non-autoprobing driver, you need to supply additional info at startup to have the driver work. More information can be found in the file INSTALL in the driver directory.

OPL3-SA2 and OPL3-SA3

According to the INSTALL file you need to supply all the information for this driver. If you initialized the card without isapnp-tools, you can probably get info from the /etc/isapnp.conf file for the following values:

```
snd_port - control port # for OPL3-SA chip
snd_wss_port - WSS port # for OPL3-SA chip (0x530,0xe80,0xf40,0x604)
snd_midi_port - port # for MPU-401 UART (0x300,0x330), -1 = disable
snd_fm_port - FM port # for OPL3-SA chip (0x388), -1 = disable
snd_irq - IRQ # for OPL3-SA chip (5,7,9,10)
snd_dma1 - first DMA # for Yamaha OPL3-SA chip (0,1,3)
snd_dma1_size - max first DMA size in kB (4-64kB)
snd_dma2 - second DMA # for Yamaha OPL3-SA chip (0,1,3), -1 = disable
snd_dma2_size - max second DMA size in kB (4-64kB)
```

You would do a "modprobe snd-opl3sa snd_port=0xNNN snd_wss_port=0x530 snd_midi_port=-1 snd_fm_port=0x388 snd_irq=5 snd_dma1=0 snd_dma1_size=NN snd_dma2=1 snd_dma2_size=NN" to load this driver (without midi-support. I am still convinced that midi-support is the thing you need when you have synthesizers and stuff and want to connect them to your Linux box. Never needed Midi-support even to play midi-files.)

Note that the "NN" values need to be supplied, only I do not know what would be reasonable values. I do not know if the dma size option is really required.

CS4232/CS4232A chips

According to the INSTALL file you need to supply the main port for this card. Note that with the driver for 3235/6/7/8/9 cards, the one below, I ended up supplying all information (except DMA-size), otherwise the driver did not work. So you may as well use the whole command line to insert the driver. If you initialized the card with the isapnp-tools, you can probably get info from the /etc/isapnp.conf file for the following values:

```
snd_port - port # for CS4232 chip (PnP setup - 0x534)
snd_cport - control port # for CS4232 chip (PnP setup - 0x120)
snd_mpu_port - port # for MPU-401 UART (PnP setup - 0x300), -1 = disable
snd_fm_port - FM port # for CS4232 chip (PnP setup - 0x388), -1 = disable
snd_jport - joystick port for CS4232 chip (PnP setup - 0x200), -1 =
disable
snd_irq - IRQ # for CS4232 chip (5,7,9,11,12,15)
snd_mpu_irq - IRQ # for MPU-401 UART (9,11,12,15)
snd_dma1 - first DMA # for CS4232 chip (0,1,3)
snd_dma1_size - max first DMA size in kB (4-64kB)
snd_dma2 - second DMA # for Yamaha CS4232 chip (0,1,3), -1 = disable
snd_dma2_size - max second DMA size in kB (4-64kB)
```

You would do a "modprobe snd-card-cs4232 snd_port=0x534 snd_cport=0x120 snd_mpu_port=-1 snd_fm_port=0x388 snd_jport=-1 snd_irq=5 snd_dma1=0 snd_dma1_size=NN snd_dma2=1 snd_dma2_size=NN" to load the driver for a "standard configured" soundcard. (Without midi-support, see the note at Yamaha OPL-3, and no joystick support). If you used different values in /etc/isapnp.conf, then you would use the values here also (Note: it can be wise to use your brains anyway ;)

Note that the "NN" values need to be supplied, only I do not know what would be reasonable values. I do not know if the dma size option is really required.

TOC

CS4235/CS4236/CS4236B/CS4237B/CS4238B/CS4239 chips

According to the INSTALL file you need to supply the main port and control ports for this card. Note that with a CS4237B card, I ended up supplying all information (except DMA-size), otherwise the driver did not work. So you may as well use the whole command line to insert the driver, and not only supply snd_port and snd_cport. If you initialized the card with the isapnp-tools, you can probably get info from the /etc/isapnp.conf file for the following values:

```
snd_port - port # for CS4232 chip (PnP setup - 0x534)
snd_cport - control port # for CS4232 chip (PnP setup - 0x120)
snd_mpu_port - port # for MPU-401 UART (PnP setup - 0x300), -1 = disable
snd_fm_port - FM port # for CS4232 chip (PnP setup - 0x388), -1 = disable
snd_jport - joystick port for CS4232 chip (PnP setup - 0x200), -1 = disable
snd_irq - IRQ # for CS4232 chip (5,7,9,11,12,15)
snd_mpu_irq - IRQ # for MPU-401 UART (9,11,12,15)
snd_dma1 - first DMA # for CS4232 chip (0,1,3)
snd_dma1_size - max first DMA size in kB (4-64kB)
snd_dma2 - second DMA # for Yamaha CS4232 chip (0,1,3), -1 = disable
snd_dma2_size - max second DMA size in kB (4-64kB)
```

You would do a "modprobe snd-card-cs4232 snd_port=0x534 snd_cport=0x120 snd_mpu_port=-1 snd_fm_port=0x388 snd_jport=-1 snd_irq=5 snd_dma1=0 snd_dma1_size=NN snd_dma2=1 snd_dma2_size=NN" to load the driver. (Without midi-support, see the note at Yamaha OPL-3, and no joystick support). Notes:

- the "NN" values need to be supplied, only I do not know what would be reasonable values.
- my CS4237B works fine without explicit dma size option.

4.4 The kerneld approach

`kerneld` is a daemon that inserts modules on request, and unloads them once they are not in use anymore. Since I have no experience with kerneld, I do not know if the information below is accurate. The info comes from the INSTALL file in the ALSA-drivers package. Excellent information about kerneld can be found in the kerneld-mini-HOWTO.

Follow these steps:

- Edit your /etc/conf.modules (see below for examples)
- Run 'modprobe snd-card' where card is name of your card [Which I find rather strange, since kerneld is supposed to load them? VS]

Example for /etc/conf.modules for Gravis UltraSound PnP soundcard:

```
alias char-major-14 snd
alias snd-minor-oss-0 snd-interwave
alias snd-minor-oss-3 snd-pcm1-oss
alias snd-minor-oss-4 snd-pcm1-oss
alias snd-minor-oss-5 snd-pcm1-oss
alias snd-minor-oss-12 snd-pcm1-oss
alias snd-card-0 snd-interwave
options snd snd_major=14 snd_cards_limit=1
options snd-interwave snd_index=1 snd_id="guspnp" snd_port=0x220 snd_irq=5
snd_dma1=5 snd_dma2=6
```

Example if you want use more soundcards in one machine (configuration below is for Sound Blaster 16 and Gravis UltraSound Classic):

```
alias char-major-14 snd
alias snd-minor-oss-0 snd-mixer
alias snd-minor-oss-3 snd-pcm1-oss
alias snd-minor-oss-4 snd-pcm1-oss
alias snd-minor-oss-5 snd-pcm1-oss
alias snd-minor-oss-12 snd-pcm1-oss
alias snd-card-0 snd-sb16
alias snd-card-1 snd-gusclassic
options snd snd_major=14 snd_cards_limit=2
options snd-sb16 snd_index=1 snd_port=0x220 snd_irq=5 snd_dma8=1 snd_dma16=5
options snd-gusclassic snd_index=2 snd_irq=11 snd_dma1=6 snd_dma2=7
```

Example if two Gravis UltraSound Classic soundcards are present in system:

```
alias char-major-14 snd
alias snd-minor-oss-0 snd-mixer
alias snd-minor-oss-3 snd-pcm1-oss
alias snd-minor-oss-4 snd-pcm1-oss
alias snd-minor-oss-5 snd-pcm1-oss
alias snd-minor-oss-12 snd-pcm1-oss
alias snd-card-0 snd-gusclassic
alias snd-card-1 snd-gusclassic
options snd snd_major=14 snd_cards_limit=2
options snd-gusclassic snd_index=1,2 snd_port=0x220,0x260 snd_irq=5,11
snd_dma1=5,6 snd_dma2=7,3
```

4.5 Backwards compatibility

If you want to preserve OSS/Free or OSS/Linux compatibility, you need to insert one more driver: the snd-pcm1-oss driver for OSS-compatibility. Issue a

```
modprobe snd-pcm1-oss
```

This will give you /dev/audio and /dev/dsp-support, just as the OSS/Free (kernel) drivers and OSS/Linux (the $25 ones) do. Note that this is only an emulation.

---

5. Testing and using

Now you should test if the sound driver really is available, then try to use it.

5.1 The /proc filesystem

You can find a lot of useful information about your system in the /proc subdirectory. /proc is a "virtual" filesystem, meaning that it does not exist in real life, but merely is a mapping to various processes and tasks in your computer. In order for /proc to work, you need to have support for it compiled into your kernel. Most linux distributions have this as a default, but if you compiled a kernel and left /proc out obviously there won't be anything in /proc.

/proc/modules gives information about loaded modules. Once the ALSA sound drivers are loaded, if you type cat /proc/modules you should see something like:

```
snd-pcm1-oss      4            0
snd-sb16          1            1
snd-sb-dsp        4    [snd-sb16]      0
snd-pcm1          4    [snd-pcm1-oss snd-sb-dsp]      0
snd-pcm           3    [snd-pcm1-oss snd-sb16 snd-sb-dsp snd-pcm1]    0
snd-mixer         3    [snd-pcm1-oss snd-sb16 snd-sb-dsp]      1
snd-mpu401-uart   1    [snd-sb16]      0
snd-midi          4    [snd-sb16 snd-sb-dsp snd-mpu401-uart]   0
snd-opl3          1    [snd-sb16]      0
snd-synth         1    [snd-sb16 snd-opl3]     0
snd-timer         1    [snd-opl3]      0
snd               8    [snd-pcm1-oss snd-sb16 snd-sb-dsp snd-pcm1 snd-pcm snd-mixer
snd-mpu401-uart snd-midi snd-opl3 snd-synth snd-timer]    0
```

If something went wrong during the installation of the driver, you will still see a couple of "snd" devices, but there won't be sound support.

For example (Note: you should never issue this command as follows, the cs4236 driver needs options):

```
win3:~# modprobe snd-card-cs4236
/lib/modules/2.0.35/misc/snd-card-cs4236.o: init_module: Device or resource busy
snd-mixer: Device or resource busy
win3:~# cat /proc/modules
snd-cs4236        2           0
snd-cs4231        3    [snd-cs4236]    0
snd-timer         1    [snd-cs4231]    0
snd-pcm1          4    [snd-cs4236 snd-cs4231] 0
snd-mixer         3    [snd-cs4236 snd-cs4231] 0
snd-pcm           3    [snd-cs4236 snd-cs4231 snd-pcm1]       0
snd-mpu401-uart   1           0
snd-midi          4    [snd-mpu401-uart]       0
snd-opl3          1           0
snd-synth         1    [snd-opl3]      0
snd-timer         1    [snd-cs4231 snd-opl3]   0
snd               8    [snd-cs4231 snd-timer snd-pcm1 snd-mixer snd-pcm]     0
```

You can check the existence of a soundcard by looking in /proc/asound/cards. For example:

```
bash$ cat /proc/asound/cards
0 [card1          : SB16 - Sound Blaster 16
                    Sound Blaster 16 at 0x220, irq 5, dma 1&5
```

TOC

In the previous example (where I forgot the options) the output would have been:

```
win3:~# cat /proc/asound/cards
--- no soundcards ---
```

A working CS4236 card would produce

```
0 [card1         ]: CS4236 - CS4237B
                    CS4237B at 0x534, irq 7, dma 1&0
```

If you checked and doublechecked your settings and still see no sound card, take a look at the troubleshooting section.

The `/proc/asound/` virtual directory shows lots of other information about the driver. Please note that /proc/asound/ will only exist after you inserted the first ALSA module. If there is no /proc/asound, it simply means that the "snd" module was not loaded properly. You can find installed cards in /proc/asound/cards, then find information about card0 in /proc/asound/0, /proc/asound/1 for card1 etcetera.

If `cat /proc/asound/` shows something like

```
ES1370 DAC2/ADC
Playback isn't active.
Record isn't active.
```

this means that your driver is ready to go, but is not doing anything right now. (So everything went well).

There is a third method to find information about the sound devices, namely if you inserted the OSS compatible driver there is a /dev/sndstat device. The ALSA drivers kindly request that you not to rely on this information as it is only there for compatibility with the OSS drivers and better information can easily be obtained from /proc/asound/.

5.2 The mixer

Once the drivers for your sound card have been installed and your /proc filesystem tells you so, you can try to make a real sound. First of all, install the utility package, or at least put the "amixer" command in some reasonable place (like /usr/local/bin). First look at the mixer settings by typing "amixer". The output from amixer can greatly differ from card to card. My Soundblaster 16 shows:

```
Master          0  % (-14.00dB) : 0  % (-14.00dB)
Bass            0  % (-14.00dB) : 0  % (-14.00dB)
Treble          0  % (-14.00dB) : 0  % (-14.00dB)
Synth           0  % (-62.00dB) : 0  % (-62.00dB)
PCM             0  % (-62.00dB) : 0  % (-62.00dB)
Line-In         0  % (-62.00dB) : 0  % (-62.00dB) Mute
MIC             0  % (-62.00dB) : 0  % (-62.00dB) Mute
CD              0  % (-62.00dB) : 0  % (-62.00dB) Mute
In-Gain         0  % (-18.00dB) : 0  % (-18.00dB)
Out-Gain        0  % (-18.00dB) : 0  % (-18.00dB)
PC Speaker      0  % (-18.00dB) : 0  % (-18.00dB)
```

My Crystal 4237B based soundcard has a lot of other options:

```
Master D        0  % (-22.00dB) : 0  % (-22.00dB) Mute
3D Center       0  % (-22.50dB) : 0  % (-22.50dB)
3D Space        0  % (-22.50dB) : 0  % (-22.50dB) Mute
Synth           0  % (-94.50dB) : 0  % (-94.50dB) Mute
FM              0  % (-94.50dB) : 0  % (-94.50dB) Mute
DSP             0  % (-94.50dB) : 0  % (-94.50dB) Mute
PCM             0  % (-94.50dB) : 0  % (-94.50dB) Mute
Line-In         0  % (-34.50dB) : 0  % (-34.50dB) Mute
MIC             0  % (-22.50dB) : 0  % (-22.50dB) Mute
CD              0  % (-34.50dB) : 0  % (-34.50dB) Mute
Record-Gain     0  % (  0.00dB) : 0  % (  0.00dB)
```

```
In-Gain        0  % (-18.00dB) : 0  % (-18.00dB)
Loopback       0  % (-94.50dB) : 0  % (-94.50dB) Mute
Mono           0  % (-45.00dB) : 0  % (-45.00dB) Mute
Aux A          0  % (-34.50dB) : 0  % (-34.50dB) Mute
```

TOC

You have noticed the "Mute" entry, the CS4237B even mutes the master channel. For the CS4237B, I would have to type `amixer "master d" unmute` to even be able to produce any sound at all. The Soundblaster does not have muted output, but `amixer master 100 unmute` would set the volume to 100% and unmute the master. You can use a number, a word like "mute" or "unmute", or both. Type `amixer "master d" 100; amixer pcm 100 unmute` to set the CS4237B card to maximum master volume and unmute PCM volume and set it to maximum. For separate L/R settings you use a colon, for example `amixer CD 25:50`. (No, I don't know which one is the left or right channel, but let me emphasize that this also depends heavily on the position of your speakers)

You would set the CD channel to record by typing `amixer cd rec` and stop the recording setting again by typing `amixer cd norec`. If you would like to record something from the microphone, you would probably use `amixer record-gain 100; amixer mic 100 rec mute`. (Using the microphone input unmuted will produce loud high-pitched sound if your mic picks up its own signal from the speakers again). Unfortunately I have not been able to change the volume of the "3d center" and "3d space" settings with amixer. If anyone succeeds please let me know. You can use alsamixer for this job.

The ALSA FAQ says that it is possible to restore mixer settings with `cat <file> > /proc/asound/#/mixerC0D0`, where <file> was obtained from /proc/asound/#/mixerC0D0. I have not been able to reproduce this as my system complains about non-existing devices.

5.3 The /dev/snd/ devices

The alsa drivers have native sound-devices in the /dev/snd/ directory. If you have one card you might see the following devices:

```
/dev/snd/pcmC0D0 - the raw audio device for the card
/dev/snd/mixerC0D0 - the mixer for card 0
/dev/snd/controlC0D0 - the control device for card 0
```

The first number means the number of the soundcard, the second number (if any) is the number of the device. A sound card with two PCM devices would have a pcmC0D0 and pcmC0D1 device. Please note: the ALSA devices have changed between the previous version. Older ALSA drivers use /dev/snd/pcm00 (first number is the card, second number is the device). If this HOWTO uses the older notation, please drop me a line so I can correct it.

Now you are ready to put any soundfile you want into the PCM device of the first card. So try to cat any textfile (any file) to /dev/snd/pcmC0D0, like this: `cat <filename> > /dev/snd/pcmC0D0`. The filename can be any file, as long as it has some length. If you have a soundfile lying around somewhere, you could try that. You could also get the file at http://www.ldp.org/sounds/english.au this is Linus Torvalds saying how to pronounce Linux.

The default setting of your sound device is 8000 Hz, 8 bit. That means that the "english.au" file mentioned above will produce speech, other test files will probably just produce noise. If you do not hear anything, check your speakers, try to run "amixer" again or consult a doctor. (Later on you can easily use the full 48 KHz, 16 bit features of your sound card, by using your favourite sound player like sox or mpg123).

If you loaded the "snd-pcm1-oss" module, you can also use the OSS-compatibility to access your sound card. The following mappings are made:

```
/dev/snd/pcmC0D0 -> /dev/audio0 (/dev/audio) -> minor 4
/dev/snd/pcmC0D0 -> /dev/dsp0 (/dev/dsp) -> minor 3
/dev/snd/pcmC0D1 -> /dev/adsp0 (/dev/adsp) -> minor 12
/dev/snd/pcmC1D0 -> /dev/audio1 -> minor 4+16 = 20
/dev/snd/pcmC1D0 -> /dev/dsp1 -> minor 3+16 = 19
/dev/snd/pcmC1D1 -> /dev/adsp1 -> minor 12+16 = 28
/dev/snd/pcmC2D0 -> /dev/audio2 -> minor 4+32 = 36
/dev/snd/pcmC2D0 -> /dev/dsp2 -> minor 3+32 = 39
/dev/snd/pcmC2D1 -> /dev/adsp2 -> minor 12+32 = 44
```

5.4 Additional information

TOC

The INSTALL file in the ALSA driver directory mentions some tricks to tell the driver which settings to use. If you need these commands it will depend on the application you use to play sound. Regular sound playing applications, like mpg123, sox (mostly called with the ``play'' command), or X11 applications like RealPlayer will probably do fine without these. I never used these anyway.

/proc/asound/#/pcm#0

```
"Playback erase" - erase all additional informations about OSS applications
"Playback <app_name> <fragments> <fragment_size> [<options>]"
"Record erase" - erase all additional informations about OSS applications
"Record <app_name> <fragments> <fragment_size> [<options>]"
```

`<app_name>` - name of application with (highter priority) or without path

`<fragments>` - number of fragments or zero if auto

`<fragment_size>` - size of fragment in bytes or zero if auto

`<options>` - optional parameters

`WR_ONLY` - if application tries open pcm device with O_RDWR driver rewrites this to O_WRONLY (playback) - good for Quake etc...

Examples:

```
echo "Playback x11amp 128 16384" > /proc/asound/0/pcm0o
echo "Playback squake 0 0 WR_ONLY" > /proc/asound/0/pcm0o
```

/proc/asound/#card#/sb16

```
"Playback 8" -> driver will use always 8-bit DMA channel for playback.
"Playback 16" -> driver will use always 16-bit DMA channel for playback.
"Playback auto" (default) -> driver will use auto mode (first opened direction will
use 16-bit DMA channel).
"Record 8" -> driver will use always 8-bit DMA channel for record.
"Record 16" -> driver will use always 16-bit DMA channel for record.
"Record auto" (default) -> driver will use auto mode (first opened direction will use
16-bit DMA channel).
```

Example: `echo "Record 16" > /proc/asound/0/sb16`

For further reference, please consult the INSTALL file.

---

6. Tips and Troubleshooting

Please take a look at the FAQ file in the sound driver directory. This section is still under construction.

6.1 Compiling the driver

Linux kernel sourcetree

If your ALSA drivers do not compile correctly and tell you things about ``version.h" or other header-files that cannot be found, this can mean that you do not have the kernel header files. Take a look at the kernel-HOWTO, unpack a recent kernel in /usr/src and issue a `make config`.

6.2 Loading the driver

Please check the following items.

Sound card compatibility

Are you 100% sure that your sound card IS supported ? Do check it again. Sometimes an X123 is not exactly an X123b and you might be wasting time. On the other hand, even a supported card can give you troubles - it took me two hours to figure out the installation of a CS4237B which was, after all, just a fine example of RTFM.

``Device busy" or ``unresolved symbols"

You might have a 2.0.x kernel with sound support compiled in, or the OSS/Lite (kernel) sound driver could be loaded (check with `cat /proc/modules`). Remove the driver or recompile the kernel (have a look at the Kernel-HOWTO).

The sound module in the 2.0 series kernel is called ``sound.o" and should **not** be active. (The ALSA driver ``snd.o" is OK, though).

If you have a 2.2.x series kernel without sound driver compiled in, the ALSA drivers will not work, too.

I know it this is confusing, so let me try to explain it one more time. If you have a **2.0.x series kernel** (the command ``uname -a" tells you something like ``Linux penguin 2.0.35 #6 Wed Sep 23 10:19:16 CEST 1998 i686 unknown") then you need to **leave out** sound drivers in the kernel.

If you have a **2.2.x series kernel** you do **need the sound drivers**. A 2.2 series kernel should be compiled **with sound support, but without any sound card driver**. So you select sound support but make sure that no specific sound card driver will be compiled.

Unresolved symbols revisited

Another source of ``unresolved symbols" messages could be a new kernel with older drivers. Please recompile the ALSA drivers after you recompile a new kernel. This will make sure that the drivers match your new kernel.

Check the PnP setup

Are you sure that your card is active? Take another look at the PnP-HOWTO and check if you activated your sound card correctly.

TOC

Are your parameters right ?

Check, doublecheck your sound card parameters. Please note: 534 is not 543, nor is 0x534 the same as 534.

Also, some sound cards must be loaded by a different name than might be expected. Take a break, a beer or whatever, and look again at your ``modprobe'' command. For example the Crystal 4232 driver should be inserted by modprobe snd-card-cs4232, not ``snd-cs4231'', and the SoundBlaster PCI 64 should be loaded with ``snd-audiopci'', not snd-es1370. (It's all in the docs, and even though I wrote the HOWTO, I once spent an evening trying to persuade snd-cs4231 to make sound).

6.3 Driver loaded... but no sound

Unmuting

The ALSA drivers can use the ``muting'' facilities that most soundcards have. If you loaded the sound drivers and everything is fine but you get nothing but silence, then you probably forgot to unmute your card. You need ``amixer'' or ``alsamixer'' for this, both from the ALSA-util package. Just typing

```
amixer -c 1 master 70:70 unmute
amixer -c 1 pcm 70:70 unmute
amixer -c 1 cd 70:70 unmute
```

should do for most applications.

OSS/Linux compatibility

If this is the first time you use the ALSA drivers and you used the built-in sound drivers before, you probably want to have backwards compatible sound (i.e. use the /dev/pcmX devices). You need to load the ``OSS compatibility driver'' for this. Do a `modprobe snd-pcm1-oss`. (See the end of the section about loading the driver). Please note: snd-pcm1-oss is not equal to snd-pcm1, you **need** snd-pcm1-oss for old-fashioned sound support.

6.4 General suggestions

Try using ``insmod''

It can always be useful to start with "insmod" instead of kerneld. Maybe you actually see the error on screen.

Read the INSTALL file.

A lot of information can be found in the INSTALL file in the drivers directory. If your driver won't work check if there is additional information available.

Debug messages

As a last resort, you can rebuild the driver and tell it to send debug information to /var/log/messages. Go to the driver-directory with `cd /usr/src/alsa-driver-....` and type:

```
./configure --with-debug=detect; make clean; make
```

Remove the driver (as far as it is active, see below for a general remove statement). Then use the "modprobe" statement you used before to insert the newly compiled driver. Look in /var/log/messages if there are any messages.

TOC

If all else fails...

If these messages doesn't help you, send a message to the ALSA users mailing list, alsa-user@alsa-project.org.

Include the following information:

- soundcard name + chip names present on your soundcard
- relevant sections in your isapnp.conf if you have ISA PnP soundcard
- your conf.modules or line which you activate ALSA driver
- all messages from /var/log/messages which should be relevant to the ALSA driver

6.5 Bug reports

If you found a bug, the ALSA developers would like to know the following things (at minimum)

1. driver + kernel version: 'cat /proc/asound/version'
2. soundcard info
   - soundcard name provided by manufacture
   - list of chips which soundcard have onboard
   - contents of 'cat /proc/asound/cards'
3. all messages from /var/log/messages which should be relevant to ALSA driver
4. problem description

6.6 Tip: playing CD's

If you use kmod/kerneld and the ALSA drivers to play CD's, then kmod/kerneld probably do not load the drivers as expected. This is due to the fact that a command line CD player only tells the CD player to start playing without using any of the devices that tell kmod/kerneld that there is sound to occur. Using modprobe may be your only solution to this problem.

6.7 Tip: installing the MIDI serial driver

Normally, the IO port of the serial device is owned by the standard serial device driver. So before you can do ``modprobe snd-serial'' we have to tell the driver to release the serial device.

Here is the procedure.

```
setserial /dev/ttyS0 uart none
modprobe snd-serial
```

(Replace /dev/ttyS0 with the appropriate /dev/ttySx device if your MIDI device uses a different serial device).

6.8 Tip: new kernel? New modules!

After you upgrade your kernel, you probably need to recompile the ALSA drivers. If they are still in the original /usr/src directory, then please do not forget to issue a `make clean` before you do the ./configure, make, make install thing.

Oh, and then there is this anomaly in kernel numbering: a ``2.2.0ac1'' kernel that is ``not a number'' - says the configure script. I think this was resolved in newer scripts, otherwise you should maybe change the kernel version in the source.

TOC

6.9 Tip: KDE and ALSA drivers

Suppose you have KDE up and running but you cannot get system sounds to work, like for opening windows, changing desktops, etc. Sound works in general. If your cd player and mp3 player and mixer all do work, then it's probably just "kwmsound" that's lacking.

So: make sure "kwmsound" is in your startscript ($KDEDIR/bin/startkde)

6.10 Tip: use the ALSA devices

If you had sound support in your Linux before, then your applications will probably all point to /dev/pcm0, /dev/audio and /dev/mixer. This is fine, if you use OSS compatibility with the snd-pcm1-oss module. It might be better, however, to use the real ALSA devices, those found in /dev/snd/.

6.11 Tip: removing all modules

Removing 10+ modules one by one is not the way to go. Luckily, all modules start with the "snd-" prefix, so a little command line programming will do. You can easily remove ALSA sound by issuing a command like:

```
cat /proc/modules|gawk '/^snd-/{print $1}|xargs -i rmmod {}
```

Juergen Kahrs wrote: ``I have a script that also removes soundcore and soundlow and sound if present and if they are not in use. This script processes /proc/modules three times so there should not be too many modules left after processing". His solution is

```
awk '/^snd/||/^sound/&&($3==0){system("rmmod " $1)}' /proc/modules /proc/modules
/proc/modules
```

Please note: if some module is dependent on another module you cannot just remove the "higher" one. This means that you might need to issue a second removal statement. (I never encountered this situation though, it seems that you can remove the ALSA modules in the order they appear in /proc/modules).

Linux Modules Installation mini-HowTo

TOC

Contents

- Purpose of this document
- WARNING
- Pre-requisites
- Recompiling the kernel for modules
- Configuring Debian or RedHat for modules
- Configuring Slackware for modules
- Configuring other distributions for modules
- Copyright and other Legalities

Purpose of this document

My experience with Linux and modules has been that the existing documents fail to provide a satisfactory explanation as to how to successfully set up Linux with modules configured and working. The procedure explained in this document has been successfully used several times, both on my own system and over the Internet to give directions to somebody trying to get some feature to work which requires a driver supplied only in module form.

My own system runs from a RedHat 4.1 distribution of Linux, and it was on this setup that I developed the procedure. I have since successfully installed it on systems running from various Slackware distributions, and on one system running from a Debian distribution, and the necessary procedure to correctly configure modules under Linux in all three is documented herein.

WARNING

I have recently used the same procedure with RedHat 4.2, but with inconsistant results on apparently identical systems. I have NOT yet determined what the problem is, so can make NO guarantees at this stage as to whether or not it will work on your system.

Pre-requisites

TOC

- Before the steps in this document can be applied, the reader MUST have a working Linux installation in which one can get to the Linux prompt as user **root** since the majority of the steps involved can only be undertaken by the said user.
- The existing kernel may be compiled either to use modules or not to use modules, and can even display error messages during the boot-up procedure as a result of modules being configured which aren't available at the moment, providing the above condition is met.
- The source tree for the current kernel is assumed to be found rooted at **/usr/src/linux** and that is also assumed to be the current directory throughout this document at the start of any sequence of commands to be issued.

Compiler Speed-up

If your machine has 16 or more Megabytes of RAM, there is a useful speed-up that can be done, which is to permit the kernel to compile two or modules in parallel. This will increase the load on the machine whilst the kernel is being recompiled, but will reduce the time during which the compilation will be taking place.

Before you can use this method, you need to check the amount of RAM present in your machine, as if you set this too high, the compilation will actually slow down. Experience has shown that the optimum value depends on the amount of RAM in your system according to the following formula, at least for systems with up to 32 Megabytes of RAM, although it may be a little conservative for systems with larger amounts of RAM:

$$N = [\text{RAM in Megabytes}] / 8 + 1$$

For the benefit of those with a dislike of maths, the values for the common amounts of RAM are as follows:

RAM size	Value to use
16 Megs	3
24 Megs	4
32 Megs	5
40 Megs	6
48 Megs	7
56 Megs	8
64 Megs	9
80 Megs	11
96 Megs	13
112 Megs	15

128 Megs	17

When you have decided on the correct number, edit the file **/usr/src/linux/Makefile** and find the line that currently reads:

```
MAKE=make
```

Replace it with one reading:

```
MAKE=make -j N
```

where **N** is the number determined above.

Recompiling the kernel for modules

The kernel can be reconfigured to use modules for everything other than the file system mounted as root (in most cases, this is the ext2 file system).

However, there are certain items that appear to be difficult to set up properly as modules, so I would recommend the following be compiled into the kernel:

- Ethernet hardware drivers.
- SCSI CD-ROM drivers.

On the other hand, there are certain driver combinations that **ONLY** work as modules, especially combinations of two or more of the following group:

- A Parallel Printer,
- A Parallel Port drive, such as the **IOMEGA** ZipDrive or JazzDrive, or the **BackPack** CD-ROM drive, and
- The **PLIP** Daemon.

You will need to decide what you are compiling into the kernel, and what as modules, but should take the above points into consideration. The actual choices are made during the compilation, by the second of the following sequence of instructions:

```
cd /usr/src/linux
make menuconfig
make dep clean modules modules_install zImage
```

Having done that, the module dependencies need to be mapped out. This is done with the following command:

```
depmod -a
```

The new kernel now needs to be inserted in the boot chain. I am assuming the reader is using LILO for

this purpose, since this is the only loader I have any experience with.

I recommend that one does NOT automatically insert the newly compiled kernel as the default Linux kernel since if it should fail, it is then extremely difficult to recover one's Linux setup without doing a complete reinstallation, which is not to be recommended. For this reason, I have the following entry in my **/etc/lilo.conf** file:

```
image=/usr/src/linux/arch/i386/boot/zImage
   label=new
   alias=n
   read-only
   vga=ask
   optional
```

This entry says that there is an OPTIONAL boot option (which will be ignored if the image in question does not exist) which boots the file **/boot/newlinux** if selected, and allows one to select the video mode it is to be booted in.

Assuming the existence of the above entry in **/etc/lilo.conf** the revised kernel is already correctly located at the end of compilation, and it can be installed via the following command:

```
lilo
```

Having done that, the reader needs to follow the further steps relevant to their selected distribution, as follows:

- Configuring Debian or RedHat for modules
- Configuring Slackware for modules
- Configuring other distributions for modules

Configuring Debian or RedHat for Modules

Prior to carrying out the steps listed here, the steps listed in Recompiling the kernel for modules are assumed to have been carried out.

The Debian and RedHat distributions have identical boot procedures, so also have identical procedures for configuring modules into them.

1. Having logged in as root, use your favourite text editor to create a new file called **/etc/rc.d/init.d/modules.init** with the following contents therein:

```
# Modules initialisation.
#
# Start up the module auto-loading daemon.
/sbin/kerneld
```

TOC

```
# Mount all currently unmounted auto-mounted partitions.
/sbin/mount -a
```

2. Having created the above file, perform the following steps whilst logged on as root:

```
cd /etc/rc.d
chmod 755 init.d/*
cd rc3.d
ln -s ../init.d/modules.init 05modules.init
```

The system can now be rebooted, and on doing so, it will be found that modules are fully implemented

---

Configuring Slackware for Modules

Prior to carrying out the steps listed here, the steps listed in Recompiling the kernel for modules are assumed to have been carried out.

1. The file **/etc/rc.d/rc.M** needs to be edited as follows:
    1. Around line 18, there is a section reading as follows:

    ```
    # Screen blanks after 15 minutes idle time.
    /bin/setterm -blank 15
    ```

    Immediately after this, insert the following paragraph, with the usual blank lines either side of it:

    ```
    # Load the kernel module auto-loader.
    /sbin/kerneld
    ```

    2. About 12 lines further down is the following:

    ```
    # if there is no /etc/HOSTNAME, fall back on this default:
    ```

    Immediately prior to this, insert the following paragraph, again with the usual blank lines either side of it:

    ```
    # Mount remaining unmounted auto-mount drives.
    /sbin/mount -a
    ```

    When those changes have been made, save the file.

No further modifications are required for Slackware.

Configuring other distributions for Modules

TOC

Prior to carrying out the steps listed here, the steps listed in Recompiling the kernel for modules are assumed to have been carried out.

The precice procedure for other distributions has not been ascertained, but is probably one of the above. To determine which one, display a directory of the contents of the **/etc/rc.d** directory, as follows:

```
cd /etc/rc.d
ls -l *.d rc.*
```

From this resulting display, you can select one of the following three options:

1. If this list includes a directory named **init.d** and some directories with names matching **rc?.d** where the question mark is replaced by single digits, and does **NOT** include a file with the name **rc.M**, that distribution can be configured for modules by following the procedure listed above for the Debian and RedHat Distributions.
2. If this list does not include a directory named **init.d** but includes a file named **rc.M** then that distribution can be configured for modules by following the procedure listed above for the Slackware distribution.
3. If this list matches neither of the above criteria, then the distribution has a boot script not covered by this HowTo. In that case, you are invited to contact the author of this document for advice.

Copyright and other Legalities

This document is covered by the terms of the **GNU General Public Licence (GPL)**, and all terms and limitations therein apply.

The author may be contacted by email at rhw@bigfoot.com.

kerneld mini-HOWTO

Version 1.7, last updated: July 19, 1997 by Henrik Storner (storner@osiris.ping.dk)

Introduction

This document explains how you can use the kerneld function in the Linux kernels. It describes

- what is kerneld
- why do I want to use it
- how to get the necessary pieces
- how to set them up
- how to tell kerneld about modules it doesn't know
- how to spy on kerneld (can be useful in setting it up).
- special kerneld uses
- Common problems and weirdness

The latest released version of this document can be found at http://eolicom.olicom.dk/~storner/kerneld-mini-HOWTO.html. Between releases of the mini-HOWTO you can find updates on my unstructured list of changes at http://eolicom.olicom.dk/~storner/kern.html

Credits

If you see things in this document that are wrong, please send me a note about it. The following people have contributed to this mini-HOWTO at some point:

- Bjorn Ekwall <bj0rn@blox.se>
- Ben Galliart <bgallia@luc.edu>
- Cedric Tefft <cedric@earthling.net>
- Brian Miller <bmiller@netspace.net.au>
- James C. Tsiao <jtsiao@madoka.jpl.nasa.gov>

I much appreciate the encouragement and suggestions sent to me by readers of the mini-HOWTO.

What is kerneld ?

kerneld is a feature introduced during the 1.3 development kernels by Bjorn Ekwall. It is included with all of the 2.0 and 2.1 kernels. It allows kernel "modules" - i.e. device drivers, network drivers, filesystems - to be loaded automatically when they are needed, rather than having to do it manually with modprobe or insmod.

And for the more amusing aspects, although these are not (yet ?) integrated with the standard kernel:

- It can be setup to run a user-program instead of the default screen blanker, thus letting you use any

program as a screen-saver.

- Similar to the screen-blanker support, you can also change the standard console "beep" into something completely different ...

kerneld consists of two separate entities:

- Support in the Linux kernel for sending requests to a daemon that a module is needed for a certain task.
- A user-level daemon that can figure out what modules must be loaded to fulfill the request from the kernel.

Both pieces must be working for the kerneld support to function - it is not enough that only one or the other has been setup.

Why do I want to use it ?

There are some good reasons for using kerneld. The ones I will mention are mine - others may want to use it for other reasons.

- If you have to build kernels for several systems that only differ slightly - different kind of network card, for instance - then you can build a single kernel and some modules, instead of having to build individual kernels for each system.
- Modules are easier for developers to test - you don't need to reboot the system to load and unload the driver. (This applies to all modules, not just kerneld-loaded ones).
- It cuts down on the kernel memory usage, meaning you have more memory available for applications. Memory used by the kernel is *never* swapped out, so if you have 100Kb worth of unused drivers compiled into your kernel, they are simply wasting RAM.
- Some of the things I use - the ftape floppy-tape driver, for instance, or iBCS - are only available as modules. But I don't want to bother with loading and unloading them whenever I need them.
- People making Linux distributions don't have to build 284 different boot images - each user loads the drivers he needs for just his hardware. This is used e.g. by RedHat 4.0 in their installation.

Of course, there are also reasons why you may not want to use it - you may prefer to have just one kernel image file with all of your drivers built in. In that case, you are reading the wrong document.

Where can I pick up the necessary pieces ?

The support in the Linux kernel was introduced with Linux 1.3.57. If you have an earlier kernel version, you will need to upgrade if you want the kerneld support. All of the major Linux ftp sites carry the kernel sources - I recommend that you upgrade to the latest stable kernel release, 2.0, now at patch-level 29:

```
ftp://sunsite.unc.edu/pub/Linux/kernel/v2.0/linux-2.0.29.tar.gz
ftp://tsx-11.mit.edu/pub/linux/sources/system/v2.0/linux-2.0.29.tar.gz
ftp://ftp.funet.fi/pub/Linux/PEOPLE/Linus/v2.0/linux-2.0.29.tar.gz
```

The user-level daemon is included with the modules-1.2.8 package, and with the newer modules-2.0 package. These are normally available from the same place as the kernel sources, but the official locations include:

```
ftp://sunsite.unc.edu/pub/Linux/kernel/v2.0/modules-2.0.0.tar.gz
ftp://tsx-11.mit.edu/pub/linux/sources/sbin/modules-2.0.0.tar.gz
```

ftp://ftp.funet.fi/pub/Linux/tools/modules-2.0.0.tar.gz

TOC

NOTE: If you want to try module-loading with the latest 2.1 **development** kernels, you should use the latest modutils- (NOT modules-) package. But see below about the problems with modules and 2.1 kernels.

How do I set it up ?

First get the necessary parts: A suitable kernel and the latest modules-utilities. Then you should install the modules-utilities. Pretty simple - just unpack the sources and run **make install**. This compiles and installs the following programs in /sbin: genksysm, insmod, lsmod, modprobe, depmod, kerneld. I recommend that you add some lines to your startup-scripts to do some necessary setup whenever you boot Linux. Add the following lines to your /etc/rc.d/rc.S file (if you are running Slackware), or to /etc/rc.d/rc.sysinit (if you are running SysVinit, i.e. Debian, RedHat, Caldera):

```
# Start kerneld - this should happen very early in the
# boot process, certainly BEFORE you run fsck on filesystems
# that might need to have disk drivers autoloaded
if [ -x /sbin/kerneld ]
then
        /sbin/kerneld
fi

# Your standard fsck commands go here
# And you mount command to mount the root fs read-write

# Update kernel-module dependencies file
# Your root-fs MUST be mounted read-write by now
if [ -x /sbin/depmod ]
then
        /sbin/depmod -a
fi
```

The first part starts kerneld itself.

The second part calls 'depmod -a' at startup. The depmod program builds a list of all available modules and analyzes their inter-dependencies, so it knows if one module needs to have another loaded before it will itself load.

NOTE: Recent versions of kerneld as an option links with the GNU dbm library, libgdbm. If you enable this when building the module-utilities, *kerneld will not start if libgdbm is not available* which may well be the case if you have /usr on a separate partition and start kerneld before /usr is mounted. The recommended solution is to move libgdbm from /usr/lib to /lib, or link kerneld statically.

Next, unpack the kernel sources, configure and build a kernel to your liking. If you have never done this before, you should definitely read the README file at the top level of the Linux sources. When you run **make config** to configure the kernel, you should pay attention to some questions that appear early on:

```
Enable loadable module support (CONFIG_MODULES) [Y/n/?] Y
```

You need to select the loadable module support, or there will be no modules for kerneld to load! Just say Yes.

```
Kernel daemon support (CONFIG_KERNELD) [Y/n/?] Y
```

This, of course, is also necessary. Then, a lot of the things in the kernel can be built as modules - you will see questions like

```
Normal floppy disk support (CONFIG_BLK_DEV_FD) [M/n/y/?]
```

where you can answer with an 'M' for 'Module'. Generally, only the drivers necessary for you to boot up your system - the harddisk driver, the driver for the root filesystem - should be built into the kernel; the rest can be built as modules.

When you have gone through the 'make config', run 'make dep', 'make clean', 'make zImage' or 'make zlilo', 'make modules' and 'make modules_install'.

Phew.

The 'make zImage' puts the new kernel image in the file arch/i386/boot/zImage. You will need to copy it where you keep your boot-image, or install it in LILO afterwards.

For more information about configuring, building and installing your own kernel, check out the Kernel-HOWTO posted regularly to comp.os.linux.answers, and available from sunsite.unc.edu in /pub/Linux/docs/HOWTO .

Trying out kerneld

Now reboot with the new kernel. When the system comes back up, you can run a 'ps -ax', and you should see a line for kerneld:

```
   PID TTY STAT  TIME COMMAND
    59  ?  S     0:01 /sbin/kerneld
```

One of the nice things with kerneld is that once you have the kernel and the daemon installed, very little setup is needed. For a start, try using one of the drivers that you built as a module - it is more likely than not that it will work without further configuration. I build the floppy driver as a module, so I could put a DOS floppy in the drive and

```
osiris:~ $ mdir a:
 Volume in drive A has no label
 Volume Serial Number is 2E2B-1102
 Directory for A:/

binuti~1 gz       1942 02-14-1996  11:35a binutils-2.6.0.6-2.6.0.7.diff.gz
libc-5~1 gz      24747 02-14-1996  11:35a libc-5.3.4-5.3.5.diff.gz
        2 file(s)        26689 bytes
```

So the floppy driver works - it gets loaded automatically by kerneld when I try to use the floppy disk.

To see that the floppy module is indeed loaded, you can run /sbin/lsmod which lists all currently loaded modules:

```
osiris:~ $ /sbin/lsmod
Module:        #pages:  Used by:
floppy            11    0 (autoclean)
```

The "(autoclean)" means that the module will automatically be removed by kerneld when it has not been used for more than one minute. So the 11 pages of memory (= 44kB, one page is 4 kB) will only be used while I access

the floppy drive - if I don't use the floppy for more than a minute, they are freed. Quite nice, if you are short memory for your applications!

How does kerneld know what module to load ?

Although kerneld comes with builtin knowledge about the most common types of modules, there are situations where kerneld will not know how to handle a request from the kernel. This is the case with things like CD-ROM drivers or network drivers, where there are more than one possible module that can be loaded.

The requests that the kerneld daemon gets from the kernel is for one of the following items:

- a block-device driver
- a character-device driver
- a binary format
- a tty line discipline
- a filesystem
- a network device
- a network service (e.g. rarp)
- a network protocol (e.g. IPX)

kerneld determines what module should be loaded by scanning the configuration file **/etc/conf.modules** There are two kinds of entries in this file: Paths (where the module-files are located), and aliases (what module should be loaded). If you don't have this file already, you could create it by running

```
/sbin/modprobe -c | grep -v '^path' >/etc/conf.modules
```

If you want to add yet another "path" directive to the default paths, you must include all the "default" paths as well, since a path directive in /etc/conf.modules will replace all the ones that modprobe knows by default!

Normally you don't want to add any paths by your own, since the built-in set should take care of all "normal" setups (and then some...), I promise!

On the other hand, if you just want to add an alias or an option directive, your new entries in /etc/conf.modules will be _added_ to the ones that modprobe already knows. If you should _redefine_ an alias or an option, your new entries in /etc/conf.modules will override the built-in ones.

Block devices

If you run '/sbin/modprobe -c', you will get a listing of the modules that kerneld knows about, and what requests they correspond to. For instance, the request that ends up loading the floppy driver is for the block-device that has major number 2:

```
osiris:~ $ /sbin/modprobe -c | grep floppy
alias block-major-2 floppy
```

Why block-major-2 ? Because the floppy devices /dev/fd* use major device 2 and are block devices:

```
osiris:~ $ ls -l /dev/fd0 /dev/fd1
brw-rw-rw-   1 root     root       2,   0 Mar  3  1995 /dev/fd0
brw-r--r--   1 root     root       2,   1 Mar  3  1995 /dev/fd1
```

TOC

Character devices

Character devices are dealt with in a similar way. E.g. the ftape floppy tape driver sits on major-device 27:

```
osiris:~ $ ls -lL /dev/ftape
crw-rw----   1 root     disk      27,   0 Jul 18  1994 /dev/ftape
```

However, kerneld does not by default know about the ftape driver - it is not listed in the output from '/sbin/modprobe -c'.

So to setup kerneld to load the ftape driver, I must add a line to the kerneld configuration file, /etc/conf.modules:

```
alias char-major-27 ftape
```

Network devices

You can also use the device name instead of the 'char-major-xxx' / 'block-major-yyy' setup. This is especially useful for network drivers. E.g. a driver for an ne2000 netcard acting as eth0 would be loaded with

```
alias eth0 ne
```

If you need to pass some options to the driver - e.g. to tell the module about what IRQ the netcard is using, you add an 'options' line:

```
options ne irq=5
```

This will cause kerneld to load the NE2000 driver with the command

```
/sbin/modprobe ne irq=5
```

Of course, the actual options available are specific to the module you are loading.

Binary formats

Binary formats are handled in a similar way. Whenever you try to run a program that the kernel does not know how to load, kerneld gets a request for "binfmt-xxx", where xxx is a number determined from the first few bytes of the executable. So, the kerneld configuration to support loading the binfmt_aout module for ZMAGIC (a.out) executables is

```
alias binfmt-267 binfmt_aout
```

since the magic number (see /etc/magic) for ZMAGIC files is 267. (If you check /etc/magic, you will see the number 0413, but /etc/magic uses octal numbers where kerneld uses decimal, and octal 413 = decimal 267). There are actually three slightly different variants of a.out executables (NMAGIC, QMAGIC and ZMAGIC), so for full support of the binfmt_aout module we need

```
alias binfmt-264 binfmt_aout  # pure executable (NMAGIC)
alias binfmt-267 binfmt_aout  # demand-paged executable (ZMAGIC)
alias binfmt-204 binfmt_aout  # demand-paged executable (QMAGIC)
```

a.out, Java and iBCS binary formats are recognized automatically by kerneld, without any configuration.

Line disciplines (slip, cslip and ppp)

Line disciplines are requested with "tty-ldisc-x", with 'x' being usually 1 (for SLIP) or 3 (for PPP). Both of these are known by kerneld automatically.

Speaking of ppp, if you want kerneld to load the bsd_comp data compression module for ppp, then you must add the following two lines to your /etc/conf.modules:

```
alias tty-ldisc-3 bsd_comp
alias ppp0 bsd_comp
```

Network protocol families (IPX, AppleTalk, AX.25)

Some network protocols can be loaded as modules as well. The kernel asks kerneld for a protocol family (e.g. IPX) with a request for "net-pf-X" where X is a number indicating what family is wanted. E.g. net-pf-3 is AX.25, net-pf-4 is IPX and net-pf-5 is AppleTalk. (These numbers are determined by the AF_AX25, AF_IPX etc. definitions in the linux source file include/linux/socket.h). So to autoload the IPX module, you would need an entr like this in /etc/conf.modules:

```
alias net-pf-4 ipx
```

See also the section below on common problems for information about how you can avoid some annoying boot-time messages related to undefined protocol families.

File systems

kerneld requests for filesystems are simply the name of the filesystem type. A common use of this would be to load the isofs module for CD-ROM filesystems, i.e. filesystems of type "iso9660":

```
alias iso9660 isofs
```

Devices requiring special configuration

Some devices require a bit on configuration beyond the normal aliasing of e.g. a device to a module.

- Character devices on major number 10: The miscellaneous devices
- SCSI devices
- Devices that require special initialization

char-major-10 : Mice, watchdogs and randomness

Hardware devices are usually identified through their major device numbers, e.g. ftape is char-major-27. However, if you look through the entries in /dev for char major 10, you will see that this is a bunch of very different devices, including

- Mice of various sorts (bus mice, PS/2 mice)
- Watchdog devices
- The kernel 'random' device

- APM (Advanced Power Management) interface

Obviously, these devices are controlled by several different modules, not a single one. Therefore, the kerneld configuration for these **misc. devices** use the major number **and** the minor number:

```
alias char-major-10-1 psaux     # For PS/2 mouse
alias char-major-10-130 wdt     # For WDT watchdog
```

You need a kernel version 1.3.82 or later to use this; earlier versions do not pass the minor number to kerneld, making it impossible for kerneld to figure out which of the misc. device modules to load.

Loading SCSI drivers: The scsi_hostadapter entry

Drivers for SCSI devices consist of a driver for the SCSI host adapter (e.g. an Adaptec 1542), and a driver for the type of SCSI device you use, e.g. a hard disk, a CD-ROM or a tape-drive. All of these can be loaded as modules. However, when you want to access e.g. the CD-ROM drive that is connected to the Adaptec card, the kernel and kerneld only knows that it needs to load the sr_mod module in order to support SCSI CD-ROM's - it does not know what SCSI controller the CD-ROM is connected to, and hence does not know what module to load to support the SCSI controller.

To resolve this, you can add an entry for the SCSI driver module to your /etc/conf.modules that tells kerneld which of the many possible SCSI controller modules it should load:

```
alias scd0 sr_mod               # sr_mod for SCSI CD-ROM's ...
alias scsi_hostadapter aha1542  # ... need the Adaptec driver
```

This only works with kernel version 1.3.82 or later.

This works if you have only one SCSI controller. If you have more than one, things become a little more difficult.

In general, you cannot have kerneld load a driver for a SCSI host adapter, if a driver for another host adapter is already installed - you must either build both drivers into your kernel (not as modules), or load the modules manually.

There *is* a way that you can have kerneld load multiple SCSI drivers. James Tsiao came up with this idea:

> *You can easily have kerneld load the second scsi driver by setting up the dependency in your modules.dep by hand. You just need an entry like:*
>
> */lib/modules/2.0.30/scsi/st.o: /lib/modules/2.0.30/scsi/aha1542.o*
>
> *To have kerneld load the aha1542.o before it loads st.o. My machine at home is set up almost exactly like the setup above, and it works fine for all my secondary scsi devices, including tape, cd-rom, and generic scsi devices. The drawback is that 'depmod -a' can't autodetect these dependencies, so the user needs to add them by hand, and not run 'depmod -a' on boot up. But once it is set up, kerneld will autoload the aha1542.o just fine.*

You should be aware, that this technique only works if you have different kinds of SCSI devices on the two controllers - e.g. hard disks on one controller, and cd-rom drives, tapes or generic SCSI devices on another.

TOC

When loading a module isn't enough: The 'post-install' entry

Sometimes, just loading the module is not enough to get things working. For instance, if you have your soundcard compiled as a module, it is often convenient to set a certain volume level. Only problem is, the setting vanishes the next time the module is loaded. Here is a neat trick from Ben Galliart (bgallia@luc.edu):

```
The final solution required installing the setmix-0.1 package
( ftp://sunsite.unc.edu/pub/Linux/apps/sound/mixers/setmix-0.1.tar.gz )

And then adding the following lines to my  /etc/conf.modules :
    post-install sound /usr/local/bin/setmix -f /etc/volume.conf
```

What this does is that after the sound module is loaded, kerneld runs the command indicated by the 'post-install sound' entry. So the sound module gets configured with the command '/usr/local/bin/setmix -f /etc/volume.conf'.

This may be useful for other modules as well, e.g. the lp module can be configured with the tunelp program by adding

```
post-install lp tunelp <options>
```

For kerneld to recognize these options, you will need a version of kerneld that is 1.3.69f or later.

NOTE: An earlier version of this mini-HOWTO mentioned a "pre-remove" option, that might be used to run a command just before kerneld removed a module. However, this has never worked and its use is therefore discouraged - most likely, this option will disappear in a future kerneld release. The whole issue of module "settings" is undergoing some change at the moment, and may look different on your system by the time you read this.

Spying on kerneld

If you have tried everything, and just cannot figure out what the kernel is asking kerneld to do, there is a way of seeing the requests that kerneld receives, and hence to figure out what should go into /etc/conf.modules: The kdstat utility.

This nifty little program comes with the modules-package, but it is not compiled or installed by default. To build it:

```
cd /usr/src/modules-2.0.0/kerneld
make kdstat
```

Then, to make kerneld display information about what it is doing, run

```
kdstat debug
```

and kerneld will start spewing messages on the console about what it is doing. If you then try and run the command that you want to use, you will see the kerneld requests; these can be put into /etc/conf.modules and aliased to the module needed to get the job done.

To turn off the debugging, run '/sbin/kdstat nodebug' .

Special kerneld uses

TOC

I knew you would ask about how to setup the screensaver module ...

The 'kerneld/GOODIES' directory in modules-package has a couple of kernel patches for screensaver- and consolebeep-support in kerneld; these are not yet part of the official kernel. So you will need to install the kernel-patches and rebuild the kernel.

To install a patch, you use the "patch" command:

```
cd /usr/src/linux
patch -s -p1 </usr/src/modules-2.0.0/kerneld/GOODIES/blanker_patch
```

Then rebuild and install the new kernel.

When the screensaver triggers, kerneld will run the command "/sbin/screenblanker" - this may be a shell script that runs your favourite screensaver.

When the kernel wants to unblank the screen, it sends a SIGQUIT signal to the process running /sbin/screenblanker. Your shell script or screensaver should trap this, and terminate. Remember to restore the screen to the original text mode!

Common problems and things that make you wonder

Why do I get "Cannot locate module for net-pf-X" messages when I run ifconfig

Around kernel version 1.3.80, the networking code was changed to allow loading protocol families (e.g. IPX, AX.25 and AppleTalk) as modules. This caused the addition of a new kerneld request: net-pf-X, where X is a number identifying the protocol (see /usr/src/linux/include/linux/socket.h for the meaning of the various numbers).
Unfortunately, ifconfig accidentally triggers these messages, so a lot of people get a couple of messages logged when the system boots and runs ifconfig to setup the loopback device. The messages are harmless, and you can disable them by adding the lines

```
alias net-pf-3 off      # Forget AX.25
alias net-pf-4 off      # Forget IPX
alias net-pf-5 off      # Forget AppleTalk
```

to /etc/conf.modules. Of course, if you do use IPX as a module, you should not add a line to disable IPX.

After starting kerneld, my system slows to a crawl when I activate my ppp-connection

There have been a couple of reports of this. It seems to be an unfortunate interaction between kerneld and the **tkPPP script** that is used on some systems to setup and monitor the PPP connection - the script apparently runs loops while running ifconfig. This triggers kerneld, to look for the net-pf-X modules (see above), keeping the system load high and possibly pouring lots of "Cannot locate module for net-pf-X" messages into the system log. There is no known workaround, other than not use tkPPP, or change it to use some other way of monitoring the connection.

TOC

kerneld does not load my SCSI driver!

Add an entry for the SCSI hostadapter to your /etc/conf.modules. See the description of the scsi_hostadapter entry above.

modprobe complains about 'gcc2_compiled' being undefined

This is a bug in the module-utilities, that show up only with binutils 2.6.0.9 and later, and it is also documented in the releasenote for the binutils. So read that. Or fetch an upgrade to the module-utilities that fix this, e.g. modules-2.0.0.

My sound driver keeps forgetting its settings for volume etc

The settings for a module are stored inside the module itself when it is loaded. So when kerneld auto-unloads a module, any settings you have made are forgotten, and the next time the module loads it reverts to the default settings.

You can tell kerneld to configure a module by running a program after the module has been auto-loaded. See the section above on the 'post-install' entry.

DOSEMU needs some modules - how can I get kerneld to load those ?

You cannot. None of the dosemu versions - official or development versions - support loading the dosemu modules through kerneld. However, if you are running kernel 2.0.26 or later, you do not need the special dosemu modules any longer - just upgrade dosemu to 0.66.1.

Why do I get "Ouch, kerneld timed out, message failed" messages ?

When the kernel sends a request off to to kerneld, it expects to receive an acknowledgement back within one second. If kerneld does not send this acknowledgement, this message is logged. The request is retransmitted, and should get through eventually.

This usually happens on systems with a very high load - since kerneld is a user-mode proces, it is scheduled just like any other proces on the system. At times of high load, it may not get to run in time to send back the acknowledgement before the kernel times out.

If this happens even when the load is light, try restarting kerneld. (Kill the kerneld proces, and start it again with the command `/usr/sbin/kerneld`. If the problem persists, you should mail a bug report to linux-kernel@vger.rutgers.edu, but **please** make sure that your version of the kernel and kerneld is up-to-date before posting about the problem.

mount doesn't wait for kerneld to load the filesystem module

There has been a number of reports that the mount(8) command does not wait for kerneld to load the filesystem module. lsmod does show that kerneld loads the module, and if you repeat the mount command immediately it will succeed. This appears to be a bug in the module-utilities version 1.3.69f that affects some Debian users - it can be fixed by getting a later version of the module-utilities.

TOC

kerneld fails to load the ncpfs module

You need to compile the ncpfs utilities with -DHAVE_KERNELD. See the ncpfs Makefile.

kerneld fails to load the smbfs module

You are using an older version of the smbmount utilities. Get the latest version (0.10 or later) from ftp://tsx-11.mit.edu/pub/linux/filesystems/smbfs/

I built everything as modules, and now my system cannot boot

kerneld fails to load the root filesystem module

You cannot modularize **everything**: The kernel must have enough drivers built in for it to be able to mount your root filesystem, and run the necessary programs to start kerneld. So you cannot modularize

- the driver for the hard disk where your root filesystem lives
- the root filesystem driver itself
- the binary format loader for init, kerneld and other programs

[Actually, this is not true. Late 1.3.x and all 2.x kernels support the use of an initial ram-disk that is loaded by LILO or LOADLIN, and it is possible to load modules from this "disk" very early in the boot process. How to do it is described in the Documentation/initrd.txt file that comes with the kernel source-files.]

kerneld will not load at boot time - complains about libgdbm

Newer versions of kerneld need the GNU dbm library, libgdbm.so, to run. Most installations have this file in /usr/lib, but you are probably starting kerneld before the /usr filesystem is mounted. One symptom of this is that kerneld will not start during boot-up (from your rc-scripts), but runs fine if you start it by hand after that system is up. The solution is to either move the kerneld startup to after your /usr is mounted, or move the gdbm library to your root filesystem, e.g. to /lib.

I get "Cannot load module xxx" but I just reconfigured my kernel without xxx support!

The Slackware installation (possibly others) builds a default /etc/rc.d/rc.modules which does an explicit modprobe on a variety of modules. Exactly which modules get modprobed depends on the original kernel's configuration. You have probably reconfigured your kernel to exclude one or more of the modules that is getting modprobed in rc.modules, thus, the error message(s). Update your rc.modules by commenting out any modules you no longer use, or remove the rc.modules entirely and let kerneld load the modules when they are needed.

I rebuilt my kernel and modules, and still get messages about unresolved symbols when booting

You probably reconfigured/rebuilt your kernel and excluded some modules. You've got some old modules that you no longer use hanging around in the /lib/modules directory. The easiest fix is to delete your /lib/modules/x.y.z directory and do a 'make modules_install' from the kernel source directory again. Note that this problem only occurs when reconfiguring your kernel without changing versions. If you see this error when moving to a newer kernel version you've got some other problem.

TOC

I installed Linux 2.1 and now I cannot load ANY module

Linux 2.1 is the current development kernel. As such, it should be expected that things break from time to time. One of the things that has changed significantly is the way modules are handled, and where the kernel and modules are loaded into memory. Also, Richard Henderson is now in charge of the module kernel development.

In brief, if you want to use modules with a 2.1 kernel, you must

- read the Documentation/Changes file and see what packages need upgrading on your system
- use the latest modutils package, available from ftp://ftp.redhat.com/pub/alphabits/ or the mirror site at ftp://tsx-11.mit.edu/pub/linux/packages/alphabits/

I would recommend using at least kernel 2.1.29, if you want to use modules with a 2.1 kernel.

What about dial-on-demand networking?

kerneld originally had some support for establishing dial-up network connections on demand; trying to send packets to a network without being connected would cause kerneld to run the **/sbin/request_route** script to setup a PPP or SLIP connection.

This turned out to be a bad idea. Alan Cox of Linux networking fame wrote on the linux-kernel mailing list, that

```
The request-route stuff is obsolete, broken and not required [...]
Its also removed from 2.1.x trees.
```

Instead of using the request-route script and kerneld, I whole heartedly advise that you install Eric Schenk's diald package, available from http://www.dna.lth.se/~erics/diald.html

---

Copyright message

This document is Copyright (c) Henrik Storner, 1996, 1997.

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu via email.

Linux Apache SSL PHP/FI frontpage mini-HOWTO

Marcus Faure, marcus@faure.de

v1.1, July 1998

*This document is about building a **multipurpose webserver** that will support dynamic web content via the **PHP/FI** scripting language, secure transmission of data based on Netscape's **SSL**, secure execution of **CGI's** and M$ **Frontpage Server Extensions***

1. Introduction

- 1.1 Description of the components
- 1.2 Working configurations
- 1.3 History

2. Component installation

- 2.1 Preparations
- 2.2 Adding PHP
- 2.3 Adding SSL
- 2.4 Adding frontpage

3. Putting it all together

- 3.1 Apache modules to try
- 3.2 Giving CGI's more security
- 3.3 Compiling and installing the server daemon
- 3.4 Adding frontpage support to a web
- 3.5 Starting the daemon
- 3.6 Some considerations left

- 3.7 Known bugs
- 3.8 The final word

1. Introduction

Before you start reading: I am not a native speaker, so there are probably spelling/grammatical errors in this document. Feel encouraged to inform me of mistakes.

1.1 Description of the components

The webserver you hopefully will get after having read this howto is composed of several parts, the original apache sources with some (well, many) patches and some external executables. I recommend using the software versions I tried, they will probably compile without greater problems and result in a fairly stable daemon. If you are courageous, you can try to compile all the latest-stuff-with-tons-of-new-features, but don't blame me if something fails ;-). However, you may report other working configurations to be included in future versions of this document. All of the steps were tested on a linux 2.0.35 box, so the howto is somewhat linux-specific, but you should be able to use it for other unixes as well.

You do not necesserily have to compile in all components. I tried to structure this howto so that you can skip the parts you are not interested in.

The document is neither a user manual to Apache, SSL, PHP/FI nor frontpage. Its prime intention is to save webservice providers some headaches when installing their server and to do my little contribution to the linux community.

PHP is a scripting language that supports dynamic HTML pages. It is a bit like Apache's SSI, but by far more complex and has database modules for many popular dbs. The GD libraries are needed by PHP.

SSL is an implementation of Netscape's Secure Socket Layer that allow secure connections over insecure networks, e.g. to transmit credit card numbers to web based forms.

frontpage is a wysiwyg web authoring tool that makes use of some server-specific extensions called webbots. Some people think frontpage is cool because you can create feedback forms and discussion webs without having to know a bit about html or cgi. It even protects the designer from uploading his/her site via ftp by using a builtin publisher. If you wish to support frontpage but do not like to setup a windows server, the apache server extensions are your choice.

1.2 Working configurations

Though this document has been downloaded some 100 times since I published it, I received only little feedback. In particular, noone told me of other working combinations. Combinations that work for me are:

- Linux 2.0.31, Apache 1.2.4, PHP 2.0.0, SSL 0.8.0, fp 98 3.0.3 (*)
- Linux 2.0.33, Apache 1.2.5, PHP 2.0.1, SSL 0.8.0, fp 98 3.0.3 (*)

- Linux 2.0.35, Apache 1.2.6, PHP 3, SSL 0.8.0, fp 98 3.0.4

(*) version 3.0.3 is not recommended

1.3 History

v0.0/Apr 98: Preview version

v1.0/Jun 98: Now using Apache 1.2.6, updated fp section, minor corrections

v1.1/Jul 98: Sgmlized and restructered version

You can find the latest version of this document at http://www.faure.de

---

2. Component installation

2.1 Preparations

You will need:

- Apache 1.2.6 http://www.apache.org/dist/apache_1_2_6.tar.gz
- PHP/FI Extensions http://php.iquest.net/files/download.phtml?/files/php-2.01.tar.gz
- GD Library http://siva.cshl.org/gd/gd.html
- SSL 0.8.0 ftp://ftp.ox.ac.uk/pub/crypto/SSL/SSLeay-0.8.0.tar.gz
- SSL patch for Apache 1.2.6 ftp://ftp.ox.ac.uk/pub/crypto/SSL/apache_1.2.6+ssl_1.17.tar.gz
- frontpage 98 server extensions and install script http://www.rtr.com/fpsupport/download.htm

Get the sources you want. Untar apche, php, gd and ssl to `/usr/src`. Untar the SSL patch to `/usr/src/apache_1.2.6`.

2.2 Adding PHP

`cd` to /usr/src/gd1.2 and type make. This will build the GD library `libgd.a`, that should be copied to `/usr/lib`. Now `cd` to `php-2.0.1` and run `./install`.

The relevant questions are:

```
Would you like to compile PHP/FI as an Apache module? [yN] y
Are you compiling for an Apache 1.1 or later server? [Yn] y
Are you using Apache-Stronghold? [yN] y
Does your Apache server support ELF dynamic loading? [yN] y
Apache include directory (which has httpd.h)? [/usr/local/include/apache]
/usr/src/apache_1.2.6/src
Would you like to build an ELF shared library? [yN] y
Additional directories to search for .h files []: /usr/src/gd1.2
Would you like the bundled regex library? [yN] n
```

Like the frontpage extensions, phtml includes a security problem because it is run under the uid of the webserver. Be sure to turn on safe mode in src/php.h and restrict the search path to a save value. There are some other options in php.h you may want to edit. If you are very concerned about security, compile php as a cgi. However, this will be a performance loss and not as smart as the module version.

Type `make` to build all files. When the compilation is done, copy `mod_php.*` and `libphp.a` to `/usr/src/apache_1.2.6/src` Add a line

```
Module php_module mod_php.o
```

TOC

to the end of `/usr/src/apache_1.2.6/src/Configuration`, add

```
-lphp -lm -lgdbm -lgd
```

to the `EXTRA_LIBS` in the same file,

```
application/x-httpd-php phtml
```

to Apache's `mime.types` and

```
AddType  application/x-httpd-php .phtml
```

to Apache's `srm.conf`.

You may also want to add `index.phtml` to `DirectoryIndex` in that file so that a file index.phtml is automatically loaded when its directory is requested.

2.3 Adding SSL

`cd /usr/src/SSL-0.8.0; ./Configure linux-elf; make; make rehash` This will create libraries needed by apache. You may issue `make test` to verify the compilation. You have to apply a patch to apache. It is important that you apply it before the frontpage patch, otherwise frontpage will not work. `cd` to `/usr/src/apache_1.2.6/src` and issue `patch < /usr/src/apache_1.2.6/SSLpatch`. Set `SSL_BASE=/usr/src/SSLeay-0.8.0` in `Configuration`. Make sure that `Module proxy_module` is disabled otherwise Apache won't compile. If you are in need of a proxy, go for Squid squid.nlanr.net

Now `make certificate` to generate `SSLconf/conf/httpsd.pem`.

2.4 Adding frontpage

Rename the `fp30.linux.tar.Z` file to `fp30.linux.tar.gz`, otherwise the install script will not find it. Run `./fp_install` to copy the extension files to `/usr/local/frontpage`. zcat can usually be invoked as /usr/bin/zcat.

You now have to apply the FP patch. `cd` to `/usr/src/apache_1.2.6/src` and type `patch < /usr/src/frontpage/version3.0/apache-fp/fp-patch-apache_1.2.5` This will create the `mod_frontpage.*` files and do some modifications to `Configuration` etc. The 1.2.5 patch will work with both apache 1.2.5 and 1.2.6. Skip the part about installing webs, you can do that later

---

3. Putting it all together

3.1 Apache modules to try

The modules I use besides SSL, PHP and frontpage are:

```
Module env_module           mod_env.o
Module config_log_module    mod_log_config.o
Module mime_module          mod_mime.o
Module negotiation_module   mod_negotiation.o
Module dir_module           mod_dir.o
Module cgi_module           mod_cgi.o
Module asis_module          mod_asis.o
Module imap_module          mod_imap.o
Module action_module        mod_actions.o
Module alias_module         mod_alias.o
Module rewrite_module       mod_rewrite.o
Module access_module        mod_access.o
Module auth_module          mod_auth.o
Module anon_auth_module     mod_auth_anon.o
Module digest_module        mod_digest.o
Module expires_module       mod_expires.o
Module headers_module       mod_headers.o
Module browser_module       mod_browser.o
```

3.2 Giving CGI's more security

If you are an ISP (you probably are when you read this) you will want to improve security. The suexec utility allows you to do so; it will execute cgi's under the UID of the webowner instead of executing it under the webservers UID. Go to `/usr/src/apache_1.2.6/support` and `make suexec`. `chmod 4711 suxec` and copy it to the location specified in `../src/httpd.h` which is `/usr/local/etc/httpd/sbin/suexec` by default. If the path seems a little cryptic to you - it did to me - edit `httpd.h` and set the path to a more comfortable value.

3.3 Compiling and installing the server daemon

Enter `/usr/src/apache_1.2.6/src` and edit `Configuration` to set all the Modules you want to include in your Apache daemon. When done, run `./Configure` and `make`. This is the last (and most complicated) compilation step, so cross your fingers. If it succeeds, `cp httpsd` to `/usr/sbin`. The daemon is somewhat big, consider this when assembling your webserver. Create the directory

TOC

`/var/httpd` with subdirectories `cgi-bin`, `conf`, `htdocs`, `icons`, `virt1`, `virt2` and `logs`. `/usr/src/apache_1.2.6/conf` edit `access.conf-dist`, `mime.types` and `srm.conf-dist` to suit your needs and copy them to `var/httpd/conf/access.conf`, `srm.conf` and `mime.types`. Copy the `httpsd.pem` you created with `make certificate` to `/var/httpd/conf`. Use the following `httpd.conf`:

```
ServerType standalone
Port 80
Listen 80
Listen 443
User wwwrun
Group wwwrun
ServerAdmin webmaster@yourhost.com
ServerRoot /var/httpd
ErrorLog logs/error_log
TransferLog logs/access_log
PidFile logs/httpd.pid
ServerName www.yourhost.com
MinSpareServers 3
MaxSpareServers 20
StartServers 3

SSLCACertificatePath /var/httpd/conf
SSLCACertificateFile /var/httpd/conf/httpsd.pem
SSLCertificateFile /var/httpd/conf/httpsd.pem
SSLLogFile /var/httpd/logs/ssl.log

<VirtualHost www.virt1.com>
SSLDisable
ServerAdmin webmaster@virt1.com
DocumentRoot /var/httpd/virt1
ScriptAlias /cgi-bin/ /var/httpd/virt1/cgi-bin/
ServerName www.virt1.com
ErrorLog logs/virt1-error.log
TransferLog logs/virt1-access.log
User virt1admin
Group users
</VirtualHost>

<VirtualHost www.virt1.com:443>
ServerAdmin webmaster@virt1.com
DocumentRoot /var/httpd/virt1
ScriptAlias /cgi-bin/ /var/httpd/virt1/cgi-bin/
ServerName www.virt1.com
ErrorLog logs/virt1-ssl-error.log
```

```
TransferLog logs/virt1-ssl-access.log
User virt1admin
Group users
SSLCACertificatePath /var/httpd/conf
SSLCACertificateFile /var/httpd/conf/httpsd.pem
SSLCertificateFile /var/httpd/conf/httpsd.pem
SSLLogFile /var/httpd/logs/virt1-ssl.log
SSLVerifyClient 0
SSLFakeBasicAuth
</VirtualHost>

<VirtualHost www.virt2.com>
SSLDisable
ServerAdmin webmaster@virt2.com
DocumentRoot /var/httpd/virt2
ScriptAlias /cgi-bin/ /var/httpd/virt2/cgi-bin/
ServerName www.virt2.com
ErrorLog logs/virt2-error.log
TransferLog logs/virt2-access.log
</VirtualHost>
```

Depending on the modules compiled in, not all directives may be available. You can retrieve a list of available directives with `httpsd -h`.

3.4 Adding frontpage support to a web

Enter `/usr/local/frontpage/version3.0/bin` and load `./fpsrvadm`. Choose `install` and `apache-fp`. The next questions should be answered the following way:

```
Enter server config filename: /var/httpd/conf/httpd.conf
Enter host name for multi-hosting []: www.virt2.com
Starting install, port: www.virt2.com:80, web: ""
Enter user's name []: virt2admin
Enter user's password:
Confirm password:
Creating root web
Recalculate links for root web
Install completed.
```

The user name must be the unix login of the webowner. The password does not necessarily have to match the system password. You have to manually add `sendmailcommand:/usr/sbin/sendmail %r` to `/usr/local/frontpage/www.virt2.com:80.conf`, otherwise your users will not be able to send web-generated eMails. `kill -HUP` your `httpsd` to make fp reread its config. You can now access `www.virt2.com` with your frontpage client.

TOC

Under some circumstances `fpsrvadm` complaints that a root web has to be installed first. This is pr useless, but you should do so to silence `fpsrvadm`.

3.5 Starting the daemon

Start Apache with `httpsd -f /var/httpd/conf/httpd.conf`. You can now access `www.virt1.com` both through http and https which is pretty cool. Of course you have to pay for a real certificate if you want to offer webwide SSL or users might laugh at you.

Copy one of the demo files from the php examples directory to `virt1` to test phtml.

3.6 Some considerations left

Do not use frontpage 97 extensions. They do not work, at least under Linux. When installing specific versions of the c++ libraries, they appear to work but your logs will soon fill with `premature end of script headers` and your mailbox will fill with complaints. Do not use frontpage 98 extensions before version 3.0.2.1330. Do not be confused, version numbers are somewhat inheterogenous. When telnetting to port 80, typing "get / http/1.0" and hitting return twice, you get a version number 3.0.4 for frontpage.

You can find out the more specific version number by executing `/usr/local/frontpage/currentversion/exes/_vti_bin/shtml.exe -version`. Older versions have a nasty bug that requires httpd.conf to be writable by the gid of the webserver. This should make you scream if you are at all concerned about security. Versions since 3.0.2.1330 are more usable.

3.7 Known bugs

When touching `Recalculate Links` in the frontpage client, the server starts a process that consumes 99% cpu cycles and some 10 mb of memory. But even for medium-sized webs and fast machines, the client sometimes recieves a timeout message, though the calculation will be finished correctly. Inform frontpage users to be patient and not to hit `Recalculate Links` several times. Inform yourself to equip the server with at least 64MB.

Please note that at the time of writing both SSL and frontpage work, but not at the same time, that means you can neither publish your web using ssl nor make use of the webbots through https. You can publish your web on port 80 and access it encrypted on port 443, but your counters etc. will be broken. I consider this a bug. This problem shall be fixed in SSL 0.9.0.

3.8 The final word

For those who think the title of this howto is nearly as long as the document: Did you ever listened to Meat Loaf?

O.K. readers, you're done for today. Feel free to send me your feedback, eternal gratitude, flowers, e
cars, oil sources etc.

Automount mini-Howto

don@sabotage.org

v0.4, 17 April 1999

This file describes the autofs automounter, how to configure it, and points out some problems to avoid.

1. Introduction

- 1.1 Automount - what and why
- 1.2 Types of automounting

2. Installation

3. Configuration

4. That long wait to unmount

5. Questions

- 5.1 I don't see /auto/floppy, or whatever mountpoint I'm looking for.
- 5.2 How do I see what's mounted?
- 5.3 I put in a win95 disk ("vfat") and it was autodetected as only a regular FAT disk.
- 5.4 My filesystem `/auto/grumblesmurf` is mounted and `kill -SIGUSR1` won't unmount it.
- 5.5 What happens if I make / the directory for the automounter
- 5.6 Can I have two map files on the same directory?
- 5.7 I'm using SuSE 6.0 and needed `---timeout` instead of `--timeout`
- 5.8 How do I set the permissions and ownership for the filesystem (IE FAT)
- 5.9 Who do I thank for autofs?
- 5.10 Where can I learn more about automounting?

1. Introduction

1.1 Automount - what and why

Automouting is the process where mounting (and unmounting) of certain filesystems is done automatically by a daemon. If the filesystem is unmounted, and a user attempts to access it, it will be automatically (re)mounted. This is especially useful in large networked environments and for crossmounting filesystems between a few machines (especially ones which are not always online). It may also be very useful for removable devices, or a few other uses, such as easy switching between a forced-on ascii conversion mount of a dos filesystem and a forced-off ascii conversion mount of the same dos fs.

1.2 Types of automounting

There are two types of automounters in linux; *AMD* and *autofs*. AMD is the automount daemon, and supposedly works like the SunOS AMD. It is implemented in user space, meaning it's not part of the kernel. It's not necessary for the kernel to understand automounting if you NFS mount to the local host, through the AMD daemon, which routes all automount filesystem traffic through the NFS system. Autofs is a newer system assisted by the kernel, meaning that the kernel's filesystem code knows where the automount mount points are on an otherwise normal underlying fs, and the automount program takes it from there. Only autofs will be described in this mini-howto.

2. Installation

Because autofs is implemented in kernel-space, your kernel must have support compiled in. In 2.0.xx it is an experimental option, but appears to be quite stable. In 2.2.xx it is a normal option.

The automount program and its configuration files are also necessary; using the rpms (from RedHat, as part of the install) is a great way to go. The automount program should be started by an rc script under the `/etc/rc.d/init.d` directory . The rpm installs this, but you will need to make sure it gets started, either by linking it from your rc?.d directory, using Redhat's control-panel, or on another distribution by getting the thing started anyway you care to. Non-rpm distros will have to do whatever's applicable to their system. And don't look too hard at what the rc script does; if you're reading this howto you probably don't want to know.

3. Configuration

Installing the RPM's will get you to this point easily enough, but here's the part you might not be sure about if you haven't done this before.

There are two files in /etc, one called `auto.master` and one called `auto.misc`. My auto.master looks like this:

```
/auto   /etc/auto.misc  --timeout 60
```

The first entry is not the mount point. It's where the set of mount points (found in the second entry) are going to be. The third option says that the mounted filesystems can try to unmount themselves 60 seconds after use. They can't unmount if being used, of course.

Auto.misc is a "map file". The map file can have any name; this one is named auto.misc because it originally controlled /misc. Multiple map files can be defined in auto.master. My auto.misc looks like this:

```
kernel          -ro,soft,intr           ftp.kernel.org:/pub/linux
cd              -fstype=iso9660,ro      :/dev/cdrom
zip             -fstype=auto            :/dev/hdd4
floppy          -fstype=vfat            :/dev/fd0
```

The first column (the "key") is the mount point. In this case it would be /auto/floppy or whatever. The middle set are the options; read the mount manpage for details on this. And the last column specifies where the fs comes from. The "kernel" entry is supposed to be an NFS mount. The : on all the other lines means its a local device.

4. That long wait to unmount

Some of you may be eyeing that 60 second timeout and thinking, *that's a long time to wait to eject a floppy.. Maybe I'll just sync the disks and pop it out mounted and nobody will notice.* Let me suggest `saner alternatives`. First of all, you can change the timeout. But that could be a little inefficient; telling the system to unmount stuff after only 15 seconds or whatever. Depending on your setup, you may be able to simply run the umount command as a normal user. But there is actually a way to ask the automount program to umount. If you send (with the program kill) the signal `SIGUSR1` to the automount process, it will unmount everything it can. But before people start making unmount buttons on their window managers, there's a little problem.

The automount process is run by root, and it will only accept signals from root. Half of the reason you're probably doing automounting is so you can mount an unmount *without* being root. It would be easy to make a suid-root C program which does the dirty deed. However, by using sudo it is possible to allow users to send the proper kill signal. The only problem is that sudo will not let you use ` to process subcommands, which you would have to do to find the current PID. You should have a program called killall, which will let you do this: (thanks for the suggestions)

```
ALL     ALL=NOPASSWD:/usr/bin/killall -USR1 automount
```

Otherwise, you would have to allow your users to send -SIGUSR1 to all processes. That has various effects on programs; it will recycle some window managers, but kills xemacs. So here's hoping there's no buffer overruns in killall...

5. Questions

5.1 I don't see /auto/floppy, or whatever mountpoint I'm looking for.

If automount is setup properly, whatever mount point you're looking for will be there if you try and use it, even though you don't see it when not in use. If you're browsing the directory with a graphical tool, you may need to type in the name manually; most programs will try what you give it, and the drive will be mounted before it notices. Unfortunately not being able to choose from the available invisible mount points is probably the major drawback of autofs. If it really bugs you, edit the configuration files. (Hint, the ones that end in .c for "configuration")

One workaround several people have tried is to create symbolic links to where automount will create something once it's mounted. This will likely prevent the program from complaining a directory doesn't exist (if the mount works, that is) but careless directory listings will cause filesystems to be mounted.

5.2 How do I see what's mounted?

The `df` command. `mount` with no options will do the same, plus show the options its mounted with.

5.3 I put in a win95 disk ("vfat") and it was autodetected as only a regular FAT disk.

This is not a problem with automount. As of this writing, the "auto" fs type does not attempt a vfat mount before it successfully mounts an msdos filesystem. VFAT is the Win95 and WinNT long filenames crammed into a FAT/MSDOS filesystem.

According to one of the authors of mount, since mount is only a wrapper around a system call which must specify the filesystem type, it's still the responsibility of the user to come up with the fs type. Having mount take a list of filesystems to try in order, rather than the current "heuristic" is under consideration. Some users have simply not compiled msdos into the kernel; this prevents it from being tested prior to vfat. This will work for most people; a few actually need msdos fs and it caused me quite some frustration to not have a module handy when I actually needed it.

I'm sure that if anyone wants to go to the effort of finding the owner(s) of the mount program, your comments would be welcome. So unless you don't compile msdos in, for now this means that you can't mount vfat unless you give up the ability to autodetect all other fs's. Hopefully it will be configurable someday. In the mean time, feel free to create multiple mount points with different fs types specified.

5.4 My filesystem `/auto/grumblesmurf` is mounted and `kill -SIGUSR1` won't unmount it.

TOC

It's being used by something. Root probably can't manually unmount it either. If you're the one who caused it to be mounted (i.e. it can't be someone else using it) look around for a shell that might be in that directory. If there are none, look for something else (particularly something that might have gone though that directory like a directory browser) that might have left an invisible foot in the door so to speak. If you've given up looking, try using the fuser program.

5.5 What happens if I make / the directory for the automounter

Oooh. Well, out of a statistical sample of only one person, none of the results were positive. You have been warned. If you want /grumblesmurf, then I suggest a symbolic link. Much safer.

5.6 Can I have two map files on the same directory?

Not as far as I know. Try using one map file, with specific options for individual entries.

5.7 I'm using SuSE 6.0 and needed `---timeout` instead of `--timeout`

Uh. Ok, I've made a note about it. Another solution to "timeout not working" problems would be to add a `-t time` option to the autofs script.

5.8 How do I set the permissions and ownership for the filesystem (IE FAT)

Check the man page for mount for some of the options, such as setting the uid=value or umask=value options. One option that appears to be missing for FAT filesystems is mode=value. Sorry. Check in with the people who do mounting.

5.9 Who do I thank for autofs?

Not me. I didn't have anything to do with it. I just wanted to bring everyone's attention to what a great job had been done with autofs, and how easy it is to use. Compared to the original perpetrators of AMD (Hint, they sell an overpriced unice with prehistoric versions of free tools) the autofs is very well documented and the implementors have my sincere thanks. Everything is stamped copyright Transmeta so sorry I can't provide a credits list, but I would bet Peter Anvin is responsible for quite a bit of it. Peter

also held a session on autofs at linuxworldexpo on March 3, 1999.

5.10 Where can I learn more about automounting?

There's a autofs tutorial at http://www.linuxhq.com/lg/issue24/nielsen.html. See also am-utils at http://www.cs.columbia.edu/~ezk/am-utils

(Thanks for these URLs)

Backup-With-MSDOS mini-HOWTO

Christopher Neufeld, neufeld@physics.utoronto.ca

v, 5 August 1997

This HOWTO describes how to use a Linux-compatible tape drive installed on an MS-DOS machine to back up the filesystem of a Linux machine.

1. Preface/Introduction

2. The technique

3. Notes

4. Copyright

1. Preface/Introduction

Earlier I posed the question on the Net, how does one back up a Linux machine to a Colorado Jumbo 250 tape drive on an MS-DOS machine. From the email I received, it seems that this is a frequently pondered problem. Now that I've figured it out, I'm posting the method. If anybody wants to massage this into a HOWTO document, let me know. I should thank Jim Nance (jlnance@isscad.com) for pointing out that an MS-DOS machine need not always be an MS-DOS machine. This technique should also work for any other tape drive supported by the ftape module, and for SCSI tape drives with suitable obvious changes (i.e. substituting /dev/st0 for /dev/ftape).

The criteria I set were that the resulting setup should be as secure as possible and should be fairly simple, and take up little or no space on the MS-DOS machine's hard drive. It should also be capable of recovering from the worst system corruptions, up to and including the theft of the hard disk, requiring a restore to a bare Linux file system. The technique described here uses no hard drive space on the MS-DOS machine, though it requires that that machine be assigned an IP#. You will need three formatted, blank 1.44MB diskettes.

2. The technique

Throughout this description I will refer to two machines as ``msdos'' and ``linux''. ``msdos'' is the name of the machine which has the tape drive and is usually running MS-DOS. ``linux'' is the Linux machine whose disk you are trying to back up or restore to the tape drive. For simplicity I will refer to the first machine as ``msdos'' even when it is booted into and running Linux. Further, all path names in this document should be considered to be relative to the Linux machine with the Search-And-Rescue (SAR) disks mounted somewhere on the system. That means that the file `/etc/passwd` is the password file for your Linux machine's hard drive, while, for instance, `/tape144/etc/passwd` is the corresponding file on the floppy disk.

I am using Karel Kubat's backup scripts, version 1.03, available at

ftp://sunsite.enc.edu:/pub/Linux/system/Backup/backup-1.03.tar.gz

Throughout this document I will refer to these simply as ``the backup scripts''. You do not have to use these scripts for your own backups to tape. I like these scripts as they use afio to form an uncompressed archive of compressed files, rather than a compressed archive of uncompressed files. The former is much safer if there is a media read error during the restore. I understand that Karel is no longer supporting backup, and now has produced 'tob', or tape oriented backup. While I haven't tried the new package myself, it cannot make a significant difference to the procedure outlined here.

First of all, obtain the ftape module. It is a part of all modern kernels, but if you are using an older kernel you can find the module at:

ftp://sunsite.unc.edu/pub/Linux/kernel/tapes/ftape-2.05.tar.gz

Next, get a Slackware boot disk (I got the net disk, but it doesn't make much difference) and the `tape144` root disk, and put the images onto 3"1/2 floppies.

The `ftape` module will only work if it is installed in the kernel which was running when you compiled it. I could not get it to work with the `ftape.o` module on the `tape144` root disk, I think because that module has been stripped of symbols and won't install. So, you now have to make a new kernel with network and ftape support, and if you're running an old kernel, a new `ftape.o`. Read the directions which ship with the `ftape` archive for directions at this stage. Remember that the kernel you compile must support the Ethernet cards on both the Linux machine and the MS-DOS machine.

Copy the newly created kernel image over top of the one on the Net boot disk. Use `/bin/cp`, do not create a boot disk with the ``dd'' command as you would to create a bootable kernel image. Write protect the boot disk, and label it: SAR#1.

Now, mount the `tape144` root disk. I'll assume that the mount point is `/tape144`, to avoid confusion in file names. We need to free some space on it, so delete the following files:

```
/tape144/bin/dialog
/tape144/bin/elvis
/tape144/bin/vi
/tape144/boot/ftape.o
```

Now, create a new file:

```
/tape144/etc/exports
```

which contains the following line:

```
/mnt    msdos(ro)
```

TOC

Where ``msdos'' should be replaced with the name or IP# of the MS-DOS machine which has the tape drive installed.

Next, so that you don't have to rely on a name server, add lines to the file `/tape144/etc/hosts` with the names and IP numbers of the Linux and MS-DOS machines. For instance, mine contains the following two lines:

```
128.100.75.114  caliban.physics.utoronto.ca caliban caliban.physics
128.100.75.111  ariel.physics.utoronto.ca ariel ariel.physics
```

Now, there's some sort of problem with the inetd configuration. We have to put the full path name of the rsh daemon in it. Change line 19 of `/tape144/etc/inetd.conf` to read:

```
shell   stream  tcp     nowait  root    /usr/etc/tcpd   /usr/etc/in.rshd
```

Add local net routing information to `/tape144/etc/rc.d/rc.inet1` to enable the MS-DOS machine to use the network. The format of this depends on your network configuration, you can just copy the appropriate format out of your Linux `/etc/rc.d/rc.inet1`. For my network, the lines that have to be added are:

```
/etc/ifconfig eth0 128.100.75.111 broadcast 128.100.75.0 netmask 255.255.255.0
/etc/route add -net 128.100.75.0 netmask 255.255.255.0
```

The IP# in the ifconfig entry is that of the MS-DOS machine.

Now, copy this file into `/tape144/etc/rc.d/rc.inet1-l`, and change the IP# in the new file to reflect that of the Linux machine rather than the MS-DOS machine.

Next, clip out lines 3 to 11 of `/tape144/etc/rc.local`. That's an if statement which executes the `rc.inet*` files. We don't want this to happen during the bootup.

Create a new file: `/tape144/root/.rhosts` containing the line:

```
linux root
```

where, again, ``linux'' is replaced with the full machine name (including domain) or the IP# of the Linux machine.

Fill in the password field in `/tape144/etc/passwd` for the root login to keep people from logging onto the MS-DOS machine while you're doing the backup. You can do this by copying the corresponding field from your Linux machine's `/etc/passwd file`.

Copy /usr/bin/rsh into /tape144/usr/bin.

Copy the following files from `/usr/etc` into `/tape144/usr/etc`:

```
in.rshd
rpc.mountd
rpc.nfsd
rpc.portmap
services
tcpd
```

Create a new script, `/tape144/bin/tapesetup`, which consists of the following: (change ``linux'' to reflect your Linux machine name).

```
#! /bin/sh

/bin/sh /etc/rc.d/rc.inet1
/bin/sh /etc/rc.d/rc.inet2
```

```
/bin/mount linux:/nfs /mnt
/bin/insmod /mnt/ftape.o
```

Note that newer kernels will not require the insmod line.

Next, create another new script, `/tape144/bin/msdosset`, as follows: (change ``linux'' to reflect your Linux machine name).

```
#! /bin/sh

/bin/sh /etc/rc.d/rc.inet1
/bin/sh /etc/rc.d/rc.inet2

mount linux:/mnt /mnt
/bin/insmod /mnt/ftape.o
```

As above, newer kernels will not require the insmod line.

Create a readable file, `/tape144/root/notes`, which contains this helpful information for use in full recovery:

```
For a full recovery to a trashed hard disk,
boot the Linux machine with the SAR disks #1 and #2
then type the following:

   /bin/sh /etc/rc.d/rc.inet1-l
   /bin/sh /etc/rc.d/rc.inet2

   /usr/etc/rpc.portmap
   /usr/etc/rpc.mountd
   /usr/etc/rpc.nfsd

Next, insert SAR disk #3 and type:

   mount /dev/fd0 /mnt

Create a new mount point, with:

   mkdir /mnt2

and mount your Linux hard disk partition on this point.
You may have to reformat the partition first, if so,
follow the directions in the Linux Installation HOWTO.
The SAR disks contain all the files necessary to do the reformat.

Finally, use disks SAR#1 and SAR#2 to boot up
the MS-DOS machine and run the /bin/msdosset script on that machine.
It will take about a minute to run that script because it is getting
an NSF file from a floppy drive, so be patient.
Now, recover the tape to /mnt2 on the Linux machine.
```

If you are using the backup scripts you will need to copy 'afio' into the `/tape144/local/bin` subdirectory. It is not necessary to have the rest of the backup script files on the recovery disks, an archive can be recovered using only 'afio' and 'gzip'.

I was unable to use the backup scripts as they come shipped. The tape archive appears to build cleanly, but it is unrecoverable. I found that removing the block size and conversion statements fixed it. Here is the patch to the

TOC

``netbackup'' script. Apply this patch to the Linux machine's hard disk copy of 'netbackup' as well as to the copy o SAR disks.

```
*** netbackup.orig      Mon Jan  9 17:22:32 1995
--- netbackup   Mon Jan  9 17:23:25 1995
***************
*** 35,41 ****
                  "'mknod", devname, "p'");
      exec ("su -", USERNAME, "-c",
                  "'rsh ", REMOTE_HOST,
!                       "\"dd", "of=" REMOTE_DEVICE, "obs=20k", "conv=sync\"",
                        "<", devname,
                "'&"
         );
--- 35,41 ----
                  "'mknod", devname, "p'");
      exec ("su -", USERNAME, "-c",
                  "'rsh ", REMOTE_HOST,
!                       "\"dd", "of=" REMOTE_DEVICE, "\"",
                        "<", devname,
                "'&"
         );
***************
*** 50,56 ****
                  "'mknod", devname, "p'");
      exec ("su", USERNAME, "-c",
                  "'rsh ", REMOTE_HOST,
!                       "\"dd", "if=" REMOTE_DEVICE, "ibs=20k", "conv=sync\"",
                        ">", devname,
                "'&"
         );
--- 50,56 ----
                  "'mknod", devname, "p'");
      exec ("su", USERNAME, "-c",
                  "'rsh ", REMOTE_HOST,
!                       "\"dd", "if=" REMOTE_DEVICE, "\"",
                        ">", devname,
                "'&"
         );
```

You have now finished your SAR disk #2. Write protect it.

Next, mount a clean, formatted disk (create it with `fdformat` and `mkfs`). Copy the `ftape.o` file onto it, and label it SAR#3. For some reason things go badly if you write protect this disk, so leave it write-enabled.

On the Linux machine, create a new directory for NFS file serving. I made a directory:

```
/nfs
```

Put the `ftape.o` (unstripped, about 500+ kB) into this subdirectory. Create an entry in your Linux's exports file `/etc/exports`:

```
/nfs    msdos(ro)
```

Note that all files in your NFS directory and it's subdirectories are not secure. Somebody else could boot the MS-DOS

machine into Linux with his own boot disks and mount this directory, so be certain that you don't put anything sensi your NFS subdirectory.

Restart your NFS daemons, `rpc.mountd` and `rpc.nfsd`. They don't seem to take kindly to a `SIGHUP` restart, so kill them and reinvoke them. If you're not activating these daemons in your `/etc/rc.d/rc.inet2` you might want to do so now.

OK, now we're all set to back up and recover. To make a full backup from the Linux machine, boot the MS-DOS machine with SAR#1. When prompted for the second disk, load SAR#2. Log in as root, and execute the script: `/bin/tapesetup`. Log out of the MS-DOS machine. If you're using the backup scripts, the netbackup command will now work. You can also use the ``-f msdos:/dev/ftape'' switch on GNU `tar`, `cpio`, or `mt`, and make your backup this way. If you have a backup program, which is only capable of writing to a local file, do the following. Assume that the backup program is called ``localbackup'' and writes to the file represented by its command line argument:

```
mknod /tmp/tapepipe p
rsh msdos dd of=/dev/ftape < /tmp/tapepipe &
localbackup /tmp/tapepipe
```

when it's done, delete `/tmp/tapepipe`.

Recovering to a live Linux machine: the `netbackup` script, `tar`, `cpio`, and so on will all work without special actions on the part of the operator. If you have a local recovery program which recovers from a file, do this:

```
mknod /tmp/tapepipe p
rsh -n msdos dd if=/dev/ftape >> /tmp/tapepipe &
localrecovery /tmp/tapepipe
```

and delete `/tmp/tapepipe` when you're done.

Notice that I'm using '`rsh`' to the root user on the MS-DOS machine. This works with a correct `.rhosts` entry. The configuration on the '`tape144`' disk allows `rsh` to root, but does not allow `telnet` or `rlogin` to root, logins are restricted to the console. This is good for security.

If you are worried about a root `.rhost` file, you can create a new user on SAR#2, ``tapeuser'', with permissions to operate the tape drive but not the disks (create a new group and put tapeuser in that group, then `chown` and `chmod` the files `/dev/rft*` and `/dev/nrft*`). Your backup program then has to know to `rsh` to that username rather than to root. Of course, now there must be an `.rhosts` file in `~tapeuser` on SAR#2. For my own use, I have chosen this course, rather than a root `.rhosts`.

Finally, the directions for a complete recovery to a trashed hard disk. This assumes that the Linux partition is completely unrecoverable. If necessary, reformat that partition as described in the Linux Installation HOWTO. Boot the Linux machine from SAR disk #1. When prompted, insert disk #2. Now, follow the directions in the file `/root/notes` (this was `/tape144/root/notes` when it was mounted on your Linux machine). Once both machines have been booted up, run the recovery routine you need. If you are running the backup scripts you can do it as follows:

1. change directory to the mount point of the hard disk partition which you will be recovering.
2. if any mounted volumes are on the backup, and you want to recover them, create the mount points within the hard disk partition and mount the volumes.
3. Enter the command:

   ```
   rsh -n msdos dd if=/dev/ftape | afio -i -v -Z -c 1024 -
   ```

   or

   ```
   rsh -n -l tapeuser msdos dd if=/dev/ftape | afio -i -v -Z -c 1024 -
   ```

or

```
mknod /tmp/backpipe p
rsh -n msdos dd if=/dev/ftape >> /tmp/backpipe &
afio -i -v -Z -c 1024 /tmp/backpipe
```

This reads the tape on the remote machine, writing the result to stdout, where `afio` picks it up. The '`-i`' switch tells it to recover the files relative to the current working directory (which is the root of the hard disk partition). '`-v`' is verbose, listing the files as they are recovered. '`-Z`' tells afio that this is an archive of individually compressed files. '`-c 1024`' tells it to use a 5 MB streaming buffer to avoid a lot of tape rewinding.

---

3. Notes

The commands listed in the `/tape144/root/notes` file could be run from a script. When I tried, I got rpc setup errors. I suspect it was just that the commands were run too quickly, and the portmapper hadn't properly installed itself. I found that typing the sequence in manually worked fine, so I've recommended that.

I think this setup is secure. Note that somebody can still get access to all your files if they go to the tape drive and pull the tape out before you get there, then then read the tape themselves. People with very sensitive data might consider encrypting the stream from the archiver. Archive to standard output and pipe the output to the encrypter, and redirect the output of the encrypter to append to the named pipe `/tmp/tapepipe` as described above. Note that errors in the recovery process will result in all files after that point being unrecoverable, as the entire archive is now a single DES-encrypted stream. It is possible to use options on afio to send each file in the archive first through gzip, then into an encryption program like des, but note that this compressing first does provide a fair amount of known plaintext for determined code breakers to work with, so a better approach might be to skip the gzip step and simply encrypt it with des, at the expense of significantly more tape area. Needless to say, DES encrypted files don't compress.

The `rc.inet1` directions I've included will allow only communication with the local network, not the rest of the world through a gateway.

During a full recovery to a blank hard disk the SAR disk #3 provides `ftape.o` to the MS-DOS machine through NFS. This is because some old versions of the `ftape` module can't control some tape drives when there is a disk mounted in the floppy drive. With newer kernels, the entire NFS stuff can be omitted.

This is very important. ***TEST*** the SAR recovery procedure. I did, but don't leave anything to chance. Make sure that you can recover at least one file from your tape to the Linux machine using only the SAR disks (i.e. without mounting the hard disk). If you can't reboot the Linux machine without inconveniencing a lot of users, change the setup information on the SAR disks to assign the ``linux" identity to another MS-DOS machine and then boot the two MS-DOS machines into Linux to make sure everything works. Then, change the ``linux" identity back again so that you have usable SAR disks.

4. Copyright

Copyright Jan 10, 1995 by Christopher Neufeld (neufeld@physics.utoronto.ca)

Modified Feb 6, 1996.

Modified Aug 5, 1997.

Battery Powered Linux Mini-HOWTO

Hanno Mueller, hanno@lava.de
http://www.lava.de/~hanno/

v, 21 December 1997

This document describes how to reduce a Linux system's power consumption by tweaking some of its configuration settings. This will be helpful for everyone who runs Linux on a portable computer system. There is also some general information about how to take care of your battery. If you are using Linux on a desktop system, you probably don't need to read all this.

1. Introduction

- 1.1 Before you ask
- 1.2 What this document is about
- 1.3 Roadmap
- 1.4 Feedback
- 1.5 Disclaimer
- 1.6 Copyright

2. General information

- 2.1 Be kind to your battery
- 2.2 Power saving - The obvious stuff

3. Advanced Power Management

- 3.1 What APM can do for you
- 3.2 How to activate APM support in Linux
- 3.3 APM support and the PCMCIA drivers
- 3.4 The apmd package

- 3.5 And if my laptop does not support APM?

4. Changing some general system settings

- 4.1 The crond daemon and atrun
- 4.2 The update / bdflush daemon
- 4.3 The syslogd daemon
- 4.4 The init command
- 4.5 The swap partition
- 4.6 The apache httpd webserver daemon
- 4.7 The XFree86 package
- 4.8 The emacs editor
- 4.9 How to find more ways to optimize

5. Appendix

- 5.1 A message to Linux distributors
- 5.2 Credits
- 5.3 About this document

---

1. Introduction

``Ages 6 and up. Batteries included."

1.1 Before you ask

This document does not describe how to install Linux on laptops, but how to optimize a ready-configured Linux for use on laptops. Please read the *Installation-HOWTO* or your distributor's handbook for help with installing Linux.

This document does not describe how to use an uninterruptable power supply and the `powerd` daemon, either (even though a ups is a big battery). Read the *UPS-HOWTO* for details about that subject.

1.2 What this document is about

More and more people own portable computers these days and in turn, more and more people install Linux on such machines.

Installing and using Linux on a laptop is usually no problem at all, so go ahead and give it a try. Unlike some other operating systems, Linux still supports and runs well on even very old hardware, so you might give your outdated portable a new purpose in life by installing Linux on it.

If you need help with installing Linux on a laptop or if you have questions about laptop hardware, you can check the excellent *Linux Laptop webpage* at `http://www.cs.utexas.edu/users/kharker/linux-laptop/` where you will find a lot of useful information and detailed help. The Linux Laptop page describes hardware configuration for specific laptop models and chipsets.

This HOWTO however will focus on the one problem that is common to all portable systems: *Power consumption*.

Yet, I have not found a Linux distribution that comes with a configuration *optimized* for laptops. Since I could not find this kind of information anywhere else, I have started to collect a few simple but effective tricks that will help you save battery power and in turn increase your system's uptime while running on battery.

(Sidenote. I received a complaint by a reader that these tips were not very effective with his laptop. So does all this really work? Yes, but don't expect miracles. I was able to increase my laptop's battery time from 90 minutes to more than 120 minutes.)

TOC

1.3 Roadmap

If you are a laptop pro, you can probably skip the General Information section. If you are a Linux pro, what you really want to know can be found in the Changing some general system settings section. If you are a Linux distributor, please read A message to Linux distributors.

1.4 Feedback

Your feedback is welcome. Please send comments to `hanno@lava.de`. Did it work on your system? Do you have new tips? Are there any outdated links or addresses in this text?

I am sorry, but I will not be able to help you with questions about specific laptop models. I don't claim to be a laptop guru, I just happen to own one laptop myself and I simply want to share the information I collected. Please check the Linux Laptop webpage first, probably someone else has already written a page dedicated to your model. Ask your manufacturer's technical support. Or go the the laptop newsgroup `comp.sys.laptops` and ask there.

1.5 Disclaimer

All methods described here were tested by me and worked fine on my laptop, unless noted otherwise. However, I cannot guarantee that any of this won't crash or seriously damage your system. Life is dangerous, so keep backup copies of your important files before playing with your Linux configuration. If things go wrong, I do not take any responsibility for your data loss. In other words: Don't sue me. Thank you.

1.6 Copyright

This document shall be distributed under the standard HOWTO-copyright notice, found in the HOWTO folder at `http://sunsite.unc.edu/mdw/linux.html`.

---

2. General information

This section describes a few technical things about laptop batteries and some general power saving tips. This information is not Linux-specific and if you are experienced with laptops, you might already know all this.

2.1 Be kind to your battery

(Please note the credits for this section.)

There are currently three types of batteries commonly used for portable computers.

- *NiCd* batteries were the standard technology for years, but today they are out of date and new laptops don't use them anymore. They are heavy and very prone to the ``memory effect''. When recharging a NiCd battery that has not been fully discharged, it ``remembers'' the old charge and continues there the next time you use it.

  The memory effect is caused by crystallization of the battery's substances and can permanently reduce your battery's lifetime, even make it useless. To avoid it, you should completely discharge the battery and then fully recharge it again at least once every few weeks.

  (A sidenote about the memory effect. James Youngman knows of a rather drastic method to - uhm - ``repair'' batteries: ``If your NiCd battery is suffering from the memory effect, remove it from your computer, hold it about 30cm above a desk or the floor, and drop it (make sure it lands flat)." He says that this will break the whiskers that have formed in the battery and that are the cause of the memory effect if your battery is already affected by it. ``I don't know if this works for non-NiCd batteries or not.")

  Cadmium is a very hazardous poison, but if returned to your dealer, the material can almost be fully recycled.

  Just in case you might be interested, here are some specs for NiCd:

  ```
  Cell voltage: 1,2 V
  Energy / mass: 40 Wh/kg
  Energy / volume: 100 Wh/l
  max. Energy: 20 Wh
  Charge temp.: 10 to 35 C (50 to 95 F)
  Discharge temp.: -20 to 50 C (-5 to 120 F)
  Storage temp.: 0 to 45 C (30 to 115 F)
  ```

- *NiMh* batteries are the current standard used in most low price laptops to date. They can be made smaller and are less affected by the memory effect than NiCd.

TOC

However, they have problems at very high or low room temperatures. And even though they less hazardous and non-poisonous substances, they cannot be fully recycled yet (but this will probably change in the future).

NiMh specs:

```
Cell voltage: 1,2 V
Energy / mass: 55 Wh/kg
Energy / volume: 160 Wh/l
max. Energy: 35 Wh
Charge temp.: 10 to 35 C (50 to 95 F)
Discharge temp.: 0 to 45 C (30 to 115 F)
Storage temp.: 0 to 30 C (30 to 85 F)
```

- The new high performance batteries use *LiIon* technology. In theory, there is no memory effect at all with these batteries, but on occasion, they seem to have similar problems. Their substances are non-hazardous to the enviroment, but they should be returned for recycling as well.

  LiIon specs:

```
Cell voltage: 3,6 V
Energy / mass: 100 Wh/kg
Energy / volume: 230 Wh/l
max. Energy: 60 Wh
Charge temp.: 0 to 45 C (30 to 115 F)
Discharge temp.: -20 to 60 C (-5 to 140 F)
Storage temp.: -20 to 60 C (-5 to 140 F)
```

Even if the battery case looks the same, you cannot just upgrade to another battery technology. The recharging process is different for the kind of battery you use.

Some manufacturers integrate the recharging circuit inside the laptop's external ac adapter, so you might just get away with buying a new power supply to upgrade. A good indication for an external recharging unit is when your ac adapter uses a proprietary connector with a lot of power lines.

Other manufacturers put the recharging unit inside the laptop case where users cannot simply replace it with a newer technology. If your ac adapter only uses two power lines to connect to the computer (just like mine), the recharging unit is probably inside the laptop.

When in doubt, ask your manufacturer if your laptop supports a more modern battery.

A battery that is not used for a long time will slowly discharge itself. And even with greatest care, a battery needs to be replaced after 500 to 1000 recharges. But still it is not recommended to run a laptop without the battery while on ac power - the battery often serves as a big capacitor to protect against voltage peaks from your ac outlet.

As the manufacturers change the shapes of their batteries every few months, you might have problems to find a new battery for your laptop in a few years from now. Buy a spare battery now - before it's out of stock.

2.2 Power saving - The obvious stuff

TOC

There are some obvious things that you can do to reduce your system's power consumption. Well, maybe not so obvious, since not very many people follow these rules...

- Decrease or turn off your display's backlight when you don't need it. By the way, tft displays use more power than dstn (so now you have a fine excuse why you bought the cheaper laptop...).

  (David Bateman tells me that using a crt screen while on battery and turning off the laptop display will extend battery time by about 30%: ``Not that this is a very useful piece of knowledge though, if you've got the crt plugged in then why not the laptop too.")
- How much processing power do you really need? I doubt that you will be doing very much more than text editing when on the road (well, at least I don't compile linux kernels then). While on battery, reducing the cpu clock speed will decrease power consumption, too. Quite a few laptops offer a cpu clock selector that will toggle between normal and slow speed.
- Turn off the cpu cooler (if you have one). Many recent laptops offer a bios option called ``cooling control". If your system's cpu is becoming too hot, this option allows you to have it cooled by a tiny fan (setting ``performance") or to have its cpu clock slowed down (setting ``silence"). To increase your uptime while on battery, use ``silence".
- Avoid using external devices (printer, crt screen, zip drive, portable camera etc.) with your computer while on battery. When connected to a standard ink jet printer, my laptop's battery time is reduced from up to 120 minutes down to 20 minutes.
- Avoid using any built in device unless necessairy: Diskette drive, harddisk, cd-rom. Especially cd-rom access will dramatically decrease your battery time.
- Pcmcia cards can also consume a lot of power, so don't leave your modem or network adapter plugged in when it is not in use. But this is different between the various pcmcia manufacturers, so check the product specs before you buy (e. g. some cards never turn themselves off even when not in use).

  (By the way, I recently read that pcmcia cards are the biggest problem for windows ce palmtops - they drain so much power that the tiny machines' little batteries have to be replaced within minutes...)
- Use simple software. A full blown multimedia application will create a lot more system load and harddisk / cd-rom activity than a small simple word processor.
- Grant Taylor has a tip for those of us who want to upgrade their system: ``Newer versions of some upgradable components consume less power. For example, IBM's Travelstar 2.5 inch 1.6 gigabyte ide harddisk drive consumes 20 percent less than the 500 megabyte toshiba harddisk my laptop came with."
- If you are yet about to buy a laptop - don't buy a laptop with a 2nd level cache if battery uptime is important. A computer with 2nd level cache is about 10% to 20% faster and it will be a lot better with multimedia applications and number crunching, but it consumes a lot of power. Bjoern Kriews tells me that he has two almost identical laptops and the one without cache ram runs 4h30 compared to 2h30 with cache.

If you already have 2nd level cache installed, turning it off will probably not help you very mu
Give it a try and write me about your experience.

- Another tip for those still buying a laptop - don't buy the latest, fastest cpu type. Usually, the older generations are optimized by the manufacturer after some time without notice. The ``new'' versions of old cpu types often create less heat and consume less power than the product's premiere version.

  There are also frankenstein laptops available that use cpus not optimized for portable systems. As I wrote this in May 97, the newest generation pentium-200 laptops ran about 20 minutes on battery and became so hot that they burnt your lap. When writing the second revision in Oct 97, pentium-233 laptops run two hours or longer without ac power. Go figure.

Well, you get the idea. Most of these are restrictions that will probably stop you from doing any serious work with your Linux system. (The best way to save power while on battery is... not to do anything at all. That increases my laptop's battery uptime by almost 100 percent.)

So let's go ahead to some other, more useful measures that will save power without disturbing your work.

---

3. Advanced Power Management

Portable systems in general, but even many desktop computers come equipped with support for apm, the ``advanced power management'' scheme. This section describes how to activate apm support in your Linux kernel. People who are experienced with Linux may find this section rather boring and want to skip to the next.

3.1 What APM can do for you

I won't describe it in detail here, check the *Linux APM drivers page* at `http://www.cs.utexas.edu/users/kharker/linux-laptop/apm.html` for more information. All that you need to know is that with the help of apm, the cpu can tell the bios when there's nothing really exciting going on so that the bios can take care of some power saving by itself - e. g. reducing the cpu clock, turning off the harddisk, turning off the display's backlight etc.

Apm is also responsible for the ``system suspend'' (or ``sleep'') mode and for the ``suspend to disk'' (or ``hiberntation'') mode. And yet another cool, though not very important feature is that with the help of apm, `shutdown -h` will not just halt your system, but also turn it off.

(By the way, most Linux systems put a `shutdown -r` in their `/etc/inittab` and map it to pressing control-alt-delete. I prefer having `shutdown -h` there, so when pressing the famous key combination, my laptop simply turns itself off.)

Not all manufacturers implement a correct apm bios, so some laptops have trouble with the Linux apm drivers (if your machine has trouble with apm, it will most likely either lock up at Linux' boot up or after returning from suspend). If you are not sure, check the Linux laptop page for your specific model.

3.2 How to activate APM support in Linux

It's easy - just recompile the Linux kernel. Check the Kernel-HOWTO if you don't know how to do that.

When the configuration script reaches the ``character devices'' section, the default setting for full apm bios support in kernel version 2.0.30 or higher is:

```
Advanced Power Management BIOS support: Yes
Ignore USER SUSPEND: No
Enable PM at boot time: Yes
Make CPU Idle calls when idle: Yes
Enable console blanking using APM: Yes
Power off on shutdown: Yes
```

Please read the configuration script's help texts. They explain in detail what each option does, so I won't

repeat them here.

If your system does not fully support the apm bios standard, some of those options might crash your system. Test all apm features with the new kernel to make sure that everything works as it should.

(A sidenote about console blanking: David Bateman tells me that you should not enable it because it can cause problems with the current version of XFree 3.2: ``The symptoms are that the screen will be blank when X starts, and it can be fixed usually by just hitting a key. It's a small but annoying problem. The next relase of XFree, will have pretty good DPMS support for a lot of laptop chipset, which should include code to turn off the LCD. Check out the manpage for `xset` in XFree 3.2A." David also notes that the lifetime of your display's backlight is determined by the number of times it's switched on and off: ``So its a compromise, lifetime of the battery versus lifetime of the backlight.")

(*Update:* With XFree 3.3, this problem still remained on my laptop. I am told that this will be fixed in a future kernel version.)

3.3 APM support and the PCMCIA drivers

After recompiling the kernel, don't forget to recompile the linux pcmcia drivers as well.

The precompiled pcmcia drivers that come with most linux distribution have apm support disabled, so that the bios can't instruct your card adapters to turn off.

Also, you must recompile the drivers if you upgrade to a new kernel version and your old kernel was compiled with module version information turned on (this option is found in the ``loadable module support'' section of the kernel configuration).

Read the PCMCIA-HOWTO for detailed instructions on how to compile the drivers or go to the *Linux PCMCIA drivers homepage* at `http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html`.

3.4 The apmd package

Now that you have APM support installed, go and get the `apmd` package from the *Linux APM drivers page*. You don't really need it, but it is a very useful collection of programs. The `apmd` daemon logs your battery's behaviour and it will send out a warning if you are on low power. The `apm` command will suspend your system with a shell command and `xapm` shows the current state of your battery.

(BTW, if you have problems with pcmcia cards after returning from suspend, you can check out an alternative `apmd` package at `http://www.cut.de/bkr/linux/apmd/apmd.html`. It unloads the pcmcia driver module before going to suspend and reloads the module on resume.)

Grant Taylor has been playing a little with the `apmd` package and came up with helpful tips.

He found that his laptop's harddisk forgets its `hdparm -S` standby period when returning from suspend: ``I modified `apmd` to reset this setting on each resume. This may be system-specific; but it's an important thing to do..."

(Note: On my own laptop, the bios takes care of the harddisk standby period and resets the value on resuming. So I could not test if this little problem is system-specific. If it happens to you as well, send a message.)

Grant also had a nice trick for screen blanking with the XFree86 package and the help of the `apmd` package, you'll find it there.

3.5 And if my laptop does not support APM?

If your computer's bios does not offer any power saving settings (even the old ones without apm should at least allow to set harddisk and display standby), you can use `hdparm -S` to define your harddisk's standby period. This will already help a lot, since harddisk activity consumes a lot of power. Your system should have `hdparm` installed, so read `man hdparm` for the command syntax.

---

4. Changing some general system settings

After I got Linux up and running on my laptop, I found it accessing the harddisk every few seconds, even when there was no user logged in to the system. The harddisk could never enter its power saving mode. Reducing harddisk activity can greatly increase the battery runtime, so this is why I collected the following recipes.

I tested all this with RedHat 4.1, the locations of some configuration settings may be different for your distribution. (If so, please let me know.)

4.1 The crond daemon and atrun

Check your `/etc/crontab` file if it starts a process every minute. You will often find `atrun` there.

With the `at` command, you can spool commands that must be invoked some time in the future. Some Linux systems use a dedicated `atd` daemon to take care of this, others (e. g. RedHat) let the `crond` daemon run `atrun` once every minute.

This is not really necessairy on most systems, since `at` commands rarely depend upon being invoked on exact time. So if you find a line like this in your `/etc/crontab`:

```
# Run any at jobs every minute
* * * * * root [ -x /usr/sbin/atrun ] && /usr/sbin/atrun
```

Then you can safely change this to:

```
# Run any at jobs every hour
00 * * * * root [ -x /usr/sbin/atrun ] && /usr/sbin/atrun
```

Read `man 5 crontab` for details. Some folks can even work fine without the `crond` daemon, so if you know what you are doing, you might want to consider disabling it completely.

4.2 The update / bdflush daemon

Linux deals with a lot of open file buffers at any given moment, so the system must make sure that file changes are saved to the harddisk as soon possible. Otherwise, those changes will be lost after a system crash.

The `update` / `bdflush` daemon takes care of this. (These are two names for the same program, so you can use either name to start the daemon). The default settings will make this daemon call `flush` every 5 seconds and `sync` every 30 seconds.

TOC

With my Fujitsu disk this caused non-stop access. (It seems that this harddisk flushes its ram cache e when nothing has changed. But this depends on your harddisk's firmware: Other people told me tha harddisk does enter its power saving mode even without the following modification.)

Since Linux does not crash very often anymore, I have changed both values to 3600 seconds (= one hour). This caused no problems at all and the constant disk access has stopped. (But if my system crashes now, there will be more broken files, of course.)

RedHat 4.1: In `/etc/inittab`, change the update call to:

```
ud::once:/sbin/update -s 3600 -f 3600
```

Suse 4.4.1: `update` is called in `/sbin/init.d/boot`.

Slackware: `update` is called in `/etc/rc.d/rc.S`.

See `man update` for details.

4.3 The syslogd daemon

The `syslogd` daemon is responsible for the various Linux system log files that are found in the `/var/log/` directory. By default `syslogd` will sync the log file each time after logging a system message.

You can turn that off by preceding the filename with a dash in `/etc/syslog.conf`. Here's an example as found in my system's `syslog.conf`:

```
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none                -/var/log/messages
```

This again means that if the system crashes, the message that reported the problem may not have been stored to disk. Dilemma...

4.4 The init command

During the bootup, the initial processes and daemons will be started using the `init` command. This command (yet again) calls `sync` before each process it creates.

You can change this by removing the `sync()` call in the source code and recompiling the command.

To avoid problems with lost file buffers, you should add a call to `sync` in your system's `/etc/rc.d/init.d/halt` script, right before the script unmounts the file systems.

TOC

4.5 The swap partition

The Linux swap partition is used to increase the physical ram space with virtual memory. This again is a possible reason for harddisk access. If your laptop already has a lot of ram or if the applications that you use are quite simple (think of `vi`), you might want to consider turning it off.

This of course depends on what you plan to do. 4 to 8 megs are not enough, you must use a swap partition then. With 8 to 16 megs, text console applications will work fine and if you can avoid using a lot of multitasking features, you can safely disable swap. The X-Windows enviroment requires a lot of ram and you should not use it without a swap partition unless you really have a lot more than 16 megs.

(Sidenote: My laptop with 16 megs and disabled swap partition can run an `emacs` session, four `bash` shells and compile a kernel without running out of memory. That's enough for me.)

If you already have installed a swap partition, you can disable it by preceding the `swapon` command that is called in `/etc/rc.d/rc.sysinit` with a hash mark. If you don't want to make it a permanent move, let the system ask during boot if you want to use the swap partition. In `/etc/rc.d/rc.sysinit` (RedHat 4.1) or `/sbin/init.d/boot` (Suse 4.4.1):

```
echo "Should the system use swap?"
echo "  0: No."
echo "  1: Yes."
/bin/echo "Your choice: \c"
read SWAPCHOICE

case "$SWAPCHOICE" in
    0)
        # Do nothing.
        echo "(Swap partitions disabled)"
        ;;
    *)
        # Start up swapping.
        echo "Activating swap partitions"
        swapon -a
esac
```

Then you can use the swap partition while on ac power and drop it while on battery.

4.6 The apache httpd webserver daemon

I am using my laptop to develop and test cgi scipts for websites, that is why I am running a local webserver on it. The standard configuration is a bit too much if all you want to do is just test a script or check a page from time to time.

In `httpd.conf`, just change the values of `MinSpareServers` and `StartServers` to `1`. This will be enough for a local test site.

TOC

If you wish to turn off the webserver's logging, you must recompile the `httpd` daemon. Read the documentation for details.

Grant Taylor recompiled apache's logging and found that this ``didn't make it stop churning the disk. So I used another, IMHO better, solution: I configured apache to run from inetd instead of standalone." Read `man inetd` for details.

4.7 The XFree86 package

Configuring XFree86 for laptops is a story of its own. And yet again, I have to refer you to the Linux Laptop page where you will find a lot of help on this.

X's console blanking only turns the screen black, but does not turn it off. As mentioned in the sidenote about console blanking, you can use `xset`'s dpms option to change this. However, this feature depends on your laptop's graphics chipset and bios.

Grant Taylor uses the following setup to send his laptop to sleep with the help of `apmd` and the screensaver:

```
# Run xscreensaver with APM program
xscreensaver -timeout 5 \
             -xrm xscreensaver.programs:apm_standby \
             -xrm xscreensaver.colorPrograms:apm_standby &
```

Where ``apm_standby` is a suid perl script that allows only certain people to run `apm -S`."

4.8 The emacs editor

Ok, `emacs` is not an editor, but a way of life. Here's a tip from Florent Chabaud: ``If you use `emacs`, perhaps you have noticed that the editor makes some automatic saves. This is of course useful and should *not* be disabled, but the default parameters can be adjusted to a laptop use.

I have put in the file `/usr/share/emacs/site-lisp/site-start.el` the two following lines:

```
(setq auto-save-interval 2500)
(setq auto-save-timeout nil)
```

This disables auto-saving based on time, and makes the auto-saving be done every 2500 keyboard actions. Of course if you are typing a text this last parameter should be reduced, but for programming it is sufficient. Since every action (up, down, left, backspace, paste, etc...) is counted, 2500 actions are reached very rapidly."

4.9 How to find more ways to optimize

TOC

If your Linux system still seems to access the harddisk too often, you can find out what is going on inside by using the `ps ax` command. This will show all running processes and their full name, sometimes it also reveals the command line arguments of each process.

Now read the `man` page of each process to find out what it does and how to change its behaviour. With this method, you will most likely find the process that is responsible. You may also find `strace` helpful.

Please send me an email if you found something new.

---

5. Appendix

5.1 A message to Linux distributors

If you happen to be a Linux distributor, thank you for reading all this. Laptops are becoming more and more popular, but still most Linux distributions are not very well prepared for portable computing. Please make this document obsolete and change this for your distribution.

- The installation routine should include a configuration, optimized for laptops. The ``mimimal install'' is often not lean enough. There are a lot of things that a laptop user does not need on the road. Just a few examples. There is no need for three different versions of vi (as found in Suse Linux). Most portable systems do not need printing support (they will never be connected to a printer, printing is usually done with the desktop system at home). Quite a few laptops do not need any network support at all.
- Don't forget to describe laptop-specific installation problems, e. g. how to install your distribution without a cd-rom drive or how to setup the plip network driver.
- Add better power management and seamless pcmcia support to your distribution. Add a precompiled kernel and an alternative set of pcmcia drivers with apm support that the user can install on demand. Include a precompiled `apmd` package with your distribution.
- Add support for dynamically switching network configurations. Most Linux laptops travel between locations with different network settings (e. g. the network at home, the network at the office and the network at the university) and have to change the network id very often. Changing a Linux system's network id is a pain with most distributions.

Please mail me if your distribution is optimized for portable computing and what kind of features you added for that. Future versions of this HOWTO will include a section where you can advertise your distribution's laptop features.

5.2 Credits

- The information about battery technology is mostly based on the article ``Stromkonserve'' by Michael Reiter, published in ``c't Magazin fuer Computertechnik'' (Heise Verlag Hannover, Germany), edition 10/96, page 204. Used by permission. Visit their website at http://www.heise.de/.
- The following people contributed to this document:

```
Frithjof Anders <anders@goethe.ucdavis.edu>
David Bateman <dbateman@ee.uts.edu.au>
Florent Chabaud <chabaud@celar.fr>
Markus Gutschke <gutschk@uni-muenster.de>
```

Kenneth E. Harker <kharker@cs.utexas.edu>
Bjoern Kriews <bkr@rrz.uni-hamburg.de>
R. Manmatha <manmatha@bendigo.cs.umass.edu>
Juergen Rink <jr@ct.heise.de>
Grant Taylor <gtaylor@picante.com>
James Youngman <JYoungman@vggas.com>

5.3 About this document

This text mentions batteries 53 times.

The current version of this and many other HOWTOs, most of them a lot more useful than this one, can be found at the main Linux documentation site http://sunsite.unc.edu/mdw/linux.html or at one of its many mirror sites.

Most of this text was written during my trips between Hamburg and Hannover on German rail. (The new ice-2 coaches have power outlets for laptops, yeah!).

And now hum along with me: ``...on the road again..."

BOCA-HOWTO

Installing a Boca 16-port serial card (Boca 2016) with Linux

TOC

Last updated 1 August 1997

This HOWTO is being maintained by David H Dennis, david@freelink.net, to try and help any other individuals who may have acquired a BocaBoard, stared at it and found out that it didn't seem to do anything.

A very high percentage of those interested in the BocaBoard are also interested in becoming Internet Service Providers (ISPs). If you are one of them, please feel free to drop by my web site, http://www.amazing.com , and check out my Inet-Access FAQ, over 9,000 lines detailing how to hook up to the net and how to make your Internet provider business succeed in these most interesting times.

Kevin Traas has issued a helpful update to this FAQ that contains a major correction to the information on `mknod`. He is also standing in as a new source for cables for the device.

W a r n i n g !

Multi-Port serial cards, such as the BocaBoard, are all but dead. If you are in business as an Internet Service Provider (ISP), you will *not* want to use any of these devices, because the new 56k modem standards will not support them. Instead, you need digital modem/terminal server combinations such as the Livingston PM3, Cisco 5200 or US Robotics Total Control. Otherwise, you simply won't be competitive in major markets. Users want high speeds; you won't be able to provide them with a BocaBoard or other multi-port serial card.

Even as a multi-port serial card, the BocaBoard is hopelessly obsolete. The overwhelming majority of people who want to buy a BocaBoard would be better off getting a accelerated card such as the Cyclades or Rocketport. (When last I looked, Rocketport was a better board, but Cyclades had the most stable kernel support). Accelerated cards use less of the CPU of your machine, and so you can support more ports per computer with less performance degradation.

That being said, perhaps you own one of these devices and want to make use of it, or perhaps you are planning an ISP based on a low-cost business model that basically means the use of cast-off equipment from others. In that case, the information in this document should be useful for you. But again, I repeat: *The person who buys a new BocaBoard at this point probably needs a few more clues before starting his ISP or other venture.*

The information here was obtained by begging many people for help. Unfortunately, I have forgotten their names due to a pressing need for disk space on my Netcom account. :-(Nonetheless, I thank them very much for their assistance when things looked blackest. This is my way of paying some of that back.

At the end of this document, there is additional information on a special $ 80 piece of equipment you will need in addition to your BocaBoard, and some comments on Boca customer service and availability.

WHAT DO YOU NEED TO DO TO INSTALL A BOCABOARD?

In outline, the following:

1. The board itself and its manuals
2. Your Linux source tree
3. Patience

Here's the basic procedure:

Recompiling The Kernel

The first step is to change your kernel so that it knows you have a BocaBoard. Unfortunately, this is not a part of the configure script; you must go in and modify the source by hand. This consists of putting the following line at the beginning of linux/drivers/char/serial.c:

```
#define CONFIG_BOCA  1
```

Well, it used to work that way. As of some kernel version or another, this has changed. To make it work with newer kernels, search for the "BOCA_FLAGS" #define in serial.c, and change the value to

```
#define BOCA_FLAGS  ASYNC_BOOT_AUTOCONF
```

You can then recompile your kernel using the instructions included with the source tree. I recommend running the new kernel from a floppy until you're very sure it works; otherwise, the procedure for getting back your system is mind-numbing at best.

At least to me, digging into my kernel and re-compiling it was quite a stressful venture! Relax; as long as you copy it to a floppy, your new kernel is completely harmless. It won't bite! Honest! :-)

Installing The Card

The default address on both the card and the Linux software for the configuration is 0x100; leave that alone. Set the IRQ on the card to Linux' default of 12. If you want to change the IRQ, you will have to search for "BOCA_FLAGS" in the file. You will find lines like this:

```
{ BASE_BAUD, 0x100, 12, BOCA_FLAGS },   /* ttyS16 */
{ BASE_BAUD, 0x108, 12, BOCA_FLAGS },   /* ttyS17 */
...
```

You can change the IRQ from 12 by changing the 12 to any valid IRQ. I have not tried this, however.

Telling Linux About Your Card

Once you have compiled your new kernel, switch off the machine and install the card. Then, turn your machine on with the new kernel floppy in the drive. If the installation succeeded, you should see all sorts of strange stuff about 16550 UARTS being connected to ttyS16-ttyS32. The system will then come up normally.

The odds are pretty good that you don't actually have entries in /dev for those lines. Remember that they start at 16 and go on to 32. If you look at the source code, you'll see why; support for other cards is included in the code for lower line numbers. Creating them is pretty simple, once you know the trick.

TOC

To create entries for dial-out lines (where you call out), type:

```
mknod /dev/cuaxx c 4 N
```

n = 64 + <line number> . The C indicates that this is a character mode device. For example, to create the first couple of lines on your board, type:

```
mknod /dev/cua16 c 4 80
mknod /dev/cua17 c 4 81
 ...
```

To create dial-in lines (where users call you), type

```
mknod /dev/ttySxx c 5 n
```

where N is the same as described above. For example, to create the first couple of lines on your BocaBoard, type:

```
mknod /dev/ttyS16 c 5 80
mknod /dev/ttyS17 c 5 81
  ...
```

It is recommended that you create both dial in and dial out lines for each port, so that you have maximum flexibility. It turns out to be very handy to call another line of your system by activating one of your lines as dial-out and calling your main number. I've done this already and it works great! It's most useful for checking how things look "on the other side of the fence"; I used it to find out how my software looked at 2400bps. (It's slower than the Linux console. A LOT slower, in fact).

Once you've finished with this, you can add entries to your inittab file in the same way as you would for a standard ttySx entry, and the modems or terminals should come up!

Possible Problems

If you have problems, you may find that this is due to conflicts between your Boca board and systems with dual IDE controllers. According to Kevin Traas:

> Is the kernel recognising the board? i.e. Right at the start of the boot process, all sixteen ports should be listed onscreen with ttySxx, I/O Port, and IRQ identified. Also, `setserial -bg ttyS*` should list *every* port in your system.
>
> By default, the BB2016 is configured to use IO Ports between 0100 and 017f. This creates problems on systems with two IDE controllers. The second controller, ide1, uses 0170-0177 which falls into this range. Big problems!!!
>
> So, the solution, in my cases, is to go with an IO base of 0200 through 027f. I haven't had any conflicts in this region; however, you may want to check you system to be sure. (run `cat /proc/ioports` - better run `cat /proc/interrupts` to make sure IRQ 12 isn't used either.)
>
> Once you've found an IO range and IRQ that isn't used, you'll have to change the jumpers on the BB2016 and edit /usr/src/linux/drivers/char/serial.c to reflect the new settings. Then,

compile the kernel and install as per normal.

THE BIG GOTCHA: Something extra you will need, and service comments

TOC

If you want to use your new Boca card with any standard DB-25 RS232 connector, you will have to get a special breakout box, which costs about $ 80. The card itself comes with a breakout box that plugs into the card through a truly formidable cable. It then supplies phone-like cables for the ports. The special $ 80 box has phone-like cables that plug into the breakout box included with the Boca card; you can then plug your modems or terminals into standard RS-232 connectors on the box. Unless you actually have a system that accepts the phone-like connectors, you should add the price of the box to the cost of the card when comparing it to other alternatives.

I got my Boca 2016 card through a special wholesale deal that I don't think many people will be able to reproduce. It was available quickly. However, the breakout box for RS-232 took about two weeks to ship. You should be aware of your need for this box before you acquire the card. The 2016 board was $ 235 and the additional box was $ 79.95.

I've heard that this breakout box is no longer available, and that now people just have special cables made. I'm not sure how you would go about doing this, though.

Only one port of my first Boca 2016 worked. I called Boca, expecting to hear a long string of questions and advice. When they said, "Linux? What's that?" I feared the worst. However, upon hearing my actual problem, they cheerfully told me that the card was defective and I should send it back; full 5-year factory warranty, 30-days exchange. I was struck by how cheery the lady was. It was as though they were expecting the worst, and very kindly making the most of it!

I had my hardware guy exchange the card. Unfortunately, the cards were back-ordered and they took about two more weeks to give me my new card. I have to say that I was quite annoyed at this, since I was anxiously chomping at the bit to get this thing up and running.

Some people have problems with ports past the first eight. I have tested my card up to port 11 and all the ports appear to be working. Boca Research quality control may be lacking; I would recommend that people with troubles with the card call Boca and deal with them directly. I suspect that if I'd exchanged the card directly through the factory I would have had better service than through my dealer. The factory people were all quite nice and eager to please.

Right now, I have only four phone lines. I bought the 16-port card for two reasons: Because it has modem control, unlike the smaller ones, and in anticipation of future expansion.

Some additional information on this subject from Kevin Traas of Baan Business Systems:

> BTW, I've used the BB2016 several times and found to work really well under Linux. I'm extremely happy with them. They perform very well, are very inexpensive, and are reliable. They claim 115200 bps per port. I've got a unique situation whereby I've got two systems with a PPP serial link between them. I'm running this link at 115200 and I experience almost 0% error rate (with hundreds of thousands of packets transmitted). My next step is to implement EQL and additional PPP links between the boxes to increase the bandwidth... We'll see how it goes.

TOC

One final note. Contrary to the suggestion in the Howto, I build my own cables for the BB2016 (they've got RJ ports on them). At first, I "dabbled" with RJ45-8 conductor plugs, but the BB2016 has RJ48-10 conductor jacks, so to get full modem support, I ended up going that route.

I now run all 10 wires between the BB2016 and my modems. It works great! No problems, no regrets! The only initial problem I had was finding a source for RJ48 plugs, 10 conductor wire, and RJ48 to DB25 adapters..... None of these are very cheap due to the low demand. Oh, yeah, and you need an RJ48 crimper as well.

If you want, I can build these cables for you and send them to you. If you're interested, e-mail me and I'll let you know pricing, etc. I found building my own was cheaper than the adapter that is mentioned in the Howto - which I couldn't find a source for anyway.

I believe the adapter mentioned in this document is no longer being manufacturered, leaving cables such as your own the only remaining way of connecting the device.

BogoMips mini-Howto

Wim van Dorst, baron@clifton.hobby.nl

v0, 8 February 1999

This text gives some information about BogoMips, compiled from various sources such as news and e-mail. This text is retrievable from the various Linux archives in .../HOWTO/mini/BogoMips. Translations are available IMHO in Japanese, Italian, German, Polish, and Portuguese from appropriate archives. An explanatory article, titled 'the Quintessential Linux Benchmark,' was published in the Linux Journal, issue January 1996. New mini-Howto entries for unlisted CPUs will be highly appreciated. They can be send per e-mail to the author.

1. Lowest and highest BogoMips ratings

- 1.1 The lowest Linux boot sequence BogoMips value
- 1.2 The highest single-CPU Linux boot sequence BogoMips value
- 1.3 The highest multiple-CPU Linux boot sequence BogoMips value
- 1.4 The highest non-Linux BogoMips value

2. What are BogoMips

3. How to estimate what the proper BogoMips rating should be

4. How to determine what the current BogoMips rating is

5. Variations in BogoMips rating

TOC

6. BogoMips ... failed

7. What about clone CPUs (Cyrix, NexGen, AMD, etc)

8. Why to pay attention to BogoMips

9. Compilation of ratings

- 9.1 Oddly or faultly configured 386 systems
- 9.2 Normal 386 systems: SX, DX, Nexgen
- 9.3 Oddly or faultly configured 486 systems
- 9.4 Normal 486 systems
- 9.5 Normal 486 variations: Cyrix/IBM, UMC
- 9.6 Oddly or faultly configured Pentium systems, or variations
- 9.7 Normal Pentium systems
- 9.8 Normal Pentium variations: MMX, Pro, II, Cyrix, AMD
- 9.9 Normal Alpha systems
- 9.10 Normal Motorola systems
- 9.11 Other Systems: Sparc, PowerPC, Mips, Intel 8088/286 ELKS
- 9.12 Normal Multi CPU systems
- 9.13 Non-Linux systems (reference only)

10. Signature

---

1. Lowest and highest BogoMips ratings

The following are the lowest and the highest BogoMips ratings, are reported directly to me, or published on Usenet.

1.1 The lowest Linux boot sequence BogoMips value

- Tim Van der Linden, `timvdl@innet.be`
- Intel 8088, at 4.77 MHz, ELKS
- 0.02 BogoMips

1.2 The highest single-CPU Linux boot sequence BogoMips value

- Ted E. Suter II, `nexion@home.com`
- AMD K6-2, at 450 MHz
- 901.12 BogoMips

1.3 The highest multiple-CPU Linux boot sequence BogoMips value

- Jeff White, `jwhite@ghq.com`
- Pentium MMX, 2 CPUs, at 233 MHz
- 927.33 BogoMips

1.4 The highest non-Linux BogoMips value

- `omega@sequent.com`
- Sequent NUMA-Q, 32 CPUs P6, at 180 MHz
- Dynix operating system
- 3776.00 BogoMips

2. What are BogoMips

>From Lars Wirzenius' `wirzeniu@kruuna.Helsinki.FI` mail of 9 September 1993, explaining Bogomips, with additional detailed information by Alessandro Rubini, `rubini@norgana.systemy.it`, and by Wim van Dorst:

> `MIPS is short for Millions of Instructions Per Second. It is a measure for the computation speed of a program. Like most such measures, it is more often abused than used properly (it is very difficult to justly compare MIPS for different kinds of computers).
>
> BogoMips are Linus's invention. The kernel (or was it a device driver?) needs a timing loop (the time is too short and/or needs to be too exact for a non-busy-loop method of waiting), which must be calibrated to the processor speed of the machine. Hence, the kernel measures at boot time how fast a certain kind of busy loop runs on a computer. "Bogo" comes from "bogus", i.e, something which is a fake. Hence, the BogoMips value gives some indication of the processor speed, but it is way too unscientific to be called anything but BogoMips.
>
> The reasons (there are two) it is printed during bootup is that a) it is slightly useful for debugging and for checking that the computers caches and turbo button work, and b) Linus loves to chuckle when he sees confused people on the news.'

BogoMips are being determined in `/usr/src/linux/init/main.c` (simple C algorithm), and the pertaining kernel variable `loops_per_sec` is used in several drivers for more serious purpose. The actual delay function `udelay()` is in assembler, and therefore each port has its own definition in `/include/asm/delay.h`. The `loops_per_sec` variable and the `udelay()` function are used in numerous drivers, see:

```
cd /usr/src/linux #or where else source is located
find . -name '*.[hcS]' -exec fgrep loops_per_sec {} /dev/null \;
find . -name '*.[hcS]' -exec fgrep udelay {} /dev/null \;
```

3. How to estimate what the proper BogoMips rating should be

>From a initiative by Ian Jackson, `ijackson@nyx.cs.du.edu`, and Przemek Klosowski, much updated and expanded by Wim van Dorst for current data, as listed below:

As a very approximate guide, the BogoMips can be calculated by:

System	BogoMips	Comparison
Intel 8088	clock * 0.004	0.02
Intel/AMD 386SX	clock * 0.14	0.8
Intel/AMD 386DX	clock * 0.18	1 (definition)
Motorola 68030	clock * 0.25	1.4
Cyrix/IBM 486	clock * 0.34	1.8
Intel Pentium	clock * 0.40	2.2
Intel 486	clock * 0.50	2.8
AMD 5x86	clock * 0.50	2.8
Mips R4000/R4400	clock * 0.50	2.8
Nexgen Nx586	clock * 0.75	4.2
PowerPC 601	clock * 0.84	4.7
Alpha 21064/21064A	clock * 0.99	5.5
Alpha 21066/21066A	clock * 0.99	5.5
Alpha 21164/21164A	clock * 0.99	5.5
Intel Pentium Pro	clock * 0.99	5.5
Cyrix 5x86/6x86	clock * 1.00	5.6
Intel Pentium II	clock * 1.00	5.6
Intel Celeron	clock * 1.00	5.6
Mips R4600	clock * 1.00	5.6
Alpha 21264	clock * 1.99	11.1
AMD K5/K6/K6-2	clock * 2.00	11.1
UltraSparc II	clock * 2.00	11.1
Pentium MMX	clock * 2.00	11.1
PowerPC 604	clock * 2.00	11.1
Motorola 68060	clock * 2.01	11.2

Note that the BogoMips calculation loop does not take full advantage of the parallelism of various processors, such as the Intel Pentium and the Alpha 21164.

Note that the BogoMips calculation loop for the non-Intel CPUs is similar but not the same.

4. How to determine what the current BogoMips rating is

There are three methods to determing the current BogoMips, viz.

1. looking in `/proc/cpuinfo`, e.g., with `` `cat /proc/cpuinfo` ``.
2. looking in the syslog output to see what was printed there during booting (if necessary retrieving the information explicitly with dmesg or syslogk) or
3. using the standalone bogomips program.

A non-determinative alternative, which is also applicable for non-Linux systems such as Crays and so, may be the standalone BogoMips program. From the readme file by Jeff Tranter, `jeff_tranter@mitel.com`:

> `Tired of rebooting your system so you can see how many BogoMIPS it's running at today? [...] "Bogomips" is a standalone program that displays your system performance using one of the world's most recognized benchmarks. It uses the same code that is used in the Linux kernel while booting, but runs as a user program. [...] Version 1.3 of BogoMIPs is now portable and should run on any system that supports an ANSI C compiler and library.'

Note that due to system load values calculated with the standalone program may be lower than registered in the list below. Intrinsically the standalone cannot give precisely similar information to the boot sequence BogoMips, since system load will compete with this program run by an ordinary user. Therefore only boot sequence BogoMips ratings are listed below.

Be aware that the file `sunsite.unc.edu:/pub/Linux/system/status/bogo-1.2.tar.gz/` contains the latest version 1.3 (sic), which is yet rather outdated.

5. Variations in BogoMips rating

>From Linus Torvalds, `torvalds@cc.helsinki.fi`, explaining about the variation one may see in the BogoMips rating, in c.o.l.development, at 28 April 1994

> `The BogoMips calculation loop is "quantizised", so you're most likely to get the exact same number all the time. You usually will get different numbers only if the speed is just on the "edge", when small variations (different time for interrupt ticks etc) will make it jump from one value to the other.'

If a kernel is not compiled specifically for the pertaining CPU, also some (even large) variations of the BogoMips ratings can occur, mainly due to erroneous alignment. This problem apparently only occurs on the various x86 CPUs (Intel and clones). Fortunately it can easily be solved: recompile the kernel specifically for your CPU.

Note that the BogoMips algorithm, contrarily to popular believe, did NOT change significantly over the various kernel versions.

6. BogoMips ... failed

Suggested by various questions on the net and private mail, e.g., by Lily, `lbliao@alumni.caltech.edu`, and by Pierre Frenkiel, `frenkiel@cdfap2.in2p3.fr`. In March 1995 they asked:

> `When I boot Linux I get the message:
>
> ```
> Calibrating delay loop.. ok - 23.96 BogoMips
> failed
> ```
>
> Where/why has the calibration delay loop failed?'

It didn't fail. If it had failed the text would have been

```
Calibrating delay loop.. failed
```

What likely did fail was a driver for some gadget which you may not have in your machine. Just after calculating the BogoMips rating all device drivers are initiated. First the SCSI devices, then Net devices, etc. Any failure is duly reported. Noteworthy is the AHA152x driver. Other effects of failing drivers (and not of failing BogoMips calculations) are systems crashes, long waits, and complete system locks.

Since Linux 1.2 many error messages have improved, so upgrade to at least that version to find out which particular driver it is that is failing.

7. What about clone CPUs (Cyrix, NexGen, AMD, etc)

Cyrix 486-like CPUs need cache enabling software, sometimes referred to as BogoBoost software. Cyrix 5x86 and 6x86 CPUs may have their BogoMips improved drastically by branch-prediction (BIOS option). Note that the performance improvement may be marginal. There are several packages available for adjusting Cyrix CPUs, such as the bogoboost patch, cx5x86mod, and set6x86, all from the normal archives, in obvious places. It is reported the Cyrix 6x86 CPUs may give better performance when the kernel is compiled with 486-optimization, instead Pentium-optimization.

NexGen 386-enhanced CPUs, marked as Nx586, are listed as 386-like, since the fact that they are performing like Pentium machines is not relevant to BogoMips.

AMD 5x86, also denoted as AMD 486DX5, are quadrupled 486/33 machines. They are fully in line with other 486 CPUs. The AMD K5 and the K6 are Pentium-like CPUs, with their own BogoMips multipliers.

8. Why to pay attention to BogoMips

Let me add that there are only two reasons for paying attention to the BogoMips rating that is presented on booting Linux:

1. To see whether it is in the proper range for the particular processor, its clock frequency, and the potentially present cache. Many CPUs are prone to faulty setups of
   - memory cache setting (write-back is wrong for BogoMips, often reported lower than 5; write-through is ok)
   - turbo-buttons (should be ON)
   - BIOS-software emulated fake cache (change it for real cache)
   - similar cache and clock related things, sometimes also BIOS-software related
2. To see whether your system is faster than mine. Of course this is completely wrong, unreliable, ill-founded, and utterly useless, but all benchmarks suffer from this same problem. So why not use it? This inherent stupidity has never before stopped people from using benchmarks, has it? :-)

Note that more serious uses for real benchmarking are addressed in the Linux Benchmarking Howto by André D. Balsa.

9. Compilation of ratings

The following table gives some reported BogoMips ratings for various systems (about 1000 entries by about 800 different persons, from about 50 different countries. Note that the ratings here are from the actual Linux booting sequence, except of course for the section on Non-Linux Systems.

9.1 Oddly or faultly configured 386 systems

System	BogoMips	Reporter
386DX/16 387 nocache	0.57	H. Peter Anvin <hpa@nwu.edu>
386DX/25	0.82	P Wright <philip.wright@purplet.demon.co.uk>
386DX/25 nocache	1.03	Mark A. Horton <mahmha@crl.com>
386SX/16	1.5	Stefan Kromer <sk@galaxy.sunflower.sub.org>
386SX/16	1.6	Bill Davidsen <davidsen@tmr.com>
386SX/20	1.87	Paul C. Dulany <pcdulany@wam.umd.edu>
386SX/20	2.45	Roger Harkess <roger@visi.com>
386DX/25(?) 128c	6.03	Chuck Meo <meo@solbourne.com>
386DX/20	13	Ed Runnion <erunnio@hubcap.clemson.edu>

9.2 Normal 386 systems: SX, DX, Nexgen

System	BogoMips	Reporter
386SX/8 undercl	1.04	Andrew Costa <c_chaos@wahnapitae.on.ca>
386SX/12	1.78	Klaus Kettner <kk@sesom.de>
386SX/16	1.99	James Vahn <jvahn@short.circuit.com>
386SX/16 Packard Bell	2.05	<root@Belvedere\%hip-hop.suvl.ca.us>
386SX/16	2.09	David E. Fox <dfox@belvedere.sbay.org>
386SX/16	2.15	W Stevens <wgsteven@math.uwaterloo.co>
386SX/16	2.2	Lech Marcinkowski <puolalm@tekla.fi>
386SX/16	2.23	Andrew Bulhak <acb@yoyo.cc.monash.edu.au>
386SX/16	2.23	Steven M. Gallo <smgallo@cs.buffalo.edu>
386SX/16	2.34	Kevin Burtch <kburtch@pts.mot.com>
386SX/16 turbo	2.38	Andrew Haylett <ajh@gec-mrc.co.uk>
386SX/16 0c	2.43	Adam Clarke <adamc@loose.apana.org.au>
386SX/16	2.49	Waymon <waymon@pacifier.com>
386SX/20	2.7	Alex Strasheim <astrashe@nyx.cs.du.edu>
386SX/20	2.70	J.L. Brothers <brothers@halcyon.com>
386SX/20	2.89	Anders Stenback <stenback@hehe.com>
386SXL/25 AMD	2.9	Vaughan R. Pratt <pratt@sunburn.stanford.edu>
386SX/25 AMD 0c	3.06	K.J. MacDonald <kenny@festival.ed.ac.uk>
386SX/25 AMD	3.38	Hamish Coleman <hamish@zot.apana.org.au>
386SX/25 0c	3.52	Rogier Wolff <r.e.wolff@et.tudelft.nl>
386SL/25 Intel	3.57	S Harris <harris@teaching.physics.ox.ac.uk>
386SX/25 AMD	3.62	S Harris <harris@teaching.physics.ox.ac.uk>
386SXL/25 AMD 0c	3.71	David E.A. Wilson <david@cs.uow.edu.au>

TOC

386SX/33	Intel	4.06	Kenneth J. Hoover <ken@psuedvax.psu.edu>
386SX/33		4.71	Alexander Komlik <apkom@l.ukrcom.kherson.ua>
386SX/40	Intel 0c	6.03	Michael Kenyon <u3g12@keele.ac.uk>
386DX/16		2.49	Mike <mike@emgee.demon.co.uk>
386DX/20	Intel	3.0	Malcolm Reeves <reeves@rocky1.usask.ca>
386DX/20	Intel	3.08	Si. Harris <harris@teaching.physics.ox.ac.uk>
386DX/20	Nec Powermate	3.22	David J Dawkins <davidd@isl.co.uk>
386DX/20	Micronics	3.25	M Haardt <u31b3hs@informatik.rwth-aachen.de>
386DX/20		3.67	Joost Helberg <jhelberg@nlsun8.oracle.nl>
386DX/25		3.91	Ian McCloghrie <imcclogh@cs.ucsd.edu>
386DX/25		3.95	Grant Edwards <grante@aquarius.rosemount.com>
386DX/25	0cache	3.96	J.O. Williams <jow@techbase.com>
386DX/25	32cache	4.53	J.M.A. Lahtinen <jmalahti@klaava.Helsinki.FI>
386DX/33		5.86	Tim Lacy <timla@microsoft.com>
386DX/33	64cache	5.99	Lars Wirzenius <wirzeniu@kruuna.Helsinki.FI>
386DX/33	Intel	5.99	Harri Pasanen <hpasanen@cs.hut.fi>
386DX/33	no387	6.03	Joel B.Levin <levin@bbn.com>
386DX/33	387	6.03	Peter Bechtold <peter@fns.greenie.muc.de>
386DX/40		6.21	J.L. Brothers <brothers@halcyon.com>
386DX/33		6.46	Dennis Robinson <djrobins@uxa.cso.uiuc.edu>
386DX/33		6.5	Dean Nelson <deannelson@aol.com>
386DX/33	387 256cache	6.65	Wim van Dorst <baron@clifton.hobby.nl>
386DX/33		6.65	Rick Lim <ricklim@opus.freenet.vancouver.bc.ca>
386DX/33		6.7	Craig Hagan <hagan@cih.com>
386DX/40		6.99	Ken Wilcox <wilcox@math.psu.edu>
386DX/40	AMD	7.76	Joe Phillips <rchandra@letter.com>
386DX/40	AMD	7.10	Kerry Person <kperson@plains.NoDak.edu>
386DX/40		7.10	D. Bikram Singh <a336dhal@cdf.toronto.edu>
386DX/40	128cache	7.23	Julian Francis Day <jfd0@aber.ac.uk>
386DX/40	bogoboosted	7.23	Pat St Jean <stjean@math.enmu.edu>
386DX/40	AMD 128cache	7.23	R.Bergs <rabe@akela.informatik.rwth-aachen.de>
386DX/40	slow DRAM	7.26	John Lockwood <lockwood@pan.vlsi.uiuc.edu>
386DX/40	128c	7.29	Karsten Friese <ftdkafr@ftd.ericsson.se>
386DX/40		7.29	E.C. Garrison <ericg@nickel.ucs.indiana.edu>
386DX/40		7.29	Darin Cowan <cowan@rubicon.org>
386DX/40		7.29	Bonne van Dijk <bonne@cs.utwente.nl>
386DX/40	AMD	7.76	Todd Lindner <tlindner@panix.com>
386DX/40		7.76	Bear Giles <bear@indra.com>
386DX/40	AMD 387 64c	7.91	<wires@gnu.ai.mit.edu>
386DX/40		7.98	Frank Pilhofer <fp@informatik.uni-frankfurt.de>
386DX/40	64c	7.98	Dean Junk <dpjunk@mm.com>
386DX/40	AMD 32c	7.98	Tommy Olsen <tommyo@ifi.uio.no>
386DX/40	AMD	7.98	James Reith <reith@racores.com>
386DX/40		7.98	Aaron T. Baldie <atb@u.washington.edu>
386DX/40	128c	7.98	John Pate <jpate@easynet.co.uk>
386DX/40		7.98	Christian Nelson <cnelson@csugrad.cs.vt.edu>
386DX/40		7.98	Alan Peckham <peckham@drei.enet.dec.com>
386DX/40		8.06	Michael Guslick <michaelg@alpha2.csd.uwm.edu>
386DX/40		8.06	Richard Brown <brown@midget.towson.edu>
386DX/40		8.06	Bill G. Bohling <bs146@tali.uchsc.edu>

TOC

Nx586/90 NexGen	67.44	<root@wgw.mnsinc.com>
Nx586/90 NexGen	67.44	Robert Gehring <rag@cs.tu-berlin.de>
Nx586/90 NexGen	67.48	David G. Eckard <dgeckard@eos.ncsu.edu>
Nx586/100 NexGen	74.34	Cameron L. Spitzer <cls@truffala.sj.ca.us>
Nx586/100 NexGen 256c	74.56	Marius Groenendijk <marius@cray-systems.lu>
Nx586/110 NexGen 256c	81.51	Michael J. Micek <mmicek@muddcs.cs.hmc.edu>
Nx586/110 NexGen	81.51	Ron Marsh <rmarsh@plains.nodak.edu>

9.3 Oddly or faultly configured 486 systems

System	BogoMips	Reporter
486DX/33 0c	1.45	Mark Gray <vatavian@gvu1.gatech.edu>
486SL/25 0c	1.95	Paraskevas Evripidou <skevos@seas.smu.edu>
486DLC/40 0c	2.45	S.Schendel <sschend@magnus.acs.ohio-state.edu>
486DX/33 128c	2.94	P.J. Nefkens <p.nefkens@student.utwente.nl>
486DX4/120 AMD	3.04	Andrew Steinbach <stei0113@maroon.tc.umn.edu>
486DX5/133 AMD	3.05	Eric Hagen <ehagen@hawaii.edu>
486DX4/100 Cyrix	3.06	Stuart Harvey <sharvey@primenet.com>
486DX5/133 AMD	3.06	Charles Galpin <chg@severn.wash.inmet.com>
486DX4/100	3.06	Bear Giles <bear@indra.com>
486DX2/80	3.08	Gerald E. Butler <gbutler@phoenix.kent.edu>
486DX4/120 AMD	3.08	Charles Hines <chuck_hines@vnet.ibm.com>
486DX4/66 256c	3.10	Riccardo Capella <mc8508@mclink.it>
486DX4/100 wb-cache	3.10	Paul Close <pdc@sgi.com>
486DX4/120	3.13	Brian Perkins <bperkins@netspace.com>
486DX4/120 AMD	3.15	<eruston@net2.intserv.com>
486DX4/100	3.17	Thomas Sudbrak <sudbrak@borneo.gmd.de>
486SLC2/50 Cyrix	3.30	Colin J. Wynne <cwynne@sage.wlu.edu>
486DX/33	3.61	Marten van de Laan <marten@cs.rug.nl>
486DX/33 noturbo	3.61	Dimitris Evmorfopoulos <devmorfo@mtu.edu>
486DX4/120	3.74	Brian Wheeler <bdwheele@indiana.edu>
486DX4/120 AMD	3.74	Frank Pilhofer <fp@informatik.uni-frankfurt.de>
486DX4/100 Cyrix 256c	4	Joel Kelso <joel@cs.murdoch.edu.au>
486DX/33 256c noturbo	4.25	Wouter Liefting <wlieftin@cs.vu.nl>
486DX/33	4.66	Mark Gray <vatavian@gvu1.gatech.edu>
486Rx2 Cyrix 25/50	4.85	<cosc19v2@menudo.uh.edu>
486SX/33 noturbo	5.21	Scott D. Heavner <sdh@fishmonger.uucp>
486DX2/66 overdrive	5.37	Jeremy Orr <jeremy@careercenter.sfsu.edu>
486DX/33	5.66	Ryan Tucker <rtucker@ttgcitn.com>
486DX2/66	5.88	P.J. Nefkens <p.nefkens@student.utwente.nl>
486DX4/100	5.94	Howard Goldstein <hg@n2wx.ampr.org>
486DX4/100 AMD	5.94	Mr Pink <vince@dallas.demon.co.uk>
486DX4/100 notebook	6.55	Thomas <tom@dirac.physik.uni-konstanz.de>
486DX4/100 notebook	6.55	Hugh McCurdy <hmccurdy@ix.netcom.com>
486SLC Cyrix	7	Pieter Verhaeghe <pive@uia.ac.be>
486SX/33	7.84	Paul Hedderly <prh6@unix.york.ac.uk>
486DLC/40	7.98	Wil Cromer <nwc2@Ra.MsState.Edu>
486DX/33 256c	8.27	Rohan Tronson <rohan@kihi.com.au>
486DX4/100	11.11	NN <usenet@uxmail.ust.hk>
486DX4/100	11.3	Earl Gooch <egooch@mc.com>

TOC

486/66 Cyrix	13.02	Mike Baptiste <baptiste@bnr.ca>
486SLC2/25	14.6	Vaughan R. Pratt <pratt@Sunburn.Stanford.ED
486DX2/66 laptop	14.46	Robert Knop <rknop@netcom.com>
486SLC2/66	18.94	<root@avalon.net>
486DX/33 turbo	19.98	C Vetter <cbvetter@informatik.th-darmstadt.de>
486DX4/75	21.5	Theo Scott <rkwtgs@pukrs3.puk.ac.za>
486DX4/75	24.13	Sherman Hsieh <shieh@csua.berkeley.edu>
486DX2/58	26.3	Vassili Leonov <leonov@iedv7.acd.com>
486DX4/100 overclock	28.67	Theo Scott <rkwtgs@pukrs3.puk.ac.za>
486DX2/80	36	Mark Lee <mlee@heartlab.rri.uwo.ca>
486DX2/80	50.08	Mark Lee <mlee@heartlab.rri.uwo.ca>
486DX4/100	60	Sebastien Dedieu <dedieu@emi.u-bordeaux.fr>
486DX2/100 overclock	60.45	Tony D Shan <tdsst9+@pitt.edu>
486DX5/133 AMD	75.40	Jeff Hyche <jwhyche@scott.net>
486DX5/133 AMD	80.08	NN <guesta@slip-29-7.ots.utexas.edu>
486DX5/133 AMD	87	John Wiggins <jwiggins@comp.uark.edu>

9.4 Normal 486 systems

System	BogoMips	Reporter
486SX/20 DECpc	9.98	Thomas Pfau <pfau@cnj.digex.com>
486SX/25	12.24	M. Buchenrieder <mibu@scrum.greenie.muc.de>
486SX/25	12.3	Darren McKay <e9bh@unb.ca>
486SX/25	12.42	Mark R. Lindsey <mlindsey@nyx.cs.du.edu>
486DX/25	12.5	Phillip Hardy <phillip@mserve.kiwi.gen.nz>
486SX/25	12.52	Emmanual Emore <emor7672@elan.rowan.edu>
486DX/33 256c	16.33	Eric Kemminan <ekemmina@pms709.ms.ford.com>
486DX/33	16.35	Christopher L. Morrow <cm43@andrew.cmu.edu>
486DX/33	16.43	Rob Janssen <pe1chl@amsat.org>
486DX/33 64cache	16.44	H. Peter Anvin <hpa@nwu.edu>
486DX/33 256c DIY	16.44	Wouter Liefting <wlieftin@cs.vu.nl>
486DX/33 Intel 128c	16.44	Rafal Kustra <g1krakow@cdf.toronto.edu>
486DX/33	16.5	Alex Freed <freed@europa.orion.adobe.com>
486DX/33	16.6	Vaughan R. Pratt <pratt@Sunburn.Stanford.EDU>
486DX/33 noturbo	16.61	C Vetter <cbvetter@informatik.th-darmstadt.de>
486DX/33	16.61	Jeffrey L. Newbern <jnewbern@athena.mit.edu>
486DX/33	16.61	Giuseppe De Marco <gdemarco@freenet.hut.fi>
486DX/33	16.61	M Heuler <heuler@informatik.uni-wuerzburg.de>
486DX/33	16.61	Frank Lofaro <ftlofaro@unlv.edu>
486SX/33	16.64	Jacob Papenfuss <webmaster@the-corridor.com>
486DX/33	16.77	Donald Lewis <dlewis@jackson.freenet.org>
486DX/33	16.77	Stephan Boettcher <staphan@alzt.tau.ac.il>
486DX/33 256c	16.77	David Manchester <mustang@tartarus.uwa.edu.au>
486DX/40	19.8	Jose Calhariz <cal@minerva.inesc.pt>
486DX/40	19.91	M Heuler <heuler@informatik.uni-wuerzburg.de>
486DX/40	19.96	David A. Ranch <dranch@ecst.csuchico.edu>
486DX/40 AMD	19.97	M Haardt <u31b3hs@informatik.RWTH-Aachen.DE>
486DX/40 Intel	19.97	Paul van Spronsen <vspr@teppic.sun.ac.za>
486DX/40	19.97	Ulf Tietz <ulf@rio70.bln.sni.de>
486DX/40	19.97	<Eberhard_Moenkeberg@p27.rollo.central.de>

TOC

486DX/40	19.97	Zoltan Lajber <lajbi@lajli.gau.hu>
486DX/40	19.97	Wim van Dorst <baron@wiesje.hobby.nl>
486DX/40 AMD	20	Chuck Munro <chuckm@canada.hp.com>
486DX/40 AMD	20.09	Pieter Eendebak <peendebak@bbsw.idn.nl>
486DX/50	24.48	Arnd Gehrmann <arnd@rea>
486DX/50 AMD	24.85	Klaas Hemstra <hst@mh.nl>
486DX/50 DTK	24.85	Randolph Christophers <randyc@lna.oz.au>
486DX/50	24.85	Kevin Lentin <kevinl@bruce.cs.monash.edu.au>
486DX2/50	24.85	Jason Matthew <jmatthew@kn.pacbell.com>
486DX2/50	24.85	Gregory P. Smith <smithgr@cs.colorado.edu>
486DX/50 VLB	24.97	Tom Miller <tvtom@en.com>
486DX/50	24.99	Jeff <css@erols.com>
486DX/50 Intel 256c	24.99	Mike <mike@emgee.demon.co.uk>
486DX/50	25	Robert Herzog <rherzog@rc1.vub.ac.be>
486DX2/50	25	M. Abrahamsson <swmike@uplift.df.lth.se>
486DX2/50	25.0	Christian Holtje <choltje@ux1.cso.uiuc.edu>
486DX2/50 DECpc	25.04	Thomas Pfau <pfau@cnj.digex.com>
486DX2/50 Eisa	25.04	John Willing <willing@cimage.com>
486DX2/50 256c	25.04	Zhou Yanmo <zhou@gauss.math.usf.edu>
486DX/50	25.04	Michael Kress <kress@hal.saar.de>
486DX2/50	25.04	Mats Wikholm <mwikholm@news.abo.fi>
486DX2/50	25.04	Jean C Delepine <delepine@linux.u-picardia.fr>
486DX/50	25.04	Jean C Delepine <delepine@linux.u-picardia.fr>
486DX/50	25.04	Kevin Burtch <kburtch@pts.mot.com>
486DX/50 notebook	25.04	Pierre Frenkiel <frenkiel@cdfap1.in2p3.fr>
486DX/50	25.10	M Heuler <heuler@informatik.uni-wuerzburg.edu>
486DX2/50	25.4	Brian Kennedy <bkenned@hubcap.clemson.edu>
486DX2/66	32	Lee Sau Dan <h9210876@khuxa.hku.hk>
486DX2/66	32.9	Frederick <niles@axp745.gsfc.nasa.gov>
486DX2/66	33	Alec Muffett <alecm@uk-usenet.uk.sun.com>
486DX2/66	33	NN <coolefa@pmifeg.com>
486DX2/66	33	Steve Tinney <sjt@enlil.museum.upenn.edu>
486DX2/66 Intel	33	Chuck Munro <chuckm@canada.hp.com>
486DX2/66 VLB	33.0	Sebastien Dedieu <dedieu@emi.u-bordeaux.fr>
486DX2/66 AMD	33.05	G. Skinner <gskinner@gwsunix1.crystalball.com>
486DX2/66 VLB	33.18	Pablo Iranzo Gomez <iranzop@gong.ci.uv.es>
486DX2/66 VLB	33.18	Dag Wieers <dag@digibel.be>
486DX2/66	33.20	Arnd Gehrmann <arnd@rea.com>
486DX2/66 Intel/PCI	33.22	C. Menke <carsten.menke@post.uni-bielefeld.de>
486DX2/66	33.22	Brian Ricker <gt2327c@prism.gatech.edu>
486DX2/66	33.22	Don Bennett < <don@engr.mun.ca>
486DX2/66	33.22	Robert Heller <heller@cs.umass.edu>
486DX2/66	33.22	Warwick Ward-Cox <wwar@lostlink.alt.za>
486DX2/66	33.22	Chien-An Chen <giant@nwu.edu>
486DX2/66 Eisa/VL	33.22	Serge <sviznyuk@magnus.acs.ohio-state.edu>
486DX2/66 AMD	33.22	Wayne Robinson <wayner@renoir.cftnet.com>
486DX2/66 Intel	33.22	Jim Barber <yeul@marsh.cs.martin.edu.au>
486DX2/66	33.22	Tom Lowery <tlowery@mcs.kent.edu>
486DX2/66	33.27	S Viznyuk <sviznyuk@magnus.acs.ohio-state.edu>
486SX2/66	33.28	Jacob Papenfuss <root@the-corridor.com>
486DX2/66	33.3	Devon Tuck <devon@netcom.com>
486DX2/66 256cache	33.4	H. Peter Anvin <hpa@nwu.edu>

TOC

486DX2/66	33.5	Jongyoon Lee <mr2@netcom.com>
486DX2/66	33.5	Petrovsky Alexey <gong@cs.msu.su>
486DX2/66	33.5	Sung Lee <slee2@umbc.edu>
486DX2/66	33.55	Gene McCulley <mcculley@greatwall.cctt.com>
486DX2/66	33.55	W. Zeilinger <wzeil@doradus.ast.univie.ac.at>
486DX2/66	33.55	Donald Lewis <dlewis@jackson.freenet.org>
486DX2/66	33.55	Eric Malkowski <malk@world.std.com>
486DX2/66 0c	33.55	Chris Petit <mystere@ix.net.com>
486DX2/66	33.55	<al-b@minster.york.ac.uk>
486DX2/66	33.55	Jesper de Jong <jesper@cas.et.tudelft.nl>
486DX2/66	33.55	John Paul Morrison <jmorriso@bogomips.com>
486DX2/66	33.55	Arash <ei39594@ios.chalmers.se>
486DX2/66	33.55	Ralph Lewis <rlewis@mail.wsu.edu>
486DX2/66	33.55	Ulisses Alonso Camaro <alonso@bebe.uv.es>
486DX2/66	33.55	Bussmann <bussmann@wolpi.infomatik.uni-bonn.de>
486DX2/66 Intel/PCI	33.55	Louis J. LaBash <labash@lcjones.aclib.siue.edu>
486DX2/66 Intel	33.55	Andrew Tubbiolo <enigma@seds.lpl.arizona.edu>
486DX2/66	33.55	W Fink <werner.fink@physik.uni-stuttgart.de>
486DX2/66 ICL	33.55	Mathias Koerber <mathias@solomon.technet.sg>
486DX2/66	33.55	Bill Pogue <gwp@dithots.dithots.org>
486DX2/66 256c	33.58	Theo Scott <rkwtgs@pukrs3.puk.ac.za>
486DX2/66	33.7	C Triantafillou <triant@pegasus.montcleair.edu>
486DX2/66 256c Intel	33.81	S Harris <harris@teaching.physics.ox.ac.uk>
486DX2/66	33.9	Magnus Back <erambk@eraj.ericsson.se>
486DX2/66 notebook	33.9	Robert A Knop <rknop@mop.caltech.edu>
486DX2/66	34.06	Al Clark <aclark@netcom.com>
486DX4/75	37.47	G Asmundarson <grettir@wordperfect.com>
486DX2/80	39.93	Andrew Tubbiolo <enigma@seds.lpl.arizona.edu>
486DX2/80 overcl/66	39.94	Mario L. Guttierez <mgutier@mentor.sdu.edu>
486DX2/80 AMD	39.94	Corey D Brenner <brenner@umr.edu>
486DX2/80	39.94	Dan Delaney <cgdela01@homer.louisville.edu>
486DX2/80	39.94	D t Haar <danny@caution.cistron.nl.mugnet.org>
486DX2/80 overcl	39.94	Peter Suetterlin <ps@kis.uni-freiburg.de>
486DX2/80 AMD	39.94	JL Gomez <kitana!sysop@caprica.com>
486DX2/80 AMD	39.94	Pete Krawczyk <pkrawczy@uiuc.edu>
486DX2/80 AMD	40	Rene Baart <baart@simplex.nl>
486DX2/80 AMD	40	Wolfgang Kalthoff <wo@rio70.bln.sni.de>
486DX2/80	40.0	Rick Brown <ccastrb@prism.gatech.edu>
486DX2/80 AMD	40.14	Jon Lewis <jlewis@inorganic5.chem.ufl.edu>
486DX2/80 AMD	40.14	Richard S. Stone <rstone@edgp.com>
486DX2/80	40.15	Oleg <oleg@hpcms.co.il>
486DX2/80 AMD	40.18	Adri Verhoef <a3@a3.xs4all.nl>
486DX2/80	40.18	Mats Andtbacka <mandtback@abo.fi>
486DX2/100 AMD overcl	49.14	Jon Lewis <jlewis@inorganic5.chem.ufl.edu>
5x86/100 AMD undercl	49.66	NN <root@tailor.aleim.net>
486DX4/100	49.66	Pete Cervasio <cervasio@airmail.net>
486DX4/100 256c	49.71	Lutz Pressler <lutz.pressler@med-stat.GWDG.de>
486DX4/100	49.71	Brett Gersekowski <bgrerseko@powerup.com.au>
486DX4/100 Intel 256c	49.77	Angelo Haritsis <ah@doc.ic.ac.uk>
5x86/100 AMB undercl	49.77	Bernd Hentig <bernd.hentig@guug.de>
486DX4/100	49.78	Aurel Balmosan <aurel@xylo.owl.de>
486DX4/100	49.87	Chris Saia <minkie@concentric.net>

TOC

486DX4/100	49.87	Aaron Brick <abrick@dcc.uchile.cl>
486DX4/100	50	Donald Lewis <dlewis@jackson.freenet.org>
486DX4/100	50.02	Peter Skov Knudsen <gogol@ask.diku.dk>
486DX4/100	50.02	Shadow Weaver <djamison@students.wisc.edu>
486DX4/100 AMD	50.3	Dave <shodan@shodan.clark.net>
486DX4/100 AMD	50.04	Tony Smolar <asmolar@fast.net>
486DX4/100	50.05	fredk <fredk@shadow.net>
486DX4/100	50.06	Ronald Prague <ronp@fisnet.net>
486DX4/100	50.08	Matt Gisher <matt@matt.fidalgo.net>
486DX4/100	50.08	Steven A. Duchene <sduchene@cis.ysu.edu>
486DX4/100	50.08	Miles O'Neal <meo@schoneal.com>
486DX4/100	50.08	Will <zxmvg07@hp12.zdv.uni-tuebingen.de>
486DX4/100	50.08	Piet de Bondt <bondt@dutiws.twi.tudelft.nl>
486DX4/100 laptop	50.08	Karl Kleinpaste <karl_kleinpaste@cs.cmu.edu>
486DX4/100 256c	50.08	Thomas Kanschik <y0000997@ws.rz.tu-bs.de>
486DX4/100	50.08	Linas Vepstas <linas@fc.net>
486DX4/100	50.08	Ed Daiga <daiga@engin.umich.edu>
486DX4/100 notebook	50.08	Gerry Quejada <fd863@cleveland.freenet.edu>
486DX4/100 AMD	50.08	B Schuller <schuller@ind136a.wi.leidenuniv.nl>
486DX4/100	50.08	J.L. Brothers <brothers@halcyon.com>
486DX4/100	50.08	David E.A. Wilson <david@cs.uow.edu.au>
486DX4/100	50.08	Mark Lumsden <root@titan2.physics.mcmaster.ca>
486DX4/100	50.08	Ashar <ashar@netcom12.netcom.com>
486DX4/100	50.08	Jacob Waltz <waltz@pcjiw.lampf.lanl.gov>
486DX4/100	50.08	Tom Sinclair <sinner@cafe.net>
486DX4/100 AMD	50.08	G. Skinner <gskinner@gwsunix1.crystalball.com>
486DX4/100 AMD	50.08	Nick Savoiu <nick@ritz.mordor.com>
486DX4/100	50.08	Thomas J Fisher <twb5odt@nmia.com>
486DX4/100	50.08	Pascal Pensa <pensa@aurora.unice.fr>
486DX4/100	50.08	Julian Bradbury <julian@xabcs.demon.co.uk>
486DX4/100	50.51	Frederic Potter <frederic@swing.ibp.fr>
486DX4/100	50.66	Bill Stegers <bill_ste@zeelandnet.nl>
486DX4/120 256c	59.1	Kevin <kalichwa@oakland.edu>
486DX4/120 AMD	59.80	Mark Tranchant <mat92@ecs.soton.ac.uk>
486DX4/120 AMD	59.80	Fred Broce <fbroce@atlanta.com>
486DX4/120 AMD	59.90	Marko Ovaska <ovaska@cc.helsinki.fi>
486DX4/120 AMD	59.80	Bob Purdon <bobp@mpx.com.au>
486DX4/120 AMD	59.80	Pat Young <dice@netbsd.warped.com>
486DX4/120	59.91	Will <zxmvg07@hp12.zdv.uni-tuebingen.de>
486DX4/120 AMD 256c	60.01	Angelo Haritsis <ah@doc.ic.ac.uk>
486DX4/120 overcl	60.45	Pascal Pensa <pensa@aurora.unice.fr>
486DX4/120	60.45	Neal Howard <neal@metronet.com>
486DX4/120 AMD	60.45	Oscar Belmar Madrid <obelmar@anakena.usach.cl>
486DX4/120	60.45	Jason Buchanan <jsb@digistar.com>
486DX4/120	60.45	Foersterling <dirk@informatik.uni-frankfurt.de>
486DX4/120	60.45	Bernd Hentig <bernd@finow.snafu.de>
5x86/133 AMD	66.15	NN <root@tailor.aleim.net>
5x86/133 AMD	66.15	Brad Wilson <bwilson@deltanet.com>
5x86/133 AMD	66.15	Paul S. Doyle <paul.doyle@mci.com>
5x86/133 AMD	66.44	P Yli-Krekola <perttu@ntcmar01ba.ntc.nokia.com>
5x86/133 AMD	66.44	V. Tailor <vtailor@ibm.net>
5x86/133 AMD	66.55	Andrew B. Cramer <cramer@ripco.com>

TOC

5x86/133 AMD		66.56	Bob Nielsen <nielsen@primenet.com>
5x86/133 AMD		66.65	Geoff Raye <gtraye@igsrsparc2.er.usgs.gov>
5x86/133 AMD		66.7	Klaas Hemstra <hst@mh.nl>
5x86/133 AMD		66.80	N.N. <vp24njcb@ubvms.cc.buffalo.edu>
5x86/133 AMD	256c	67.10	Vasily Lewis
5x86/133 AMD		67.10	James Reith <reith@racores.com>
5x86/133 AMD	256c	67.10	Yves Rougy <yrougy%siam@cal.fr>
5x86/133 AMD	256c	67.10	Peter A. Koren <pkoren@lvdc20.dseg.ti.com>
5x86/133 AMD	256c	67.10	Wim Joppe <joppe@xs4all.nl>
5x86/133 AMD	256c	67.10	Gunnar Stefansson <gunnars@rhi.hi.is>
5x86/133 AMD	256c	67.10	Vernard Martin <vernard.martin@cc.gatech.edu>
5x86/150 AMD	overcl	74.75	Sergio Riveros <riveros@musca.unm.edu>
5x86/150 AMD	overcl	74.75	Arthur K. Chan <artchan@cs.ucr.edu>
5x86/160 AMD	overcl	79.87	M.Suencksen <msuencks@techfak.uni-bielefeld.de>
5x86/160 AMD	overcl	79.87	J. Chris Hammond <cosmo@pcisys.net>
5x86/160 AMD	overcl	79.87	Bird Chen <luca@linux.taiwan.hp.com>
5x86/160 AMD	overcl	79.89	Geir Skaugen <geir.skaugen@energy.sintef.no>
5x86/160 AMD	overcl	79.89	Martin Vernon <martin@gw6hva.demon.co.uk>
5x86/160 AMD	overcl	79.92	T. Zerucha <zerucha@shell.portal.com>
5x86/160 AMD	overcl	80.36	Paul Colucci <pcolucci@acsu.buffalo.edu>
5x86/160 AMD	overcl	80.36	Steinar Haug <sthaug@nethelp.no>
5x86/160 AMD	overcl	80.36	James Daniel <triadmin@bga.com>
5x86/160 AMD		80.36	David H.S. Oh <david@std.net>

9.5 Normal 486 variations: Cyrix/IBM, UMC

System	BogoMips	Reporter
486DLC/33	9.42	Dennis Robinson <djrobins@uxa.cso.uiuc.edu>
486DLC/33 387DX/40	9.47	Denis Solaro <drzob@vectrex.login.qc.ca>
486DLC/33 Cyrix wb	9.5	M. Asplund <matt@xenon.cchem.berkely.edu>
486DLC/33 Cyrix 386	11.2	Alex Freed <freed@europa.orion.adobe.com>
486DLC/40 256c	11.33	Schendel <sschend@magnus.acs.ohio-state.edu>
486Dx/40 Cyrix	11.73	Malcolm Bremer <malcolm@strw.LeidenUniv.nl>
486DRx2/40 Cyrix	13.10	Christopher Lau <clau@acs.ucalgary.ca>
486DX/33 Cyrix	13.21	M Haardt <u31b3hs@informatik.RWTH-Aachen.DE>
486DLC/40 bogoboost	13.21	Harry Pasanen <ps@tekla.fi>
486DLC/40 487 Cyrix	13.21	Ian A. Verschuren <iav@po.CWRU.Edu>
486DCL Cyrix	13.3	Tracer Bullet P.I. <ges@earth.baylor.edu>
486DLC/40	13.31	Adam Frampton <frampton@access2.digex.net>
486DLC/40	13.31	Rick Chow <crc@cacs.usl.edu>
486SLC-S/33	13.51	Brad Pepers <pepersb@cuug.ab.ca>
486DLC/40 no Cxpatch	15.47	Sergei O. Naoumov <serge@envy.astro.unc.edu>
486DLC/40 TI 128c	15.97	Philip K. Roban <phil@seal.micro.umn.edu>
486DLC/40 Cyrix	15.97	L.J. LaBash <labash@lcjones.aclib.siue.edu>
486DRx2/40	15.99	Christopher Lau <lauc@fusion.cuc.ab.ca>
486DX2/66 IBM no-FF	19	NN <coolefa@pmifeg.com>
486SLC2/66 IBM 64c	18.95	Sujat Jamil <sujat@shasta.ee.umn.edu>
486SLC2/66 IBM 128c	18.95	Sujat Jamil <sujat@shasta.ee.umn.edu>
486SLC2/66	19.02	Harry Mangalam <mangalam@uci.edu>
486SLC/50	19.28	Sion Arrowsmith <sion@bast.demon.co.uk>

TOC

486BL3/75 IBM 256c	21.40	Anders Stenback <stenback@kuai.se>
486BL3/75 IBM 256c	21.50	Ming S. Chan <ming.chan@canrem.com>
486DX2/66 Cyrix 128c	26.63	Derek Kwan <dkwan@zeus.UWaterloo.ca>
486DX2/66 Cyrix	26.63	Adrian Parker <adrian@willen.demon.co.uk>
486DX2-S/66 256c	26.63	Jean-Marc Wislez <JeanMarc.Wislez@rug.ac.be>
486DX2/66 Cyrix	26.63	Curran W. Fey <fey@biotech.washington.edu>
486BL3/100 IBM 256c	28.36	Anders Stenback <stenback@kuai.se>
486SX-S/33 UMC 0c	20.20	Hynek Med <xmedh02@manes.vse.cz>
486SX-S/40 UMC 0c	26.52	Hynek Med <xmedh02@manes.vse.cz>
486SX-U5/40 UMC 0c	26.63	Dusan Mihajlovic <zdule@herkules.co.yu>

9.6 Oddly or faultly configured Pentium systems, or variations

System	BogoMips	Reporter
Pentium/66	2.18	Bob Myers <root@shyguy.lonestar.org>
Pentium/90 notebook	9.5	Mark Maybee <markm@cs.colorado.edu>
6x86/120 Cyrix	52.32	Joel Boring <dwild@eskimo.com>
Pentium/83 Overdrive	82.85	Brian Smith <smithb@laraby.tiac.net>
Pentium/83 Overdrive	83.32	Scott Francis <mord@netcom.com>
Pentium/83 Overdrive	82.94	G. Spiegelberg <greg@owens.ridgecrest.ca.us>
Pentium/83 Overdrive	83.35	Jacek Polewczak <jacek.polewczak@csun.edu>
6x86/120 P120+overcl	104.86	Howard Poe <falcor@kingsnet.com>
Pentium MMX/263 overcl	392.40	John Appleby <jma24@cam.ac.uk>
AMD K6-2/300	399.95	Chris Ebenezer <chriseb@nortelworks.com>
Pentium MMX/231	419.43	Juan Domenech <domenech@mail.seric.es>
Pentium MMX/263 overcl	435.87	Juan Domenech <domenech@mail.seric.es>
SMP4 Pentium Pro/200	700.13	R. Carrico <robert_carrico@themoneystore.com>

9.7 Normal Pentium systems

System	BogoMips	Reporter
Pentium/60	23	Chien-An Chen <giant@nwu.edu>
Pentium/60	23.96	Joost Helberg <jhelberg@nlsun8.oracle.nl>
Pentium/60	23.96	Ulf Tietz <ulf@rio70.bln.sni.de>
Pentium/60 Gateway	23.96	Manoj Kasichainula <mvkasich@eos.ncsu.edu>
Pentium/60	23.96	Pierre Frenkiel <frenkiel@cdfap1.in2p3.fr>
Pentium/60	23.96	Tim Oosterbroek <tim@astro.uva.nl>
Pentium/60 NCR 3455	24	Mathias Koerber <mathias@solomon.technet.sg>
Pentium/60	24	Joe Sloan <jjs@engr.ucr.edu>
Pentium/60	24.0	Mark H. Wood <mwood@indyvax.iupui.edu>
Pentium/60	24.13	R.M. van Rijswijk <rijswijk@cs.utwente.nl>
Pentium/66	25	Chuck Munro <chuckm@canada.hp.com>
Pentium/66	26.63	Jason M. Naughton <jnaughto@ee.ryerson.ca>
Pentium/66	26.84	Kelly Carmichael <kcarmich@cln.etc.bc.ca>
Pentium/75 256c	29.5	Chris Dodd <cdodd@super.win.or.jp>
Pentium/75	29.79	Scott M. Grim <sgrim@netwalk.com>

TOC

Pentium/75	29.80	Torbjoern Kristoffersen <torbkris@online
Pentium/75	29.95	Steve Martin <smartin@mrg.uswest.com>
Pentium/75	30.22	Zoran Marjanski <bagzor@enterprise.ca>
Pentium/75	30.22	Andrew Buckby <c4ab1@dmu.ac.uk>
Pentium/75	30.22	Stuart Nuttall <u9230106@sys.uea.ac.uk>
Pentium/90 notebook	32.73	Rich Neves <neves@cs.colorado.edu>
Pentium/90	34.07	A. Galbraith <ajgalb@postman.essex.ac.uk>
Pentium/90 zappa 256c	35.6	Sebastien Dedieu <dedieu@emi.u-bordeaux.fr>
Pentium/90	35.84	Jason Nunn <jsno@stormfront.com.au>
Pentium/90	35.88	Joe Anderson <CPJJA@ttacs3.ttu.edu>
Pentium/90	35.88	Warwick Allison <warwick@cs.uq.oz.au>
Pentium/90	35.94	Erik Walthinsen <omega@teleport.com>
Pentium/90	36	Joe Sloan <jjs@engr.ucr.edu>
Pentium/90	36	Larry Auton <lda@nfa.research.att.com>
Pentium/90	36	Richard Knipe <knipe@lobby.ti.com>
Pentium/90	36.0	Werner Almesberger <almesber@lrc.epfl.ch>
Pentium/90	36.06	Rob J. Nauta <rob@iaehv.nl>
Pentium/90	36.08	Leung Hon Wa <cshwleun@cs.citu.edu.hk>
Pentium/90 Zeos	36.08	Chris Laurel <claurel@mr.net>
Pentium/90	36.08	Ronald Prague <ronp@fishnet.net>
Pentium/90	36.08	Adrian Blues <adrian@hypereality.co.uk>
Pentium/90	36.08	Pak Yin Tam "Fred" <ptam@eesun1.tamu.edu>
Pentium/90	36.08	Jason Heiss <jheiss@cco.caltech.edu>
Pentium/90	36.08	Tim Krantz <tek@dsinc.com>
Pentium/90 Gateway	36.08	Pete Stewart <stewart@bae.bellcore.com>
Pentium/90	36.08	Tom Manos <tmanos@infi.net>
Pentium/90	36.08	Richard Mundell <R.Mundell@uea.ac.uk>
Pentium/90	36.08	NN <usenet@uxmail.ust.hk>
Pentium/90	36.08	Yavuz Batmaz <yavuzb@knidos.cc.metu.edu.tr>
Pentium/90	36.08	Alan Skelley <askelley@gpu.srv.ualberta.ca>
Pentium/90	36.08	Ralph Sims <ralphs@locus.halcyon.com>
Pentium/90	36.08	Julian Bradbury <julian@xabcs.demon.co.uk>
Pentium/90	36.08	Dan Langrill <langrlld@mussel.cig.mot.com>
Pentium/90	36.08	Ravi Krishna Swamy <rkswamy@eos.ncsu.edu>
Pentium/90	36.08	J.L. Brothers <brothers@halcyon.com>
Pentium/90 Micronics	36.09	Andrew Brown <andrewbrown@acm.org>
Pentium/90	36.27	Giao H. Phan <giao@concrete.resnet.upenn.edu>
Pentium/90 Plato	36.4	Joe Valenzuela <smarm@ibm.net>
Pentium/90	36.5	Mike Kelleher <mikejk@umfacad.maine.edu>
Pentium/90	36.9	Ted Gaunt <tgaunt@pms701.pms.ford.com>
Pentium/100 overcl	38.39	Stefan Onderka <stefan.onderka@metronet.de>
Pentium/100	39.52	Adrian Blues <adrian@hypereality.co.uk>
Pentium/100 overcl	39.73	Phil Howard <phil@clr.com>
Pentium/100	39.73	<Peter.Weiss@informatik.uni-oldenburg.de>
Pentium/100 0c	39.73	Jason Crawford <jasonc1@gramercy.ios.com>
Pentium/100 overcl	39.83	Stefan Onderka <stefan.onderka@metronet.de>
Pentium/100	39.93	Tom Miller <tvtom@en.com>
Pentium/100 Cyrix	39.94	Mike Holland <myk@cygnus.uwa.edu.au>
Pentium/100	39.94	KAZ Vorpal <kaz@upx.net>
Pentium/100 overcl	39.94	Donar G.E. Alofs <donar@cs.vu.nl>
Pentium/100	39.94	Larry Snyder <larry@trauma.iag.net>
Pentium/100	39.94	Ian Hill <ian@hecate.phy.queensu.ca>

TOC

Pentium/100		39.94	John Crawford <link@spu.edu>
Pentium/100		39.94	Jered <jered@mit.edu>
Pentium/100	overcl	39.94	Ian <irs2@tweedledum.amp.york.ac.uk>
Pentium/100		39.94	Brian McGhee <brianm@iceonline.com>
Pentium/100		39.94	M Skjelland <morten.skjelland@pvv.unit.no>
Pentium/100		39.96	Dan Kha <dkha@yorku.ca>
Pentium/100		39.98	Phillipe Charon <charron@ecoledoc.ibp.fr>
Pentium/100		40.03	<bon@elektron.ikp.physik.th-darmstadt.de>
Pentium/100		40.08	Ronny Spiegel <rspiegel@htwm.de>
Pentium/100		40.18	David Baldwin <davidb@exis.net>
Pentium/100		40.18	<habibie@catevr.fiu.edu>
Pentium/120	Cyrix	47.8	Simon Ho <simon@epsilon.win-uk.net>
Pentium/120		47.82	Jorge Juan-Chico <jjchico@imse.cnm.es>
Pentium/120	Cyrix	47.92	Joel N. Squire <squire@colorado.edu>
Pentium/120		47.93	Umberto d'Ortona <umberto@grenet.fr>
Pentium/120	Cyrix	47.93	Jim T. Polk <jtpolk@cris.com>
Pentium/120		47.93	Jon Trowbridge <trow@mcs.com>
Pentium/120		47.98	Craig Bates <cbates@psu.edu>
Pentium/120		47.92	Aaron Brick <abrick@dcc.uchile.cl>
Pentium/120	Cyrix	48	Steve <horne@mhd2.pfc.mit.edu>
Pentium/120		48.00	Michael Wazenski <mwazenski@dsrnet.com>
Pentium/120	Intel	48.02	Scott M. Grim <sgrim@netwalk.com>
Pentium/120	Cyrix	48.27	Glenn T. Jayaputera <gjt@budgie.apana.org.au>
Pentium/120		48.27	Roman Mitnitski <mitnits@shany.net>
Pentium/120		48.27	Peter Walsh <pwalsh@rain.org>
Pentium/120	Cyrix	48.2	Viznyuk <sviznyuk@magnus.acs.ohio-state.edu>
Pentium/120		49.27	Simon Hargrave <simon@revell.demon.co.uk>
Pentium/133		53.04	Wayne Roberts <wroberts@aug.com>
Pentium/133		53.04	Gregory Travis <greg@indiana.edu>
Pentium/133	overcl	53.04	A. Kunigelis <algikun@santaka.sc-uni.ktu.lt>
Pentium/133	Intel	53.04	Jimmie Farmer <calvin@malchick.com>
Pentium/133		53.25	Chuck Mattern <cmattern@mindspring.com>
Pentium/133		53.26	Glenn Holt <gholt@lsil.com>
Pentium/133		53.26	Heikki Levanto <heikki@lsd.ping.dk>
Pentium/133		53.26	Chaim Tarshish <chaim@ipl.med.nyu.edu>
Pentium/133		53.26	Mitchell B. Hamm <hamm@one.net>
Pentium/133		53.26	Donald Lewis <dlewis@jackson.freenet.org>
Pentium/133		53.26	Jon Trowbridge <trow@kremlin.emccta.com>
Pentium/133		53.26	Charny Peete Mitchell <cpmiche@eos.ncsu.edu>
Pentium/133	256c	53.26	David Wuertele <dave@gctech.com>
Pentium/133	256c	53.68	Guiseppe Miceli <ferdy@ccii.unipi.it>
Pentium/133		53.68	Michael Kress <kress@hal.saar.de>
Pentium/150	Intel	59.80	Joel D. Young <jdyoung@afit.af.mil>
Pentium/150		60.01	Joost de Greef <joost@stack.nl>
Pentium/150	overcl	60.21	Duarte Cordeiro <l38404@alfa.ist.utl.pt>
Pentium/166		66.16	Pedro Soria-Rodriguez <sorrodp@wpi.edu>
Pentium/166		66.35	K. Visweswaran <kviswesw@lehman.com>
Pentium/166		66.36	T. Endo <enchan@trc.rwcp.or.jp>
Pentium/166		66.44	Donald Lewis <dlewis@jackson.freenet.org>
Pentium/166		66.76	F. Baitinger <baiti@herrenberg.netsurf.de>
Pentium/166		67.10	Jon Trowbridge <trow@mcs.com>
Pentium/166		67.10	Dylan <dylan@ert.com>

TOC

Pentium/166 512c	67.10	Dirk Freese <freese@infra.de>
C6/200 CenterHauls	79.46	James Rotenberry <rotenber@jmr.lasx.com>
Pentium/200	79.66	Piete Brooks <Piete.Brooks@cl.cam.ac.uk>
Pentium/200	79.69	Timm Gleason <timm@bess.net>
Pentium/200	79.89	Dave S. Baker <dave@acedia.demon.co.uk>
Pentium/200	78.87	Nick D'Apice <ndapice@erols.com>
Pentium/200	80.08	Anders Stenback <stenback@hehe.com>
Pentium/200	80.08	Steve Freeland <sfreel@cs.mcgill.ca>
Pentium/200	81.92	Steve Baur <steve@xemacs.org>

9.8 Normal Pentium variations: MMX, Pro, II, Cyrix, AMD

System	BogoMips	Reporter
Pentium MMX/133	265.77	Ron Peters <rpeters@f15fast.al.intel.com>
Pentium MMX/150	307.53	Sami Sihvonen <buggy@fix.no>
Pentium MMX/166 notebk	331.75	n.n. <visionary@aura.title14.com>
Pentium MMX/166	331.78	Rob Janssen <pe1chl@amsat.org>
Pentium MMX/166	331.78	Dave Page <dave@vale-housing.co.uk>
Pentium MMX/166	331.78	Matthew C. Sell <amtmcs@amsta.leeds.ac.uk>
Pentium MMX/166	333.41	Sjoelie <patrick@sjoel.xs4all.nl>
Pentium MMX/166	333.41	Viet Yen Nguyen <vt@multiweb.nl>
Pentium MMX/166	333.41	Marcin Owsiany <porridge@lo4.ids.bielsko.pl>
Pentium MMX/180	358.81	David Efflandt <efflandt@xnet.com>
Pentium MMX/200	398.13	Andy Saunders <andi@numenor.oucs.ox.ac.uk>
Pentium MMX/200	398.13	A. James Lewis <james@vrtx.net>
Pentium MMX/200	398.13	Sean Horan <sean@psy.ed.asu.edu>
Pentium MMX/200	398.19	Fabio Comolli <comolli@tin.it>
Pentium MMX/200	398.95	Reinhold J. Gerharz <rgerharx@erols.com>
Pentium MMX/200	398.95	Eric Beymer <beymer@soundex.com>
Pentium MMX/200	398.95	Duane Steel <dsteele@direct.ca>
Pentium MMX/200	398.95	<ndpilatz@undergrad.math.uwaterloo.ca>
Pentium MMX/200	400	Rob Jokinen <rjokinen@rt66.com>
Pentium MMX/200	400.59	Paul Black <paul@darwin.demon.co.uk>
Pentium MMX/200	400.59	Juho Cederstrom <cederstrom@kolumbus.fi>
Pentium MMX/200	400.59	Bart <bart@aceonline.com.au>
Pentium MMX/210 overcl	416.97	John Saunders <john@nlc.net.au>
Pentium MMX/225 overcl	448.92	Ingo Reimann <reimann@uni-muenster.de>
Pentium MMX/225 overcl	448.92	Mattias Hembruch <mghembru@ece.uwaterloo.ca>
Pentium MMX/225 overcl	448.92	Larry Lade <lade@btigate.com>
Pentium MMX/230 overcl	461.85	Rene Stanneveld <edge@huizen.nhkanaal.nl>
Pentium MMX/233	463.67	Rui M.B. Machado <rui@nifdl7.fd.ul.pt>
Pentium MMX/233	463.67	Rob Janssen <pe1chl@amsat.org>
Pentium MMX/233	465.31	Per Andersson <ppan@cntw.com>
Pentium MMX/233	465.31	Jose Maria Perez Box <jmpbox@etsin.upm.es>
Pentium MMX/233	466.94	Van Overbruggen <marcel@johannes.iconnect.nl>
Pentium MMX/233	466.94	Erwin Lubbers <erwin@box.nl>
Pentium MMX/250 overcl	498.07	Maarten van Rossum <m@vr.xs4all.nl>
Pentium MMX/266 overcl	525.93	Frank <r2@xs4all.nl>

TOC

Pentium Pro/133	132.88	John D. Sundberg <jdsundberg@mmm.com>
Pentium Pro/150	149.50	Rogier Wolff <wolff@bitwizard.nl>
Pentium Pro/180	179.61	Chuck Fee <fee@ch4549.org>
Pentium Pro/200	197.42	Michael Griffith <grif@cs.ucr.edu>
Pentium Pro/200	197.42	Curtis Varner <carner@cs.ucr.edu>
Pentium Pro/200	198.84	Erik Max Francis <max@alcyone.com>
Pentium Pro/200	198.84	Marc Winkler <marcus@healthchex.com>
Pentium Pro/200	199.04	V. Bostrom <Vareck_Bostrom@ccm.jf.intel.com>
Pentium Pro/200	199.06	Justin Clancy <justin@hippos.demon.co.uk>
Pentium Pro/200	199.06	Glenn Lamb <mumford@netcom15.netcom.com>
Pentium Pro/200	199.06	Laszlo Herczeg <las@light-house.com>
Pentium Pro/200	199.07	Stefan <boresch@schuber.u-strasbg.fr>
Pentium Pro/200	199.07	Greg Fausak <lgfausak@august.com>
Pentium Pro/200	199.07	Chris Jones <chris@planetsymphone.com>
Pentium Pro/200	199.07	Matthew S. Crocker <matthew@crocker.com>
Pentium Pro/200	199.95	Reinhold J. Gerharz <rgerharx@erols.com>
Pentium Pro/200	200.32	Gil Megidish <gmegidis@ort.org.il>
Pentium Pro/200	200.32	Jose Navarro <jnavarro@aoc.nrao.edu>
Pentium Pro/200	200.32	<Eric_Zucker@om.hp.com>
Pentium Pro/200	200.32	Wayne Scott <wscott@ichips.intel.com>
Pentium Pro/200	200.32	Adrian L. Hosey <ahosey@cs.indiana.edu>
Pentium Pro/233 overcl	234.43	S. Curtarolo <auro@spiro.fisica.unipd.it>
Pentium II/233	233.47	Torbjoern Kristoffersen <torbkris@online.no>
Pentium II/233	233.47	Sorin Mitran <mitran@irs.fzk.de>
Pentium II/266	265.42	Jon Trowbridge <trow@kremlin.emccta.com>
Pentium II/266	265.42	A. Hochwimmer <a.hochwimmer@auckland.ac.nz>
Pentium II/266	265.44	Nick Ullman <nick@avenza.com>
Pentium II/266	266.24	Pieter Eendebak <p.t.eendebak@phys.uu.nl>
Pentium II/266	267.06	James McKinnon <jmack@phys.ualberta.ca>
Pentium II/266	267.06	Greg <gl2hughes@undergrad.math.uwaterloo.ca>
Pentium II/266	267.06	Pros Robaer <pros@innet.be>
Pentium II/266	272.79	Frank Hale <frankhale@worldnet.att.net>
Pentium II/300 overcl	299.01	Martin Lathoud <nytral@endirect.qc.ca>
Pentium II/448 overcl	447.28	Mathieu Bois <mathieu.bois@nomura.co.uk>
Pentium II/450	447.28	Mathieu Bois <mathieu.bois@nomura.co.uk>
Pentium II/466	462.03	Bjorn Lindgren <bjorn@chiba.cx>
Intel Celeron/333	332.6	Florian <florian@egon-bbs.n-a-m.de>
Cyrix 5x86/100	100.16	NN <root@anxa04.cc.ic.ac.uk>
Cyrix 5x86/100	100.19	Valient Gough <vgough@teton.mines.edu>
Cyrix 5x86/100	100.47	C.Chan <chan@alfrothul.uchicago.edu>
Cyrix 5x86/120 P150+	119.60	Wynstan Tong <wynstan@eecg.toronto.ca>
Cyrix 5x86/120 P150+	119.60	Joel N. Squire <squire@colorado.edu>
Cyrix 5x86/120 P150+	119.83	Leland Olds <olds@eskimo.com>
Cyrix 5x86/120 P150+	119.83	NN <root@anxa04.cc.ic.ac.uk>
Cyrix 5x86/120 P150+	120.68	C.Chan <chan@alfrothul.uchicago.edu>
Cyrix 5x86/120 P150+	122.01	Andre Coetzee <acoetzee@ctcc.gov.za>
Cyrix 6x86/100	99.42	Stig M. Valstad <svalstad@sn.no>
Cyrix 6x86/110 P133+	109.77	Matthew Flint <matthew@philtrum.demon.co.uk>

TOC

Cyrix 6x86/110 P133+	109.77	John Merriam <suprnaut@esslink.com>
Cyrix 6x86/110 P133+	109.77	Keith Smith <keith@ksmith.com>
Cyrix 6x86/120 P150+	119	Jean-Claude Gouiran <jcg13@ibm.net>
Cyrix 6x86/120 P150+	119.60	Taso Lyristis <taso@remus.rutgers.edu>
Cyrix 6x86/120 P150+	119.60	Hrvoje Stipetic <stipe@zemris.fer.hr>
Cyrix 6x86/120	119.60	Yakko J. Warner <yakko@wtower.com>
Cyrix 6x86/120 P150+	119.60	B. James Philippe <bryan@terran.org>
Cyrix 6x86/120 P150+	119.83	Roger Merchberger <zmerch@northernway.net>
Cyrix 6x86/120	119.83	Daniel Wold <danw@panix.com>
Cyrix 6x86/120	120	John C. Beasley <beaslej1@nevada.edu>
Cyrix 6x86/120 P150+	120.01	Jay Thorne <jay@result.com>
Cyrix 6x86/120 P150+	120.01	Jeawan Kim <jaewan@harc.edu>
Cyrix 6x86/120 P150+	120.91	Cymen <cymen@ziplink.net>
Cyrix 6x86/133 P166+	132.71	Holger Kemper <hok@balu.ping.de>
Cyrix 6x86/133 P166+	132.71	Hrvoje Stipetic <stipe@zemris.fer.hr>
Cyrix 6x86/133 P166+	132.82	Alex Liffers <aliffers@tartarus.uwa.edu.au>
Cyrix 6x86/133 P166+	132.82	Brian C. Theobald <theobald@nortel.ca>
Cyrix 6x86/133 P166+	132.88	Alvaro Lopes <alvieboy@utad.pt>
Cyrix 6x86/133 P166+	132.88	Craig Andersen <andersen@fastlane.net>
Cyrix 6x86/133 P166+	133.73	C. Drews <drews_c@informatik.fh-hamburg.de>
Cyrix 6x86/133 P166+	133.12	Daniel Gritter <dgritt47@calvin.edu>
Cyrix 6x86/150 P200+	149.50	Evan L. Schemm <elschemm@mtu.edu>
Cyrix 6x86/150 P200+	149.50	Steven Rainwater <srainwater@ncc.com>
Cyrix 6x86/150 P200+	149.91	Sid Boyce <szb50@amdahl.com>
Cyrix 6x86/150 P200+	149.91	Eric Haas <haas@andrew.cmu.edu>
IBM 6x86/150 P200	149.96	M.D. Guardia <mikedg@ghostbbs.pandora.it>
Cyrix 6x86MX/166 A-Step	166.71	David Anderson <rovaughn@infoave.net>
Cyrix 6x86MX/188 P233	186.78	Dominic Baines <rdab100@hermes.cam.ac.uk>
Cyrix 6x86MX/188 P233	187.19	Dominic Baines <rdab100@hermes.cam.ac.uk>
K5/75 AMD	149.91	Simon Karpen <slk@linux-shell.net>
K5/90 AMD	179.40	<root@krabi.mbp.ee>
K5/90 AMD	179.40	Ken Edwards <edwards@thor.xon.cuug.ab.ca>
K5/90 AMD	179.40	Marcin Owsiany <porridge@lo4.ids.bielsko.pl>
K5/90 AMD	179.81	Marcin Owsiany <porridge@lo4.ids.bielsko.pl>
K5/90 AMD	180.22	Hector DC Gonzalez <turbo@linux.lsl.com.mx>
K5/90 AMD	181.00	Drew Golden <golden@platinum.nb.net>
K5/100 AMD PR133	198.66	Trond Solem <trond.solem@homemail.com>
K5/100 AMD PR133	199.07	Henri Jamgotchian <hjamgot@planete.net>
K5/100 AMD PR133	199.48	Dark Mind <root@dmh.ml.org>
K5/100 AMD PR133	199.88	J. Grassel <grassel@heart.cas.und.nodak.edu>
K5/100 AMD PR133	199.88	Berend Reitsma <berend@united-info.com>
K5/100 AMD PR133	200.29	Tilman Sommer <sommer@vsun02.ag01.kodak.com>
K5/100 AMD PR133	200.32	Carlo Politi <cpoliti@mare.gol.grosseto.it>
K5/100 AMD PR133	199.07	Franco De Angelis <fda@ied.unipr.it>
K5/100 AMD PR133	199.07	HaJo Simons <hajo@frodo.com>
K5/116 AMD PR166	233.47	Hans-Joachim Baader <hans@grumber.ika.de>
K5/116 AMD PR166	233.47	Sean Kelly <s.kelly@newcastle.ac.uk>
K5/120 AMD PR133 overcl	239.21	Chris Harshman <harshman@paradigm.uor.edu>
AMD K6/166	332.60	David Parsons <orc@pell.chi.il.us>
AMD K6/166	331	Bill Petersen <brp@cuberramp.net>

AMD K6/200	398.85	n.n. <uh886@freenet.victoria.bc.ca>
AMD K6/200	398.89	Ian Hanschen <hanschen@uwyo.edu>
AMD K6/200	398.98	Philip Lijnzaad <lijnzaad@columba.ebi.ac.uk>
AMD K6/200	399.77	Murtaza Amiji <murti@wpi.edu>
AMD K6/200	399.78	Dan Hetzel <dan@icor.fr>
AMD K6/200	400.58	Paco Culebras Amigo <paco@hades.udg.es>
AMD K6/200	400.59	Sverre H. Huseby <s.h.huseby@usit.uio.no>
AMD K6/200	400.59	Steve Conley <sconley@muck.leonine.com>
AMD K6/200	400.59	Mark Lehrer <edge@dux.raex.com>
AMD K6/200	400.59	Chris Esser <ksqueak@erols.com>
AMD K6/200	400.59	Federico Pellegrin <fede@triangolo.it>
AMD K6/200	400.59	Howard Poe <falcor@kingsnet.com>
AMD K6/200	400.64	Fabrizio Santini <sanfff@tin.it>
AMD K6/208 overcl	416.15	Jani Halme <jaadha@utu.fi>
AMD K6/208 overcl	416.15	Donnie Savage <dsavage@cris.com>
AMD K6/208	417.97	J.F. Ursetto <ursetto@uiuc.edu>
AMD K6/225	440	n.n. <uh886@freenet.victoria.bc.ca>
AMD K6/225 overcl	448.92	Paco Culebras Amigo <paco@hades.udg.es>
AMD K6/225 undercl	448.92	Henrik Storner <storner@image.dk>
AMD K6/225 overcl	450.56	M. Cramer <mccramer@stuttgart.netscape.de>
AMD K6/233	463.67	Aduanne Carter <drow@blazenet.net>
AMD K6/233	465.31	Michael Haardt <haardt@gmd.de>
AMD K6/233	466.5	R. Garcia <rgarciaitt@aol.com>
AMD K6/233 overcl	466.84	Francesco <root@fly.cnuce.cnr.it>
AMD K6/233 overcl	466.94	Paco Culebras Amigo <paco@hades.udg.es>
AMD K6/233	466.94	Howard Poe <hpoe@nyx.net>
AMD K6/233	466.94	Andreas Haumer <andreas@xss.co.at>
AMD K6/233	466.94	Damien Castelltort <eznerald@mail.mnet.fr>
AMD K6/233	466.94	G. Cantallops Ramis <gcantallopsr@jet.se>
AMD K6/233	466.94	Juergen Hammelmann <juergen@neptun.stgt.de>
AMD K6/233	466.94	Chris Gushue <seymour@iname.com>
AMD K6/239	478.41	Torbjoern Kristoffersen <torbkris@online.no>
AMD K6/239	478.41	Mark Neill <btech@styx.phy.vanderbilt.edu>
AMD K6/239	478.41	Louis van Dompselaar <etaoin@xs4all.nl>
AMD K6/266	534.12	Andreas Czerniak <cognac@toppoint.de>
AMD K6-2/300	598.02	Pasa Guglielmo <gpasa@omedia.ch>
AMD K6-2/300	599.65	Ivo Plana <ipv@tinet.fut.es>
AMD K6-2/300	599.65	Juan Domenech <domenech@mail.seric.es>
AMD K6-2/300	599.65	Ashley Penney <ashp@bastard.co.uk>
AMD K6-2/300	599.65	Rob <im@home.com>
AMD K6-2/300	599.65	Stephen Jenuth <jenuths@homacjen.ab.ca>
AMD K6-2/308	614.40	Erv Walter <walter@chem.wisc.edu>
AMD K6-2/333	663.55	<kuhn@ernie.lpr.e-technik.tu-muenchen.de>
AMD K6-2/333	665.19	A. Bihlmaier <stormtank@gmx.net>
AMD K6-2/338 overcl	675.02	Nicola Fabiano <mx8644@mclink.it>
AMD K6-2/350	699.60	Juan Domenech <domenech@mail.seric.es>
AMD K6-2/400 overcl	799.54	Ivo Plana <ipv@tinet.fut.es>
AMD K6-2/400 overcl	801.18	Phillip Deackes <gsmh@gmx.net>
AMD K6-2/400 overcl	801.18	Jon Riekenberg <jrr8208@ksu.edu>
AMD K6-2/400 overcl	801.18	Ryan Bethke <immortal26@prodigy.net>
AMD K6-2/400	801.18	Wolfgang Zekoll <wzk@um1.pce.de>
AMD K6-2/400	801.18	Ted E. Suter <nexion@home.com>

```
AMD K6-2/400            801.29  Birko Bergt <birko.bergt@cdc.com>
AMD K6-2/440            878.18  Steve Bourland <sbourlan@ecn.purdue.edu>
AMD K6-2/450 overcl     901.12  Ted E. Suter <nexion@home.com>
```

TOC

9.9 Normal Alpha systems

```
System              BogoMips  Reporter
21064/150 Jensen        148.37  Linus Torvalds <torvalds@cc.helsinki.fi>
21064/150 Jensen        149.49  J.L. Brothers <brothers@halcyon.com>
21064/150 Jensen        148.89  Martin Osterman <ost@comnets.rwth-aachen.de>

21064A/275 Cabriolet    271.58  <michal@ellpspace.math.ualberta.ca>
21064A/275 Cabriolet    272     Linus Torvalds <torvalds@cc.helsinki.fi>
21064A/275              272     Stephen Gaudet <sjg@tiac.net>
21064A/275 Cabriolet    272.63  Jay Estabrook <jestabro@amt.tay1.dec.com>
21064A/275 Cabriolet    273.37  David Mosberger-Tang <davidm@cs.arizona.edu>
21064A/275              274.11  Kevin Jacobs <jacobs@eek.cwru.edu>
21064A/300 Cabriolet    298     Jay Estabrook <jestabro@amt.tay1.dec.com>
21064A/300              298     Stephen Gaudet <sjg@tiac.net>

21066/166               162.53  Phil Bostley <pbostley@qualcomm.com>
21066/166               163.05  Matthew Jacob <mjacob@feral.com>
21066/166               163.05  Jon Spreha <jxs557@psu.edu>
21066/166               164.59  David Mosberger-Tang <davidm@cs.arizona.edu>
21066/166 Multia        164.63  Rudolf Gabler <rug@usm.uni-muenchen.de>
21066/166               165     Gareth Bult <gareth@ftech.net>
21066/166               165.04  Craig Ruff <cruff@ncar.ucar.edu>
21066/200               196.9   Danny ter Haar <danny@cistron.nl>
21066/200 UDB overcl    198     Kari Davidsson <d154402@cs.tut.fi>

21066A/233 UDB          229.63  Toon van der Pas <toon@vdpas.hobby.nl>
21066A/233 AS400        230.16  Ophir Ronen <ophir@connectsoft.com>
21066A/233 AS200 Avanti 230.16  B. James Phillippe <bryan@terran.org>
21066A/233 NoName       230.67  T. Bogendoerfer <tsbogend@bigbug.franken.de>
21066A/233 UDB          230.68  Ted Schipper <ted@tedux.hobby.nl>
21066A/233 NoName       230.76  Mikael Nykvist <viper@ludd.luth.se>
21066A/233 UDB          231.21  Eric Smith <eric@goonsquad.spies.com>
21066A/233 NoName       231.21  Jay Estabrook <jestabro@amt.tay1.dec.com>
21066A/266 NoName ov.cl 261.62  Andreas Johansson <ajo@ludd.luth.se>
21066A/266 UDB overcl   261.62  Michael Brennen <mbrennen@fni.net>
21066A/266 NoName ov.cl 262.14  Wim van Dorst <baron@clifton.hobby.nl>
21066A/266 Multia       264     Joshua Grubman <joshg@dn.net>
21066A/284 NoName ov.cl 281.0   <imakino@gloria.cord.edu>
21066A/297 NoName ov.cl 293.6   <imakino@gloria.cord.edu>
21066A/300 UDB ov.cl    294.65  Topi Kanerva <tkanerva@nks.oulu.fi>

21164/266 EB164         265.29  Jay Estabrook <jestabro@amt.tay1.dec.com>
21164/300 EB164         297.79  Hilarius <maurice@ellpspace.math.ualberta.ca>
21164/300 XLT Alcor     297.79  Dave Wreski <dave@nic.com>
21164/300 AS1000        297.79  Salvador Pinto Abreu <spa@sc.uevora.pt>
```

TOC

21164/333	Alcor	331.35	Linus Torvalds <torvalds@cs.helsinki.fi>
21164/333	Alcor	331.35	David Mosberger-Tang <davidm@azstarnet.co
21164/366	PC164 Durango	363.85	Geerten Kuiper <geerten@bart.nl>
21164/433	PC164	429.89	Paul D. Robertson <proberts@clark.net>
21164/433	PC164	429.91	Bernd Meyer <bmeyer@cs.monash.edu.au>
21164/433	Maita	429.91	Bill Broadley <bill@math.ucdavis.edu>
21164/433	PC164	430.96	<rainer.landes@physik.uni-karlsruhe.de>
21164/433	EB164	430.96	<Metod.Kozelj@rzs-hm.si>
21164/433	PC164	430.96	H. Sumargo <habibie@robotic.eng.fiu.edu>
21164/433	PC164	431.94	Timm Gleason <timm@bess.com>
21164/466	PC164	464.51	Daryll Strauss <daryll@d2.com>
21164/500		497	Alex Butcher <alex@asimov.annex.co.uk>
21164/500	PC164	497.02	C.J. Grayce <cgrayce@wasatch.ps.uci.edu>
21164/500		497.05	Heiner Kruener <hk@martian.ping.de>
21164/500	AS500	497.03	Jim Nance <jlnance@avanticorp.com>
21164/500	P7	497.03	Jan guldentops <jacko@ba.be>
21164/500	EB164 Durango	497.43	Kevin Jacobs <jacobs@eek.cwru.edu>
21164A/500	PWS	497.02	Robert Harley <robert.harley@inria.fr>
21164A/500	PC164	497.02	Stephen Oberski <sfo@deterministic.com>
21164A/533	PC164LX	529.53	Harvey J. Stein <hjstein@bfr.co.il>
21164PC/533	PC164SX	529.53	Danny ter Haar <dth@cistron.nl>
21164PC/533	PC164SX	529.53	Shane Sturrock <sss@holyrood.ed.ac.uk>
21164PC/533	PC164SX	529.53	Wim van Dorst <baron@clifton.hobby.nl>
21164A/533	PC164LX	530.57	Ronny Ranerup <ronny@axis.com>
21164A/533	PC164LX	530.57	Bill Broadley <bill@math.ucdavis.edu>
21164A/600	PC164LX	595.59	L.F. Donaldson <donaldlf@cs.rose-hulman.edu>
21164A/600	PC164LX	595.59	<stepan@wesley.informatik.uni-freiburg.de>
21164A/600	PC164	597.60	Fabrizio Santini <sanfff@tin.it>
21164A/666	PC164LX	662.70	<stepan@wesley.informatik.uni-freiburg.de>
21264/400	Pass-1	794.82	Jay Estabrook <jay.estabrook@digital.com>

9.10 Normal Motorola systems

System		BogoMips	Reporter
68030/16	Atari Falcon	3.90	Jay T. Millar <jmillar@eaglequest.com>
68030/16	Atari Falcon	3.95	J.L. Brothers <brothers@halcyon.com>
68030/16	Atari Falcon	3.98	<Roman.Hodek@informatik.uni-erlangen.de>
68030/20	MacII Si	4.88	Bill Maloy <maloy@goldinc.com>
68030/20	0c	4.92	Chris Nadigh <chrnadig@iiic.ethz.ch>
68030	Amiga 3000	6.08	Andy Wick <awick@vt.edu>
68030/30	Amiga 4000	6.09	Karsten Merker <km@golf.dinet.com>
68030/25	Amiga 3000	6.12	Glen Hewlett <hewlett@planeteer.com>
68030/25	Amiga 3000	6.21	Hamish Macdonald <hamish@bnr.ca>
68030/25	Amiga 3000	6.21	J.L. Brothers <brothers@halcyon.com>
68030/32	Atari Falcon	7.91	Franz Korntner <fkorntne@bazis.nl>
68030	Atari TT	7.96	<schwab@issan.informatik.uni-dortmund.de>
68030/32	Atari MegaST	7.98	E.J. van den Bussche <busscheh@ksepl.nl>
68030/33	Atari TT	7.98	<Roman.Hodek@informatik.uni-erlangen.de>

TOC

68030	Atari TT	7.98	Wayne Booth <trek@ihgp114r.ih.att.com>
68030/60		8.06	Korey Budgen <kbudgen@st.nepean.uws.edu.a
68030/33	Amiga 2000	8.14	W. Haidinger <e9225662@stud1.tuwien.ac.a>
68030/48	32c	11.89	Martin Rogge <Martin_Rogge@ki.maus.de>
68030/50	Atari	12	<Roman.Hodek@informatik.uni-erlangen.de>
68030/50	Amiga 1200	12.36	Chris Sumner <chris@ganymede.sonnet.co.uk>
68030/50	Amiga 1200	12.4	Richard Jerome <etlrdje@tigger.ericsson.se>
68030/50	32c	12.42	Michael Plonus <michi@pluto.ping.de>
68030/50	Amiga 1200	12.33	Detrix <detrix@popd.ix.netcom.com>
68030/50	Amiga 1200	12.36	E.J.M. Brocklesby <ejb@klamath.demon.co.uk>
68040/24	Amiga 4000-40	16.6	Hamish Macdonald <hamish@bnr.ca>
68040/24	Amiga 4000-20	16.60	J.L. Brothers <brothers@halcyon.com>
68040/25	Amiga 4000-040	16.61	<Geert.Uytterhoeven@cs.ku-leuven.ac.be>
68040/25	Amiga 4000	16.61	Lawrence <lawrenc@nextwork.rose-hulmand.edu>
68040/60	Amiga 4000-40	18.99	Darren Enns <dmenns@surf.pangea.ca>
68040/32	Medusa T40	21.25	<Hartmut.Koptein@et-inf.fho-emden.de>
68040	Amiga 2000	21.86	Ron Flory <rjflory@feist.com>
68060/50	Amiga 1200	99.53	baba <baba@pa.yokogawa.jp>
68060/50	Amiga 4000	99.74	Stefan Tauche <otauche@uni-paderborn.de>
68060/50	Amiga 4000	100.16	Jan Johansson <jj@mordor.it.kth.se>
68060/66	Amiga CS MkII	132.71	Paul Hill <paul@lagernet.clara.co.uk>

9.11 Other Systems: Sparc, PowerPC, Mips, Intel 8088/286 ELKS

System	BogoMips	Reporter
Sparc sun4c	17.94	J.L. Brothers <brothers@halcyon.com>
Sparc SLC/20 S1	19.86	Simon Karpen <slk@linux-shell.net>
Sparc sun4c/25	24.88	Paolo <paolo0@yahoo.com>
Sparc IPX 4c	39.83	Paul D. Robertson <proberts@clark.net>
Sparc SS10 super50 4m	39.93	Juan Cespedes <cespedes@etsit.upm.es>
Sparc microS/50 SS-LX	49.76	Will Shaw <romulan@netwatch.clemson.edu>
Sparc SS20	49.86	Gary A. Donahue <lordgad@webspan.net>
Sparc hyperS Classic 4m	49.86	Juan Cespedes <cespedes@etsit.upm.es>
Sparc Voyager portable	59.80	Edward Austin <eastin@shl.com>
Sparc SS5 Netra	68	Craig Falconer <cf@papanui.school.nz>
Sparc SS50	74.95	Kaz <kaz@latte.cafe.net>
Sparc SS10 hyperS	99.73	Thomas B. Fox <tfox@oliverdesign.com>
Sparc SS5/66	109.77	Lance S. Nehring <lnehrin@uswest.com>
Sparc SS4/110	109.77	Chris Sylvain <csylvain@umms-itg.ab.umd.edu>
Sparc hSparc/150 Sun4m	150.32	Tethys <tethys@ml.com>
Sparc Ultra1 sun4u	284.05	Iban Cardona <icc@seric.es>
Sparc UltraII/270	539.03	Pieter Krul <pieter@dare.demon.nl>
Sparc UltraII/360	719.26	Bjoern Augustsson <d3august@dtek.chalmers.se>
PowerPC 601/60 Mac6100	45.24	J.L. Brothers <brothers@halcyon.com>

PowerPC 601/66 Mac	51.62	Fred Klein <klein@des3.u-strasbg.fr>
PowerPC 601/60 Mac6100	59.38	Kent Radek <goo@itd.sterling.com>
PowerPC 601/75 Mac7200	73.93	Dimitris Tsifakis <jimmy@typhoeus.dg.uoa.gr>
PowerPC 601/100 Mac7500	98.91	Russ Hoffman <reh@fore.com>
PowerPC 601/110 Mac8100	108	Charles Eicher <ceicher@inav.net>
PowerPC 603/100	66.56	A. Costa <c_chaos@chaosnet.wahnapitae.on.ca>
PowerPC 604/100 PPS6050	99.74	Evaldas Darcianovas <evaldas@isi.kvm.lt>
PowerPC 604/100	199	Hamish Marson <hamish@aixrules.nz.ibm.com>
PowerPC 604/100 PPS7248	199.48	Evaldas Darcianovas <evaldas@isi.kvm.lt>
PowerPC 604/132 Mac7500	263.78	Patrick Murmann <murmann@mi.uni-erlangen.de>
PowerPC 604/133 MOT PS	266.24	Christoper Harrel <cnh@eng.mindspring.net>
PowerPC 604/150 Mac9500	297.73	Jean-Philippe Lord <jpl@binex.com>
PowerPC 604e/195	359.63	Michael Potts <pottsjam@flyernet.udayton.edu>
PowerPC 604/200 Mac8600	320	Julien Sebot <sebot@lri.fr>
PowerPC 604e/200 SM4000	399.12	Roland Kuhn <kuhn@physik.uni-kassel.de>
PowerPC 750/266 Mac	532.48	Fritz Anderson <fritza@tezcat.com>
Mips R4000/100	48.30	J.L. Brothers <brothers@halcyon.com>
Mips R4000/100 Magnum	50.03	Andreas Busse <andy@soft-n-hard.de>
Mips R4400/134 Acer Pica	67.10	Andreas Busse <andy@soft-n-hard.de>
Mips R4400/134 Acer Pica	67.10	Ralf Baechle <ralf@waldorf-gmbh.de>
Mips R4600/133 Tyne	133.12	Ralf Baechle <ralf@waldorf-gmbh.de>
Mips R4600/133 RM200	133.08	Ralf Baechle <ralf@julia.de>
Mips R5000/150 Indy	154.83	Ralf Baechle <ralf@julie.de>
Intel 8088/4.77	0.02	Tim Van der Linden <timvdl@innet.be>
Intel 8088/10	0.05	Tim Van der Linden <timvdl@innet.be>
Intel 8086	0.5	Kin Lau <gabe@zot.io.org>
Intel 286 Tandy	0.75	Joey Hess <joey@kite.ml.org>
Intel 286	0.99	Anders Stenback <stenback@hehe.com>
Intel 286/8 VAXMate	1.03	Andrew Costa <c_chaos@wahnapitae.on.ca>
Intel 286/10 Commodore	1.30	Hans-Joachim Baader <hans@grumber.inka.de>
Intel 286 PS2	2.32	Morillas C.H. Antonio <morilla@fie.us.es>
Intel 286 PS2	2.34	Joey Hess <joey@kite.ml.org>

9.12 Normal Multi CPU systems

System	BogoMips	Reporter
SMP2 Pentium/75	59.50	Michael Engel <engel@unix-ag.uni-siegen.de>
SMP2 Pentium/90	71.68	Edwin Whitelaw <elw@ivc.com>
SMP2 Pentium/90	71.98	Daniel Luhde-Thompson <dl10010@cam.ac.uk>
SMP2 Pentium/90	72.08	Alan Cox <alan@cymru.net>
SMP2 Pentium/100	79.46	Lam Dang <dangit@ix.netcom.com>
SMP2 Pentium/100	80.08	Christian Tan <pigeon@xs4all.nl>
SMP2 Pentium/100	80.08	McNalley <jmcnalle@attila.stevens-tech.edu>
SMP2 Pentium/120	95.85	Maurice de Bijl <maurice.d.bijl@topic.nl>
SMP2 Pentium/133	106.09	Maurice de Bijl <maurice.d.bijl@topic.nl>

TOC

```
SMP2 Pentium/166          133.53  <Reinhard.Simkovics@jk.uni-linz.ac.at>
SMP2 Pentium/200          158.72  Ed Hoppitt <edward@dcs.rhbnc.ac.uk>

SMP2 Pentium MMX/200      794.62  Kristian Koehntopp <kris@koehntopp.de>
SMP2 Pentium MMX/200      796.26  Eric Clark <eclark@opencominc.com>
SMP2 Pentium MMX/233      927.33  Jeff White <jwhite@ghq.com>

SMP2 Pentium Pro/150      299.00  Steven Gallo <smgallo@numenor.csgeeks.org>
SMP2 Pentium Pro/150      299.01  Eric van Dijken <E.vanDijken@PTT-Telecom.nl>
SMP2 Pentium Pro/166      331.78  Eric van Dijken <E.vanDijken@PTT-Telecom.nl>
SMP2 Pentium Pro/180      358.81  Eric van Dijken <E.vanDijken@PTT-Telecom.nl>
SMP2 Pentium Pro/180      358.81  Frankie East <fae2401@rit.edu>
SMP2 Pentium Pro/180      358.81  James K. Wiggs <wiggs@wolfenet.com>
SMP2 Pentium Pro/200      396.25  Will Shaw <shaww@dialup.dstm.com>
SMP2 Pentium Pro/200      398.13  David Konerding <dek@cgl.ucsf.edu>
SMP2 Pentium Pro/200      398.13  B. Heinen <benedikt.heinen@infrasys.ascom.ch>
SMP2 Pentium Pro/200      398.13  Ramon Huerta <huerta@minerva.ii.uam.es>
SMP2 Pentium Pro/200      398.14  C.-A. Possamai <camille@sugiton.cnrs-mrs.fr>
SMP2 Pentium Pro/200      398.14  Leland <llucius@millcom.com>
SMP2 Pentium Pro/200      398.14  John Lellis <lellis@dmccorp.com>
SMP2 Pentium Pro/200      398.14  Jim Gifford <jim@mail.rath.peachnet.edu>
SMP2 Pentium Pro/200      398.14  Manuel Galan <root@ulpgc.es>
SMP2 Pentium Pro/200      398.6   Bill Davidsen <davidsen@tmr.com>
SMP2 Pentium Pro/200      398.95  Fons Rademakers <f.rademakers@cern.ch>
SMP2 Pentium Pro/200      398.98  Greg Fausak <glfausak@august.com>
SMP2 Pentium Pro/200      400.18  Attila Karpati <karpati@cs.elte.hu>
SMP2 Pentium Pro/233      464.49  F. Baitinger <baiti@zenon.toplink.net>
SMP4 Pentium Pro/133      532.07  Erik Walthinsen <omega@sequent.com>
SMP4 Pentium Pro/200      794.62  Kenneth Hedlund <c415khd@nll.se>
SMP4 Pentium Pro/200      796.28  John Pelan <j.pelan@am.qub.ac.uk>

SMP2 Pentium II/233       466.94  Eli Kane <eli@crl.nmsu.edu>
SMP2 Pentium II/266       530.84  Shon Martin <cshoon@oberlin.edu>
SMP2 Pentium II/266       534.12  n.n. <service@cprompt.sk.ca>
SMP2 Pentium II/266       534.12  Emmanuel Tychon <manu@acm.org>
SMP2 Pentium II/274       546.41  Richard Jellinek <rj@suse.de>
SMP2 Pentium II/300       597.20  Paul Sheer <psheer@obsidian.co.za>
SMP2 Pentium II/300       598.02  Hans Ameel <hans@pla.to>
SMP2 Pentium II/300       599.76  S. Curtarolo <auro@spiro.fisica.unpd.it>
SMP2 Pentium II/333 ov    665.73  S. Curtarolo <auro@spiro.fisica.unpd.it>
SMP2 Pentium II/400       796.26  Michael Necaise <necaise@jlab.org>

SMP2 Celeron/450          894.57  Stephan Eisler <stephan@lingo.gun.sub.org>
```

9.13 Non-Linux systems (reference only)

```
System                      OS      BogoMips Reporter
Z80 TRS80 model 4           TRSDOS     0.004 <cervasio@airmail.net>
68000/8 Macintosh Classic   MacOS      0.53  <jimmy@typhoeus.dg.uoa.gr>
```

TOC

68020/20 Sun 3	SUNOS	2.0	<korpela@ssl.berkeley.edu>
68020/16 Macintosh LC	MacOS	3.09	<jimmy@typhoeus.dg.uoa.gr>
68020/25 Sun 3-180	SUNOS	4.0	<korpela@ssl.berkeley.edu>
Sparc Sun 3-60	SUNOS	4.00	<tranter@software.mitel.com>
Sparc Sun 3-80	SUNOS	4.00	<tranter@software.mitel.com>
68030/25 Macintosh IIci	NetBSD	5.62	<kirk-corey@uiowa.edu>
68040/25 Macintosh LC475	MacOS	12.27	<jimmy@typhoeus.dg.uoa.gr>
68040/33 HP9000-280	HP-UX	14	<niles@axp745.gsfc.nasa.gov>
68040/25 Apollo 5500	DomainOS	14.00	<kirk-corey@uiowa.edu>
Sparc Sun 4-280	SUNOS	16.0	<korpela@ssl.berkeley.edu>
Sparc Sun IPC/40	Solaris	16.00	<gong@cs.msu.su>
68040/25 HP9000-425	DomainOS	16.00	<kirk-corey@uiowa.edu>
68040 NextStep	(?)	16.26	<petergun@coffeehaus.com>
Sparc Sun SS1	SUNOS	18.00	<tranter@software.mitel.com>
Sparc Sun SS1	SUNOS	18.00	<jimmy@typhoeus.dg.uoa.gr>
IBM RS6000 320	AIX	18.00	<mg@svtpc3.fb12.tu-berlin.de>
Sparc Sun SS1+	SUNOS	19	<swmike@uplift.df.lth.se>
Sparc Sun SS1+	SUNOS	24.00	<tranter@software.mitel.com>
Sparc Sun IPC	SUNOS	24.00	<tranter@software.mitel.com>
IBM RS6000 320H	AIX	24.00	<mg@svtpc3.fb12.tu-berlin.de>
Sparc Sun Sparcstation2	SUNOS	26.00	<gong@cs.msu.su>
SparcClassic/50	Solaris	32.00	<gong@cs.msu.su>
Sparc Sun ELC	SUNOS	32.00	<tranter@software.mitel.com>
HP-PA 9000-720	HP-UX	32.00	<metod.kozelj@rzs-hm.si>
Sparc Sun SS10	SUNOS	34.00	<tranter@software.mitel.com>
Sparc Sun SS10	SUNOS	34.00	<makler@man.torun.pl>
Pentium/100	Win-NT	36.21	<marcus@healthchex.com>
Sparc Sun Sparcstation2	SUNOS	38.0	<korpela@ssl.berkeley.edu>
Mips R4000/100 Indy	(?)	48.00	<p.verwer@organon.akzonobel.nl>
Mips R4000/100 SGI IndySC	Irix	48.00	<lziegler@csbsju.edu>
HP-PA 9000-720	HP-UX	48.00	<metod.kozelj@rzs-hm.si>
Sparc SS10	SUNOS	48.00	<makler@man.torun.pl>
IBM RS-6000-250/66	AIX	52.00	<gl23@e4.hrz.uni-giessen.de>
Sparc Sun Sparcstation10d	SUNOS	54.0	<korpela@ssl.berkeley.edu>
Sparc SS1000 2CPU	SUNOS	58.00	<gong@cs.msu.su>
Sparc SS20	SUNOS	60.00	<makler@man.torun.pl>
IBM RS6000-250/80	AIX	62.00	<gl23@e4.hrz.uni-giessen.de>
Alpha 21064/133 Decstation	OSF1	64	<niles@axp745.gsfc.nasa.gov>
Alpha 21064/133 Dec3000	OSF1	66.00	<daniels@helplaxp1.harvard.edu>
Sparc Sun SS5	SUNOS	68.00	<tranter@software.mitel.com>
Sparc Sun SS20	SUNOS	72.00	<tranter@software.mitel.com>
Sparc Sun SS20/712	Solaris	74	<spohr@qmos.com>
Mips R4400/150 Challenge	(?)	74.00	<p.verwer@organon.akzonobel.nl>
Mips R4400/150 Indigo2Extr	(?)	74.00	<p.verwer@organon.akzonobel.nl>
HP-PA 9000-715	HP-UX	74.00	<metod.kozelj@rzs-hm.si>
Sparc SS5 sun4m	SUNOS	84.00	<chrisv@allegria.com>
Sparc SS1000E	SUNOS	84.00	<makler@man.torun.pl>
Alpha	OSF1	92.00	<petergun@coffeehaus.com>
Mips R4400/200 Indigo2Extr	(?)	98.00	<p.verwer@organon.akzonobel.nl>
HP-PA 9000-735/99	HP-UX	98.00	<lankhors@cs.rug.nl>
Sparc SS5	SUNOS	104.00	<makler@man.torun.pl>
Sparc Sun SS4/110	Solaris	108	<spohr@qmos>

TOC

Sparc Sun SS4	Solaris	108.00	<jimmy@typhoeus.dg.uoa.gr>
Sparc Sun SS5	SUNOS	110	<swmike@uplift.df.lth.se>
Alpha 21064A/233	OSF1	114	<niles@axp745.gsfc.nasa.gov>
Alpha 3000 21064/300	OSF1	120.00	<gl23@e4.hrz.uni-giessen.de>
HP-PA 700/125	HP-UX	122	<niles@axp745.gsfc.nasa.gov>
HP-PA 9000-735/125	HP-UX	122.00	<lankhors@cs.rug.nl>
Mips R4600/133 SGI Indy	Irix	132	<lziegler@csbsju.edu>
MIPS R5000/150	Irix	148.00	<nachiket@indimax.hydroqual.com>
IBM 6x86	OS/2	149.13	<mikedg@ghostbbs.pandora.it>
Alpha	OSF1	180.0	<mauger@ensinfo.univ-nantes.fr>
MIPS R10000/195	Irix	194.00	<nachiket@indimax.hydroqual.com>
Alpha 21164/333 AS500	OSF1	222.00	<daniels@helplaxp1.harvard.edu>
Sparc Ultra-1 sun4u	Solaris	254.00	<alternat@rwth-aachen.de>
PPC604/133 IBM RS6000	AIX	254.00	<kirk-corey@uiowa.edu>
IBM RS-6000 43P powerpc	AIX	260.00	<gl23@e4.hrz.uni-giessen.de>
Sparc U-1 sun4u	SunOS	284.00	<gl23@e4.hrz.uni-giessen.de>
Alpha 21164/433 DPW433au	OSF1	286.00	<daniels@helplaxp1.harvard.edu>
HP9000-C160/160	HP-UX	316.00	<nicolai@prz.tu-berlin.de>
Sparc Sun US1/170	Solaris	330	<spohr@qmos>
Sparc Sun US1/167	Solaris	330	<jimmy@typhoeus.dg.uoa.gr>
Sparc Ultra-1 sun4u	SUNOS	334.00	<chrisv@allegria.com>
Alpha 3000-600S	VMS	348.61	<metod.kozelj@rzs-hm.si>
Alpha server 1000-4/200	VMS	397.68	<metod.kozelj@rzs-hm.si>
Sparc Sun-Ultra30/248	Solaris	482.00	<marc.Quinton@stna.dgac.fr>
Sparc US2/296 2cpu	SUNOS	596.00	<manu@acm.org>
Cray J90 Y-MP/100 16cpu	Unicos	912.00	<lankhors@cs.rug.nl>
Sequent Symmetry 6xP5/166	Dynix	984.00	<omega@sequent.com>
Sequent Symmetry 16xP5/66	Dynix	1056.00	<omega@sequent.com>
HP9000-C160/160	HP-UX	1278.00	<nicolai@prz.tu-berlin.de>
Sequent NUMA-Q 12x P6/180	Dynix	1416.00	<omega@sequent.com>
Sequent NUMA-Q 32x P6/180	Dynix	3776.00	<omega@sequent.com>

10. Signature

My son Roderick and his sister Isolde are doing well. Isolde, 4yrs, is at school, and already knows how to write her name. Her brother Roderick, 2yrs, loudly acknowledges that he knows his letters too: P of papa, R of Roderick!

Met vriendelijke groeten, Wim van Dorst.

```
-----------------------------------------------------------------------
Wim van Dorst, Clifton Scientific Text Services, tel/fax +31 355 242 319
                    This sentence have three erors
```

Bridging mini-Howto

Christopher Cole cole@coledd.com

v1.11, 7 September 1998

This document describes how to setup an ethernet bridge. What is an ethernet bridge? An ethernet bridge is a device that controls data packets within a subnet in an attempt to cut down the amount of traffic. A bridge is usually placed between two separate groups of computers that talk within themselves, but not so much with the computers in the other group. A good example of this is to consider a cluster of Macintoshes and a cluster of unix machines. Both of these groups of machines tend to be quite chatty amongst themselves, and the traffic they produce on the network causes collisions for the other machines who are trying to speak to one another. A bridge would be placed between these groups of computers. The job of the bridge is then to examine the destination of the data packets one at a time and decide whether or not to pass the packets to the other side of the ethernet segment. The result is a faster, quieter network with less collisions.

1. Setup

2. Common problems

1. Setup

1. Get ``Bridge Config'':

   ftp://shadow.cabi.net/pub/Linux/BRCFG.tgz

2. Enable multiple ethernet devices on your machine by adding this to your `/etc/lilo.conf`, and re-run `lilo`:

   ```
   append = "ether=0,0,eth1"
   ```

   If you have three interfaces on your bridge, use this line instead:

   ```
   append = "ether=0,0,eth1 ether=0,0,eth2"
   ```

   More interfaces can be found by adding more ether statements. By default a stock Linux kernel probes for a single ethercard, and once one is found the probe ceases. The above append statement tells the kernel to keep probing for more ethernet devices after the first one is found. Alternatively, the boot parameter can be used instead:

   ```
   linux ether=0,0,eth1
   ```

   Or, with 3 interfaces, use:

   ```
   linux ether=0,0,eth1 ether=0,0,eth2
   ```

3. Recompile the kernel with `BRIDGING` enabled.
4. A bridge should not have an IP address. It CAN, but a plain bridge doesn't need one. To remove the IP address from your bridge, go to `/etc/sysconfig/network-scripts/` (for a RedHat system) and copy `ifcfg-lo0` to `ifcfg-eth0` & `ifcfg-eth1`. In these 2 eth files, change the line containing ``DEVICE=lo'' to ``DEVICE=eth0'' and ``DEVICE=eth1''. Other distributions may deviate from this, do what you need to do! If there are more than 2 interfaces to this bridge, be sure to make the corresponding configurations to those, as well.
5. Reboot, so you are running the new kernel with bridging in it, and also to make sure that an IP addresses are not bound to the network interfaces.
6. Once the system is back up, put the ethernet cards into promiscuous mode, so they will look at every packet that passes by its interface:

   ```
   ifconfig eth0 promisc ; ifconfig eth1 promisc
   ```

All interfaces which are connected to network segments to be bridged are to be put into promiscuous mode.

7. Turn bridging ON using the `brcfg` program:

```
brcfg -ena
```

8. Verify that there is different traffic on each interface:

```
tcpdump -i eth0      (in one window)
tcpdump -i eth1      (in another window)
```

9. Run a sniffer or `tcpdump` on another machine to verify the bridge is separating the segment correctly.

---

2. Common problems

1. **Question**

   I get the message

   ```
   ioctl(SIOCGIFBR) failed: Package not installed
   ```

   What does this mean?

   Answer

   You don't have bridging capability in your kernel. Get a 2.0 or greater kernel, and recompile with the `BRIDGING` option enabled.

2. **Question**

   Machines on one side cannot ping the other side!

   Answer

   - Did you enable bridging using ``brcfg -ena''? (`brcfg` should say ``bridging is ENABLED'')
   - Did you put the interfaces into promiscuous mode? (issue the ``ifconfig'' command. The ``PROMISC'' flag should be on for both interfaces.)
   - If using multiple-media interface adapters, make sure that the correct one is enabled. You may need to use the config/setup program that came with the network interface card.

3. **Question**

   I cannot `telnet`/`ftp` from the bridge! Why?

   Answer

   This is because there is no IP address bound to any of bridge interfaces. A bridge is to be a transparent part of a network.

4. **Question**

   What do I need to set up in the way of routing?

   Answer

   Nothing! All routing intelligence is handled by the bridging code in the kernel. To see the ethernet addresses as they are learned by the bridge, use the `brcfg` program in debug mode:

   ```
   brcfg -deb
   ```

TOC

5. **Question**

   The bridge appears to work, but why doesn't ``traceroute'' show the bridge as a part of the path?

   Answer

   Due to the nature of a bridge, a ``traceroute'' should NOT show the bridge as a part of the path. A bridge is to be a transparent component of the network.

6. **Question**

   Is it necessary to compile `IP_FORWARD` into the kernel?

   Answer

   No. The bridging code in the kernel takes care of the packet transport. `IP_FORWARD` is for a gateway which has IP addresses bound to its interfaces.

7. **Question**

   Why are the physical ethernet addresses for port 1 and port 2 the same according to the ``brcfg'' program? Shouldn't they be different?

   Answer

   No. Every port on a bridge intentionally is assigned the same physical ethernet address by the bridging code.

8. **Question**

   Bridging does not appear to be an option when performing a make config on the kernel. How does one enable it?

   Answer

   During the kernel config, answer 'Y' to the question, ``Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [Y/n/?]''.

9. **Question**

   Too many hubs (4 or more) chained one after another (in series) cause timing problems on an ethernet. What effect does a bridge have in a subnet that is layered with hubs?

   Answer

   A bridge resets the 3/4/5 hubs rule. A bridge does not deal with packets the way a hub does, and is therefore not a contributor to timing problems on a network.

10. **Question**

    Can a bridge interface to both 10Mb and 100Mb ethernet segments? Will such a configuration slow down the rest of the traffic on the high speed side?

    Answer

    Yes, a bridge can tie together a 10Mb segment with a 100Mb segment. As long as the

network card on the fast network is 100Mb capable, TCP takes care of the rest. While it true that the packets from a host in the 100Mb network communicating to a host in the 10Mb network are moving at only 10Mb/s, the rest of the traffic on the fast ethernet is not slowed down.

Linux Bridge+Firewall Mini-HOWTO version 1.2.0

Peter Breuer (ptb@it.uc3m.es)

v, 19 December 1997

1. Introduction

2. What and Why (and How?)

- 2.1 What
- 2.2 Why
- 2.3 How?

3. BRIDGING

- 3.1 Software
- 3.2 Prior Reading.
- 3.3 Boot configuration
- 3.4 Kernel configuration
- 3.5 Network addresses
- 3.6 Network routing
- 3.7 Card configuration
- 3.8 Additional routing
- 3.9 Bridge Configuration
- 3.10 Try it out
- 3.11 Checks

4. FIREWALLING

- 4.1 Software and reading
- 4.2 Preliminary checks
- 4.3 Default rule
- 4.4 Holes per address
- 4.5 Holes per protocol
- 4.6 Checks

1. Introduction

You should look at the original Bridging mini-HOWTO by Chris Cole for a different perspective on this. He is chris@polymer.uakron.edu. The version of his HOWTO that I have based this document on (alternatively, ripped off) is 1.03 dated Aug 23 1996.

2. What and Why (and How?)

2.1 What

A bridge is an intelligent connecting wire betwen two network cards. A firewall is an intelligent insulator.

2.2 Why

You might want a bridge if you have several computers:

1. to save the price of a new hub when you just happen to have an extra ethernet card available.
2. to save the bother of learning how to do IP-forwarding and other tricks when you _have_ two cards in your computer.
3. to avoid maintenance work in the future when things change around!

``Several computers'' might be as few as three if those are routing or bridging or just moving around the room from time to time! You also might want a bridge just for the fun of finding out what it does. 2 was what I wanted a bridge for.

If you are really interested in 1, you have to be one of the very few. Check the NET-2-HOWTO and the Serial-HOWTO for better tricks.

You want a firewall if

1. you are trying to protect your network from external accesses, or
2. you are trying to deny access to the world outside from your network.

Curiously, I needed 2 here too. Policy at my university presently is that we should not act as internet service providers to undergraduates.

2.3 How?

I started out bridging the network cards in a firewalling machine and ended up firewalling without having cut the bridge. It seems to work and is more flexible than either configuration alone. I can take down the firewall and keep bridging or take down the bridge when I want to be more circumspect.

I would guess that the bridge code lives just above the physical device layer and the firewalling code lives one layer higher up, so that the bridging and firewalling configurations effectively act as though they are running connected together ``in sequence'' and not ``in parallel'' (ouch!). Diagram:

```
-> Bridge-in -> Firewall-in -> Kernel -> Firewall-out -> Bridge-out ->
```

There is no other way to explain how one machine can be a ``conductor'' and an ``insulator'' at the same time. There are a few caveats but I'll come to those later. Basically you must route packets that you want to firewall. Anyway, it all seems to work together nicely for me. Here is what you do ...

3. BRIDGING

3.1 Software

Get the bridge configuration utility from Alan Cox's home pages. This is the same reference as in Chris' document. I just didn't realize that it was an ftp and not an http URL ...

3.2 Prior Reading.

Read the Multiple Ethernet HOWTO for some advice on getting more than one network card recognized and configured.

Yet more details of the kind of boot magic that you may need are in the Boot Prompt HOWTO.

You may be able to get away without the NET-2 HOWTO. It is a good long read and you will have to pick from it the details you need.

3.3 Boot configuration

The reading material above will tell you that you need to prepare the kernel to recognize a second ethernet device at boot up by adding this to your **/etc/lilo.conf**, and then re-run **lilo**:

```
append = "ether=0,0,eth1"
```

Note the "eth1". "eth0" is the first card. "eth1" is the second card. You can always add the boot parameters in your response to the line that lilo offers you. This is for three cards:

```
linux ether=0,0,eth1 ether=0,0,eth2
```

I use **loadlin** to boot my kernel from DOS:

```
loadlin.exe c:\vmlinuz root=/dev/hda3 ro ether=0,0,eth1 ether=0,0,eth2
```

Note that this trick makes the kernel probe at bootup. That will not happen if you load the ethernet drivers as **modules** (for safety since the probe order can't be determined) so if you use modules you will have to add the appropriate IRQ and port parameters for the driver in your **/etc/conf.modules**. I have at least

```
alias eth0 3c509
alias eth1 de620
options 3c509 irq=5 io=0x210
options de620 irq=7 bnc=1
```

You can tell if you use modules by using ``ps -aux'' to see if **kerneld** is running and checking that there are .o files in a subdirectory of your **/lib/modules** directory. You want the directory named with what uname -r tells

TOC

you. If you have kerneld and/or you have a foo.o then edit **/etc/conf.modules** and read the man page for depmod carefully.

Note also that until recently (kernel 2.0.25) the **3c509** driver could not be used for more than one card if used as a module. I have seen a patch floating around that fixes the oversight. It may be in the kernel when you read this.

3.4 Kernel configuration

Recompile the kernel with bridging enabled.

```
CONFIG_BRIDGE=y
```

I also compiled with firewalling and IP-forwarding and -masquerading and the rest enabled. Only if you want firewalling too ...

```
CONFIG_FIREWALL=y
CONFIG_NET_ALIAS=y
CONFIG_INET=y
CONFIG_IP_FORWARD=y
CONFIG_IP_MULTICAST=y
CONFIG_IP_FIREWALL=y
CONFIG_IP_FIREWALL_VERBOSE=y
CONFIG_IP_MASQUERADE=y
```

You don't need all of this. What you do need apart from this is the standard net configuration:

```
CONFIG_NET=y
```

and I do not think you need worry about any of the other networking options. I have any options that I did not actually compile into the kernel available through kernel modules that I can add in later.

Install the new kernel in place, rerun lilo and reboot with the new kernel. Nothing should have changed at this point!

3.5 Network addresses

Chris says that a bridge should not have an IP address but that is not the setup to be described here.

You are going to want to use the machine for connecting to the net so you need an address and you need to make sure that you have the loopback device configured in the normal way so that your software can talk to the places they expect to be able to talk to. If loopback is down the name resolver or other net sevices might fail. See the NET-2-HOWTO, but your standard configuration should already have done this bit:

```
ifconfig lo 127.0.0.1 route add -net 127.0.0.0
```

You will have to give addresses to your network cards. I altered the /etc/rc.d/rc.inet1 file in my slackware (3.x) to setup two cards and you should also essentially just look for your net configuration file and double or treble the number of instructions in it. Suppose that you already have an address at

```
192.168.2.100
```

(that is in the private net reserved address space, but never mind - it won't hurt anybody if you use this address by mistake) then you probably already have a line like

```
ifconfig eth0 192.168.2.100 netmask 255.255.255.0 metric 1
```

in your configuration. The first thing you are going to probably want to do is cut the address space reached by this card in half so that you can eventually bridge or firewall the two halves. So add a line which reduces the mask to address a smaller number of machines:

```
ifconfig eth0 netmask 255.255.255.128
```

Try it too. That restricts the card to at most the address space between .0 and .127.

Now you can set your second card up in the other half of the local address space. Make sure that nobody already has the address. For symmetry I set it at 228=128+100 here. Any address will do so long as it is not in the other card's mask, and even then, well, maybe. Avoid special addresses like .0, .1, .128 etc. unless you really know what you are doing.

```
ifconfig eth1 192.168.2.228 netmask 255.255.255.128 metric 1
```

That restricts the second card to addresses between .128 and .255.

3.6 Network routing

This is where I have to announce the caveats in the bridging + firewalling scheme: you cannot firewall packets which are not routed. No routes, no firewall. At least this appears to be true in the 2.0.30 and more recent kernels. The firewalling filters are closely involved with the ip-forwarding code.

That does not mean that you cannot bridge. You can bridge between two cards and firewall them from a third. You can have only two cards and firewall both of them against an outside IP such as a nearby router, provided that the router is routed by you to exactly one of the cards.

In other words, since I will be doing firewalling, so I want to precisely control the physical destination of some packets.

I have the small net of machines attached to a hub hanging off eth0, so I configure a net there:

```
route add -net 192.168.2.128 netmask 255.255.255.128 dev eth0
```

The 128 would be 0 if I had a full class C network there. I don't, by definition, since I just halved the address space. The "dev eth0" is not necessary here because the cards address falls within the mask, but it may be necessary for you. One might need more than one card holding up this subnet (127 machines on one segment, oh yeah) but those cards would be being bridged under the same netmask so that they appear as one to the routing code.

On the other card I have a line going straight through to a big router that I trust.

TOC

```
                                                  client 129
          __                                            |    __
client 1   \    .0                      .128            |   /   net 1
client 2 --- Hub - eth0 - Kernel - eth1 - Hub - Router ---  net 2
client 3 __/       .100            .228         .2 |   \__ net 3
                                                  |
                                                  client 254
```

I attach the address of the router to that card as a fixed ("static") route because it would otherwise fall within the first cards netmask and the kernel would be thinking wrongly about how to send packets to the big router. I will want to firewall these packets and that is another reason fow wanting to route them specifically.

```
route add 192.168.2.2 dev eth1
```

I don't need it, since I don't have any more machines in that half of the address space, but I declare a net also on the second card. Separating my interfaces into two sets via routing will allow me to do very tight firewalling eventually , but you can get away with far less routing than this.

```
route add -net 192.168.2.128 netmask 255.255.255.128 dev eth1
```

I also need to send all non-local packets out to the world so I tell the kernel to send them to the big router

```
route add default gw 192.168.2.2
```

3.7 Card configuration

So much was standard networking setup, but we are bridging so we also have to listen on both (?) cards for packets that are not aimed at us. The following should go into the network configuration file.

```
ifconfig promisc eth0 ifconfig promisc eth1
```

The man page says allmulti=promisc, but it didn't work for me.

3.8 Additional routing

One thing that I noticed was that I had to put at least the second card into a mode where it would respond to the big router's questions about which machines I was hiding in my local net.

```
ifconfig arp eth1
```

For good measure I did this to the other card too.

```
ifconfig arp eth0.
```

TOC

3.9 Bridge Configuration

Put bridging enabling on and into your configuration file:

```
brcfg -enable
```

You should have been trying this out in real time all along, of course! The bridge configure will bring up some numbers. You can experiment with turning on and off the ports one at a time

```
brcfg -port 0 -disable/-enable
 brcfg -port 1 -disable/-enable
```

You get status reports anytime by just running

```
brcfg
```

without any parameters. You will see that the bridge listens,learns, and then does forwarding. (I don't understand why the code repeats the same hardware addresses for both my cards, but never mind .. Chris' howto say that is OK)

3.10 Try it out

If you are still up and running as things are, try out your configuration script for real by taking down both cards and then executing it:

```
ifconfig eth0 down ifconfig eth1 down /etc/rc.d/rc.inet1
```

With any luck the various subsystems (**nfs**, **ypbind**, etc.) won't notice. *Do not try this unless you are sitting at the keyboard!*

If you want to be more careful than this, you should take down as many daemons as possible beforehand, and unmount nfs directories. The worst that can happen is that you have to reboot in single-user mode (the "**single**" parameter to **lilo** or **loadlin**), and take out your changes before rebooting with things the way they were before you started.

3.11 Checks

Verify that there is different traffic on each interface:

```
tcpdump -i eth0
(in one window)

tcpdump -i eth1
(in another window)
```

You should get used to using **tcpdump** to look for things that should not be happening or that are happening and should not.

For instance look for packets that have gone through the bridge to the second card from the internal net. He am looking for packets from the machine with address .22:

```
tcpdump -i eth1 -e host 192.168.2.22
```

Then send a ping from the .22 host to the router. You should see the packet reported by tcpdump.

At this stage you should have a bridge ready that also has two network addreses. Test that you can ping them from outside and inside your local net, and that you can telnet and ftp around between inside and outside too.

---

4. FIREWALLING

4.1 Software and reading

You should read the Firewall-HOWTO.

That will tell you where to get **ipfwadm** if you don't already have it. There are other tools you can get but I made no progress until I tried **ipfwadm**. It is nice and low level! You can see exactly what it is doing.

4.2 Preliminary checks

You have compiled IP-forwarding and masquerading into the kernel so you will want to check that the firewall is in its default (accepting) state with

```
ipfwadm -I -l ipfwadm -O -l ipfwadm -F -l
```

That is respectively, "display the rules affecting the .." incoming or outgoing or forwarding (masquerading) ".. sides of the firewall". The "-l" means "list".

You might have compiled in accounting too:

```
ipfwadm -A -l
```

You should see that there are no rules defined and that the default is to accept every packet. You can get back to this working state anytime with

```
ipfwadm -I -f
ipfwadm -O -f
ipfwadm -F -f
```

The "-f" means "flush". You may need to use that.

4.3 Default rule

I want to cut the world off from my internal net and do nothing else, so I will want to give as a last (default) rule that the firewall should ignore any packets coming in from the internal net and directed to outside. I put all the rules (in this order) into **/etc/rc.d/rc.firewall** and execute it from **/etc/rc.d/rc.local** at bootup.

```
ipfwadm -I -a reject -S 192.168.2.0/255.255.255.128 -D 0.0.0.0/0.0.0.0
```

The "-S" is the source address/mask. The "-D" is the destination address/mask.

This format to is rather long-winded. **Ipfwadm** is intelligent about network names and some common abbreviations. Check the man pages.

TOC

It is possibly more convenient to put some or all of these rules on the outgoing half of the firewall by using instead of "-I", but I'll state the rules here all formulated for the incoming half.

4.4 Holes per address

Before that default rule, I have to place some rules that serve as exceptions to this general denial of external services to internal clients.

I want to treat the firewall machines address on the internal net specially. I will stop people logging in to the firewall machine unless they have special permission, but once they are there they should be allowed to talk to the world.

```
ipfwadm -I -i accept -S 192.168.2.100/255.255.255.255 \
 -D 0.0.0.0/0.0.0.0
```

I also want the internal clients to be able to talk to the firewalling machine. Maybe they can persuade it to let them get out!

```
ipfwadm -I -i accept -S 192.168.2.0/255.255.255.128 \
 -D 192.168.2.100/255.255.255.255
```

Check at this point that you can get in to the clients from outside the firewall via **telnet**, but that you cannot get out. That should mean that you can just about make first contact, but the clients cannot send you any prompts. You should be able to get all the way in if you use the firewall machine as a staging post. Try **rlogin** and **ping** too, with **tcpdump** running on one card or the other. You should be able to make sense of what you see.

4.5 Holes per protocol

I went on to relax the rules protocol by protocol. I want to allow pings from the outside to the inside to get an echo back, for instance, so I inserted the rule:

```
ipfwadm -I -i accept -P icmp -S 192.168.2.0/255.255.255.128 \
 -D 0.0.0.0/0.0.0.0
```

The "`-P icmp`" works the protocol-specific magic.

Until I get hold of an **ftp** proxy I am also allowing ftp calls out with port-specific relaxations. This targets ports 20 21 and 115 on outside machines.

```
ipfwadm -I -i accept -P tcp -S 192.168.2.0/255.255.255.128 \
 -D 0.0.0.0/0.0.0.0 20 21 115
```

I could not make **sendmail** between the local clients work without a nameserver. Rather than set up a nameserver right then on the firewall, I just lifted the firewall for tcp domain service queries precisely aimed at the nearest existing nameserver and put its address in the clients **/etc/resolv.conf** ("`nameserver 123.456.789.31`" on a separate line).

```
ipfwadm -I -i accept -P tcp -S 192.168.2.0/255.255.255.128 \
```

```
-D 123.456.789.31/255.255.255.255 54
```

You can find which port number and protocol a service requires with **tcpdump**. Trigger the service with a an **ftp** or a **telnet** or whatever to or from the internal machine and then watch for it on the input and output ports of the firewall with **tcpdump**:

```
tcpdump -i eth1 -e host client04
```

for example. The **/etc/services** file is another important source of clues. To let **telnet** and **ftp** IN to the firewall from outside, you have to allow the local clients to call OUT on a specific port. I understand why this is necessary for **ftp** - it's the server that establishes the data stream in the end - but I am not sure why **telnet** also needs this.

```
ipfwadm -I -i accept -P tcp -S 192.168.2.0/255.255.255.128 ftp telnet \
 -D 0.0.0.0/0.0.0.0
```

There is a particular problem with some daemons that look up the hostname of the firewalling machine in order to decide what is their networking address. **Rpc.yppasswdd** is the one I had trouble with. It insists on broadcasting information that says it is outside the firewall (on the second card). That means the clients inside can't contact it.

Rather than start IP aliasing or change the daemon code, I mapped the name to the inside card address on the clients in their **/etc/hosts**.

4.6 Checks

You want to test that you can still **telnet**, **rlogin** and **ping** from the outside. From the inside you should be able to **ping** out. You should also be able to **telnet** to the firewall machine from the inside and the latter should be able to do anything.

That is it. At this point you probably want to learn about **rpc**/**Y**ellow **P**ages and the interaction with the password file. The firewalled network wants to run without its unprivileged users being able to log on to the firewall - and thus get out. Some other HOWTO!

Bzip2 mini-HOWTO

David Fetter, david@fetter.org

v2.00, 22 August 1999

This document tells how to use the new bzip2 compression program. The local copy of the sgml at the current site is here, and the "author-itative" sgml is here.

1. Introduction

- 1.1 **Revision History**

2. Getting bzip2

- 2.1 Bzip2-HOWTO in your language
- 2.2 Getting bzip2 precompiled binaries
- 2.3 Getting bzip2 sources
- 2.4 Compiling bzip2 for your machine

3. Using bzip2 by itself

4. Using bzip2 with tar

- 4.1 Easiest to set up:
- 4.2 Easy to set up, fairly easy to use, no need for root privileges:
- 4.3 Also easy to use, but needs root access.

5. Using bzip2 with less

TOC

6. Using bzip2 with emacs

- 6.1 Changing emacs for everyone:
- 6.2 Changing emacs for one person:

7. Using bzip2 with wu-ftpd

8. Using bzip2 with grep

9. Using bzip2 with Netscape under the X.

10. Using bzip2 to recompress other compression formats

---

1. Introduction

Bzip2 is a groovy new algorithm for compressing data. It generally makes files that are 60-70% of the size of their gzip'd counterparts.

This document will take you through a few common applications for bzip2.

Future versions of the document will have applications of libbzip2, the bzip2 C library which bzip2's author, Julian Seward has kindly written. The bzip2 manual, which includes low-level information about the library, can be found here.

Future versions of the document may also include a summary of the discussion over whether (and how) bzip2 should be used in the Linux kernel.

1.1 Revision History

v2.00

Changed the Using bzip2 with less section so .tar.bzip2 files can actually be read. Thanks to Nicola Fabiano for the correction.

Updated buzzit utility.

Updated tar information.

v1.92

Updated the Getting bzip2 binaries section, including adding S.u.S.E.'s.

v1.91

Corrected a typo and clarified some shell idioms in the section on using bzip2 with tar. Thanks to Alessandro Rubini for these.

Updated the buzzit tool not to stomp on the original bzip2 archive.

Added bgrep, a zgrep-like tool.

v1.9

Clarified the gcc 2.7.* problem. Thanks to Ulrik Dickow for pointing this out.

Added Leonard Jean-Marc's elegant way to work with tar.

TOC

Added Linus Ãkerlund's Swedish translation.

Fixed the wu-ftpd section per Arnaud Launay's suggestion.

Moved translations to their own section.

v1.8

Put buzzit and tar.diff in the sgml where they belong. Fixed punctuation and formatting. Thanks to Arnaud Launay for his help correcting my copy. :-)

Dropped xv project for now due to lack of popular interest.

Added teasers for future versions of the document.

v1.7

Added buzzit utility. Fixed the patch against gnu tar.

v1.6

Added TenThumbs' Netscape enabler.

Also changed lesspipe.sh per his sugestion. It should work better now.

v1.5

Added Arnaud Launay's French translation, and his wu-ftpd file.

v1.4

Added Tetsu Isaji's Japanese translation.

v1.3

Added Ulrik Dickow's .emacs for 19.30 and higher.

(Also corrected jka-compr.el patch for emacs per his suggestion. Oops! Bzip2's doesn't yet(?) have an "append" flag.)

v1.2

Changed patch for emacs so it automagically recognizes .bz2 files.

v1.1

Added patch for emacs.

v1.0

Round 1.

---

2. Getting bzip2

Bzip2's home page is at The UK home site. The United States mirror site is here.

2.1 Bzip2-HOWTO in your language

French speakers may wish to refer to Arnaud Launay's French documents. The web version is here, and you can use ftp here Arnaud can be contacted by electronic mail at this address

Japanese speakers may wish to refer to Tetsu Isaji's Japanese documents here. Isaji can be reached at his home page, or by electronic mail at this address.

Swedish speakers may wish to refer to Linus Ãkerlund's Swedish documents here. Linus can be reached by electronic mail at this address.

2.2 Getting bzip2 precompiled binaries

See the home sites.

2.3 Getting bzip2 sources

They come from the Official sites (see Getting Bzip2 for where.

2.4 Compiling bzip2 for your machine

If you have gcc 2.7.*, change the line that reads

```
CFLAGS = -O3 -fomit-frame-pointer -funroll-loops
```

to

```
CFLAGS = -O2 -fomit-frame-pointer
```

that is, replace -O3 with -O2 and drop the -funroll-loops. You may also wish to add any -m* flags (like -m486, for example) you use when compiling kernels.

Avoiding -funroll-loops is the most important part, since this will cause many gcc 2.7's to generate wrong code, and all gcc 2.7's to generate slower and larger code. For other compilers (lcc, egcs, gcc 2.8.x) the default CFLAGS are fine.

After that, just `make` it and install it per the README.

3. Using bzip2 by itself

Read the Fine Manual Page :)

4. Using bzip2 with tar

Listed below are three ways to use bzip2 with tar, namely

4.1 Easiest to set up:

This method requires no setup at all. To un-tar the bzip2'd tar archive, foo.tar.bz2 in the current directory, do

```
/path/to/bzip2 -cd foo.tar.bz2 | tar xf -
```

or

```
tar --use-compress-prog=bzip2 xf foo.tar.bz2
```

These work, but can be a PITA to type often.

4.2 Easy to set up, fairly easy to use, no need for root privileges:

Thanks to Leonard Jean-Marc for the tip. Thanks also to Alessandro Rubini for differentiating bash from the csh's.

In your .bashrc, you can put in a line like this:

```
alias btar='tar --use-compress-program /usr/local/bin/bzip2 '
```

In your .tcshrc, or .cshrc, the analogous line looks like this:

```
alias btar 'tar --use-compress-program /usr/local/bin/bzip2'
```

4.3 Also easy to use, but needs root access.

Update your tar to GNU's newest version, which is currently 1.13.10. It can be found at GNU's ftp site or any mirror.

5. Using bzip2 with less

To uncompress bzip2'd files on the fly, i.e. to be able to use "less" on them without first bunzip2'ing them, you can make a lesspipe.sh (man less) like this:

```
#!/bin/sh
# This is a preprocessor for 'less'.  It is used when this environment
# variable is set:   LESSOPEN="|lesspipe.sh %s"

  case "$1" in
  *.tar) tar tvvf $1 2>/dev/null ;; # View contents of various tar'd files
  *.tgz) tar tzvvf $1 2>/dev/null ;;
# This one work for the unmodified version of tar:
  *.tar.bz2) bzip2 -cd $1 $1 2>/dev/null | tar tvvf - ;;
#This one works with the patched version of tar:
# *.tar.bz2) tyvvf $1 2>/dev/null ;;
  *.tar.gz) tar tzvvf $1 2>/dev/null ;;
  *.tar.Z) tar tzvvf $1 2>/dev/null ;;
  *.tar.z) tar tzvvf $1 2>/dev/null ;;
  *.bz2) bzip2 -dc $1  2>/dev/null ;; # View compressed files correctly
  *.Z) gzip -dc $1  2>/dev/null ;;
  *.z) gzip -dc $1  2>/dev/null ;;
  *.gz) gzip -dc $1  2>/dev/null ;;
  *.zip) unzip -l $1 2>/dev/null ;;
  *.1|*.2|*.3|*.4|*.5|*.6|*.7|*.8|*.9|*.n|*.man) FILE=`file -L $1` ; # groff src
    FILE=`echo $FILE | cut -d ' ' -f 2`
    if [ "$FILE" = "troff" ]; then
      groff -s -p -t -e -Tascii -mandoc $1
    fi ;;
  *) cat $1 2>/dev/null ;;
#  *) FILE=`file -L $1` ; # Check to see if binary, if so -- view with 'strings'
#    FILE1=`echo $FILE | cut -d ' ' -f 2`
#    FILE2=`echo $FILE | cut -d ' ' -f 3`
#    if [ "$FILE1" = "Linux/i386" -o "$FILE2" = "Linux/i386" \
#         -o "$FILE1" = "ELF" -o "$FILE2" = "ELF" ]; then
#      strings $1
#    fi ;;
  esac
```

6. Using bzip2 with emacs

6.1 Changing emacs for everyone:

I've written the following patch to jka-compr.el which adds bzip2 to auto-compression-mode.

Disclaimer: I have only tested this with emacs-20.2, but have no reason to believe that a similar approach won't work with other versions.

To use it,

1. Go to the emacs-20.2/lisp source directory (wherever you untarred it)
2. Put the patch below in a file called jka-compr.el.diff (it should be alone in that file ;).
3. Do

```
patch < jka-compr.el.diff
```

4. Start emacs, and do

```
M-x byte-compile-file jka-compr.el
```

5. Leave emacs.
6. Move your original jka-compr.elc to a safe place in case of bugs.
7. Replace it with the new jka-compr.elc.
8. Have fun!

```
--- jka-compr.el        Sat Jul 26 17:02:39 1997
+++ jka-compr.el.new    Thu Feb  5 17:44:35 1998
@@ -44,7 +44,7 @@
 ;; The variable, jka-compr-compression-info-list can be used to
 ;; customize jka-compr to work with other compression programs.
 ;; The default value of this variable allows jka-compr to work with
-;; Unix compress and gzip.
+;; Unix compress and gzip.  David Fetter added bzip2 support :)
 ;;
 ;; If you are concerned about the stderr output of gzip and other
 ;; compression/decompression programs showing up in your buffers, you
@@ -121,7 +121,9 @@


 ;;; I have this defined so that .Z files are assumed to be in unix
-;;; compress format; and .gz files, in gzip format.
+;;; compress format; and .gz files, in gzip format, and .bz2 files,
+;;; in the snappy new bzip2 format from http://www.muraroa.demon.co.uk.
```

```
+;;; Keep up the good work, people!
 (defcustom jka-compr-compression-info-list
   ;;[regexp
   ;; compr-message  compr-prog  compr-args
@@ -131,6 +133,10 @@
      "compressing"    "compress"     ("-c")
      "uncompressing"  "uncompress"   ("-c")
      nil t]
+    ["\\.bz2\\'"
+     "bzip2ing"        "bzip2"         ("")
+     "bunzip2ing"      "bzip2"         ("-d")
+     nil t]
     ["\\.tgz\\'"
      "zipping"        "gzip"         ("-c" "-q")
      "unzipping"      "gzip"         ("-c" "-q" "-d")
```

6.2 Changing emacs for one person:

Thanks for this one go to Ulrik Dickow, ukd@kampsax.dk, Systems Programmer at Kampsax Technology:

To make it so you can use bzip2 automatically when you aren't the sysadmin, just add the following to your .emacs file.

```
;; Automatic (un)compression on loading/saving files (gzip(1) and similar)
;; We start it in the off state, so that bzip2(1) support can be added.
;; Code thrown together by Ulrik Dickow for ~/.emacs with Emacs 19.34.
;; Should work with many older and newer Emacsen too.  No warranty though.
;;
(if (fboundp 'auto-compression-mode) ; Emacs 19.30+
    (auto-compression-mode 0)
  (require 'jka-compr)
  (toggle-auto-compression 0))
;; Now add bzip2 support and turn auto compression back on.
(add-to-list 'jka-compr-compression-info-list
             ["\\.bz2\\(~\\|\\.~[0-9]+~\\)?\\'"
              "zipping"        "bzip2"         ()
              "unzipping"      "bzip2"         ("-d")
              nil t])
(toggle-auto-compression 1 t)
```

7. Using bzip2 with wu-ftpd

Thanks to Arnaud Launay for this bandwidth saver. The following should go in /etc/ftpconversions to do on-the-fly compressions and decompressions with bzip2. Make sure that the paths (like /bin/compress) are right.

```
:.Z:  :  :/bin/compress -d -c %s:T_REG|T_ASCII:O_UNCOMPRESS:UNCOMPRESS
:   : :.Z:/bin/compress -c %s:T_REG:O_COMPRESS:COMPRESS
:.gz: :  :/bin/gzip -cd %s:T_REG|T_ASCII:O_UNCOMPRESS:GUNZIP
:   : :.gz:/bin/gzip -9 -c %s:T_REG:O_COMPRESS:GZIP
:.bz2: :  :/bin/bzip2 -cd %s:T_REG|T_ASCII:O_UNCOMPRESS:BUNZIP2
:   : :.bz2:/bin/bzip2 -9 -c %s:T_REG:O_COMPRESS:BZIP2
:   : :.tar:/bin/tar -c -f - %s:T_REG|T_DIR:O_TAR:TAR
:   : :.tar.Z:/bin/tar -c -Z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS
:   : :.tar.gz:/bin/tar -c -z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP
:   : :.tar.bz2:/bin/tar -c -y -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+BZIP2
```

8. Using bzip2 with grep

The following utility, which I call bgrep, is a slight modification of the zgrep which comes with Linux. You can use it to grep through files without bunzip2'ing them first.

```
#!/bin/sh

# bgrep -- a wrapper around a grep program that decompresses files as needed
PATH="/usr/bin:$PATH"; export PATH

prog=`echo $0 | sed 's|.*/||'`
case "$prog" in
        *egrep) grep=${EGREP-egrep}     ;;
        *fgrep) grep=${FGREP-fgrep}     ;;
        *)      grep=${GREP-grep}       ;;
esac
pat=""
while test $# -ne 0; do
  case "$1" in
  -e | -f) opt="$opt $1"; shift; pat="$1"
           if test "$grep" = grep; then  # grep is buggy with -e on SVR4
             grep=egrep
           fi;;
  -*)      opt="$opt $1";;
   *)      if test -z "$pat"; then
             pat="$1"
           else
             break;
           fi;;
  esac
  shift
done

if test -z "$pat"; then
  echo "grep through bzip2 files"
  echo "usage: $prog [grep_options] pattern [files]"
  exit 1
fi

list=0
silent=0
op=`echo "$opt" | sed -e 's/ //g' -e 's/-//g'`
case "$op" in
  *l*) list=1
esac
```

```
case "$op" in
  *h*) silent=1
esac

if test $# -eq 0; then
  bzip2 -cd | $grep $opt "$pat"
  exit $?
fi

res=0
for i do
  if test $list -eq 1; then
    bzip2 -cdfq "$i" | $grep $opt "$pat" > /dev/null && echo $i
    r=$?
  elif test $# -eq 1 -o $silent -eq 1; then
    bzip2 -cd "$i" | $grep $opt "$pat"
    r=$?
  else
    bzip2 -cd "$i" | $grep $opt "$pat" | sed "s|^|${i}:|"
    r=$?
  fi
  test "$r" -ne 0 && res="$r"
done
exit $res
```

9. Using bzip2 with Netscape under the X.

tenthumbs@cybernex.net says:

> I also found a way to get Linux Netscape to use bzip2 for Content-Encoding just as it uses gzip. Add this to $HOME/.Xdefaults or $HOME/.Xresources
>
> I use the -s option because I would rather trade some decompressing speed for RAM usage. You can leave the option out if you want to.

```
Netscape*encodingFilters:      \
        x-compress :  : .Z     : uncompress -c  \n\
        compress   :  : .Z     : uncompress -c  \n\
        x-gzip     :  : .z,.gz : gzip -cdq      \n\
        gzip       :  : .z,.gz : gzip -cdq      \n\
        x-bzip2    :  : .bz2   : bzip2 -ds \n
```

10. Using bzip2 to recompress other compression formats

The following perl program takes files compressed in other formats (.tar.gz, .tgz. .tar.Z, and .Z for this iteration) and repacks them for better compression. The perl source has all kinds of neat documentation on what it does and how it does what it does. This latest version takes files as input on the command line. Without command line arguments, it tries to repack every file in the current working directory.

```
#!/usr/bin/perl -w

#######################################################
#                                                     #
# This program takes compressed and gzipped programs  #
# in the current directory and turns them into bzip2  #
# format.  It handles the .tgz extension in a         #
# reasonable way, producing a .tar.bz2 file.          #
#                                                     #
#######################################################
$counter = 0;
$saved_bytes = 0;
$totals_file = '/tmp/machine_bzip2_total';
$machine_bzip2_total = 0;

@raw = (defined @ARGV)?@ARGV:<*>;

foreach(@raw) {
    next if /^bzip/;
    next unless /\.(tgz|gz|Z)$/;
    push @files, $_;
}
$total = scalar(@files);

foreach (@files) {
    if (/tgz$/) {
        ($new=$_) =~ s/tgz$/tar.bz2/;
    } else {
        ($new=$_) =~ s/\.g?z$/.bz2/i;
    }
    $orig_size = (stat $_)[7];
    ++$counter;
    print "Repacking $_ ($counter/$total)...\n";
    if ((system "gzip -cd $_ |bzip2 >$new") == 0) {
        $new_size = (stat $new)[7];
        $factor = int(100*$new_size/$orig_size+.5);
        $saved_bytes += $orig_size-$new_size;
```

```
        print "$new is about $factor% of the size of $_.
:",($factor<100)?')':'(',"\n";
        unlink $_;
    } else {
        print "Arrgghh!  Something happened to $_: $!\n";
    }
}
print "You've "
    , ($saved_bytes>=0)?"saved ":"lost "
    , abs($saved_bytes)
    , " bytes of storage space :"
    , ($saved_bytes>=0)?")":("
    , "\n"
    ;

unless (-e '/tmp/machine_bzip2_total') {
    system ('echo "0" >/tmp/machine_bzip2_total');
    system ('chmod', '0666', '/tmp/machine_bzip2_total');
}


chomp($machine_bzip2_total = `cat $totals_file`);
open TOTAL, ">$totals_file"
     or die "Can't open system-wide total: $!";
$machine_bzip2_total += $saved_bytes;
print TOTAL $machine_bzip2_total;
close TOTAL;

print "That's a machine-wide total of ",`cat $totals_file`," bytes saved.\n";
```

Cable-Modem-mini-HOWTO

Vladimir Vuksan, vuksan@veus.hr

v4.3, 22 June 1999

This document attempts to answer basic questions on how to connect your Linux box to cable modem or cable Internet provider.

1. Introduction

- 1.1 New Versions of this Document
- 1.2 Feedback
- 1.3 Contributors
- 1.4 Standard Disclaimer
- 1.5 Copyright Information

2. Setting up your Ethernet Card

3. Regular Cable Modem ISPs

- 3.1 MediaOne Express
- 3.2 @Home
- 3.3 RoadRunner
- 3.4 Rogers Wave
- 3.5 Sunflower Cablevision
- 3.6 Jones Intercable
- 3.7 GTE Worldwind
- 3.8 SpeedChoice, Phoenix, Arizona
- 3.9 Cedar Falls Cybernet, Iowa
- 3.10 Telstra Bigpond Cable, Australia
- 3.11 Fibertel, Buenos Aires, Argentina

- 3.12 Videotron (Videon), Canada
- 3.13 Telekabel (Teleweb), Austria
- 3.14 Tebecai, Netherlands
- 3.15 A2000, Netherlands
- 3.16 Shaw Cable, Canada
- 3.17 Cogeco Cable, Canada
- 3.18 Optimum Online, New York and Connecticut
- 3.19 Singapore Cable Vision, Singapore
- 3.20 Cable Wanadoo, France
- 3.21 Prime Cable Expressnet, Las Vegas, NV
- 3.22 TVD, Belgium
- 3.23 Telenet Vlaanderen, Belgium
- 3.24 Total-Web, United States
- 3.25 CyberCable, Paris, France
- 3.26 StjärnTV, Stockholm Sweden

4. Hybrid Cable modem ISPs

- 4.1 Adelphia Powerlink
- 4.2 LinkExpress, Brasil
- 4.3 ExpressNet, Maryland
- 4.4 Charter Pipeline, Riverside, CA
- 4.5 Chambers Cable, Chico, CA / Fundy Cable, New Brunswick
- 4.6 Smyrna Cable, Atlanta, GA

1. Introduction

The main goal of this document is to get your system running with your cable modem, and cable internet provider. Unfortunately, many ISPs that provide cable modem services, give you Windows and Macintosh software only.

This document attempts to explain how to setup some cable modems and internet providers in Linux, the tricks to get them working correctly, and the traps not to fall down. It is hoped that this document will assist you, however we make no claims for the validity of the information contained within.

1.1 New Versions of this Document

New versions of this document will be periodically posted to *comp.os.linux.answers*. They will also be added to the various anonymous FTP sites who archive such information, including:

`ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/`

In addition, you should generally be able to find this document on the Linux Documentation Project page via:

`http://metalab.unc.edu/LDP/`

1.2 Feedback

Feedback is most certaintly welcome for this document. Without your submissions and input, this document wouldn't exist. So, please post your additions, comments and criticisms to `vuksan@veus.hr`.

1.3 Contributors

The following people have contributed to this mini-HOWTO.

- Dan Sullivan, dsulli@home.com
- Andrew Novick
- Michael Strates

1.4 Standard Disclaimer

No liability for the contents of this documents can be accepted. Use the concepts, examples and other content at your own risk. As this is a new edition of this document, there may be errors and inaccuracies, that may of course be damaging to your system. Proceed with caution, and although this is highly

unlikely, I don't take any responsibility for that.

Also bear in mind that this is NOT official information. Obtaining official information is usually an impossibility with many ISPs. Much content in this document are assumptions, which appear to work for people. Use the information at your own risk.

1.5 Copyright Information

This document is copyrighted (c)1998 Vladimir Vuksan and distributed under the following terms:

- Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.
- All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.
- If you have questions, please contact, the Linux HOWTO coordinator, at

linux-howto@metalab.unc.edu

2. Setting up your Ethernet Card

All of the setups below use ethernet cards (network cards) to connect you somehow to the Internet. That is why first we need to check if your ethernet card is working and most importantly can be used (read is supported) in Linux. There is a comprehensive Ethernet HOWTO at http://metalab.unc.edu/LDP/HOWTO/Ethernet-HOWTO.html if you would like to read it otherwise try this.

Boot into Linux, During boot up a message like this should appear...

```
eth0: 3c509 at 0x300 tag 1, 10baseT port, address 00 20 af ee 01 23,
IRQ 10.
3c509.c:1.07 6/15/95 becker@cesdis.gsfc.nasa.gov
```

If you missed it type `dmesg`.

If you see a message like that you are set and you can go to the next section. If you can't see a message like this there are two possible explanations, your ethernet card is PNP (plug-n-play) and you need to use tools such as isapnptools to get it recognized (I am not quite sure on this because I don't have a single PNP card so correct me if I am wrong). The other explanation is that you need to set up your card.

Most cards today come with DOS programs that are used to setup your card. For example to get my 3COM 3c509 to work all I needed to do is boot into DOS and use a utility to configure my card. There is usually a Auto Configure option. If that does not solve your problem try changing the IRQ for the card using the same utility. I find that usually IRQs 10,11 and 12 work well. If none of this solves your problem please read the Ethernet HOWTO referenced above or post to a newsgroup such as comp.os.linux.setup or comp.os.linux.networking.

3. Regular Cable Modem ISPs

If you think you have the card recognized you have to now look at the entry for your ISP. I have sorted the information according to a provider because setups are mostly ISP specific.

3.1 MediaOne Express

MediaOne Express is a Internet cable service provided by MediaOne. The hardware setup consists of a cable modem produced by LanCity or General Instruments which plugs into an ethernet card using a 10BaseT (UTP-45) cable. Assignment of IP addresses and other networking information is done using DHCP which stands for Dynamic Host Configuration Protocol. The only thing you need to do is read the DHCP mini-HOWTO and configure your system appropriately. There is no other necessary configuration. DHCP mini-HOWTO can be found at http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html

Information about MediaOne Service can be found at http://www.mediaone.com.

3.2 @Home

@Home uses a similar setup to MediaOne. However, there are a few fundamental differences, outlined by contributors. Since @Home spans different geographic locations you might get assigned different kind of equipment and have slightly different kind of setups.

Before you try anything, go to control panel, network, and properties for your network card. Write down all of the information. You will need it later.

TCI, the company that runs @Home issues a Etherlink III 3c509b NIC for all of their customers. What TCI does not tell you is that when they install your ethernet card, it is in PnP mode. Now in Slakware, if you uncomment the proper line for this card, everything will appear to be working fine. There will be no system problems, but the 'PC' light on your CyberSURFR modem will never turn on. If you are using Slakware, and are having this problem, reboot in DOS and skip the next paragraph.

In RedHat 5, your system will have some trouble autodetecting the card. If you try to pass the paramaters manually, the system will hang. This should be obvious that your card is not setup properly. Before wasting anymore time, reboot in DOS (This is a must because as of 12/25/97, there is no utility written for Linux to turn off PnP and turn on ISA.)

TCI does not give out a utility disk for your ethernet card, so you must download the utility from one of 3Com's sites. Here is a link to 3COM's page for driver download.

http://support.3com.com/infodeli/tools/nic/index.htm

Once you have downloaded your driver files you will need to run them and disable the PNP mode of

your network card.

What you've now done will make your ethernet card 'broken' in Windows 95. You'll need to go to Control Panel, Network, and remove the network card and the adapter. Reboot your computer, and again go back to control panel. Go to add/remove new hardware, and have it autodetect. It will automatically setup the correct i/o address for you. You will most likely need to reboot again. Now you should be in Windows 95, with the 'PC' light on your cable modem on. You will also notice that none of your internet applications seem to work, you can't ping, and you can't resolve DNS. You now must go back to control panels, network, and click on properties for your network card (not the adapter). Re-enter all the data you wrote down, and reboot.

With a little luck, your ethernet card should be working in Windows 95, and ready to rock in Linux.

If you live Hampton Roads, VA or Phoenix, AZ you should read a little note from Mark Solomon

With the @home service in Hampton Roads, VA, it is absolutly neccessary to run dhcpcd-0.70 (or higher) that supports the "-h" option to specify the hostname of your computer. Without this switch the @home dhcpcd server will not assign addresses.

More information on setting up dhcpcd-0.70 and @Home service with Intel Ether Express cards can be found at http://www.monmouth.com/~jay/Linux/

Notes for Baltimore, MD and Colleyville,TX:

In Baltimore subscribers are issued Intel Ether Express Pro 10 nics and a static IP number.

Notes for Milpitas, CA, USA (Joe Byrne byrne@well.com):

@Home service assignes 3Com 509b cards. IP addresses assigned statically.

Information about @Home Service can be found at http://www.home.com.

3.3 RoadRunner

RoadRunner is an Internet cable service provided by Excalibur Group (Time Warner). The hardware setup consists of a cable modem produced by Motorola and Toshiba which plugs into an ethernet card using a 10BaseT (UTP-45) cable. From what I can gather RR uses DHCP for IP assignment. In order to set up Linux to use DHCP you need to read the DHCP mini-HOWTO and configure your system appropriately. DHCP mini-HOWTO can be found at http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html

If this doesn't work out for you you should check out http://www.math.uakron.edu/RoadRunner/ for Akron, Ohio and http://people.qualcomm.com/karn/rr/index.html for San Diego, California. It might help solve your problem.

Another good site is http://www.vortech.net/rrlinux/.

Information about RoadRunner Service can be found at http://www.rr.com.

3.4 Rogers Wave

The hardware setup consists of a cable modem produced by LanCity which plugs into an ethernet card using a 10BaseT (UTP-45) cable.

When the cable modem is installed by Rogers Wave technicians you are assigned a static IP address. They should also provide you with information on your subnet mask, router (gateway) numbers and DNS numbers.

Other information about Rogers Wave Service can be found at http://www.rogerswave.ca.

3.5 Sunflower Cablevision

This information is provided by Andrew Novick:

"I recently saw your cable modem howto and I have an addition. Sunflower Cable is a company stricly in Lawrence KS, however we have a rather large Linux community because of the University of Kansas. On our local LUG mailing list, we are starting to get more and more questions on how to configure their linux machine for the cable modem. It is just regular static addressing, and the modem is made by Zenith."

To configure your Linux box make sure you get all the pertinent information from the Cablevision tech support or use these.

- IP address: Assigned by SunFlower Cablevision
- Subnet mask: 255.255.255.0
- Gateway (router) address: 24.124.11.254
- Hostname: Assigned by SunFlower Cablevision
- Domain name: lawrence.ks.us
- Primary DNS server (nameserver): 24.124.0.1
- Secondary DNS server (nameserver): 24.124.0.6

Have all those numbers written down before you proceed. To register, visit

http://www.sunflower.com

To register in the lawrence.ks.us domain contact Stephen Spencer at

gladiatr@artorius.sunflower.com

Under RedHat use Control Panel and Network Configuration to put in these numbers. Just say Add Interface, Device type=Ethernet, Device name=eth0 (this is zero not O in eth0) then fill out all the fields. Then click activate.

In Slackware type netconfig. When you are finished reboot and you should be up and running.

If this doesn't work make sure you do network card troubleshooting from the beginning of this document.

3.6 Jones Intercable

This information is provided by Bob Kimble:

Jones Internet Cable supplies a Hybrid cable modem that connects to your machine / network via 10BaseT ethernet. They provide you with a static IP address for your machine, and another static IP address for the modem. The modem acts as a gateway to their network. Your IP address and the modem IP address are on the same network and have the same network mask (in my case 255.255.255.0 -- 24 bits). They also provide two DNS IP addresses which you enter into your configuration. I just entered the numbers when I installed Red Hat Linux 5 and it worked like a champ from the beginning. Since then I have configured my Linux machine to enable IP masquerading and domain name services, and it now acts as a router for my entire private network. My other machines are connected via a private network using the addresses 192.168.0.x. They are running Windows 95, Windows NT (Intel and Alpha) and OS/2. My Linux machine has two IP addresses -- the one from the cable company and one from the 192.168.0.x private network. Everything works like a champ. All six machines can browse the web simultaneously. My kids are even able to connect to their favorite game site, "The Realm" from Sierra.

3.7 GTE Worldwind

This information is provided by Mike Hughes:

The information you provided for Rogers Wave works for GTE WorldWind cable modem services also. However, GTE's service is EXTREMELY slow, barely faster than ISDN. more information at

http://www.psilord.com

3.8 SpeedChoice, Phoenix, Arizona

According to Micah peenchee@asu.edu:

Just thought I'd let you know that in my area (Phoenix, AZ, usa) there is a company called speedchoice that provides cable modem service. The service uses a hybrid cable modem and the set up is almost identical to that of Jones intercable described in the howto. See Jones Cable. For any other issues mail Micah.

3.9 Cedar Falls Cybernet, Iowa

This information is provided thanks to Joe Breu breu@cfu.net

We are an ISP in Cedar Falls, Iowa that uses the Zenith HomeWorks Universal over our own Hybrid Fiber/Coax system. Our system uses no proprietary connection software and is straight TCP/IP connections. We do use DHCP, but will offer static IP addresses to customers requesting them because they cannot use DHCP.

3.10 Telstra Bigpond Cable, Australia

TOC

This information is provided by Geoff Conway gconway@vic.bigpond.net.au

Disclaimer:

The information contained herein is in no way the responsibility of Big Pond Cable, Telstra Multimedia, Telstra, IBM Global Services Australia or any related company. Any loss or consequential damage associated with the attempted installation of the Big Pond Cable Linux software is the PERSONAL responsibility of whomsoever follows these instructions.

Big Pond Cable does not currently support Linux - so do not report faults to them regarding any aspect of the Linux installation and/or operation.

Big Pond Cable Linux Installation Notes. V 1.1 (12/03/98)

Change Note 1.1 12/3/98 - Corrected spelling of dhcpcd & rrdhcpcd

Linux is not officially supported by Big Pond Cable. However, the TMM cable infrastructure does allow Linux to be used - with the appropriate login client. The following installation notes should allow you to install the Linux software without too many difficulties.

The full client distribution of rrclientd was obtained from:

ftp://ftp.vortech.net/pub/rrlinux/rrclientd-1.3.tar.gz

I am running RedHat version 5 of Linux - but the documentation indicates that it will work with other releases.

Note that there is also more information in the rrclientd release than is documented here.

Procedure:

- Copy the distribution file rrclientd-1.3.tar.gz to your Linux PC.
- Login as root for all of the following :-
- Expand the distribution (with gunzip) & then untar it.
- In directory rrclientd-1.3/bin you will find all the executables that you need: fetchmail (not used) kdestroy kinit rdate rrclientd rrdhcpcd rrpasswd
- Rename /sbin/dhcpcd to /sbin/dhcpcd.orig. Copy & rename rrdhcpcd to /sbin/dhcpcd
- Copy kdestroy,kinit,rdate,rrclientd,rrdhcpd,rrpasswd to /usr/local/bin
- Set up your ethernet card to use dhcp & restart the system. If the dhcp configuration is ok - an IP address will be assigned - with the details being places in /etc/dhcpc/hostinfo-eth0 & /etc/dhcpc/resolv.conf

  You may see a startup warning about not finding dhcp.conf - this doesn't seem to matter.

  ** At this point you should be able to ping proxy-server successfully.

  (If you can't then you won't be able to login later)

- Copy the krb5.ini file (from your Win95 \netmanag directory) to /etc/krb5.conf

  You will need to make the following changes (adding 2 new sections)

  The example shown is for the Melbourne krb5.ini file - Sydney users will need to substitute appropriately.

  This is also documented in the distribution's examples/ sub-directories.

```
[libdefaults]
        ticket_lifetime = 600
        default_realm = c3.telstra-mm.net.au

** new *1
       kdc_req_checksum_type = 2
       ap_req_checksum_type = 2
       safe_req_checksum_type = 3
       ccache_type = 2
** end of new *1

[realms]
c3.telstra-mm.net.au = {
        kdc = dce-server
        admin_server = dce-server
        default_domain = c3.telstra-mm.net.au
        }

wfh.c3.telstra-mm.net.au = {
        kdc = wfh.c3.telstra-mm.net.au
        admin_server = wfh.c3.telstra-mm.net.au
        default_domain = c3.telstra-mm.net.au
        }

wfh1.c3.telstra-mm.net.au = {
        kdc = wfh1.c3.telstra-mm.net.au
        admin_server = wfh1.c3.telstra-mm.net.au
        default_domain = c3.telstra-mm.net.au
        }

wfh2.c3.telstra-mm.net.au = {
        kdc = wfh2.c3.telstra-mm.net.au
        admin_server = wfh2.c3.telstra-mm.net.au
        default_domain = c3.telstra-mm.net.au
        }
```

TOC

```
wfh3.c3.telstra-mm.net.au = {
        kdc = wfh3.c3.telstra-mm.net.au
        admin_server = wfh3.c3.telstra-mm.net.au
        default_domain = c3.telstra-mm.net.au
        }

wfh4.c3.telstra-mm.net.au = {
        kdc = wfh4.c3.telstra-mm.net.au
        admin_server = wfh4.c3.telstra-mm.net.au
        default_domain = c3.telstra-mm.net.au
        }

wfh5.c3.telstra-mm.net.au = {
        kdc = wfh5.c3.telstra-mm.net.au
        admin_server = wfh5.c3.telstra-mm.net.au
        default_domain = c3.telstra-mm.net.au
        }

** new *2

[domain_realm]
    .c3.telstra-mm.net.au = c3.telstra-mm.net.au

** end *2
```

- Save your existing services file /etc/services to /etc/services.orig

  Copy the services file from examples/Nassau/services to /etc/services
- Create a new file /etc/rrpasswd - with the contents being your current BPC password
- Change the PATH environment variable as follows

  `PATH=$PATH:/usr/local/bin ; export PATH` (This is most likely not necessary)
- Login to BPC as follows

  ```
  rrclientd -u <username> /etc/rrpasswd dce-server
  ```

  so if you're username jsmith

  ```
  rrclientd -u jsmith /etc/rrpasswd dce-server
  ```

  If this is successful you'll see a message in /var/log/messages indicating it was ok.

  `tail /var/log/messages` will display the end of the file.
- Logout of BPC by entering `rrclientd -k`
- To make this automatic - you'll need to set the PATH & invoke rrclientd each time the system boots.

- The rrpasswd command has NOT been tried.
- That's it - setup your Linux Netscape proxies & go for it !!
- According to the documentation the client will logout & then login automatically at 3am. This has not been checked.

Problems

If you don't setup the PATH properly - then the login will fail with GSS_API complaints about non-supported checksum type.

If you don't make the needed changes to krb5.conf you'll also get a similar error.

If you are using your PC to dual/triple boot between W95/NT/Linux - the PC's CMOS clock will be altered by Linux - giving continual Kerberos errors if you try & login using NT. You need to login using Win95 first (which will fix the CMOS clock settings), then you can get in via NT.

Make sure you setup you PC time/timezone correctly in Linux. Note also that the /var/log/messages timestamps are a bit strange :) (possibly referenced to GMT)

3.11 Fibertel, Buenos Aires, Argentina

This information is provided by Pablo Godel

My name is Pablo Godel and want report that I'm using the cable service of Fibertel in Buenos Aires, Argentina and it works perfectly with Linux.

They gave me a static IP. The brand of the cablemodem is COM21 and the model is ComPort.

I connect it to the nic properly, configured in Linux and it worked perfectly.

More information about Fibertel can be found at http://www.fibertel.com.ar

3.12 Videotron (Videon), Canada

I don't have much information about Videotron except the fact that they use DHCP for the assignment of IP addresses and other networking information. Just read the DHCP mini-HOWTO (http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html) and configure your system appropriately. There is no other necessary configuration. Also important thing to note is that the Ethernet card that Videotron distributes is a NE2000 compatible PCI card for which you need to use

ne2k-pci driver

Additional information from Philip Gwyn (gwynp@artware.qc.ca):

They use Motorola CyberSURFR (sic) modems. This will give you an ethernet NIC when you sign up. They gave me a TMC NE2000 PCI clone, which Linux recognises as a "RealTek RTL-8029". It couldn't configure it propperly however, until I downloaded the manufacturer's driver disk and changed the media type to "auto-config".

While they can't guarantee to get it working with Linux, they will do a bit of hand holding so it works. Many of the tech-support have Linux-clues, as it were, even asking questions like "what kernel are you running?" "do you have the latest version of dhcpcd?". When their DHCP server broke, they even suggested I delete the cache files in /etc/dhcpcd to start over from zero, as it were. This worked.

Additional information from Mihai Petre (mihaip@videotron.ca):

Yes they are using DHCP for the tcp settings. They have also included dhcpcd on their ftp server at (ftp://ftp.videotron.ca/pub/linux/). You can also try posting your problems on news.powersurfr.com newsgroup videon.linux.

Their only limitation regarding making a server is a 1 Gig upload limit and a 6 Gig download limit.

Information about Videotron can be found at http://www.videotron.ca

3.13 Telekabel (Teleweb), Austria

According to Andreas Kostyrka:

- You may want to mention, that the Austrian Telekabel (Teleweb) ISP works with Linux. It seems quite similiar like MediaOne Express (3c509+dhcp, etc.)
- Second there are 3COM configuration utilities for Linux (But don't fool around with them on a busy system, 3c509 may lock the bus if touched the wrong way :( ):
  ftp://ftp.redhat.com/pub/contrib/hurricane/SRPMS/3c5x9utils-1.0-3.src.rpm
  ftp://ftp.redhat.com/pub/contrib/readmes/3c5x9utils-1.0-1.README

Information about Telekabel can be found at http://www.telekabel.at/

3.14 Tebecai, Netherlands

According to Frodo Looijaard:

Tebecai is yet another provider which uses a LANcity cable modem connected to a 10BaseT ethernet card. DHCP is used for configuration (see the DHCP mini-HOWTO at http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html). A step-by-step guide to install the cable modem under Linux can be found at http://huizen.dds.nl/~frodol/

(in dutch), but it is really very straightforward. You must only remember that your IP-address is not visible from the Internet (it is on the private 10.x.y.z subnet), so you can not setup a publicly available server.

Information about Tebecai can be found at http://www.tebenet.nl (in dutch only).

3.15 A2000, Netherlands

This information is provided by Johan List J.A.List@speed.A2000.nl:

Basically the way to go is the same as with Tebecai. A2000 provides cable internet access by means of a LANCity cable modem, connected to an 10BaseT ethernet card. This also works well for the "Vortex/Boomerang" cards by 3COM (I've got a 3COM Boomerang Fast Etherlink XL 10/100Mb TX Ethernet Adapter), providing you compile the Vortex/Boomerang drivers. (See the Linux Ethernet-HOWTO)

Setting up access can be done with DHCP (See the DHCP-Mini-HOWTO). A Dutch guide to setting up Internet access for A2000 is available at http://agvk.a2000.nl/LINUX/index.html.

Contrary to Tebecai, your IP-number *is* visible from the Internet, so take your precautions regarding security and safety when setting up a Linux machine using A2000 Internet access!!

3.16 Shaw Cable, Canada

According to Peng F. Mok pmok@shaw.wave.ca:

I recently signed up for a cable modem service from Shaw Cable here in Canada, which they have christened `Shaw Wave'. They also appear to have another service called `Shaw@Home' which they are using in some locations. Shaw has been upgrading their cable network for about a year and a half now, and now offer cable-modem service to a number of locations across Canada. Information about the `Shaw Wave' service can be found at http://www.shaw.wave.ca while information about the `Shaw@Home' service can be found at http://shaw.home.com General information about Shaw Cable can be found at http://www.shaw.ca

I thought I'd just drop you a line to inform you that your DHCP mini-HOWTO, (http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html) was very useful in helping me set up Linux to work with my cable-modem service, and that you might want to add `Shaw Wave' (and perhaps `Shaw@Home') to your Cable-Modem mini-HOWTO as another entry explaining how to set up Linux with the service. I don't have specific information on the `Shaw@Home' service yet, but from conversations I've had with Shaw technical support it seems that `Shaw@Home' is a partnership with `@Home' and involves the same features and setup procedure as that already described for `@Home' in the Cable-Modem mini-HOWTO document.

In both services Shaw Cable techs will come over and bring you two pieces of hardware -- a Motorola CyberSURFR cable-modem and either a 3Com EtherLink III 16-Bit ISA 3C509B-TPO NIC or an EtherLink XL PCI 3C900-TPO NIC. You have your choice of either an ISA or a PCI card depending on your needs, and these models only have the RJ-45 (UTP) connectors. The ISA cards come with PnP-mode enabled by default, so it may be necessary to boot into DOS, disable PnP operation, and configure the card to some base I/O address and IRQ setting which are available. Once this is done Linux should have no problems detecting the NIC at boot-time. I'm not sure about what needs to be done in the case of a PCI card since I don't currently have a PC which supports either PCI or PnP.

Note from another user:

Shaw now also issues SMC PCI ethernet cards. They give these out without boxes or manuals. I found out that these are the SMC EtherPower PCI RJ45 card (model 8432T). They use the DEC 21041 chip so the tulip ethernet driver is needed for it.

If you need to set up your POP3 mailboxes you can do that at following URL:

https://profile.home.net/Users/menu.htm

and you can login and setup the POP3 mailboxes that way.

Oh, and for the record, Shaw's technical support is horrendous. The best place to go for information is the athome.users-unix newsgroup (which is not even mentioned by any of the documentation) or please consult http://www.ee.ualberta.ca/~pmok/linux/

3.17 Cogeco Cable, Canada

This information is provided thanks to Terry O'Grady tograd y@cgocable.net

I have a Cable Modem through a company called Cogeco Cable located in various parts of Canada. They are part of the "Wave" system which includes Rogers Cable and Shaw Cable. The technical setup is different for each provider though. They supplied me with a D-Link ethernet card (the version of that has changed since then but I believe are still using D-Link) and a Zenith modem. The Ip setup is easy since they use DHCP so all I did was install the dhcp daemon (http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html) and that was that. If you like you can list my e-mail address tograd y@cgocable.net for anyone with questions.

3.18 Optimum Online, New York and Connecticut

This information is provided by Seth Greenfield (islesfan@nassau.cv.net):

Optimum Online uses DHCP, and rrclientd in Linux, by John Clark. Check out http://www.netaxis.com/~wharris/optimum/index.html for instructions on how to set up your service with Linux.

Tell people who had private ips who have public ips now to change their /etc/resolv.conf to look like this... domain nassau.cv.net (or optonline.net depenind on the users mood) nameserver 167.206.112.3 nameserver 167.206.112.4 Other Than that its the same setup procedure note: the rrclientd software will work if you tell it dce-server and you specify the domain as optonline.net

3.19 Singapore Cable Vision, Singapore

This information is provided by Jieyao (jieyao@letterbox.com):

SCV provides Motorola Cybersurfer modem connected to the network card via UTP cable. The IP setup

is easy since they use DHCP so all you need to do is install the dhcp daemon (http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html). If you can't make access the first time, turn the modem off then on again.

3.20 Cable Wanadoo, France

This information is provided by Jerome SAUTRET (Jerome.Sautret@wanadoo.fr):

I just read your Cable Modem HOWTO. I live in France, and I use Cable Wanadoo, the cable service of France Telecom, which is the main telecom operator in France. It is available in a few cities in France at the moment, like Angers and Metz. This service provides a dynamic IP address via DHCP. It uses a COM 21 modem plugged in a 10BaseT Ethernet card. The IP setup is easy just read the DHCP mini HOWTO (http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html).

3.21 Prime Cable Expressnet, Las Vegas, NV

This information is provided by jedi jedi@penguin.lcvm.com:

They use the Com21 which can either be connected directly to your 10baseT input or inserted into the downlink input on your router. Static IPs are available for $10 per month and the usual address assignment is through DHCP (http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html)

More information can be obtained from http://penguin.lvcm.com

3.22 TVD, Belgium

This information is provided by Pierre-Yves Keldermans pykeldermans@usa.net:

At home, my cable-TV company is "TVD", it is the first company to offer internet on the cable in Belgium.

Hardware : LanCity cable modem & 10-Base-T NIC (DLink ISA if you buy it from TVD)

Config : DHCP (http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html)

Prices :

- Cheap : for home use, real IP address but dynamic, DHCP expires every 10 min, 1 user only (theorically ...), no problem with firewall, the web server on my computer is even reachable from outside..., full speed FROM internet, small speed TO internet.
- No so cheap : for small office use, same as 1) but not limited to 1 user and more speed TO internet.
- Expensive : for WWW servers & ... , Static IP addresses and reserved bandwidth TO internet following price.

Speed :

>From TVD's mirror site : up to 250Kbytes/sec, very nice (and YES, they have some Linux mirrors redhat ... :-)) >From internet : variable but rather good if the remote server isn't overloaded.

For more info : http://www.tvd.be and http://www.tvd.net

3.23 Telenet Vlaanderen, Belgium

This information provided by Karel Goderis (karel.goderis@pandora.be):

Operator: Telenet Vlaanderen - Operational in flemish speaking (northern) part of Belgium

Hardware: Motorola CyberSURFR Wave Cable Modem using an RJ-45 Ethernet crosscable to a PC

Software: Standard config is Windoze + Modified Netscape for newbie installations, although Linux i386 support is there under the form of mirrored redhat.com software on the internal ftp servers

IP Setup: DHCP address assignment (http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html) , single address only. Outgoing : all ports accepted, except mandatory use of Netscape Proxy on port 8080, and thus port 80 blocked. Incoming : ports 0-1024 blocked, but re-allocation of ports on Linux works fine.

Pricing: one-off installation : BF 10000 (+-$ 250) + BF 3000 ($75) deposit for the cable modem (+$25 for Ethernet NIC if not present in pc) monthly rental : BF 1500 ($40)

Services: redhat.com netscape.com mirrors, quake I and II servers, proxy, mail relay and pop account (4 aliases) and the other usual stuff you need to survive on the net.

Caveats: Telenet states that you can download 300Mbytes/month, but this rule is not enforced unless there is a obvious abuse (i.e. you download 300Mbytes/*day* ;-))

More info at http://www.pandora.be/ or http://www.telenet.be/

3.24 Total-Web, United States

This information is provided by iota (iota@inaxx.net):

I work for Convergence.com, a cable internet company who provides the network monitoring, technical support, and advice to cable providers (who, in turn, provide the internet service to their customers). Most of our markets are through Cablevision, one of the more popular cable providers in the eastern US. The service name that they operate their cable internet service under is "Total-Web". Customers are provided with a static IP; simply set this up like you would any other ethernet device under Linux. The modems we use are LanCity LCP's and COM21 ComPort's, but these devices should be transparent to your computer.

Total-Web is available in limited areas, including: Miami Beach, FL; Gwinnett County and Roswell, GA; Cookeville, Lebanon, and Columbia, TN; and many other test markets.

3.25 CyberCable, Paris, France

TOC

This information is provided by David Monniaux:

CyberCable gives you an ethernet card if you need one. It is a cheap NE2000 clone, ISA-Pnp or PCI I think. They only know how to set up the stuff for Windows, but reports from other people say that they let you use their access with other systems, provided you do the software setup yourself.

They use DHCP (http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html). DHCPcd in RedHat worked immediately. In RedHat's netcfg, this means selecting "DHCP configuration". There are still some problems sometimes: DHCP outputs some error messages, but things still work afterwards generally (?). This seems related to rebooting; it looks like the cable system doesn't reallocate the line for one minute after reboot.

CyberCable's web site is at http://www.cybercable.fr.

3.26 StjärnTV, Stockholm Sweden

This information is provided by Fredrik Staxaeng (fstx@algorithmica.se) :

StjärnTV sells a Bay Networks Versalar Cable Modem 100, and they include a NetGear ethernet card in the initial fee. Just set your interface to DHCP (http://metalab.unc.edu/LDP/HOWTO/mini/DHCP.html). Info about their service is available at http://www.starport.se

---

4. Hybrid Cable modem ISPs

If you think you have the card recognized you have to now look at the entry for your ISP. I have sorted the information according to a provider because setups are mostly ISP specific.

This section is for people who are using so-called "hybrid" cable modems. Hybrid modems are modems that need two hookups (connections), one to the (TV) cable and one to the phone line. TV cable is used for downloading while phone line is used for uploading.

4.1 Adelphia Powerlink

Check out the following URL for directions on how to set up Linux with Powerlink
http://home.adelphia.net/~siglercm/sb1000.html

4.2 LinkExpress, Brasil

This information is provided by Rodrigo Severo (rodrigo@who.net):

First of all, let me tell you that here we have the MMD Cable Modem from General Instrument. We use SurfBoard 1000 ISA board for download and a regular telephone modem for upload. I would prefer to use an external board like the SurfBoard 1200 which is available only to corporate users, i.e., willing to pay US$ 200,00 instead of the regular US$ 30,00 so I found out this driver for the internal ISA board. For home users, Linkexpress (my ISP) just installs and supports the internal ISA board - Surfboard 1000. If you want to use it, you have to install Windows 95/98 on your computer and let the guy from Linkexpress install the equipment. After that, make your Linux installation as you like.

I started from the files I downloaded from http://home.adelphia.net/~siglercm/.

Here is the relevant data:

```
ISP: LinkExpress <http://www.linkexpress.com.br>
DNS: 200.252.88.20
Frequency: 351 MHz
Phone number: 321 3300
City: Brasilia
Province: Distrito Federal
Upload speed: regular 33.6K (just the download goes through the
cable modem)
```

During a download from a local tucows mirror I got 70KB 300KB per second. From distant sites I managed to get 30KB/s a few times.

More information about LinkExpress can be found at http://www.linkexpress.com.br

4.3 ExpressNet, Maryland

I recently accuired an expressnet cable modem for the maryland area. it is a com21 one way modem and I had a hell a time making it work so id like to share my knowledge in your faq maybe? well heres what I learned inorder for the PPP connection to authenticate the user must be running PAP which consists of editing the /etc/ppp/pap-secrets file accordingly

```
#/etc/ppp/pap-secrets
#this is the PAP secrets file for PPP
#the quotes are required on both
"username"      *       "password"
```

after that they must create a PPP-on script what ever name it must execute this command

```
exec /usr/sbin/pppd debug persist /dev/ttyS1 38400 0.0.0.0:0.0.0.0 connect "chat -v
TIMEOUT 3 ABORT 'BUSY' ABORT 'NOANSWER' '' ATH TIMEOUT 30 'OK' ATDTTELEPHONE CONNECT
''"
```

this must be done with no carrage returns either

next step is to modify the /etc/ppp/ip-up.local file if it does not exist it should be created it should read the following

```
#!/bin/bash
#/etc/ppp/ip-up.local
#this will set up the route to the ppp device as default everytime the modem
#authenticates dont include it if you do not want this option
route add default ppp0
```

then the user must configure their ethernet card on box I have a 3c905

I configured it the following way

```
ifconfig eth0 up
ifconfig eth0 10.0.0.1 broadcast 10.0.0.15 netmask 255.255.255.240
```

then I added some more routes to the kernel routing table as follows

```
route add -host 10.0.0.1 eth0
route add -net 10.0.0.0 eth0
```

all of these commands can be added into a script file as follows

```
#!/bin/bash
#This is a script file for establishing the cable modem IF device properties as
#well as the route properties
ifconfig eth0 up
ifconfig eth0 10.0.0.1 broadcast 10.0.0.15 netmask 255.255.255.240
route add -host 10.0.0.1 eth0
route add -net 10.0.0.0 eth0
```

thats all and the cable modem connection is setup fast as hell I might add.

Contributers: Chris (chris@wrm.grdn.net) & Mike Milbert (mike@milbert.com)

TOC

4.4 Charter Pipeline, Riverside, CA

This information is provided by Gabriel Peters (gpx1@earthlink.net):

(I have Charter Pipeline, Powered by Earthlink, Riverside, CA) The modem is a Com21 ComPORT 2000.. connected to the computer via 10 BaseT ethernet cable to a Linksys 10/100 ethernet card (Cable modem, ethernet cabling, and ethernet card supplied) The ethernet card driver that I had to compile into the kernel was for the DEC Tulip. auto-detected the card and set it up nicely.

This is the information I needed:

```
eth0 IP address - 10.0.0.1
DNS Servers - 207.217.126.81, 207.217.120.83
Subnet Mask - 255.255.255.240
Gateway: None
Your hostname should be CBL-(your username).hs.earthlink.net
```

Then you need to configure PPP to dial up your access number as normal.. What I had to do to get it to work was this: I typed 'ifconfig eth0 down' to shutdown the ethernet, ppp-go to dial in, once it reported my IP addresses, i typed 'ifconfig eth0 up' and voila, it worked perfectly.

Editor's comment:

Each time PPP link is brought up or down pppd executes scripts /etc/ppp/ip-up (link up) and /etc/ppp/ip-down (link down) so in order to have Ethernet network go up and down with PPP link simply add

ifconfig eth0 up

before 'exit 0' statement in /etc/ppp/ip-up and ifconfig eth0 down in ip-down.

4.5 Chambers Cable, Chico, CA / Fundy Cable, New Brunswick

This information is provided by Brian Moore (bem@cmc.net):

For those using Chambers Cable in Chico, CA, the product is the Scientific Atlanta data Xcellerator(tm) modem. Mike Cumings of Cal State University wrote a nifty driver for it, available at http://www.ecst.csuchico.edu/~mcumings/cablemodem/. This should also work for others using the same modem, such as Fundy Cable of New Brunswick.

4.6 Smyrna Cable, Atlanta, GA

This information is provided by Blake Sorensen (librarian@unseen.net):

I have Smyrna Connect, supported by Smyrna Cable in Atlanta, GA. They are currently (June, 1999) using half-duplex but are scheduled to have full-duplex within six months. The Cable Modem is a ComPort Com21. Here is the configuration stuff I needed to get my linux box running as my dialup.

My eth0 device is a 3com ISA card set to IP 10.0.0.1, Bcast 10.0.0.255, Mask 255.255.255.0. The machine is set to the

TOC

hostname Smyrna418.smyrnacable.net where Smyrna418 is my username. I don't think this is that important, but I have fiddled with it to see if it will still work once I change the hostname.

My modem is an external 56K X2 US Robotics on /dev/ttyS0.

I also have eth1 (a DEC tulip based pci card) set up as 192.168.0.1 as the gateway for the rest of my network to masquerade behind.

Smyrna Connect does not give you DNS info for the cablemodem since you are supposed to use the Windows PPP feature of using the default DNS for the server you dial in to. However, they do have one that works at 209.116.152.252.

I use a ppp connection script to dial in, but the guts of it is this:

```
/usr/sbin/pppd modem /dev/ttyS0 persist mru 1000 asyncmap 0 \
-detach crtscts user Smyrna??? defaultroute connect '/usr/sbin/chat \
ABORT BUSY ABORT ERROR "" ATZ OK ATDT7704365664 CONNECT' \
57600 0.0.0.0:0.0.0.0 &
```

I keep the persist in there since Smyrna Connect has a habit of dropping the connection every once in awhile, and this way it automatically dials back in. You will need to replace the Smyrna??? in the above command with your own username and put the line:

```
Smyrna??? Smyrna??? password
```

in the file /etc/ppp/pap-secrets.

The Clock Mini-HOWTO

Ron Bean, rbean@execpc.com

v2, July 1999

How to set and keep your computer's clock on time.

1. Introduction

2. Basic Timekeeping under Linux

3. Software

- 3.1 Clock(8) and Hwclock(8)
- 3.2 Adjtimex(8)
- 3.3 Xntpd and ntpd: the Network Time Protocol
- 3.4 Chrony

4. Radio Clocks

- 4.1 CHU and the "Gadget Box"
- 4.2 WWV and the "Most Accurate Clock"
- 4.3 GPS and the "Totally Accurate Clock"
- 4.4 Low-frequency Time Signals: DCF77, MSF(Rugby), WWVB

5. Detailed instructions for clock(8)

- 5.1 Checking your installation
- 5.2 Measuring your clock's drift rate
- 5.3 Example

1. Introduction

The Real-Time-Clock (RTC) chips used on PC motherboards (and even expensive workstations) are notoriously inaccurate, usually gaining or losing a consistent amount of time each day. Linux provides a simple way to correct for this in software, which can make the clock *very* accurate, even without an external time source. But most people don't know how to set it up, for several reasons:

- It's not mentioned in most of the general "how to set up linux" documentation, and it can't be set up automatically without an external time source, so the default is not to use it.
- If you type "`man clock`" you may get the man page for `clock(3)`, which is not what you want (try "`man 8 clock`" or "`man 8 hwclock`"-- some distributions search in numerical order if you don't give a section number, others search in the order specified in `/etc/man.config`).
- Most people don't seem to care what time it is anyway.
- Those few who do care often want to sync the system clock to an external time source, such as a network time server or radio clock. This makes the accuracy of the RTC irrelevant.

This mini-HOWTO describes the low-tech approach (which can be very accurate by itself), and provides pointers to several more sophisticated options. In most cases the documentation is well written, so I'm not going to repeat that information here, but I've included detailed instructions for the old `clock(8)` program at the end for anyone still running an older system.

Note

> You must be logged in as "**root**" to run any program that affects the RTC or the system time. Keep this in mind when setting up any of the programs described here.

Note

> If you run more than one OS on your machine, you should only let one of them set the CMOS clock, so they don't confuse each other. This includes the twice-a-year adjustment for Daylight Savings Time.

If you run a dual-boot system that spends a lot of time running Windows, you may want to check out some of the clock software available for that OS instead. Follow the links on the NTP website at http://www.eecis.udel.edu/~ntp/software.html. Many of the radio clocks mentioned here include software for Windows.

2. Basic Timekeeping under Linux

A Linux system actually has two clocks: One is the battery powered "Real Time Clock" (also known as the "RTC", "CMOS clock", or "Hardware clock") which keeps track of time when the system is turned off but is not used when the system is running. The other is the "system clock" (sometimes called the "kernel clock" or "software clock") which is a software counter based on the timer interrupt. It does not exist when the system is not running, so it has to be initialized from the RTC (or some other time source) at boot time. References to "the clock" in the `ntpd` documentation refer to the system clock, not the RTC.

The two clocks will drift at different rates, so they will gradually drift apart from each other, and also away from the "real" time. The simplest way to keep them on time is to measure their drift rates and apply correction factors in software. Since the RTC is only used when the system is not running, the correction factor is applied when the clock is read at boot time, using `clock(8)` or `hwclock(8)`. The system clock is corrected by adjusting the rate at which the system time is advanced with each timer interrupt, using `adjtimex(8)`.

A crude alternative to `adjtimex(8)` is to have `chron` run `clock(8)` or `hwclock(8)` periodically to sync the system time to the (corrected) RTC. This was recommended in the `clock(8)` man page, and it works if you do it often enough that you don't cause large "jumps" in the system time, but `adjtimex(8)` is a more elegant solution. Some applications may complain if the time jumps backwards.

The next step up in accuracy is to use a program like `ntpd` to read the time periodically from a network time server or radio clock, and continuously adjust the rate of the system clock so that the times always match. If you always have a network connection at boot time, you can ignore the RTC completely and use `ntpdate` (which comes with the `ntpd` package) to initialize the system clock from the time server-- either a local server on a LAN, or a remote server on the internet. But if you sometimes don't have a network connection, or if you need the time to be accurate during the boot sequence before the network is active, then you need to maintain the time in the RTC as well.

It might seem obvious that in this case you would want to sync the RTC to the (corrected) system clock. But this turns out to be a bad idea if the system is going to stay shut down longer than a few minutes, because it interferes with the programs that apply the correction factor to the RTC at boot time.

If the system runs 24/7 and is always rebooted immediately whenever it's shut down, then you can just set the RTC from the system clock right before you reboot. The RTC won't drift enough to make a difference in the time it takes to reboot, so you don't have to know its drift rate.

Of course the system may go down unexpectedly, so some versions of the kernel sync the RTC to the system clock every 11 minutes if the system clock has been adjusted by another program. The RTC won't drift enough in 11 minutes to make any difference, but if the system is down long enough for the RTC to drift significantly, then you have a problem: the programs that apply the drift correction to the RTC need to know **exactly** when it was last reset, and the kernel doesn't record that information anywhere.

Some unix "traditionalists" might wonder why anyone would run a linux system less than 24/7, but s of us run dual-boot systems with another OS running some of the time, or run Linux on laptops that have to be shut down to conserve battery power when they're not being used. Other people just don't like to leave machines running unattended for long periods of time (even though we've heard all the arguments in favor of it). So, the "every 11 minutes" feature becomes a bug.

This "feature/bug" appears to behave differently in different versions of the kernel (and possibly in different versions of `xntpd` and `ntpd` as well), so if you're running both `ntpd` and `hwclock` you may need to test your system to see what it actually does. If you can't keep the kernel from resetting the RTC, you might have to run without a correction factor.

The part of the kernel that controls this can be found in `/usr/src/linux-2.0.34/arch/i386/kernel/time.c` (where the version number in the path will be the version of the kernel you're running). If the variable `time_status` is set to `TIME_OK` then the kernel will write the system time to the RTC every 11 minutes, otherwise it leaves the RTC alone. Calls to `adjtimex(2)` (as used by `ntpd` and `timed`, for example) may turn this on. Calls to `settimeofday(2)` will set `time_status` to `TIME_UNSYNC`, which tells the kernel not to adjust the RTC. I have not found any real documentation on this.

If you don't need sub-second accuracy, `hwclock(8)` and `adjtimex(8)` may be all you need. It's easy to get enthused about radio clocks and such, but I ran the old `clock(8)` program for years with excellent results. On the other hand, if you have several machines on a LAN it can be handy to have them automatically sync their clocks to each other.

---

3. Software

3.1 Clock(8) and Hwclock(8)

All linux distributions install either the old `clock(8)` or the newer `hwclock(8)`, but without a correction factor. Some may also install `adjtimex(8)`, or they may include it on the CD as optional (or you can download it from the usual places). Some distributions also include a graphical clock setting program that runs in an X-window, but they're designed for interactive use and the system will still install `clock(8)` or `hwclock(8)` for use in the startup scripts.

`Clock(8)` requires you to calculate the correction factor by hand, but `hwclock(8)` calculates it automatically whenever you use it to reset the clock (using another program to set the time will interfere with the drift correction, so always use the same program if you're using a correction factor). If you have an older system that still uses `clock(8)` and you want to upgrade, you can find `hwclock(8)` in the "`util-linux`" package, version 2.7 or later. See the man page for more information.

Note

> The man page for `hwclock(8)` may be called "`clock`" for backward compatibility, so try both names. `Hwclock(8)` will respond to commands written for `clock(8)`, but the result may not be the same-- in particular, "`hwclock -a`" is not quite the same as "`clock -a`", so if you're upgrading to `hwclock` I'd suggest replacing all references to "`clock`" in your startup scripts to use `hwclock`'s native commands instead.

The startup scripts vary from one distribution to another, so you may have to search a bit to find where it sets the clock. Typical locations are `/etc/rc.local`, `/etc/rc.d/rc.sysinit`, `/etc/rc.d/boot`, or some similar place. The correction factor for the RTC is stored in `/etc/adjtime`.

When you're setting the clock to determine the drift rate, keep in mind that your local telephone time announcement may or may not be accurate. If you don't have a shortwave radio or GPS receiver, you can hear the audio feed from WWV by calling (303)499-7111 (this is a toll call to Boulder, Colorado). It will cut you off after three minutes, but that should be long enough to set the clock. USNO and Canada's CHU also have telephone time services, but I prefer WWV's because there is more time between the announcement and the "beep".

If you get bizarre results from the RTC you may have a hardware problem. Some RTC chips include a lithium battery that can run down, and some motherboards have an option for an external battery (be sure the jumper is set correctly). The same battery maintains the CMOS RAM, but the clock takes more power and is likely to fail first. Bizarre results from the system clock may mean there is a problem with interrupts.

TOC

3.2 Adjtimex(8)

`Adjtimex(8)` allows the user to adjust the kernel's time variables, and therefore change the speed of the system clock (you must be logged in as "**root**" to do this). It is cleverly designed to compare the system clock to the RTC using the same correction factor used by `clock(8)` or `hwclock(8)`, as stored in `/etc/adjtime`. So, once you've established the drift rate of the RTC, it's fairly simple to correct the system clock as well. Once you've got it running at the right speed, you can add a line to your startup scripts to set the correct kernel variables at boot time. Since `adjtimex(8)` was designed to work with `clock(8)` or `hwclock(8)`, it includes a work-around for the "every 11 minutes" bug.

After you've installed `adjtimex(8)` you can get more information on setting it up by typing "`man 8 adjtimex`" (there is also an `adjtimex(2)` man page, which is not what you want) and by reading the `README` file in `/usr/doc/adjtimex-1.3/README` (where the version number in the path will be the current version of `adjtimex(8)`).

3.3 Xntpd and ntpd: the Network Time Protocol

`Xntpd` (NTPv3) has been replaced by `ntpd` (NTPv4); the earlier version is no longer being maintained.

`Ntpd` is the standard program for synchronizing clocks across a network, and it comes with a list of public time servers you can connect to. It can be a little more complicated to set up than the other programs described here, but if you're interested in this kind of thing I highly recommend that you take a look at it anyway. The "home base" for information on `ntpd` is the NTP website at http://www.eecis.udel.edu/~ntp/ which also includes links to all kinds of interesting time-related stuff (including software for other OS's). Some linux distributions include `ntpd` on the CD.

A relatively new feature in `ntpd` is a "burst mode" which is designed for machines that have only intermittent dial-up access to the internet.

`Ntpd` includes drivers for quite a few radio clocks (although some appear to be better supported than others). Most radio clocks are designed for commercial use and cost thousands of dollars, but there are some cheaper alternatives (discussed in later sections). In the past most were WWV or WWVB receivers, but now most of them seem to be GPS receivers. NIST has a PDF file that lists manufacturers of radio clocks on their website at http://www.boulder.nist.gov/timefreq/links.htm (near the bottom of the page). The NTP website also includes many links to manufacturers of radio clocks at http://www.eecis.udel.edu/~ntp/hardware.htm and http://www.eecis.udel.edu/~mills/ntp/refclock.htm. Either list may or may not be up to date at any given time `:-)`. The list of drivers for `ntpd` is at http://www.eecis.udel.edu/~ntp/ntp_spool/html/refclock.htm.

`Ntpd` also includes drivers for several dial-up time services. These are all long-distance (toll) calls, so be sure to calculate the effect on your phone bill before using them.

3.4 Chrony

TOC

`Xntpd` was originally written for machines that have a full-time connection to a network time server or radio clock. In theory it can also be used with machines that are only connected intermittently, but Richard Curnow couldn't get it to work the way he wanted it to, so he wrote "`chrony`" as an alternative for those of us who only have network access when we're dialed in to an ISP (this is the same problem that `ntpd`'s new "burst mode" was designed to solve). The current version of `chrony` includes drift correction for the RTC, for machines that are turned off for long periods of time.

You can get more information from Richard Curnow's website at http://www.rrbcurnow.freenet.co.uk/chrony/index.html. He distributes the program as source code only, but Debian has been including a binary in their "unstable" collection. The source file is also available at the usual linux archive sites.

---

4. Radio Clocks

4.1 CHU and the "Gadget Box"

CHU, the Canadian shortwave time station near Ottawa, is similar to WWV in the US but with one important difference: in addition to announcing the time in both French and English, it also broadcasts the current time once per minute using the old "Bell 103" (300 baud) modem tones. These tones are very easy to decode, and Bill Rossi realised that you don't even need a modem-- all you need is a shortwave radio and a sound card. If you're able to receive the signal from CHU, this may be the cheapest radio clock available. Shortwave reception varies throughout the day, but Bill claims that by changing frequencies twice a day (morning and evening) he gets almost 24-hour coverage. CHU broadcasts on 3.33, 7.335, and 14.670 MHz.

For more information see Bill Rossi's website at http://www.rossi.com/chu/. The source file is also available at the usual linux archive sites. For information on CHU's time services see http://www.nrc.ca/inms/time/ctse.html.

The NTP website has plans for a "gadget box" that decodes the CHU time broadcast using an inexpensive 300 baud modem chip and any shortwave radio, at http://www.eecis.udel.edu/~ntp/ntp_spool/html/gadget.htm. The plans include a Postscript image of a 2-sided custom printed circuit board, but you have to make the board yourself (or find someone who can make it for you).

`Ntpd` includes a driver (type 7) for CHU receivers, which works either with modems like the "`gadget box`", or by feeding the audio directly into the mic input of a Sun SPARCstation (or any other machine with "compatible audio drivers").

4.2 WWV and the "Most Accurate Clock"

You may have heard about Heathkit's "Most Accurate Clock", which received and decoded the time signal from WWV and had an optional serial port for connecting to a computer. Heathkit stopped selling kits a long time ago, but they continued to sell the factory-built version of the clock until 1995, when it was also discontinued. For Heathkit nostalgia (not including the clock) see http://www.heathkit-museum.com. The Heathkit company still exists, selling educational materials. See http://www.heathkit.com.

According to Dave Mills, Heathkit's patent on the "Most Accurate Clock" is due to expire soon, so maybe someone out there would like to clone it as a single-chip IC.

The NTP website has a DSP program (and a PDF file describing it) at http://www.eecis.udel.edu/~mills/resource.htm that decodes the WWV time signal using a shortwave radio and the TAPR/AMSAT DSP-93, a DSP kit which is no longer available. It was based on the Texas

TOC

Instruments TMS320C25 DSP chip. The TAPR website at http://www.tapr.org includes a lot of information on homebrew DSP programming.

`Ntpd` includes a driver (type 6) for the IRIG-B and IRIG-E time codes, using `/dev/audio` on a Sun SPARCstation, with a note that it is "likely portable to other systems". WWV uses the IRIG-H time code.

WWV is run by NIST, which has a website at http://www.boulder.nist.gov/timefreq/index.html. This site includes the text of "Special Publication 432", which describes their time and frequency services, at http://www.boulder.nist.gov/timefreq/pubs/sp432/sp432.htm. WWV broadcasts on 2.5, 5, 10, 15, and 20 Mhz.

4.3 GPS and the "Totally Accurate Clock"

GPS signals include the correct time, and some GPS receivers have serial ports. `Ntpd` includes drivers for several GPS receivers. The 1PPS feature ("One Pulse Per Second", required for high accuracy) usually requires a separate interface to connect it to the computer.

TAPR (Tuscon Amateur Packet Radio) makes a kit for an interface called "TAC-2" (for "Totally Accurate Clock") that plugs into a serial port and works with any GPS receiver that can provide a 1PPS output-- including some "bare board" models that can be mounted directly to the circuit board. For more information see their website at http://www.tapr.org. The price (as of June 1999) is around $140, not including the GPS receiver. The kit does not include any enclosure or mounting hardware.

The CHU "gadget box" (described in another section) can also be used as an interface for the 1PPS signal. The NTP website has a discussion of this at http://www.eecis.udel.edu/~ntp/ntp_spool/html/pps.htm.

4.4 Low-frequency Time Signals: DCF77, MSF(Rugby), WWVB

These low-frequency stations broadcast a time code by simply switching the carrier on and off. Each station uses its own coding scheme, and summaries are available on the NTP website at http://www.eecis.udel.edu/~mills/ntp/index.htm (near the bottom of the page). DCF77 in Germany broadcasts on 77.5kHz. MSF in England (also called "Rugby", which apparently refers to its location) and WWVB in Colorado both broadcast on 60 kHz.

Inexpensive receivers that can plug into a serial port are reported to be available in Europe. `Ntpd` includes drivers for a couple of MSF receivers.

A number of companies in the US sell relatively inexpensive clocks that have built-in WWVB receivers (including several analog wall clocks), but I'm only aware of two that can be connected to a computer:

The Ultralink Model 320 sells for about $120 (as of June 1999) and has a serial interface and a straightforward ASCII command set, so it shouldn't be too hard to program. It draws 1mA from the serial port for power. The antenna can be up to 100 feet away from the computer, and the unit contains its own

clock to maintain the time if it loses the signal. They also sell a "bare board" version for about $80 th designed to work with the "BASIC Stamp" series of microcontrollers. See http://www.ulio.com/timepr.html.

Arcron Technology sells a desk clock with an optional serial port for about $130, including software for Windows. See http://www.arctime.com

Reception of WWVB varies, but there are plans to increase its broadcast power, in several stages. You can follow its progress on NIST's website at http://www.boulder.nist.gov/timefreq/wwvstatus.html.

---

5. Detailed instructions for clock(8)

This section is retained from the earlier version of this mini-HOWTO, and is provided for anyone still using the old `clock(8)` program. Everything you need to know is in the man page, but the following discussion will walk you through the process.

Note

> You must be **root** to run "`clock`", or any other program that affects either the system time or the CMOS clock.

5.1 Checking your installation

Check your system startup files for a command like "`clock -a`" or "`clock -ua`". Depending on which distribution you're using, it might be in `/etc/rc.local`, or `/etc/rc.d/rc.sysinit`, or some similar place.

If it says "`clock -s`" or "`clock -us`", change the "`s`" to an "`a`", and then check to see if you have the file `/etc/adjtime`, which contains a single line that looks something like this:

```
0.000000 842214901 0.000000
```

These numbers are the correction factor (in seconds per day), the time the clock was last corrected (in seconds since Jan 1, 1970), and the partial second that was rounded off last time. If you don't have this file, login as **root** and create it, with a single line that looks like this (all zeros):

```
0.0 0 0.0
```

Then run "`clock -a`" or "`clock -ua`" manually from the shell to update the 2nd number (use the "`u`" if your clock is set to Universal instead of local time).

5.2 Measuring your clock's drift rate

First, you need to know what time it is `:-)`. Your local time of day number may or may not be accurate. My favorite method is to call WWV's voice announcment at (303)499-7111 (this is a toll call). If you have access to a network time server, you can use the `ntpdate` program from the `xntpd` package (use the `-b` flag to keep the kernel from messing with the CMOS clock). Otherwise use "`date -s hh:mm:ss`" to set the kernel time by hand, and then "`clock -w`" to set the CMOS clock from the kernel clock. You'll need to remember when you last set the clock, so write down the date someplace where you won't lose it. If you used `ntpdate`, do "`date +%s`" and write down the number of seconds since Jan 1,1970.

TOC

Then come back some days or weeks later and see how far the clock has drifted. If you're setting the clock by hand, I'd recommend waiting at least two weeks, and only calculate the drift rate to the nearest .1 sec/day. After several months you could get to the nearest .01 sec/day (some people claim more accuracy than that, but I'm being conservative here). If you use `ntpdate` you shouldn't have to wait that long, but in any case you can always fine-tune it later.

You can have cron run "`clock -a`" at regular intervals to keep the system time in line with the (corrected) CMOS time. This command will also be run from your startup file every time you boot the system, so if you do that often (as some of us do), that may be enough for your purposes.

Note that certain programs may complain if the system time jumps by more than one second at a time, or if it jumps backwards. If you have this problem, you can use `xntpd` or `ntpdate` to correct the time more gradually.

5.3 Example

To set time

Login as **root**. Dial (303)499-7111 (voice), listen for time announcement. Then type:

```
date -s hh:mm:ss
```

but don't press enter until you hear the beep. (You could use "`ntpdate`" here instead of "`date`", and skip the phone call) This sets the "kernel time". Then type:

```
clock -w
```

This sets the CMOS time to match the kernel time. Then type:

```
date +%j
```

(or "`date +%s`" if you used "`ntpdate`" instead of "`date`" above) and write down the number it gives you for next time.

To reset time and check drift rate

Find the date you wrote down last time. Login as **root** Then type:

```
clock -a
```

This sets the kernel time to match the current CMOS time. Dial (303)499-7111 (voice), listen for announcement. Then type:

```
date
```

and press enter when you hear the beep, but while you're waiting, write down the time they announce, and don't hang up yet. This tells you what time your machine thought it was, when it should have been exactly on the minute. Now type in

```
date hh:mm:00
```

using the minute *after* the one that was just announced, and press enter when you hear the beep again (now you can hang up). For `hh` use the local hour. This sets the "kernel time". Then type:

```
clock -w
```

which writes the new (correct) time to the CMOS clock. Now type:

```
date +%j
```

(or "`date +%s`" if that's what you used before)

You now have three numbers (two dates and a time) that will allow you to calculate the drift rate.

Calculating the correction factor

When you ran "`date`" on the minute, was your machine slow or fast? If it was fast, you'll have to subtract some number of seconds, so write it down as a negative number. If it was slow, you have to add some seconds, so write it down as positive.

Now subtract the two dates. If you used "`date +%j`", the numbers represent the day-of-year (1-365, or 1-366 in leap years). If you've passed Jan 1 since you last set the clock you'll have to add 365 (or 366) to the 2nd number. If you used "`date +%s`" then your number is in seconds, and you'll have to divide it by 86400 to get days.

If you already had a correction factor in `/etc/adjtime`, you'll have to account for the number of seconds you've already corrected. If you've overcorrected, this number will have the opposite sign of the one you just measured; if you've undercorrected it will have the same sign. Multiply the old correction factor by the number of days, and then add the new number of seconds (signed addition-- if the two numbers have the same sign, you'll get a larger number, if they have opposite signs, you'll get a smaller number).

Then divide the total number of seconds by the number of days to get the new correction factor, and put it in `/etc/adjtime` in place of the old one. Write down the new date (in seconds or days) for next time.

Here's what my `/etc/adjtime` looks like:

```
-9.600000 845082716 -0.250655
```

(note 9.6 seconds per day is nearly five minutes per month!)

COFFEE-HOWTO

Georgatos Photis, < gef@ceid.upatras.gr>

v0.5, 15 January 1998

One of the most bothering remarks on software, I have ever heard, is weather this or that thing can make coffee. So, Linux DOES make coffee. And it tastes good, instead!

For a long time, humanity was wondering how could a computer make coffee... People need coffee to get awake and stay asleep in front of the computer for a long time. Everyone knows that coding is better at night...

The main problem is how to control the coffee machine with the computer, so that it will be controlled by software. This generally means an ON/OFF switch implemented as a circuit which controls the coffee-machine's power supply.

1. Menu

- 1.1 French
- 1.2 Nescaffe
- 1.3 frappe
- 1.4 freddo
- 1.5 cappuccino (To be added)
- 1.6 espresso

2. Electronic circuit

- 2.1 Driving voltage 0-5V from the computer
- 2.2 Controlling with a Relay
- 2.3 Controlling with TRIAC #1
- 2.4 Controlling with TRIAC #2

TOC

3. Software

- 3.1 Software
- 3.2 Device driver
- 3.3 Connecting with the Internet

4. Overdose symptoms

5. Expansions

6. References

7. etc

- 7.1 preface
- 7.2 Authorship and maintenance
- 7.3 Copyrights
- 7.4 Credits

---

1. Menu

1.1 French

Popular coffe among programmers because doesn't need a lot of care, like all commercial software. Its exciting taste has inspired thousands of programmers in writing incredible software, written in the very first ours of a day. Windows for example are written at 5:00 o'clock in the morning, favouring to this coffee! Result is guaranteed.

1.2 Nescaffe

Nescaffe is a rather strong coffee, made by pouring hot water in a mixture of coffee, sugar and some water. You usually take 1 spoon of coffee and 1 spoon of sugar with just a bit of water, to mix it. In the meantime you should have the water boiling. As soon as the water is hot enough, you mix them all together and preferably add milk. Although you can use something simpler than a coffee-machine to boil the water, I have seen this scheme a lot of times...

1.3 frappe

Popular variation of the above mentioned coffee. Actually, it doesn't need any coffee-machine, rather a refridgerator to get cold water and ice-cubes.

1.4 freddo

This is a difficult one, read coffee-faq (see references)

1.5 cappuccino (To be added)

1.6 espresso (To be added)

2. Electronic circuit

A general diagram is like this:

```
--------- 0-5V  --------- ~220V  ----------------
|  PC   |===>===|Circuit|========|Coffee-Machine|
---------       ---------        ----------------
```

The concept is that we take a controling voltage from the computer, which drives an electrically isolated circuit with Relay or Triac.

You must choose a Relay circuit, if you have a big coffee-machine (greater than 200W or so), otherwise you can use a triac-based one.

All circuits presented are at least once tested, but it's YOURS RESPONSIBILITY the results. If you have no experience with electronics you should NOT try with these, otherwise you may get a bad one...

You should be very careful while experimenting with 220V, and there is no obsolence in using an appropriate fuse.

2.1 Driving voltage 0-5V from the computer

Here is a simple example to get a voltage 0-5V from the parallel port of the computer.

```
      Back View          -----    Pin 10 - ACK
      Male DB-25         |   |    Pin  9 - D7
      Connector          |   |                          Pin 2 - D0
                         v   v                          v   Pin 1 -
~Strobe
       ____________________________________________________________
      /                                                            \
      \     13  12  11  10   9   8   7   6   5   4   3   2   1     /
       \                                                           /
        \     25  24  23  22  21  20  19  18  17  16  15  14     /
         \__________________________________________________________/
```

Pin 1 is Strobe (inverse logic)

Pins 2-9 is DATA BUS's signals, exactly what was written to the parallel port's latches with an OUTB command.

Pin 10 is the acknowledge signal (ACK), controlled by you, so that you can produce an interrupt to the CPU.

Pins 18-25 are short-circuited and this is the ground (GND).

In detail:

```
<= in   DB25    Cent    Name of         Reg
=> out  pin     pin     Signal          Bit     Function Notes
------  ----    ----    --------        ---     ------------------------------
=>       1       1      -Strobe         C0-     Set Low pulse >0.5 us to send
=>       2       2      Data 0          D0      Set to least significant data
=>       3       3      Data 1          D1      ...
```

TOC

```
=>       4       4      Data 2          D2      ...
=>       5       5      Data 3          D3      ...
=>       6       6      Data 4          D4      ...
=>       7       7      Data 5          D5      ...
=>       8       8      Data 6          D6      ...
=>       9       9      Data 7          D7      Set to most significant data
<=      10      10      -Ack            S6+ IRQ Low Pulse ~ 5 uS, after accept
<=      11      11      +Busy           S7-     High for Busy/Offline/Error
<=      12      12      +PaperEnd       S5+     High for out of paper
<=      13      13      +SelectIn       S4+     High for printer selected
=>      14      14      -AutoFd         C1-     Set Low to autofeed one line
<=      15      32      -Error          S3+     Low for Error/Offline/PaperEnd
=>      16      31      -Init           C2+     Set Low pulse > 50uS to init
=>      17      36      -Select         C3-     Set Low to select printer
==      18-25   19-30,  Ground
```

2.2 Controlling with a Relay

The simplest circuit that somebody can build is:

```
                                Vcc
                                 |
                                 +------+
                                 |    __|__
                               Relay   /^\  Diode 1N4002
                                Coil  /---\
                                 |      |
                                 +------+
                                 |
                              | /
                  4.7K    B |/  C
parallel port >-\/\/\/\/---|        NPN Transistor: BC547A or 2N2222A
data pi                     |\  E
                              | V
                                 |
parallel port >--------------+
ground pin                       |
                              Ground
```

Connect Vcc with the same voltage as the relay type (usually 5 or 12V). Obviously, the relay's specifications should be reasonable for your coffee-machine.

Barmen, usually, tend to put the relay AFTER the transistor, at the emitter (E) pin instead of the collector (C) pin. This is a bad practice because it biases the transistor badly, and may result in bad coffee :-). Diode 1N4002 is useful to protect the transistor from the relay's currents. If you don't use it the transistor will become darker and smelling...

2.3 Controlling with TRIAC #1

If you only want a simple circuit, you can use Motorola's triac driver MOC301[012], together with a general purpose TRIAC like SC141D. This method has the advantage that you don't need an extra power supply.

For non-inductive loads this is the circuitry:

TOC

```
       270     1 +-------+ 6    180
  +5v -VAVAVA-----+       +----VAVAVA-----+-------------- Line Hot
                2 |  MOC  |               |
  TTL in ---------+ 3012  +nc            VA  SC141D
                  |       | 4           / |
                nc+       +------------/  |
                  +-------+               +----\/\/\/---- Line Neutral
                                                LOAD
```

If you are going to work with 220V, prefer a 3021. Inductive loads should be used in conjuction with bypass capacitors, see *Motorola Application Note AN-780*. Coffee-machines are mainly resistive loads and not inductive (like a motor), but who knows what's yours...

2.4 Controlling with TRIAC #2

```
+5VDC
|    180                        180            2.2k
+---/\/\/\----+-----+   +----/\/\/-+--/\/\/\---+-------> 120V
              |    1|   |6         |           |         Hot
              |    +=====+         |           | MT1
              |    | MC  | TRIAC   |          +-+
              |    | 3032| Driver  |        G | | TRIAC
              |    +=====+         |         /| |
              \    2|   |4         |        / +-+
       2N3904  |----+   |          |        |  | MT2
              /     |   +--------- | -------+  |
             V      \              |        |  |
             |      /              |        \  |
             |      \ 43    .01u  ---   10k /  |
             |      /       500V  ---       \  |
             |      |              |        /  |
             +------+              |        |  |            Neutral
             |                     +--------+--+---o    o--> 120V
             /                                        load
 >-/\/\--|  2N3904
             \
              V
              |
             ---
            ///
You should change resistors accordingly for 220V.
```

Circuit description:

The MC3032 is an optoisolator TRIAC driver. The 180-ohm resistor sets the current for the LED emitter in the optoisolator. Change the value of this resistor - if necessary - to get reasonable current (e.g., 15 mA).

Note that you cannot test this circuit without a load. The TRIAC will not switch unless connected to an AC voltage source, so you can't test it for simple switching w/o applying AC and a load. Note the 500V rating on the .01 cap.

---

3. Software

3.1 Software

You will have to build an executable that will perform like this:

- Get permissions to use I/O address space, by calling kernel, with the command **ioperm**: eg *ioperm(BASE, range ,1);*
- Perform an out request instruction, to set the 0-5V voltage to the parallel port, eg *outb(1, BASE);*
- Wait for enough time so that coffee gets ready. It would be nice if that time is read by looking at the command line.
- Then it will turn off the coffee-machine: outb(0 , BASE);
- Before ending it should give back the paraller port with a *ioperm(BASE, range, 0);*

Change BASE = 0x3bc for /dev/lp0, 0x378 for /dev/lp1, and 0x278 for /dev/lp2, range=8.

It would be nice if you had that program setuid, so that everybody can drink coffee!

3.2 Device driver

Just read kernel hacker's guide, implement a device driver (it could even be user space i think). Please, compile it as a module, so that we won't need a kernel compile in every update. Then write:

```
echo cappuccino >/dev/coffee
```

And you will have a hot cup of coffee in 1 minute. Remember to give the right permission to /dev/coffee, depending on whether you want only root making coffee or not.

The advantage of this method is that it supports feedback from the coffee-machine by using the ACK of parallel port, so that smart coffee-machines could produce an interrupt.

Do it as homework.

3.3 Connecting with the Internet

If you have implemented the C programme (see above), you just have to write down a simple CGI script to turn ON and OFF the coffee-machine. You should write some nice webpages, explaining how to make coffee, and put them at an **apache** web server...

4. Overdose symptoms

- excitement
- nervousness
- insomnia
- tachycardia or cardiac arhythmia
- gastrointestinal disturbance
- restlessness

5. Expansions

These are our ideas:

- All hardware and software described here, can be expanded so that it will support toast, beaf, applepies, etc.
- Cluster with 8 coffee-machines. This will let you have coffee even when the one gets off. Of course there will be a perfomance hit.
- Parallel vector coffee-machine will be a future release.
- If you want the maximum automation you'll need more circuits and censors, so that you can control water flow, temperature, coffee quantity etc.
- In the near future we will implement SNMP features.
- Serial coffee-machine at 115Kbps.

6. References

- http://daisy.uwaterloo.ca/~alopez-o/caffaq.html This is Internet's **Coffee-FAQ**
- http://lonestar.texas.net/~andrew/f_pc_.htm A lot of circuits in ASCII. Some of them are for parallel port.
- http://shell.rmi.net/~hisys/parport.html Whatever you wanted to learn about a parallel port and didn't dare to ask.
- http://sunsite.unc.edu/LDP/HOWTO/mini/IO-Port-Programming Programming of I/O ports under popular operating system Linux.
- http://www.redhat.com:8080/HyperNews/get/khg.html How to write your own device drivers. Come on, do it!
- http://www.hut.fi/Misc/Electronics/circuits/parallel_output.html Tomi Engdahl's web page is a *must see* for everyone who likes electronics.
- http://www.yahoo.com/Computers_and_Internet/Internet/Entertainment/Interesting_Devices_Connected_to_the_Net/Coffee_Machines/ Coffee-machines on-line. Unfortunatelly, there are no tests.
- http://www.cs.su.oz.au/~bob/coffee.html This coffee-machine offers only cappuccino. It should be upgraded!
- http://einstein.et.tudelft.nl/~janssen/ Hot coffee from Netherlands.
- http://circe.chinalake.navy.mil/cgi-bin/spion/snapit.cgi
- http://www.cl.cam.ac.uk/coffee/coffee.html
- http://www.menet.umn.edu/coffeecam/

7. etc

7.1 preface

This document was initially written in hellenic (greek), originating from a small debate in the linux-greek-users list , whether linux can make coffee or not. It has been an article in our beautiful online magazine on Linux called "magaz": http://www.linux.gr/magaz. Remember, that magaz is greek.

7.2 Authorship and maintenance

My name is Georgatos Photis, still-yet-foralongmaybe student at Computer Engineering and Informatics department in University of Patras. Usually, I am pretty busy with the greek documentation and webpages maintainance, but I can accept submissions to this HOWTO, as long as you don't be anxious on the changes.

7.3 Copyrights

The casual copyright with everything you get with linux... To find it, you'll have to read all of them and count the most common. Otherwise, no, you can't copy it.

7.4 Credits

- Christofer Colombus. His real name is Cristobal Colon. He was the 1st one who brought coffee from America, which was for first discovered (among Europeans) by him in 1492.
- Kostas Lialiambis is the one who dared saying that he can't make coffee with his Linux box!
- Panagiotis Vrionis, Yannakopoulos Haralambos, for giving me interesting and humoristing notes.

mini-HOWTO install Cyrus IMAP Server
Kevin Mitchell (kevin@iserv.net)
v0.9 21.01.98

This documentation is to offer some help in installing the Cyrus IMAP Server on a Linux machine.

1. Introduction

I would like to thank Bob Anderson (boba@iserv.net) and Jorge Paramo (jorge@iserv.net) for helping me in my various Linux adventures.

What is IMAP and why should I use it?

IMAP (Internet Message Access Protocol) is a method of accessing electronic mail or bulletin board messages that are kept on a mail server. IMAP is seen by many as a replacement for POP (Post Office Protocol) mail. IMAP allows users to access their e-mail from different computers without having to download it. This method of mail access is more secure and offers many benefits to the end user.

A more indepth answer can be found at:
http://www.imap.org/whatisIMAP.html

A comparision of IMAP & POP can be found here:
http://www.imap.org/imap.vs.pop.brief.html

Why use the Cyrus server?

Cyrus is designed to be run on a server where normal users are not permitted to log in. Cyrus also seems to be among the two most popular IMAP servers for Unix. The University of Washington IMAP Server is the other.
ftp://ftp.cac.washington.edu/imap/imap.tar.Z

2. My System Details

I have successfully installed Cyrus on both 486DX66 and Pentium machines running Linux Kernel 2.1.79 and 2.0.33 respectively. These machines were originally based upon Slackware 3.4.

3. Tcl Installation

Make sure that you have Tcl installed on your machine before attempting to install Cyrus - otherwise you will not be able to use the Cyrus Admin Tool (cyradm).

You can find the latest tcl source at
ftp://ftp.sunlabs.com/pub/tcl/

After installation, make sure that the file libtcl.a is found in the

/usr/local/lib/ directory. Tcl 8.0 makes a file called libtcl8.0.a to which you should create a soft link using the following command:

```
# ln -s libtcl8.0.a libtcl.a
```

4. makedepend Installation

Check your system for the makedepend command. If you don't have it, don't worry - it comes with the Cyrus source. (I didn't have it on my installations of Slackware 3.4)

To install makedepend, extract the Cyrus distribution, change to the makedepend directory, and enter the following sequence of commands:

```
./configure
make
cp ./makedepend /usr/local/bin/makedepend
```

5. Cyrus Installation

Follow the instructions that come with the Cyrus distribution carefully. You can find a copy of them online at: http://andrew2.andrew.cmu.edu/cyrus/imapd/install.html

Some tips for some of the steps:

If you are running Slackware 3.4 (with Shadow Passwords), make sure that you configure in this way:

```
./configure --with-login=unix_pwcheck
```

make is pretty straight forward:

```
make depend
make all CFLAGS=-O
```

Step 1: When adding the user cyrus, I locked down the account to help minimize any security holes.

Step 3: I edited the /etc/syslog.conf file rather than copy theirs.

Step 9: With Linux, make sure to start up pwcheck in this way otherwise the server will not work right:

```
umask 0;/usr/cyrus/bin/pwcheck &
umask 022
```

Then add it to a startup script like this:

```
if [ -f /usr/cyrus/bin/pwcheck ]; then
echo -n "Starting pwcheck for imap"
umask 0;/usr/cyrus/bin/pwcheck &
umask 022
```

```
      fi
```

I put mine in /etc/rc.d/rc.local and it works well.

Step 12: When editing the /etc/inetd.conf, make sure to include the TCP Wrappers in your line like this:

```
imap    stream  tcp     nowait  cyrus   /usr/sbin/tcpd  /usr/cyrus/bin/imapd imap
```

And don't forget to kill -HUP inetd after you are done making your addition:

```
      # ps ax | grep inetd
         61  ?  S    0:00 /usr/sbin/inetd
      # kill -HUP 61
```

6. Sendmail Configuration

Download the sendmail source if you don't already have it. Besides running IMAP, you can do fun things like setting up anti-spam measures.

Here is my mc file. It will deliver mail to the IMAP location unless there is an entry of that username in the /etc/sendmail.cN file. This allows system accounts like root to keep their mail in a spool; however, user accounts use IMAP by default. Do not simply copy & paste this code because sendmail will compain about spaces being used rather than tabs:

```
divert(-1)
#
#       (C) Copyright 1995 by Carnegie Mellon University
#
#                       All Rights Reserved
#
# Permission to use, copy, modify, and distribute this software and its
# documentation for any purpose and without fee is hereby granted,
# provided that the above copyright notice appear in all copies and that
# both that copyright notice and this permission notice appear in
# supporting documentation, and that the name of CMU not be
# used in advertising or publicity pertaining to distribution of the
# software without specific, written prior permission.
#
# CMU DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING
# ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL
# CMU BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR
# ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
# WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
# ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
# SOFTWARE.
#
#       Contributed to Berkeley by John Gardiner Myers .
#
#       This sample mc file is for a site that uses the Cyrus IMAP server
#       exclusively for local mail.
```

```
#

divert(0)dnl
VERSIONID(`@(#)cyrusproto.mc    8.3 (Carnegie Mellon) @(#)cyrusproto.mc 8.3')
OSTYPE(linux)
define(`confBIND_OPTS',`-DNSRCH -DEFNAMES')
FEATURE(nouucp)
FEATURE(nocanonify)
FEATURE(always_add_domain)
MAILER(smtp)
MAILER(local)
MAILER(cyrus)

define(`confLOCAL_MAILER',`cyrus')

LOCAL_RULE_0
R$=N                            $: $#local $: $1
R$=N < @ $=w . >                $: $#local $: $1
Rbb + $+ < @ $=w . >            $#cyrusbb $: $1

LOCAL_CONFIG
FN /etc/sendmail.cN

# end of mc file
```

After making the /etc/sendmail.cf file, create a file called /etc/sendmail.cN and put in the username of accounts that do not wish to use the IMAP delivery method:

```
root
majordom
stan
mothra
```

After installing Sendmail 8.8.8 I also installed mail.local as my local delivery program for these other accounts. There is a trick to making mail.local. Change to the mail.local directory in the sendmail source and do the following:

```
cp Makefile Makefile.orig
cp Makefile.dist Makefile
make
cp mail.local /bin/mail.local
chmod 4555 /bin/mail.local
```

After all of this is done, go ahead and restart sendmail.

Don't forget to finish the Cyrus installation instructions.

7. Setting up mailboxes

Make sure to follow the tests for the IMAP server. If everything looks acceptable, go ahead and set up some mailboxes.

TOC

8. Disclaimers

No guarantees, no money back, use at your own risk.

9. Sources

Required Packages

- The Cyrus home page is http://andrew2.andrew.cmu.edu/cyrus/imapd/
- You can download the latest version from ftp://ftp.andrew.cmu.edu/pub/cyrus-mail/
- The Tcl home page is http://sunscript.sun.com/
- You can download the latest tcl source from ftp://ftp.sunlabs.com/pub/tcl/
- The Sendmail home page is http://www.sendmail.org/
- You can download the latest version from ftp://ftp.sendmail.org/ucb/src/sendmail/

DHCP mini-HOWTO (DHCPd/DHCPcd)

Vladimir Vuksan, vuksan@veus.hr

v4.3, 22 June 1999

This document attempts to answer basic questions on how to setup your Linux box to serve as a DHCP server or a DHCP client.

1. Introduction

- 1.1 Standard Disclaimer
- 1.2 New Versions of this Document
- 1.3 Feedback
- 1.4 Contributors
- 1.5 Copyright Information

2. DHCP protocol

3. Client Setup

- 3.1 Downloading Client Daemon
- 3.2 Slackware
- 3.3 RedHat 5.x and RedHat 6.x
- 3.4 RedHat 4.x and Caldera OpenLinux 1.1/1.2
- 3.5 Debian
- 3.6 LinuxPPC and MkLinux
- 3.7 Token Ring networks
- 3.8 Tying it all together
- 3.9 Various notes
- 3.10 Troubleshooting

4. DHCP Server Setup

- 4.1 DHCP server for UNIX
- 4.2 Network Configuration.
- 4.3 Options for DHCPd
- 4.4 Starting the server

---

1. Introduction

1.1 Standard Disclaimer

No liability for the contents of this documents can be accepted. Use the concepts, examples and other content at your own risk. As this is a new edition of this document, there may be errors and inaccuracies, that may of course be damaging to your system. Proceed with caution, and although this is highly unlikely, I don't take any responsibility for that.

Also bear in mind that this is `NOT` official information. Much content in this document are assumptions, which appear to work for people. Use the information at your own risk.

1.2 New Versions of this Document

New versions of this document will be periodically posted to *comp.os.linux.answers*. They will also be added to the various anonymous FTP sites who archive such information, including:

`ftp://metalab.unc.edu/pub/Linux/docs/HOWTO`

In addition, you should generally be able to find this document on the Linux Documentation Project page via:

`http://metalab.unc.edu/LDP/`

Chinese version of the HOWTO can be found at:

http://www.linux.org.tw/CLDP/mini/DHCP.html

Japanese version at:

http://www.linux.or.jp/JF/JFdocs/DHCP.html

Permission is hereby granted to all individuals who want to translate the document into their own language. I would only ask that you provide a link back to this document.

1.3 Feedback

Feedback is most certaintly welcome for this document. Without your submissions and input, this document wouldn't exist. So, please post your additions, comments and criticisms to `vuksan@veus.hr`.

1.4 Contributors

This document has been modified from the original version by Paul Makeev.

The following people have contributed to this mini-HOWTO.

- Heiko Schlittermann
- Jonathan Smith
- Dan Khabaza
- Hal Sadofsky
- Henrik Stoerner
- Paul Rossington

1.5 Copyright Information

This document is copyrighted (c)1998 Vladimir Vuksan and distributed under the following terms:

- Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.
- All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.
- If you have questions, please contact the Linux HOWTO coordinator at

linux-howto@metalab.unc.edu

---

2. DHCP protocol

DHCP is Dynamic Host Configuration Protocol. It is used to control vital networking parameters of hosts (running clients) with the help of a server. DHCP is backward compatible with BOOTP. For more information see RFC 2131 (old RFC 1531) and other. (See Internet Resources section at the end of the document). You can also read DHCP FAQ (http://web.syr.edu/ jmwobus/comfaqs/dhcp.faq.html).

This mini-HOWTO covers both the DHCP _SERVER_ daemon as well as DHCP _CLIENT_ daemon. Most people need the client daemon which is used by workstations to obtain network information from a remote server. The server daemon is used by system administrators to distribute network information to clients so if you are just a regular user you need the _CLIENT_ daemon.

3. Client Setup

3.1 Downloading Client Daemon

2.0.x kernels

No matter what distribution you are using you will need to download the DHCP client daemon for Linux. The package you need to download is called dhcpcd and the current version is 0.70. You can read the description of the package here. (ftp://metalab.unc.edu/pub/Linux/system/network/daemons/dhcpcd-0.70.lsm)

2.2.x kernels

Due to changes in ipv4 network package in 2.1.x and 2.2.x kernels (e.g. way it sets the defaults for several fields) dhcpcd doesn't work properly. Unless you are using RedHat 6.x (which comes with dhcpcd 1.3.x and 2.2.x kernel) you will need to download an alternative version of dhcpcd (1.3.x) which is a modified version that has been written by Sergei Viznyuk `sergei@phystech.com` in order to avoid mentioned problems. You can download it at:

- ftp://phystech.dyn.ml.org/pub/
- http://www.cps.msu.edu/~dunham/out/dhcpcd-1.3.6.tar.gz

Then follow the instructions. They should be the same.

3.2 Slackware

You can download the latest copy of the DHCPcd from any Metalab mirror or following:

- ftp://ftp.cdrom.com/pub/linux/metalab/system/network/daemons/
- ftp://metalab.unc.edu/pub/Linux/system/network/daemons
- ftp://ftp.kobe-u.ac.jp/pub/PC-UNIX/Linux/network/dhcp (Primary site in Japan)

Download the latest version of dhcpcd.tar.gz

- Unpack it

  ```
  tar -zxvf dhcpcd-0.70.tar.gz
  ```
- cd into the directory and make dhcpcd

  ```
  cd dhcpcd-0.70
  ```

  ```
  make
  ```
- Install it (you have to run the following command as root)

  ```
  make install
  ```

This will create the directory /etc/dhcpc where DHCPcd will store the DHCP information and dhcpcd file will be copied into /usr/sbin.

In order to make the system initialize using DHCP during boot type:

```
cd /etc/rc.d
```

```
mv rc.inet1 rc.inet1.OLD
```

This will move the old network initialization script into rc.inet1.OLD. You now need to create the new rc.inet1 script. Following code is all you need:

```
#!/bin/sh
#
# rc.inet1      This shell script boots up the base INET system.

HOSTNAME=`cat /etc/HOSTNAME` #This is probably not necessary but I
                             #will leave it in anyways

# Attach the loopback device.
/sbin/ifconfig lo 127.0.0.1
/sbin/route add -net 127.0.0.0 netmask 255.0.0.0 lo

# IF YOU HAVE AN ETHERNET CONNECTION, use these lines below to configure the
# eth0 interface. If you're only using loopback or SLIP, don't include the
# rest of the lines in this file.

/usr/sbin/dhcpcd
```

Save it and reboot your computer.

When you are finished go the last step

3.3 RedHat 5.x and RedHat 6.x

DHCPcd configuration under RedHat 5.0+ is really easy. All you need to do is start the Control Panel by typing

```
control-panel
```

- Select "Network Configuration"
- Click on Interfaces
- Click Add
- Select Ethernet
- In the Edit Ethernet/Bus Interface select **"Activate interface at boot time"** as well as select **DHCP** as **Interface configuration protocol**

When you are finished go the last step

3.4 RedHat 4.x and Caldera OpenLinux 1.1/1.2

DHCPcd is included in the standard RedHat distribution as an RPM and you can find it on your distribution's CD-ROM in RPMS directory or you can download it from:

ftp://ftp.redhat.com/pub/redhat/redhat-4.2/i386/RedHat/RPMS/dhcpcd-0.6-2.i386.rpm

and install it with

```
rpm -i dhcpcd-0.6-2.i386.rpm
```

Alternatively you can compile your own version by following the steps outlined in the Slackware.

```
This information was provided to me by nothing nothing@cc.gatech.edu
```

```
Removed my static ip and name from /etc/resolv.conf. However, I
did leave in the search line and my two nameserver lines (for some reason my
dhcpcd never creates a /etc/dhcpc/resolv.conf, so I have to use a static
/etc/resolv.conf).

In /etc/sysconfig/network I removed the HOSTNAME and GATEWAY
entries. I left the other entries as is
(NETWORKING, DOMAINNAME, GATEWAYDEV).

In /etc/sysconfig/network-scripts/ifcfg-eth0 I removed the IPADDR,
NETMASK, NETWORK, and BROADCAST entries. I left DEVICE and ONBOOT as is.
I changed the BOOTPROTO line to BOOTPROTO=dhcp.

Save the file. Reboot your computer.
```

When you are finished go the last step

3.5 Debian

There is a deb package of DHCPcd at

ftp://ftp.debian.org/debian/dists/slink/main/binary-i386/net/dhcpcd_0.70-4.deb

or you can follow the Slackware installation instructions. To unpack the deb package type

```
dpkg -i /where/ever/your/debian/packages/are/dhcpd*deb
```

It appears that there isn't a need for any DHCPcd configuration because:

```
From: Heiko Schlittermann (heiko@os.inf.tu-dresden.de)
```

The dhcpcd package installs it's startup script as usual for debian packages in /etc/init.d/<package_name>, here as /etc/init.d/dhcpcd, and links this to the various /etc/rc?.d/ directories.

The contents of the /etc/rc?.d/ dirs is then executed at boot time.

If you don't reboot after installing you should consider starting the daemon manually:

```
/etc/init.d/dhcpcd start
```

When you are finished go the last step

3.6 LinuxPPC and MkLinux

TOC

Following section has been written by R. Shapiro

Versions 0.65 and 0.70 of Yoichi Hariguchi's dhcpcd should work properly in MkLinux and in linuxppc kernel 2.1.24, with the following caveats:

* If you want, or need, to build the executable from sources, note that the ppc linux compilers assume that 'char' is 'unsigned char' while the Hariguchi sources assume 'char' is 'signed char'. To build from sources you must edit the Makefile so that CFLAGS includes the option "-fsigned-char".

* The current stable release of linuxppc [aka linux-pmac] is 2.1.24 and requires the

2.1 patch (http://www.cro.net/ vuksan/dhcppatch). Both the DR2.1 and DR3.0 releases of MkLinux use a 2.0 kernel (2.0.33) and do not require this patch, although it's harmless to apply it. Note that the dhcpcd rpm on the linuxppc cd-rom does not include the 2.1 patch and therefore will not work with the linux on that cd! It will work with MkLinux however.

* In linuxppc 2.1.24, you'll see a router warning shortly after dhcpcd starts up. You can ignore this.

* The Hariguchi dhcpcd takes awhile, about 30 seconds, to make its initial connection to the server and to set up routing. In linuxppc 2.1.24, the warning mentioned above is an indication that the routing is ready.

For later linuxppc kernels, no version of the Hariguchi dhcpcd will work: you **must** use Sergei Viznyuk's version instead (current release is 1.3.9: see above for url). Unfortunately the Viznyuk dhcpcd is written for glibc 2, which linuxppc 2.1.1xx isn't. As a result, compiling it is a bit tricky - contact me for details. Once compiled, however, it works fine on late kernels (and not at all in MkLinux or linuxppc 2.1.24).

As far as Viznyuk's version of dhcpcd is concerned I have a Viznyuk dhcpcd (v1.3.7) executable that works in recent linuxppc kernels: 2.1.102, 103, 115, and 119 have been tested. It's possible to build this from sources, but I don't know the details. The Viznyuk dhcpcd doesn't work in 2.1.24, but in that kernel the patched Hariguchi dhcpcd works. The Hariguchi dhcpcd can be built easily from sources.

Short summary:

```
MkLinux:  Hariguchi: yes; Viznyuk, no
2.1.24:   Hariguchi: yes if patched (easy to build); Viznyuk: no
2.1.102+: Hariguchi: no; Viznyuk: yes (tricky to build)
```

Note that the Viznyuk dhcpcd writes into /etc/resolv.conf directly (after renaming the existing one), so there's no need to copy or link it from /etc/dhcpc. Also note that it's typically installed into /sbin, not /usr/sbin, and that the command lines options are slightly different from the Hariguchi version. These differences may require small changes to ifup, if you're starting dhcpcd that way.

If you want a precompiled dhcpcd for linuxppc, send mail to

reshapiro@mediaone.net.

I've also made binary RPMs available in

ftp://ftp.linuxppc.org//pub/linuxppc/contrib/linuxppc-R4/RPMS/ Don't use dhcpcd-1.3.8-2.ppc.rpm in that directory, it's broken. The reliable versions here are dhcpcd-0.70-0.ppc.rpm (for linuxppc 2.1.24), and dhcpcd-1.3.8-3.ppc.rpm

(linuxppc 2.1.102 and up). An rpm for 1.3.9 should show up shortly. I also have a modified 1.3.9 which include -c command-file option, as in 0.65 and 0.70 (the standard Viznyuk dhcpcd doesn't include this.)

TOC

3.7 Token Ring networks

If you are trying to run dhcpcd on the Token Ring Network it will not work. This is the solution provided to me by Henrik Stoerner `(henrik_stoerner@olicom.dk)`

The problem is that dhcpcd only knows about Ethernet cards. If it finds a Token-Ring card, it refuses to do anything with it and reports "interface is not ethernet".

The solution is to apply a simple patch to the dhcpcd sources. I have put up a small web page with the patch, RedHat RPM-files and a precompiled binary at http://eolicom.olicom.dk/~storner/dhcp/

The patch has been sent to the dhcpcd maintainer, so hopefully it will be included in a future release of dhcpcd.

3.8 Tying it all together

After your machine reboots your network interface should be configured. Type:

```
ifconfig
```

You should get something like this

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
          UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
          RX packets:302 errors:0 dropped:0 overruns:0 frame:0
          TX packets:302 errors:0 dropped:0 overruns:0 carrier:0 coll:0

eth0      Link encap:Ethernet  HWaddr 00:20:AF:EE:05:45
          inet addr:24.128.53.102  Bcast:24.128.53.255  Mask:255.255.254.0
          ^^^^^^^^^^^^^^^^^^^^^^^^
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:24783 errors:1 dropped:1 overruns:0 frame:1
          TX packets:11598 errors:0 dropped:0 overruns:0 carrier:0 coll:96
          Interrupt:10 Base address:0x300
```

If you have some normal number under inet. addr you are set. If you see 0.0.0.0 don't despair, it is a temporary setting before dhcpcd acquires the IP address. If even after few minutes you are seeing 0.0.0.0 please check out troubleshooting. DHCPcd is a daemon and will stay running as long as you have your machine on. Every three hours it will contact the DHCP server and try to renew the IP address lease. It will log all the messages in the syslog (on Slackware /var/adm/syslog, RedHat/OpenLinux /var/log/syslog).

One final thing. You need to specify your nameservers. There are two ways to do it, you can either ask your provider to provide you with the addresses of your name server and then put those in the /etc/resolv.conf or DHCPcd will obtain the list from the DHCP server and will build a resolv.conf in /etc/dhcpc. I decided to use DHCPcds resolv.conf by doing the following:

Back up your old /etc/resolv.conf

```
mv /etc/resolv.conf /etc/resolv.conf.OLD
```

If directory /etc/dhcpc doesn't exist create it

```
mkdir /etc/dhcpc
```

Make a link from /etc/dhcpc/resolv.conf to /etc/resolv.conf

```
ln -s /etc/dhcpc/resolv.conf /etc/resolv.conf
```

If that doesn't work try this (fix suggested by nothing@cc.gatech.edu with a little amendment by Henrik Stoerner)

This last step I had to perform only because my dhcpcd doesn't create an /etc/dhcpc/resolv.conf. In /etc/sysconfig/network-scripts/ifup I made the following changes (which are a very poor hack, but they work for me):

```
elif [ "$BOOTPROTO" = dhcp -a "$ISALIAS" = no ]; then
    echo -n "Using DHCP for ${DEVICE}... "
    /sbin/dhcpcd -c /etc/sysconfig/network-scripts/ifdhcpc-done ${DEVICE}
    echo "echo \$$ > /var/run/dhcp-wait-${DEVICE}.pid; exec sleep 30" | sh

    if [ -f /var/run/dhcp-wait-${DEVICE}.pid ]; then
        ^^^^
        echo "failed."
        exit 1
```

I changed to:

```
elif [ "$BOOTPROTO" = dhcp -a "$ISALIAS" = no ]; then
    echo -n "Using DHCP for ${DEVICE}... "
    /sbin/dhcpcd
    echo "echo \$$ > /var/run/dhcp-wait-${DEVICE}.pid; exec sleep 30" | sh

    if [ ! -f /var/run/dhcp-wait-${DEVICE}.pid ]; then
        ^^^^^^
       echo "failed."
       exit 1
```

Notice the ! (bang) in `if [ ! -f /var/run/dhcp-wait-${DEVICE}.pid ];`

Now sit back and enjoy :-).

3.9 Various notes

Following step(s) are not necessary but might be useful to some people

a) If you need network connectivity only occasionally you can start dhcpcd from the command line (you have to be root to do this) with:

```
/usr/sbin/dhcpcd
```

When you need to down (turn off) the network type

```
/usr/sbin/dhcpcd -k
```

3.10 Troubleshooting

If you have followed the steps outlined above and you are unable to access the network there are several possible explanations:

I. Your network card is not configured properly.

During the boot up process your Linux will probe your network card and should say something along these lines:

```
eth0: 3c509 at 0x300 tag 1, 10baseT port, address  00 20 af ee 11 11, IRQ 10.
3c509.c:1.07 6/15/95 becker@cesdis.gsfc.nasa.gov
```

If a message like this doesn't appear your ethernet card might not be recognized by your Linux system. If you have a generic ethernet card (a NE2000 clone) you should have received a disk with DOS utilities that you can use to set up the card. Try playing with IRQs until Linux recognizes your card (IRQ 9,10,12 are usually good).

II. Your DHCP server supports RFC 1541/My DHCP server is Windows NT

Try running dhcpcd by typing

```
dhcpcd -r
```

Use ifconfig to check if your network interface is configured (wait few seconds for the configuration process, initally it will say Inet.addr=0.0.0.0)

If this solves your problem add the "-r" flag to the boot up scripts ie. instead of `/sbin/dhcpcd` you will have `/sbin/dhcpcd -r`

For example under RedHat edit script /etc/sysconfig/network-scripts/ifup and change the following

---

```
        IFNAME=$[ {DEVICE} \
        "/sbin/dhcpcd -r -c /etc/"- etc etc.
```

III. During bootup I get error message "Using DHCP for eth0 ... failed" but my system works fine.

You are most likely using RedHat and you haven't followed instructions carefully :-). You are missing the ! (bang) in one of the if statements. Jump here and check how to fix it.

IV. My network works for few minutes and then stops responding

There are some reports of gated (gateway daemon) screwing up routing on Linux boxes which results in problem described above. Check if gated is running

```
ps -auxww | grep gate
```

If it is try removing it with RedHat's RPM manager or removing the entry in /etc/rc.d/

V. My ethernet card is recognized during boot up but I still get "NO DHCPOFFER" message in my logs. I also happen to have a PCMCIA ethernet card.

You need to make sure that you have the 10BaseT port ("phone" plug) on your network card activated. Best way to verify it is to check what kind of connector your card is configured for during bootup e.g.

```
eth0: 3c509 at 0x300 tag 1, 10baseT port, address  00 20 af ee 11 11, IRQ 10.
                            ^^^^^^^^^^^^
3c509.c:1.07 6/15/95 becker@cesdis.gsfc.nasa.gov
```

I have received reports of laptop users having this kind of problems due to the PCMCIA utilities (specifically ifport) that would set the connector type to 10Base2 (thinnet). You have to make sure you use 10BaseT for your connection. If you are not reconfigure the card and restart the computer.

VI. My DHCP client broadcasts requests but no one answers (Contributed by Peter Amstutz)

On some systems, you need to include some hostname for your machine as part of the request. With dhcpcd, do this with 'dhcpcd -h foohost' Probably the hostname wanted will be your account username on the network.

VII. I have followed all the steps but still my machine is not able to connect

The cable modem will usually memorize the ethernet address of your network card so if you connect a new computer or switch network cards you will somehow have to "teach" your cable modem to recognize the new computer/card. Usually you can turn of the modem and bring it back up while computer is on or you will have to call tech support and tell them that you have changed a network card in the computer.

You have firewall rules (ipfwadm rules) that disallow port 67/68 traffic used by DHCP to distribute configuration info. Check your firewall rules carefully.

VIII. I have MediaOne Express service and I still can't connect.

It appears that MediaOne has been using adding some things to DHCP that shouldn't be there. Supposedly this is not a problem anymore but if you experience outages check for these things. If you are (un)lucky to have Windows NT on your machine if you go into Event Viewer you will see a warning like this.

DHCP received an unknown option 067 of length 005. The raw option data is given below.

```
0000: 62 61 73 69 63 basic
```

If this is the problem go to ftp://vanbuer.ddns.org/pub/ and either download a binary or get the source for the change.

---

4. DHCP Server Setup

4.1 DHCP server for UNIX

There are several DHCP servers available for U*X-like OSes, both commercial and free. One of the more popular free DHCP servers is Paul Vixie/ISC DHCPd. Currently the latest version is 2.0 (suggested for most users) but 3.0 is in beta testing. You can get them from

ftp://ftp.isc.org/isc/dhcp/

RedHat distribution (version 5.2) includes a binary RPM with DHCP server. The RPM is called dhcp-2.0b1pl6-2.platform.rpm. To install it simply type

```
rpm -i dhcp-2.0b1pl6-2.i386.rpm
```

You can skip following section if you used this procedure.

After you download it you need to unpack it. After you do cd into the distribution directory and type:

```
./configure
```

It will take some time to configure the settings. After it is done type:

```
make
```

and

```
make install
```

4.2 Network Configuration.

When done with installation type ifconfig -a. You should see something like this:

```
eth0      Link encap:10Mbps Ethernet  HWaddr 00:C0:4F:D3:C4:62
          inet addr:183.217.19.43  Bcast:183.217.19.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2875542 errors:0 dropped:0 overruns:0
          TX packets:218647 errors:0 dropped:0 overruns:0
          Interrupt:11 Base address:0x210
```

If it doesn't say MULTICAST you should reconfigure your kernel and add multicast support. On most systems you will not need to do this.

Next step is to add route for 255.255.255.255. Quoted from DHCPd README:

"In order for dhcpd to work correctly with picky DHCP clients (e.g., Windows 95), it must be able to send

packets with an IP destination address of 255.255.255.255. Unfortunately, Linux insists on changing 255.255.255.255 into the local subnet broadcast address (here, that's 192.5.5.223). This results in a DHCP protocol violation, and while many DHCP clients don't notice the problem, some (e.g., all Microsoft DHCP clients) do. Clients that have this problem will appear not to see DHCPOFFER messages from the server."

TOC

Type:

```
route add -host 255.255.255.255 dev eth0
```

If you get a message

```
"255.255.255.255: Unknown host"
```

You should try adding the following entry to your /etc/hosts file:

```
255.255.255.255 all-ones
```

Then, try:

```
route add -host all-ones dev eth0
```

or

```
route add 255.255.255.0 dev eth0
```

eth0 is of course the name of the network device you are using. If it differs change appropriately.

4.3 Options for DHCPd

Now you need to configure DHCPd. In order to do this you will have to create or edit /etc/dhcpd.conf.

Most commonly what you want to do is assign IP addresses randomly. This can be done with settings as follows

---

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "mydomain.org";

subnet 192.168.1.0 netmask 255.255.255.0 {
   range 192.168.1.10 192.168.1.100;
   range 192.168.1.150 192.168.1.200;
}
```

This will result in DHCP server giving a client an IP address from the range 192.168.1.10-192.168.1.100 or 192.168.1.150-192.168.1.200. It will lease an IP address for 600 seconds if the client doesn't ask for specific time frame. Otherwise the maximum (allowed) lease will be 7200 seconds. The server will also "advise" the client that it should use 255.255.255.0 as its subnet mask, 192.168.1.255 as its broadcast address, 192.168.1.254

as the router/gateway and 192.168.1.1 and 192.168.1.2 as its DNS servers.

You can also assign specific IP addresses based on clients ethernet address e.g.

```
host haagen {
   hardware ethernet 08:00:2b:4c:59:23;
   fixed-address 192.168.1.222;
}
```

This will assign IP address 192.168.1.222 to a client with ethernet address 08:00:2b:4c:59:23.

If you need to specify a WINS server for your Windows clients you will need to include the netbios-name-servers option e.g.

```
option netbios-name-servers 192.168.1.1;
```

You can also mix and match e.g. you can have certain clients getting "static" IP addresses (e.g. servers) and others being alloted dynamic IPs (e.g. mobile users with laptops). There are a number of other options e.g. nis server addresses, time server addresses etc., if you need any of those options please read the dhcpd.conf man page.

4.4 Starting the server

There is only one thing to do before starting the server. In most cases DHCP installation doesn't create a dhcpd.leases files. This file is used by DHCPd to store information about current leases. It is in the plain text form so you can view it during the operation of DHCPd. To create dhcpd.leases type

```
touch /etc/dhcpd.leases
```

on the command line. This will create an empty file (file size = 0). You do not need to make any changes to it it should be empty. If you get a message saying that file exists simply ignore it and go to the next step.

You can now invoke the DHCP server. Simply type (or include in the bootup scripts)

```
/usr/sbin/dhcpd
```

If you want to verify that everything is working fine you should first turn on the debugging mode and put the server in foreground. You can do this by typing

```
/usr/sbin/dhcpd -d -f
```

Then boot up one of your clients and check out the console of your server. You will see a number of debugging messages coming up.

Linux DPT Hardware RAID HOWTO

Ram Samudrala (me@ram.org)

v1.4, 9 April 1999

How to set up hardware RAID under Linux.

1. Introduction

2. Supported controllers

- 2.1 DPT controllers
- 2.2 ICP vortex controllers

3. What hardware should be used?

- 3.1 Controller type
- 3.2 Enclosure type

4. Installation

- 4.1 Installing and configuring the hardware
- 4.2 Configuring the kernel
- 4.3 Bootup messages

5. Usage

- 5.1 fdisk, mke2fs, mount, etc.
- 5.2 Hotswapping
- 5.3 Performance

TOC

6. Features in the EATA DMA driver

7. Troubleshooting

- 7.1 Upon bootup, no SCSI hosts are detected
- 7.2 RAID configuration shows up as N different disks
- 7.3 Machine/controller is shut down in the middle of a format
- 7.4 If all fails...

8. References

9. Acknowledgements

---

1. Introduction

This document describes how to set up SCSI hardware RAID, focusing mainly on host-based adapters from DPT, though the principles applied here are fairly general.

Use the information below at your own risk. I disclaim all responsibility for anything you may do after reading this HOWTO. The latest version of this HOWTO will always be available at http://www.ram.org/computing/linux/dpt_raid.html.

For the purposes of this HOWTO, I am assuming you have only a Linux system running. Also, note that I've only tried this out with the DPT Smartcache IV PM2144UW and PM3334UW controllers, with DPT (SmartRAID tower) and Wetex enclosures, and I have no experience with other setups. So things may be different for your setup.

2. Supported controllers

Currently the only well-supported host-based hardware RAID controller (i.e, a controller for which there exists a driver under Linux) is one that is made by DPT. However, there exist other host-based and scsi-to-scsi controllers which may work under Linux. These include the ones made by Syred, ICP-Vortex, and BusLogic. In addition, there exist plenty of SCSI-to-SCSI controllers. See the RAID solutions for Linux page for more info.

If, in the future, there is support for other controllers, I will do my best to incorporate that information into this HOWTO. Please send me any such information you think is appropriate for this HOWTO.

2.1 DPT controllers

This document is currently DPT-oriented. Essentially all the SmartRAID IV controllers are supported.

2.2 ICP vortex controllers

ICP vortex has a complete line of disk array controllers which support Linux. The ICP driver is in the Linux kernel since version 2.0.31 All major Linux Distributors S.u.S.e., LST Power Linux, Caldera and Red Hat support the ICP controllers as boot/installation controllers. The RAID system can easily be configured with our ROMSETUP (you do not have to boot MS-DOS for configuration!).

With the monitoring utility GDTMON it is possible to manage the complete ICP RAID system during operation (check transferrates, set parameters for the controller and hard disks, exchange defective hard disks, etc.). Currently available are: 1 and 2 channel wide and ultra SCSI controller for RAID 0 and RAID 1 1, 2, 3 and 5 chn. wide and ultra SCSI controller for RAID 0, 1, 4, 5 and 10 1 and 2 channel wide and ultra2 LVDS SCSI controller for RAID 0 and RAID 1 1, 2, 3 and 5 chn. wide and ultra2 LVDS SCSI controller for RAID 0, 1, 4, 5 and 10 1 and 2 port Fibre Channel contollers for RAID 0, 1, 4, 5 and 10 Pretty soon there will be also 64-bit controllers available.

3. What hardware should be used?

3.1 Controller type

Given all these options, if you're looking for a RAID solution, you need to think carefully about what you want. Depending on what you want to do, and which RAID level you wish to use, some cards may be better than others. SCSI-to-SCSI adapters may not be as good as host-based adapters, for example (see the DPT comparison between host-based and SCSI-to-SCSI adapters for why this is the case). Michael Neuffer (neuffer@kralle.zdv.uni-mainz.de), the author of the EATA-DMA driver, has a nice discussion about this on his Linux High Performance SCSI and RAID page.

3.2 Enclosure type

The enclosure type affects the hot swapability of the drive, the warning systems (i.e., whether there will be indication of failure, and whether you will know which drive has failed), and what kind of treatment your drive receives (for example, redundant cooling and power supplies). We used the DPT supplied enclosures for an initial RAID-5 system of 18 GB, but they are expensive. We are now using a Wetex enclosure (http://www.wetex.com/), which have the same features as the DPT ones, but are about 1/4 of the cost. The Wetex enclosure system (14 bays) we use is composed of two RAID-5 arrays, with sizes of 45 GB and 63 GB.

4. Installation

4.1 Installing and configuring the hardware

Refer to the instruction manual to install the card and the drives. For DPT, since a storage manager for Linux doesn't exist yet, you need to create a MS-DOS-formatted disk with the system on it (usually created using the command "format /s" at the MS-DOS prompt). You will also be using the DPT storage manager for MS-DOS, which you should probably make a copy of for safety.

Once the hardware is in place, boot using the DOS system disk. Replace the DOS disk with the storage manager. And invoke the storage manager using the command:

```
a:\ dptmgr
```

Wait a minute or so, and you'll get a nice menu of options. Configure the set of disks as a hardware RAID (single logical array). Choose "other" as the operating system.

The MS-DOS storage manager is a lot easier to use with a mouse, and so you might want to have a mouse driver on the initial system disk you create.

Technically, it should be possible to run the SCO storage manager under Linux, but it may be more trouble than its worth. It's probably more easier to run the MS-DOS storage manager under Linux.

4.2 Configuring the kernel

You will need to configure the kernel with SCSI support and the appropriate low level driver. See the Kernel HOWTO for information on how to compile the kernel. Once you choose "yes" for SCSI support, in the low level drivers section, select the driver of your choice (EATA DMA or EATA ISA/EISA/PCI for most EATA DMA compliant (DPT) cards, EATA PIO for the very old PM2001 and PM2012A from DPT). Most drivers, including the EATA DMA and EATA ISA/EISA/PCI drivers, should be available in recent kernel versions.

Once you have the kernel compiled, reboot, and if you've set up everything correctly, you should see the driver recognising the RAID as a single SCSI disk. If you use RAID-5, you will see the size of this disk to be 2/3 of the actual disk space available.

4.3 Bootup messages

The messages you see upon bootup if you're using the EATA DMA driver should look something like this:

```
EATA (Extended Attachment) driver version: 2.59b
developed in co-operation with DPT
(c) 1993-96 Michael Neuffer, mike@i-Connect.Net
Registered HBAs:
HBA no. Boardtype    Revis  EATA Bus  BaseIO IRQ DMA Ch ID Pr QS  S/G IS
scsi0 : PM2144UW     v07L.Y 2.0c PCI  0xef90  11 BMST 1  7  N  64 252 Y
scsi0 : EATA (Extended Attachment) HBA driver
scsi : 1 host.
  Vendor: DPT       Model: RAID-5            Rev: 07LY
  Type:   Direct-Access                      ANSI SCSI revision: 02
```

```
Detected scsi disk sda at scsi0, channel 0, id 8, lun 0
scsi0: queue depth for target 8 on channel 0 set to 64
scsi : detected 1 SCSI disk total.
SCSI device sda: hdwr sector= 512 bytes. Sectors= 35591040 [17378 MB] [17.4 GB]
```

(The above display is for a setup with a single DPT SCSI controller, configured as RAID-5, with three disks of 9 GB each.)

The messages you see upon bootup if you're using the EATA ISA/EISA/PCI driver should look something like this:

```
aic7xxx: <Adaptec AHA-294X SCSI host adapter> at PCI 15
aic7xxx: BIOS enabled, IO Port 0x7000, IO Mem 0x3100000, IRQ 15, Revision B
aic7xxx: Single Channel, SCSI ID 7, 16/16 SCBs, QFull 16, QMask 0x1f
EATA0: address 0x7010 in use, skipping probe.
EATA0: 2.0C, PCI 0x7410, IRQ 11, BMST, SG 252, MB 64, tc:y, lc:y, mq:62.
EATA0: wide SCSI support enabled, max_id 16, max_lun 8.
EATA0: SCSI channel 0 enabled, host target ID 6.
EATA/DMA 2.0x: Copyright (C) 1994-1997 Dario Ballabio.
scsi0 : Adaptec AHA274x/284x/294x (EISA/VLB/PCI-Fast SCSI) 4.1.1/3.2.1
scsi1 : EATA/DMA 2.0x rev. 3.11.00
scsi : 2 hosts.
scsi0: Scanning channel A for devices.
  Vendor: IBM OEM   Model: DFHSS2F           Rev: 1818
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi disk sda at scsi0, channel 0, id 0, lun 0
  Vendor: SEAGATE   Model: ST41650       TX  Rev: DG01
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi disk sdb at scsi1, channel 0, id 0, lun 0
  Vendor: TEAC      Model: FC-1     GF   00  Rev: RV L
  Type:   Direct-Access                      ANSI SCSI revision: 01 CCS
Detected scsi removable disk sdc at scsi1, channel 0, id 3, lun 0
  Vendor: SONY      Model: CD-ROM CDU-541    Rev: 2.6a
  Type:   CD-ROM                             ANSI SCSI revision: 02
Detected scsi CD-ROM sr0 at scsi1, channel 0, id 5, lun 0
EATA0: scsi1, channel 0, id 0, lun 0, cmds/lun 21, sorted, tagged.
EATA0: scsi1, channel 0, id 3, lun 0, cmds/lun 21, sorted.
EATA0: scsi1, channel 0, id 5, lun 0, cmds/lun 21, sorted.
scsi : detected 1 SCSI cdrom 3 SCSI disks total.
SCSI device sda: hdwr sector= 512 bytes. Sectors= 4404489 [2150 MB] [2.2 GB]
SCSI device sdb: hdwr sector= 512 bytes. Sectors= 2779518 [1357 MB] [1.4 GB]
SCSI device sdc: hdwr sector= 256 bytes. Sectors= 4160 [1 MB] [0.0 GB]
```

(The above display is for a setup wih two SCSI controllers, DPT PM3224W and and Adaptec AHA2940.)

5. Usage

5.1 fdisk, mke2fs, mount, etc.

You can now start treating the RAID as a regular disk. The first thing you'll need to do is partition the disk (using fdisk). You'll then need to set up an ext2 filesystem. This can be done by running the command:

```
% mkfs -t ext2 /dev/sdxN
```

where /dev/sdxN is the name of the SCSI partition. Once you do this, you'll be able to mount the partitions and use them as you would any other disk (including adding entries in /etc/fstab).

5.2 Hotswapping

We first tried to test hotswapping by removing a drive and putting it back in the DPT-supplied enclosure/tower (which you buy for an additional cost). Before we could carry this out to completion, one of the disks failed (as I write this, the beeping is driving me crazy). Even though one of the disks failed, all the data on the RAID drive is accessible.

Instead of replacing the drive, we just went through the motions and put the same drive back in. The drive rebuilt itself and everything seems to be okay. During the time the disk had filed, and during the rebuilding process, all the data was accessible. Though it should be noted that if another disk had failed, we'd have been in serious trouble.

5.3 Performance

Here's the output of the Bonnie program, on a 2144 UW with 9x3=17 GB RAID 5 setup, using the EATA DMA driver. The RAID is on a dual processor Pentium Pro machine running Linux 2.0.33. For comparison, the Bonnie results for the IDE drive on that machine are also given.

```
            -------Sequential Output-------- ---Sequential Input--  --Random--
            -Per Char- --Block--- -Rewrite-- -Per Char- --Block---  --Seeks---
         MB K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU   /sec  %CPU
RAID    100  9210 96.8  1613  5.9   717  5.8  3797 36.1 90931 96.8 4648.2 159.2
IDE     100  3277 32.0  6325 23.5  2627 18.3  4818 44.8 59697 88.0  575.9  16.3
```

Some people have disputed the above timings (and rightly so---I've been unable to try it out on our machines since they're completely loaded) because the size of the file used may have led to it being cached (resulting in an unusually good performance report). Here are some timings with a 3344 UW controller:

```
         -------Sequential Output-------- ---Sequential Input-- --Random--
         -Per Char- --Block--- -Rewrite-- -Per Char- --Block--- --Seeks---
      MB K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU  /sec %CPU
    1000  1714 17.2  1689  6.0  1200  5.7  5263 40.2  7023 12.1  51.3 2.2
```

And here are some timings on a host-based RAID system:

```
      -------Sequential Output-------- ---Sequential Input-- --Random--
      -Per Char- --Block--- -Rewrite-- -Per Char- --Block--- --Seeks---
MB    K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU /sec %CPU
64    7465 100.0 70287 98.7 37012 97.7 8074 99.2 *****100.3 ***** 196.6
128   7289 99.3  67595 98.5 35294 98.6 7792 97.6 *****100.3 ***** 195.8
256   7222 98.8  44844 69.6 16096 51.8 5787 72.7 ***** 99.8 ***** 85.2
512   7138 98.4  13871 23.2 7888  29.3 7183 89.3 16488 27.2 1585. 11.5
```

```
1024 6908 95.8  12270 21.5 7161  25.4 7373 90.4 16527 28.2 123.8 1.8
2047 6081 84.1  12664 22.6 7191  25.6 7289 89.5 16573 28.5 75.0  1.2

***** results exceed column width (> 100 MB/sec, > 10000 seeks/sec)

host:   Dual PII 400 MHz, 2 x U2W, 512 MB RAM, no internal disks
RAID:   IFT 3102 UA 128 MB Cache, RAID-5, 6 x 9 GB
OS:     SuSE Linux 6.0 with Kernel 2.2.3
```

6. Features in the EATA DMA driver

This section describes some of the commands available under Linux to check on the RAID configuration. Again, while references to the eata_dma driver is made, this can be used to check up on any driver.

To see the configuration for your driver, type:

```
% cat /proc/scsi/eata_dma/N
```

where N is the host id for the controller. You should see something like this:

```
EATA (Extended Attachment) driver version: 2.59b
queued commands:         353969
processed interrupts:    353969

scsi0 : HBA PM2144UW
Firmware revision: v07L.Y
Hardware Configuration:
IRQ: 11, level triggered
DMA: BUSMASTER
CPU: MC68020 20MHz
Base IO : 0xef90
Host Bus: PCI
SCSI Bus: WIDE Speed: 10MB/sec.
SCSI channel expansion Module: not present
SmartRAID hardware: present.
    Type: integrated
    Max array groups:              7
    Max drives per RAID 0 array:   7
    Max drives per RAID 3/5 array: 7
Cache Module: present.

    Type: 0
    Bank0: 16MB without ECC
    Bank1: 0MB without ECC
    Bank2: 0MB without ECC
    Bank3: 0MB without ECC
Timer Mod.: present
NVRAM     : present
SmartROM  : enabled
Alarm     : on
Host<->Disk command statistics:
```

```
            Reads:      Writes:
      1k:           0            0
      2k:           0            0
      4k:           0            0
      8k:           0            0
     16k:           0            0
     32k:           0            0
     64k:           0            0
    128k:           0            0
    256k:           0            0
    512k:           0            0
   1024k:           0            0
  >1024k:           0            0
  Sum   :           0            0
```

To get advanced command statistics, type:

```
% echo "eata_dma latency" > /proc/scsi/eata_dma/N
```

Then you can do a:

```
% cat /proc/scsi/eata_dma/N
```

to get more detailed statistics.

To turn off advanced command statistics, type:

```
% echo "eata_dma nolatency" > /proc/scsi/eata_dma/N
```

---

7. Troubleshooting

7.1 Upon bootup, no SCSI hosts are detected

This could be due to several reasons, but it's probably because the appropriate driver is not configured in the kernel. Check and make sure the appropriate driver (EATA-DMA or EATA ISA/EISA/PCI for most DPT cards) is configured.

7.2 RAID configuration shows up as N different disks

The RAID has not been configured properly. If you're using a DPT storage manager, you need to configure the RAID disks as a single logical array. Michael Neuffer (neuffer@kralle.zdv.uni-mainz.de) writes: "When you configure the controller with the SM start it with the parameter /FW0 and/or select Solaris as OS. This will cause the array to be setup to be managed internally by the controller."

7.3 Machine/controller is shut down in the middle of a format

As stated in the DPT manual, this is clearly a no-no and might require the disks to be returned to the manufactured, since the DPT Storage Manager might not be able format it. However, you might be able to perform a low level format on it, using a program supplied by DPT, called clfmt in their utilities page (http://www.dpt.com/techsup/sr4utils.htm). Read the instructions after unzipping the clfmt.zip file on how to use it (and use it wisely). Once you do the low level format, you might be able to treat the disks like new. Use this program carefully!

7.4 If all fails...

Read the SCSI-HOWTO again. Check the cabling and the termination. Try a different machine if you have access to one. The most common cause of problems with SCSI devices and drivers is because of faulty or misconfigured hardware. Finally, you can post to the various newsgroups or e-mail me, and I'll do my best to get back to you.

8. References

The following documents may prove useful to you as you set up RAID:

- DPT Technology Library
- EATA-DMA homepage
- Linux Disk HOWTO
- Linux Kernel HOWTO
- Linux SCSI HOWTO
- Multi Disk System Tuning HOWTO
- RAID Solutions for Linux

9. Acknowledgements

The following people have been helpful in getting this HOWTO done:

- Andreas Koepf (A_Koepf@icp-vortex.com)
- Boris Fain (fain@zen.stanford.edu)
- Dario Ballabio (Dario_Ballabio@milano.europe.dg.com)
- Heiko Rommel (Heiko.Rommel@Uni-Bielefeld.DE)
- Jos Vos (jos@xos.nl)
- Michael Neuffer (neuffer@kralle.zdv.uni-mainz.de)
- Ralph Wallace (rwallace@rwallace.interaccess.com)
- Russell Brown (russell@lutton.lls.com)
- Syunsuke Ogata (Syunsuke_Ogata@appear.ne.jp)
- Tom Brown (tbrown@baremetal.com)

Setting up diald for Linux - an example

TOC

GNU Copyright by Harish Pillay h.pillay@ieee.org 1996

The following is an example of a setup I have running on many machines that allow automatic installation of diald upon boot-up. When a connection request comes along, it automatically dials up my preferred ISP, Pacific Internet here in Singapore.

Many thanks to Eric Schenk for the super tool diald.

/etc/rc.d/rc.local

```
#! /bin/sh
# Put any local setup commands in here
# Running gpm
echo "Running gpm..."
gpm -t ms &

# starting innd
/etc/rc.d/rc.news

# starting CERN httpd
echo "Starting CERN httpd with proxy and caching."
/usr/local/bin/httpd

# loading modules that are needed
/etc/rc.d/rc.modules

# starting diald
echo "Starting diald daemon to Pacific Internet ..."
cd /usr/lib/ppp
/usr/lib/ppp/diald.pacific.internet
```

/usr/lib/ppp/diald.pacific.internet

```
/usr/sbin/diald /dev/ttyS1 /dev/ttyS1 lock debug 20 -m ppp local 127.0.0.2 \
  remote 127.0.0.3 defaultroute modem crtscts \
  connect "chat -v -f /usr/lib/ppp/pppchat.pi" \
  dynamic -- debug noipdefault
```

/usr/lib/ppp/pppchat.pi

TOC

```
ABORT "NO CARRIER"
ABORT BUSY
"" ATZ OK
ATm1s50=255s111=0DT1-800-555-1212
CONNECT ""
login MYLOCINNAME
ssword MYPASSWORD
```

Blind PPP script

- Fire up minicom.
- Connect up to your provider.
- invoke PPP on the other end.
- Suspend minicom (alt-A-J).
- invoke the following script

```
#!/bin/sh
#       Set up a PPP link blindly - script called blind.ppp

DEVICE=ttyS1
(
        stty 38400 -tostop
        pppd -detach debug noipdefault defaultroute mru 1500 /dev/$DEVICE &
        exit 0
) < /dev/$DEVICE > /dev/$DEVICE
```

My /etc/diald.conf file (unchanged from stock distribution):

```
# This is a pretty complicated set of filter rules.
# (These are the rules I use myself.)
#
# I've divided the rules up into four sections.
# TCP packets, UDP packets, ICMP packets and a general catch all rule
# at the end.
#-----------------------------------------------------------------------------
# Rules for TCP packets.
#-----------------------------------------------------------------------------
# General comments on the rule set:
#
# In general we would like to treat only data on a TCP link as signficant
# for timeouts. Therefore, we try to ignore packets with no data.
# Since the shortest possible set of headers in a TCP/IP packet is 40 bytes.
```

TOC

```
# Any packet with length 40 must have no data riding in it.
# We may miss some empty packets this way (optional routing information
# and other extras may be present in the IP header), but we should get
# most of them. Note that we don't want to filter out packets with
# tcp.live clear, since we use them later to speedup disconnects
# on some TCP links.
#
# We also want to make sure WWW packets live even if the TCP socket
# is shut down. We do this because WWW doesn't keep connections open
# once the data has been transfered, and it would be annoying to have the link
# keep bouncing up and down every time you get a document.
#
# Outside of WWW the most common use of TCP is for long lived connections,
# that once they are gone mean we no longer need the network connection.
# We don't neccessarily want to wait 10 minutes for the connection
# to go down when we don't have any telnet's or rlogin's running,
# so we want to speed up the timeout on TCP connections that have
# shutdown. We do this by catching packets that do not have the live flag set.

# --- start of rule set proper ---

# When initiating a connection we only give the link 15 seconds initially.
# The idea here is to deal with possibility that the network on the opposite
# end of the connection is unreachable. In this case you don't really
# want to give the link 10 minutes up time. With the rule below
# we only give the link 15 seconds initially. If the network is reachable
# then we will normally get a response that actually contains some
# data within 15 seconds. If this causes problems because you have a slow
# response time at some site you want to regularly access, you can either
# increase the timeout or remove this rule.
accept tcp 15 tcp.syn

# Keep named xfers from holding the link up
ignore tcp tcp.dest=tcp.domain
ignore tcp tcp.source=tcp.domain

# (Ack! SCO telnet starts by sending empty SYNs and only opens the
# connection if it gets a response. Sheesh..)
accept tcp 5 ip.tot_len=40,tcp.syn

# keep empty packets from holding the link up (other than empty SYN packets)
ignore tcp ip.tot_len=40,tcp.live

# make sure http transfers hold the link for 2 minutes, even after they end.
# NOTE: Your /etc/services may not define the tcp service www, in which
# case you should comment out the following two lines or get a more
# up to date /etc/services file. See the FAQ for information on obtaining
# a new /etc/services file.
accept tcp 120 tcp.dest=tcp.www
accept tcp 120 tcp.source=tcp.www
```

TOC

```
# Once the link is no longer live, we try to shut down the connection
# quickly. Note that if the link is already down, a state change
# will not bring it back up.
keepup tcp 5 !tcp.live
ignore tcp !tcp.live

# an ftp-data or ftp connection can be expected to show reasonably frequent
# traffic.
accept tcp 120 tcp.dest=tcp.ftp
accept tcp 120 tcp.source=tcp.ftp

#NOTE: ftp-data is not defined in the /etc/services file provided with
# the latest versions of NETKIT, so I've got this commented out here.
# If you want to define it add the following line to your /etc/services:
# ftp-data        20/tcp
# and uncomment the following two rules.
#accept tcp 120 tcp.dest=tcp.ftp-data
#accept tcp 120 tcp.source=tcp.ftp-data

# If we don't catch it above, give the link 10 minutes up time.
accept tcp 600 any

# Rules for UDP packets
#
# We time out domain requests right away, we just want them to bring
# the link up, not keep it around for very long.
# This is because the network will usually come up on a call
# from the resolver library (unless you have all your commonly
# used addresses in /etc/hosts, in which case you will discover
# other problems.)
# Note that you should not make the timeout shorter than the time you
# might expect your DNS server to take to respond. Otherwise
# when the initial link gets established there might be a delay
# greater than this between the initial series of packets before
# any packets that keep the link up longer pass over the link.

# Don't bring the link up for rwho.
ignore udp udp.dest=udp.who
ignore udp udp.source=udp.who
# Don't bring the link up for RIP.
ignore udp udp.dest=udp.route
ignore udp udp.source=udp.route
# Don't bring the link up for NTP or timed.
ignore udp udp.dest=udp.ntp
ignore udp udp.source=udp.ntp
ignore udp udp.dest=udp.timed
ignore udp udp.source=udp.timed
# Don't bring up on domain name requests between two running nameds.
ignore udp udp.dest=udp.domain,udp.source=udp.domain
# Bring up the network whenever we make a domain request from someplace
# other than named.
```

TOC

```
accept udp 30 udp.dest=udp.domain
accept udp 30 udp.source=udp.domain
# Do the same for netbios-ns broadcasts
# NOTE: your /etc/services file may not define the netbios-ns service
# in which case you should comment out the next three lines.
ignore udp udp.source=udp.netbios-ns,udp.dest=udp.netbios-ns
accept udp 30 udp.dest=udp.netbios-ns
accept udp 30 udp.source=udp.netbios-ns
# keep routed and gated transfers from holding the link up
ignore udp tcp.dest=udp.route
ignore udp tcp.source=udp.route
# Anything else gest 2 minutes.
accept udp 120 any

# Catch any packets that we didn't catch above and give the connection
# 30 seconds of live time.
accept any 30 any
```

My diald.defs file (unchanged from stock distribution):

```
# Define default protocol rules
prule tcp tcp 9:12:13:14:15:16:17:18:19:+0:+1:+2:+3:9:9:9
prule udp udp 9:12:13:14:15:16:17:18:19:+0:+1:+2:+3:9:9:9
prule icmp icmp 9:12:13:14:15:16:17:18:19:9:9:9:9:9:9:9
prule any any 9:12:13:14:15:16:17:18:19:9:9:9:9:9:9:9
# Define the internet packet header fields.
var ip.ihl 0(24)&0xf
var ip.version 0(28)&0xf
var ip.tos 1(24)&0xff
var ip.tot_len 2(16)&0xffff
var ip.id 4(16)&0xffff
var ip.frag_off 6(16)&0x3fff
var ip.ttl 8(24)&0xff
var ip.protocol 9(24)&0xff
var ip.check 10(16)&0xffff
var ip.saddr 12
var ip.daddr 16
# Define the TCP packet header fields.
var tcp.source +0(16)&0xffff
var tcp.dest +2(16)&0xffff
var tcp.seq +4
var tcp.ack_seq +8
var tcp.doff +12(28)&0xf
var tcp.fin +13(24)&0x1
var tcp.syn +13(25)&0x1
var tcp.rst +13(26)&0x1
var tcp.psh +13(27)&0x1
var tcp.ack +13(28)&0x1
var tcp.urg +13(29)&0x1
var tcp.live +127
```

```
# Define the UDP packet header fields.
var udp.source +0(16)&0xffff
var udp.dest +2(16)&0xffff
var udp.len +4(16)&0xffff
var udp.check +6(16)&0xffff
# Define the ICMP packet header fields.
var icmp.type +0(24)&0xff
var icmp.code +1(24)&0xff
var icmp.checksum +2(16)&0xffff
var icmp.echo.id +4(16)&0xffff
var icmp.echo.sequence +6(16)&0xffff
var icmp.gateway +4
```

My setup

- 486/66 with 20 MB ram and plenty of disk space :-)
- 16550 UARTs a plenty
- diald 0.11, pppd 2.2.0d, kernel 1.3.95 modularized.

Notes

- Make sure you change the /dev/ttyS1 above to match your requirements.
- Ensure that you can actually connect up with PPP by using the blind.ppp script.

Questions? Email Harish Pillay -> h.pillay@ieee.org

Go back to the Harish's Home Page

Last updated June 3rd 1996

Linux Ext2fs Undeletion mini-HOWTO

Aaron Crane, aaronc@pobox.com

v1.3, 2 February 1999

Picture this. You've spent the last three days with no sleep, no food, not even a shower. Your hacking compulsion has at last paid off: you've finished that program that will bring you world-wide fame and recognition. All that you still need to do is tar it up and put it on Metalab. Oh, and delete all those Emacs backup files. So you say `rm * ~`*. And too late, you notice the extra space in that command. You've just deleted your magnum opus! But help is at hand. This document presents a discussion of how to retrieve deleted files from a Second Extended File System. Just maybe, you'll be able to release that program after all...*

1. Introduction

- 1.1 Revision history
- 1.2 Canonical locations of this document

2. How not to delete files

3. What recovery rate can I expect?

4. So, how do I undelete a file?

5. Unmounting the file system

6. Preparing to change inodes directly

7. Preparing to write data elsewhere

8. Finding the deleted inodes

9. Obtaining the details of the inodes

10. Recovering data blocks

- 10.1 Short files
- 10.2 Longer files

11. Modifying inodes directly

12. Will this get easier in future?

13. Are there any tools to automate this process?

14. Colophon

15. Credits and Bibliography

16. Legalities

1. Introduction

This mini-Howto attempts to provide hints on how to retrieve deleted files from an ext2 file system. It also contains a limited amount of discussion of how to avoid deleting files in the first place.

I intend it to be useful certainly for people who have just had, shall we say, a little accident with `rm`; however, I also hope that people read it anyway. You never know: one day, some of the information in here could save your bacon.

The text assumes a little background knowledge about UNIX file systems in general; however, I hope that it will be accessible to most Linux users. If you are an outright beginner, I'm afraid that undeleting files under Linux *does* require a certain amount of technical knowledge and persistence, at least for the time being.

You will be unable to recover deleted files from an ext2 file system without at least read access to the raw device on which the file was stored. In general, this means that you must be root, but some distributions (such as Debian GNU/Linux) provide a `disk` group whose members have access to such devices. You also need `debugfs` from the `e2fsprogs` package. This should have been installed by your distribution.

Why have I written this? It stems largely from my own experiences with a particularly foolish and disastrous `rm -r` command as root. I deleted about 97 JPEG files which I needed and could almost certainly not recover from other sources. Using some helpful tips (see section Credits and Bibliography) and a great deal of persistence, I recovered 91 files undamaged. I managed to retrieve at least parts of five of the rest (enough to see what the picture was in each case). Only one was undisplayable, and even for this one, I am fairly sure that no more than 1024 bytes were lost (though unfortunately from the beginning of the file; given that I know nothing about the JFIF file format I had done as much as I could).

I shall discuss further below what sort of recovery rate you can expect for deleted files.

1.1 Revision history

The various publicly-released revisions of this document (and their publication dates) are as follows:

- v1.0 on 18 January 1997
- v1.1 on 23 July 1997 (see section Changes in version 1.1)
- v1.2 on 4 August 1997 (see section Changes in version 1.2)
- v1.3 on 2 February 1999 (see section Changes in version 1.3)

TOC

Changes in version 1.1

What changes have been made in this version? First of all, the thinko in the example of file recovery has been fixed. Thankyou to all those who wrote to point out my mistaek; I hope I've learned to be more careful when making up program interaction.

Secondly, the discussion of UNIX file system layout has been rewritten to be, I hope, more understandable. I wasn't entirely happy with it in the first place, and some people's comments indicated that it wasn't clear.

Thirdly, the vast uuencoded gzipped tarball of `fsgrab` in the middle of the file has been removed. The program is now available on my website and on Metalab (and mirrors).

Fourthly, the document has been translated into the Linux Documentation Project SGML Tools content markup language. This markup language can be easily converted to any of a number of other markup languages (including HTML and LaTeX) for convenient display and printing. One benefit of this is that beautiful typography in paper editions is a much more achievable goal; another is that the document has cross-references and hyperlinks when viewed on the Web.

Changes in version 1.2

This revision is very much an incremental change. It's here mainly to include changes suggested by readers, one of which is particularly important.

The first change was suggested by Egil Kvaleberg `egil@kvaleberg.no`, who pointed out the `dump` command in `debugfs`. Thanks again, Egil.

The second change is to mention the use of `chattr` for avoiding deleting important files. Thanks to Herman Suijs `H.P.M.Suijs@kub.nl` for mentioning this one.

The abstract has been revised. URLs have been added for organisations and software. Various other minor changes have been made (including fixing typos and so on).

Changes in version 1.3

Though it is the first release in 17 months, there is very little that is new here. This release merely fixes a few minor errors (typos, dangling URLs, that sort of thing -- especially the non-link to the Open Group), and updates a few parts of the text that have become hopelessly out-of-date, such as the material on kernel versions and on `lde`. Oh, and I've changed `Sunsite' to `Metalab' throughout.

This release is anticipated to be the last one before release 2.0, which will hopefully be a full Howto. I have been working on some substantial changes which will justify an increment of the major version number.

1.2 Canonical locations of this document

TOC

The latest public release of this document should always be available in on the Linux Documentation Project site (and mirrors).

The latest release is also kept on my website in several formats:

- SGML source. This is the source as I have written it, using the SGML Tools package.
- HTML. This is HTML, automatically generated from the SGML source.
- Plain text. This is plain text, which is also automatically generated from the SGML source.

---

2. How not to delete files

It is vital to remember that Linux is unlike MS-DOS when it comes to undeletion. For MS-DOS (and its bastard progeny Windows 95), it is generally fairly straightforward to undelete a file - the `operating system' (I use the term loosely) even comes with a utility which automates much of the process. For Linux, this is not the case.

So. Rule number one (the prime directive, if you will) is:

KEEP BACKUPS

no matter what. Think of all your data. Perhaps, like me, you keep several years' of accumulated email, contacts, programs, papers on your computer. Think of how your life would be turned upside down if you had a catastrophic disk failure, or if -- heaven forbid! -- a malicious cracker wiped your disks. This is not unlikely; I have corresponded with a number of people in just such a situation. I exhort all right-thinking Linux users to go out and buy a useful backup device, work out a decent backup schedule, and to *stick to it*. Myself, I use a spare hard disk on a second machine, and periodically mirror my home directory onto it over the ethernet. For more information on planning a backup schedule, read Frisch (1995) (see section Bibliography and Credits).

In the absence of backups, what then? (Or even in the presence of backups: belt and braces is no bad policy where important data is concerned.)

Try to set the permissions for important files to 440 (or less): denying yourself write access to them means that `rm` requires an explicit confirmation before deleting. (I find, however, that if I'm recursively deleting a directory with `rm -r`, I'll interrupt the program on the first or second confirmation request and reissue the command as `rm -rf`.)

A good trick for selected files is to create a hard link to them in a hidden directory. I heard a story once about a sysadmin who repeatedly deleted `/etc/passwd` by accident (thereby half-destroying the system). One of the fixes for this was to do something like the following (as root):

```
# mkdir /.backup
# ln /etc/passwd /.backup
```

It requires quite some effort to delete the file contents completely: if you say

```
# rm /etc/passwd
```

then

```
# ln /.backup/passwd /etc
```

will retrieve it. Of course, this does not help in the event that you overwrite the file, so keep backups anyway.

On an ext2 file system, it is possible to use ext2 attributes to protect things. These attributes are manipulated with the `chattr` command. There is an `append-only' attribute: a file with this attribute may be appended to, but may not be deleted, and the existing contents of the file may not be overwritten. If a directory has this attribute, any files or directories within it may be modified as normal, but no files may be deleted. The `append-only' attribute is set with

```
$ chattr +a FILE...
```

There is also an `immutable' attribute, which can only be set or cleared by root. A file or directory with this attribute may not be modified, deleted, renamed, or (hard) linked. It may be set as follows:

```
# chattr +i FILE...
```

The ext2fs also provides the `undeletable' attribute (`+u` in `chattr`). The intention is that if a file with that attribute is deleted, instead of actually being reused, it is merely moved to a `safe location' for deletion at a later date. Unfortunately this feature has not yet been implemented in mainstream kernels; and though in the past there has been some interest in implementing it, it is not (to my knowledge) available for any current kernels.

Some people advocate making `rm` a shell alias or function for `rm -i` (which asks for confirmation on *every* file you delete). Indeed, the Red Hat distribution does this by default for all users, including root. Personally, I cannot stand software which won't run unattended, so I don't do that. There is also the problem that sooner or later, you'll be running in single-user mode, or using a different shell, or even a different machine, where your `rm` function doesn't exist. If you expect to be asked for confirmation, it is easy to forget where you are and to specify too many files for deletion. Likewise, the various scripts and programs that replace `rm` are, IMHO, very dangerous.

A slightly better solution is to start using a package which handles `recyclable' deletion by providing a command not named `rm`. For details on these, see Peek, et al (1993) (see section Bibliography and Credits). These however still suffer from the problem that they tend to encourage the user to have a nonchalant attitude to deletion, rather than the cautious approach that is often required on Unix systems.

---

3. What recovery rate can I expect?

That depends. Among the problems with recovering files on a high-quality, multi-tasking, multi-user operating system like Linux is that you never know when someone wants to write to the disk. So when the operating system is told to delete a file, it assumes that the blocks used by that file are fair game when it wants to allocate space for a new file. (This is a specific example of a general principle for Unix-like systems: the kernel and the associated tools assume that the users aren't idiots.) In general, the more usage your machine gets, the less likely you are to be able to recover files successfully.

Also, disk fragmentation can affect the ease of recovering files. If the partition containing the deleted files is very fragmented, you are unlikely to be able to read a whole file.

If your machine, like mine, is effectively a single-user workstation, and you weren't doing anything disk-intensive at the fatal moment of deleting those files, I would expect a recovery rate in the same ball-park as detailed above. I retrieved nearly 94% of the files (and these were binary files, please note) undamaged. If you get 80% or better, you can feel pretty pleased with yourself, I should think.

4. So, how do I undelete a file?

The procedure principally involves finding the data on the raw partition device and making it visible again to the operating system. There are basically two ways of doing this: one is to modify the existing file system such that the deleted inodes have their `deleted' flag removed, and hope that the data just magically falls back into place. The other method, which is safer but slower, is to work out where the data lies in the partition and write it out into a new file on another file system.

There are some steps you need to take before beginning to attempt your data recovery; see sections Unmounting the file system, Preparing to change inodes directly and Preparing to write data elsewhere for details. To find out how to actually retrieve your files, see sections Finding the deleted inodes, Obtaining the details of the inodes, Recovering data blocks and Modifying inodes directly.

5. Unmounting the file system

Regardless of which method you choose, the first step is to unmount the file system containing the deleted files. I strongly discourage any urges you may have to mess around on a mounted file system. This step should be performed *as soon as possible* after you realise that the files have been deleted; the sooner you can unmount, the smaller the chance that your data will be overwritten.

The simplest method is as follows: assuming the deleted files were in the `/usr` file system, say:

```
# umount /usr
```

You may, however, want to keep some things in `/usr` available. So remount it read-only:

```
# mount -o ro,remount /usr
```

If the deleted files were on the root partition, you'll need to add a `-n` option to prevent mount from trying to write to `/etc/mtab`:

```
# mount -n -o ro,remount /
```

Regardless of all this, it is possible that there will be another process using that file system (which will cause the unmount to fail with an error such as `Resource busy'). There is a program which will send a signal to any process using a given file or mount point: `fuser`. Try this for the `/usr` partition:

```
# fuser -v -m /usr
```

This lists the processes involved. Assuming none of them are vital, you can say

```
# fuser -k -v -m /usr
```

to send each process a `SIGKILL` (which is guaranteed to kill it), or for example,

```
# fuser -k -TERM -v -m /usr
```

to give each one a `SIGTERM` (which will normally make the process exit cleanly).

6. Preparing to change inodes directly

My advice? Don't do it this way. I really don't think it's wise to play with a file system at a low enough level for this to work. This method also has problems in that you can only reliably recover the first 12 blocks of each file. So if you have any long files to recover, you'll normally have to use the other method anyway. (Although see section Will this get easier in future? for additional information.)

If you feel you must do it this way, my advice is to copy the raw partition data to an image on a different partition, and then mount this using loopback:

```
# cp /dev/hda5 /root/working
# mount -t ext2 -o loop /root/working /mnt
```

(Note that obsolete versions of `mount` may have problems with this. If your `mount` doesn't work, I strongly suggest you get the latest version, or at least version 2.7, as some very old versions have severe security bugs.)

Using loopback means that if and when you completely destroy the file system, all you have to do is copy the raw partition back and start over.

7. Preparing to write data elsewhere

If you chose to go this route, you need to make sure you have a rescue partition somewhere -- a place to write out new copies of the files you recover. Hopefully, your system has several partitions on it: perhaps a root, a `/usr`, and a `/home`. With all these to choose from, you should have no problem: just create a new directory on one of these.

If you have only a root partition, and store everything on that, things are slightly more awkward. Perhaps you have an MS-DOS or Windows partition you could use? Or you have the ramdisk driver in your kernel, maybe as a module? To use the ramdisk (assuming a kernel more recent than 1.3.48), say the following:

```
# dd if=/dev/zero of=/dev/ram0 bs=1k count=2048
# mke2fs -v -m 0 /dev/ram0 2048
# mount -t ext2 /dev/ram0 /mnt
```

This creates a 2MB ramdisk volume, and mounts it on `/mnt`.

A short word of warning: if you use `kerneld` (or its replacement `kmod` in 2.2.x and later 2.1.x kernels) to automatically load and unload kernel modules, then don't unmount the ramdisk until you've copied any files from it onto non-volatile storage. Once you unmount it, `kerneld` assumes it can unload the module (after the usual waiting period), and once this happens, the memory gets re-used by other parts of the kernel, losing all the painstaking hours you just spent recovering your data.

If you have a Zip, Jaz, or LS-120 drive, or something similar, it would probably be a good choice for a rescue partition location. Otherwise, you'll just have to stick with floppies.

The other thing you're likely to need is a program which can read the necessary data from the middle of the partition device. At a pinch, `dd` will do the job, but to read from, say, 600 MB into an 800 MB partition, `dd` insists on reading but ignoring the first 600 MB. This takes a not inconsiderable amount of time, even on fast disks. My way round this was to write a program which will seek to the middle of the partition. It's called `fsgrab`; you can find the source package on my website or on Metalab (and mirrors). If you want to use this method, the rest of this mini-Howto assumes that you have `fsgrab`.

If none of the files you are trying to recover were more than 12 blocks long (where a block is usually one kilobyte), then you won't need `fsgrab`.

If you need to use `fsgrab` but don't want to download and build it, it is fairly straightforward to translate an `fsgrab` command-line to one for `dd`. If we have

```
fsgrab -c count -s skip device
```

then the corresponding (but typically much slower) `dd` command is

```
dd bs=1k if=device count=count skip=skip
```

I must warn you that, although `fsgrab` functioned perfectly for me, I can take no responsibility for it performs. It was really a very quick and dirty kludge just to get things to work. For more details on lack of warranty, see the `No Warranty' section in the `COPYING` file included with it (the GNU General Public Licence).

---

8. Finding the deleted inodes

The next step is to ask the file system which inodes have recently been freed. This is a task you can accomplish with `debugfs`. Start `debugfs` with the name of the device on which the file system is stored:

```
# debugfs /dev/hda5
```

If you want to modify the inodes directly, add a `-w` option to enable writing to the file system:

```
# debugfs -w /dev/hda5
```

The `debugfs` command to find the deleted inodes is `lsdel`. So, type the command at the prompt:

```
debugfs:  lsdel
```

After much wailing and grinding of disk mechanisms, a long list is piped into your favourite pager (the value of `$PAGER`). Now you'll want to save a copy of this somewhere else. If you have `less`, you can type `-o` followed by the name of an output file. Otherwise, you'll have to arrange to send the output elsewhere. Try this:

```
debugfs:  quit
# echo lsdel | debugfs /dev/hda5 > lsdel.out
```

Now, based only on the deletion time, the size, the type, and the numerical permissions and owner, you must work out which of these deleted inodes are the ones you want. With luck, you'll be able to spot them because they're the big bunch you deleted about five minutes ago. Otherwise, trawl through that list carefully.

I suggest that if possible, you print out the list of the inodes you want to recover. It will make life a lot easier.

9. Obtaining the details of the inodes

`debugfs` has a `stat` command which prints details about an inode. Issue the command for each inode in your recovery list. For example, if you're interested in inode number 148003, try this:

```
debugfs:  stat <148003>
Inode: 148003   Type: regular    Mode:  0644   Flags: 0x0   Version: 1
User:   503   Group:   100   Size: 6065
File ACL: 0    Directory ACL: 0
Links: 0   Blockcount: 12
Fragment:  Address: 0    Number: 0    Size: 0
ctime: 0x31a9a574 -- Mon May 27 13:52:04 1996
atime: 0x31a21dd1 -- Tue May 21 20:47:29 1996
mtime: 0x313bf4d7 -- Tue Mar  5 08:01:27 1996
dtime: 0x31a9a574 -- Mon May 27 13:52:04 1996
BLOCKS:
594810 594811 594814 594815 594816 594817
TOTAL: 6
```

If you have a lot of files to recover, you'll want to automate this. Assuming that your `lsdel` list of inodes to recover in is in `lsdel.out`, try this:

```
# cut -c1-6 lsdel.out | grep "[0-9]" | tr -d " " > inodes
```

This new file `inodes` contains just the numbers of the inodes to recover, one per line. We save it because it will very likely come in handy later on. Then you just say:

```
# sed 's/^.*$/stat <\0>/' inodes | debugfs /dev/hda5 > stats
```

and `stats` contains the output of all the `stat` commands.

10. Recovering data blocks

This part is either very easy or distinctly less so, depending on whether the file you are trying to recover is more than 12 blocks long.

10.1 Short files

If the file was no more than 12 blocks long, then the block numbers of all its data are stored in the inode: you can read them directly out of the `stat` output for the inode. Moreover, `debugfs` has a command which performs this task automatically. To take the example we had before, repeated here:

```
debugfs:  stat <148003>
Inode: 148003   Type: regular    Mode:  0644   Flags: 0x0   Version: 1
User:   503   Group:   100   Size: 6065
File ACL: 0    Directory ACL: 0
Links: 0   Blockcount: 12
Fragment:  Address: 0    Number: 0    Size: 0
ctime: 0x31a9a574 -- Mon May 27 13:52:04 1996
atime: 0x31a21dd1 -- Tue May 21 20:47:29 1996
mtime: 0x313bf4d7 -- Tue Mar  5 08:01:27 1996
dtime: 0x31a9a574 -- Mon May 27 13:52:04 1996
BLOCKS:
594810 594811 594814 594815 594816 594817
TOTAL: 6
```

This file has six blocks. Since this is less than the limit of 12, we get `debugfs` to write the file into a new location, such as `/mnt/recovered.000`:

```
debugfs:  dump <148003> /mnt/recovered.000
```

Of course, this can also be done with `fsgrab`; I'll present it here as an example of using it:

```
# fsgrab -c 2 -s 594810 /dev/hda5 > /mnt/recovered.000
# fsgrab -c 4 -s 594814 /dev/hda5 >> /mnt/recovered.000
```

With either `debugfs` or `fsgrab`, there will be some garbage at the end of `/mnt/recovered.000`, but that's fairly unimportant. If you want to get rid of it, the simplest method is to take the `Size` field from the inode, and plug it into the `bs` option in a `dd` command line:

```
# dd count=1 if=/mnt/recovered.000 of=/mnt/resized.000 bs=6065
```

Of course, it is possible that one or more of the blocks that made up your file has been overwritten. If so, then you're out of luck: that block is gone forever. (But just imagine if you'd unmounted sooner!)

10.2 Longer files

TOC

The problems appear when the file has more than 12 data blocks. It pays here to know a little of how UNIX file systems are structured. The file's data is stored in units called `blocks'. These blocks may be numbered sequentially. A file also has an `inode', which is the place where information such as owner, permissions, and type are kept. Like blocks, inodes are numbered sequentially, although they have a different sequence. A directory entry consists of the name of the file and an inode number.

But with this state of affairs, it is still impossible for the kernel to find the data corresponding to a directory entry. So the inode also stores the location of the file's data blocks, as follows:

- The block numbers of the first 12 data blocks are stored directly in the inode; these are sometimes referred to as the *direct block*s.
- The inode contains the block number of an *indirect block*. An indirect block contains the block numbers of 256 additional data blocks.
- The inode contains the block number of a *doubly indirect block*. A doubly indirect block contains the block numbers of 256 additional indirect blocks.
- The inode contains the block number of a *triply indirect block*. A triply indirect block contains the block numbers of 256 additional doubly indirect blocks.

Read that again: I know it's complex, but it's also important.

Now, the kernel implementation for all versions up to and including 2.0.36 unfortunately zeroes all indirect blocks (and doubly indirect blocks, and so on) when deleting a file. So if your file was longer than 12 blocks, you have no guarantee of being able to find even the numbers of all the blocks you need, let alone their contents.

The only method I have been able to find thus far is to assume that the file was not fragmented: if it was, then you're in trouble. Assuming that the file was not fragmented, there are several layouts of data blocks, according to how many data blocks the file used:

0 to 12

The block numbers are stored in the inode, as described above.

13 to 268

After the direct blocks, count one for the indirect block, and then there are 256 data blocks.

269 to 65804

As before, there are 12 direct blocks, a (useless) indirect block, and 256 blocks. These are followed by one (useless) doubly indirect block, and 256 repetitions of one (useless) indirect block and 256 data blocks.

65805 or more

The layout of the first 65804 blocks is as above. Then follow one (useless) triply indirect block and 256 repetitions of a `doubly indirect sequence'. Each doubly indirect sequence consists of a (useless) doubly indirect block, followed by 256 repetitions of one (useless) indirect block and 256 data blocks.

Of course, even if these assumed data block numbers are correct, there is no guarantee that the data in them is intact. In addition, the longer the file was, the less chance there is that it was written to the file system without appreciable fragmentation (except in special circumstances).

You should note that I assume throughout that your blocksize is 1024 bytes, as this is the standard value. If your blocks are bigger, some of the numbers above will change. Specifically: since each block number is 4 bytes long, blocksize/4 is the number of block numbers that can be stored in each indirect block. So every time the number 256 appears in the discussion above, replace it with blocksize/4. The `number of blocks required' boundaries will also have to be changed.

Let's look at an example of recovering a longer file.

```
debugfs:  stat <1387>
Inode: 148004   Type: regular    Mode:  0644   Flags: 0x0   Version: 1
User:   503   Group:   100   Size: 1851347
File ACL: 0    Directory ACL: 0
Links: 0   Blockcount: 3616
Fragment:  Address: 0    Number: 0    Size: 0
ctime: 0x31a9a574 -- Mon May 27 13:52:04 1996
atime: 0x31a21dd1 -- Tue May 21 20:47:29 1996
mtime: 0x313bf4d7 -- Tue Mar  5 08:01:27 1996
dtime: 0x31a9a574 -- Mon May 27 13:52:04 1996
BLOCKS:
8314 8315 8316 8317 8318 8319 8320 8321 8322 8323 8324 8325 8326 8583
TOTAL: 14
```

There seems to be a reasonable chance that this file is not fragmented: certainly, the first 12 blocks listed in the inode (which are all data blocks) are contiguous. So, we can start by retrieving those blocks:

```
# fsgrab -c 12 -s 8314 /dev/hda5 > /mnt/recovered.001
```

Now, the next block listed in the inode, 8326, is an indirect block, which we can ignore. But we trust that it will be followed by 256 data blocks (numbers 8327 through 8582).

```
# fsgrab -c 256 -s 8327 /dev/hda5 >> /mnt/recovered.001
```

The final block listed in the inode is 8583. Note that we're still looking good in terms of the file being contiguous: the last data block we wrote out was number 8582, which is 8327 + 255. This block 8583 is a doubly indirect block, which we can ignore. It is followed by up to 256 repetitions of an indirect block (which is ignored) followed by 256 data blocks. So doing the arithmetic quickly, we issue the following commands. Notice that we skip the doubly indirect block 8583, and the indirect block 8584 immediately (we hope) following it, and start at block 8585 for data.

```
# fsgrab -c 256 -s 8585 /dev/hda5 >> /mnt/recovered.001
# fsgrab -c 256 -s 8842 /dev/hda5 >> /mnt/recovered.001
# fsgrab -c 256 -s 9099 /dev/hda5 >> /mnt/recovered.001
# fsgrab -c 256 -s 9356 /dev/hda5 >> /mnt/recovered.001
# fsgrab -c 256 -s 9613 /dev/hda5 >> /mnt/recovered.001
# fsgrab -c 256 -s 9870 /dev/hda5 >> /mnt/recovered.001
```

Adding up, we see that so far we've written 12 + (7 * 256) blocks, which is 1804. The `stat' results for the inode gave us a `blockcount' of 3616; unfortunately these blocks are 512 bytes long (as a hangover from

TOC

UNIX), so we really want 3616/2 = 1808 blocks of 1024 bytes. That means we need only four more bloc
The last data block written was number 10125. As we've been doing so far, we skip an indirect block (nu
10126); we can then write those last four blocks.

```
# fsgrab -c 4 -s 10127 /dev/hda5 >> /mnt/recovered.001
```

Now, with some luck the entire file has been recovered successfully.

---

11. Modifying inodes directly

This method is, on the surface, much easier. However, as mentioned above, it cannot yet cope with files longer than 12 blocks.

For each inode you want to recover, you must set the usage count to one, and set the deletion time to zero. This is done with the `mi` (modify inode) command in `debugfs`. Some sample output, modifying inode 148003 from above:

```
debugfs:  mi <148003>
                          Mode    [0100644]
                       User ID    [503]
                      Group ID    [100]
                          Size    [6065]
                 Creation time    [833201524]
            Modification time     [832708049]
                  Access time     [826012887]
                 Deletion time    [833201524] 0
                    Link count    [0] 1
                   Block count    [12]
                    File flags    [0x0]
                     Reserved1    [0]
                      File acl    [0]
                 Directory acl    [0]
             Fragment address     [0]
              Fragment number     [0]
                Fragment size     [0]
              Direct Block #0     [594810]
              Direct Block #1     [594811]
              Direct Block #2     [594814]
              Direct Block #3     [594815]
              Direct Block #4     [594816]
              Direct Block #5     [594817]
              Direct Block #6     [0]
              Direct Block #7     [0]
              Direct Block #8     [0]
              Direct Block #9     [0]
             Direct Block #10     [0]
             Direct Block #11     [0]
               Indirect Block     [0]
        Double Indirect Block     [0]
        Triple Indirect Block     [0]
```

That is, I set the deletion time to 0 and the link count to 1 and just pressed return for each of the other fields. Granted, this is a little unwieldy if you have a lot of files to recover, but I think you can cope. If you'd wanted chrome, you'd have used a graphical `operating system' with a pretty `Recycle Bin'.

By the way: the `mi` output refers to a `Creation time' field in the inode. This is a lie! (Or misleading, anyway.) The fact of the matter is that you cannot tell on a UNIX file system when a file was created. The `st_ctime` member of a `struct stat` refers to the `inode change time', that is, the last time when any inode details were changed. Here endeth today's lesson.

Note that more recent versions of `debugfs` than the one I'm using probably do not include some of the fields in the listing above (specifically, `Reserved1` and (some of?) the fragment fields).

Once you've modified the inodes, you can quit `debugfs` and say:

```
# e2fsck -f /dev/hda5
```

The idea is that each of the deleted files has been literally undeleted, but none of them appear in any directory entries. The `e2fsck` program can detect this, and will add a directory entry for each file in the `/lost+found` directory of the file system. (So if the partition is normally mounted on `/usr`, the files will now appear in `/usr/lost+found` when you next mount it.) All that still remains to be done is to work out the name of each file from its contents, and return it to its correct place in the file system tree.

When you run `e2fsck`, you will get some informative output, and some questions about what damage to repair. Answer `yes' to everything that refers to `summary information' or to the inodes you've changed. Anything else I leave up to you, although it's usually a good idea to say `yes' to all the questions. When `e2fsck` finishes, you can remount the file system.

Actually, there's an alternative to having `e2fsck` leave the files in `/lost+found`: you can use `debugfs` to create a link in the file system to the inode. Use the `link` command in `debugfs` after you've modified the inode:

```
debugfs:  link <148003> foo.txt
```

This creates a file called `foo.txt` in what `debugfs` thinks is the current directory; `foo.txt` will be your file. You'll still need to run `e2fsck` to fix the summary information and block counts and so on.

---

12. Will this get easier in future?

Yes. In fact, I believe it already has. Although as of this writing, current stable kernels (in the 2.0.x series) zero indirect blocks, this does not apply to development kernels in the 2.1.x series, nor to the stable 2.2.x series. As I write this on 2 February 1999, kernel 2.2.1 was released a few days ago; Linux vendors are likely to start producing distributions containing and supporting 2.2.x kernels a month or two from now.

Once the indirect-zeroing limitation has been overcome in the production kernels, a lot of my objections to the technique of modifying inodes by hand will disappear. At the same time, it will also become possible to use the `dump` command in `debugfs` on long files, and to conveniently use other undeletion tools.

13. Are there any tools to automate this process?

As it happens, there are. Unfortunately, I believe that they currently suffer from the same problem as the manual inode modification technique: indirect blocks are unrecoverable. However, given the likelihood that this will shortly no longer be a problem, it's well worth looking these programs out now.

I have written a tool called `e2recover`, which is essentially a Perl wrapper around `fsgrab`. It makes a reasonable amount of effort to deal with zeroed indirect blocks, and seems to work fairly well as long as there was no fragmentation. It also correctly sets the permissions (and when possible the ownership) of recovered files, and even makes sure that recovered files have the correct length.

I originally wrote `e2recover` for the forthcoming major update to this Howto; unfortunately this means that much of the useful documentation for `e2recover` is scheduled for inclusion in that update. Be that as it may, it should be useful now; it can be downloaded from my web site, and soon from Metalab.

Scott D. Heavner is the author of `lde`, the Linux Disk Editor. It can be used as both a binary disk editor, and as an equivalent to `debugfs` for ext2 and minix file systems, and even for xia file systems (though xia support is no longer available in 2.1.x and 2.2.x kernels). It has some features for assisting undeletion, both by walking the block list for a file, and by grepping through disk contents. It also has some fairly useful documentation on basic file system concepts, as well as a document on how to use it for undeletion. Version 2.4 of `lde` is available on Metalab and mirrors, or on the author's web site.

Another possibility is offered by the GNU Midnight Commander, `mc`. This is a full-screen file management tool, based AFAIK on a certain MS-DOS program commonly known as `NC'. `mc` supports the mouse on the Linux console and in an xterm, and provides virtual file systems which allow tricks like `cd`-ing to a tarfile. Among its virtual file systems is one for ext2 undeletion. It all sounds very handy, although I must admit I don't use the program myself -- I prefer good old-fashioned shell commands.

To use the undeletion feature, you have to configure the program with the `--with-ext2undel` option; you'll also need the development libraries and include files that come with the `e2fsprogs` package. The version provided in Debian GNU/Linux is built in this way; the same may apply to packages for other Linux distributions. Once the program is built, you can tell it to `cd undel:/dev/hda5`, and get a `directory listing' of deleted files. Like many current undeletion tools, it handles zeroed indirect blocks poorly -- it typically just recovers the first 12k of long files.

The current version may be downloaded from the Midnight Commander ftp site.

14. Colophon

I intend to produce regular updates to this document as long as I have both enough time to do it, and something interesting to say. This means that I am eager to hear comments from readers. Could my writing be clearer? Can you think of something that would make matters easier? Is there some new tool that does it all automatically? Whatever. If you have something to say about this document or about the `fsgrab` or `e2recover` tools, drop me a line on aaronc@pobox.com.

15. Credits and Bibliography

> `If I have seen farther than others, it is because I was standing on the shoulders of giants.' (Isaac Newton)

This mini-Howto was originally derived from a posting in the `comp.os.linux.misc` newsgroup by Robin Glover `swrglovr@met.rdg.ac.uk`. I would like to thank Robin for graciously allowing me to rework his ideas into this mini-Howto.

I would also like to take this opportunity to thank once again all the people who've written to me about the Howto. Receiving grateful comments makes the effort worth while.

Some bibliographic references:

- **Frisch**, Æleen (1995), *Essential System Administration*, second edition, O'Reilly and Associates, Inc., ISBN: 1-56592-127-5.
- **Garfinkel**, Simson, Daniel **Weise** and Steven **Strassmann** (1994), *The Unix-Haters Handbook*, IDG Books, ISBN: 1-56884-203-1. Much of this book is merely the adolescent whinings of people who think that *their* operating system was so much better than Unix, and much of the rest simply doesn't apply if you have a well-written user-space such as GNU. But there is some wheat among the chaff; for example, the discussion of how easy it is to delete files under Unix is well worth reading.
- **Glover**, Robin (31 Jan 1996), *HOW-TO : undelete linux files (ext2fs/debugfs)*, comp.os.linux.misc Usenet posting.
- **Peek**, Jerry, Tim **O'Reilly**, Mike **Loukides** et al (1993), *UNIX Power Tools*, O'Reilly and Associates, Inc./Random House, Inc., ISBN: 0-679-79073-X. Second edition, 1998.

16. Legalities

All trademarks are the property of their respective owners. Specifically:

- *MS-DOS* and *Windows* are trademarks of Microsoft.
- *UNIX* is a trademark of the Open Group.
- *Linux* is a trademark of Linus Torvalds in the USA and some other countries.

This document is Copyright © 1997, 1999 Aaron Crane aaronc@pobox.com. It may be freely redistributed in its entirety, including the whole of this copyright notice, but may not be changed without permission from either the author or the Linux Documentation Project HOWTO Coordinator. Dispensation is granted for copying small verbatim portions for the purposes of reviews or for quoting; in these circumstances, sections may be reproduced in the presence of an appropriate citation but without this copyright notice.

The author requests but does not require that parties intending to sell copies of this document, whether on computer-readable or human-readable media, inform either him or the Linux HOWTO Coordinator of their intentions.

The Linux HOWTO Coordinator is currently Tim Bynum linux-howto@metalab.unc.edu.

Linux simple fax printer server mini-HOWTO (faxsrv-mini-HOWTO)

Erez Strauss <erez@newplaces.com>

v1.0, 8 November 1997

No warranties. *Comments are always welcome. This document describes in details one of the simplest ways to setup fax server on your Linux system. The fax is available to the users on the local host and to network users.*

1. Introduction

2. Questions & Answers

- 2.1 What is a fax printer server ?
- 2.2 How to set it up ?
- 2.3 How do I use it, for the server ?
- 2.4 Where do I specify the target Fax number ?
- 2.5 How do I use it from other Unixes on the net ?
- 2.6 Caldera, LPRng users

3. Latest version, Contacting the author.

1. Introduction

To get a working printer/fax on the linux machine you should have some software and fax modem.

The Printing software will be using the fax software as a print filter. The efax will find the fax number from the print Job number and will send the fax. The efax software was written be Ed Casas <edc@cce.com>.

The rest of the document is build as Q&A.

2. Questions & Answers

2.1 What is a fax printer server ?

A fax printer server is a setup of few programs: efax, and the print server, in such a way that sending a fax from the computer is as simple as sending printout to a printer.

2.2 How to set it up ?

Setting efax as fax print server includes few problems As I worked it out few times, I decided to collect this wisdom in this small mini-HOWTO, comments are welcome at `<erez@newplaces.com>`. I describe them here and the solutions, and all the instruction in short steps:

The efax Software

make sure you have the efax package.

On RPM based system use the command 'rpm -qv efax'.

You can get the efax sources in tar.gz format from sunsite:
ftp://sunsite.unc.edu/pub/Linux/apps/serialcomm/fax/efax08a.tar.gz or binary rpm package:
ftp://ftp.redhat.com/pub/redhat/redhat-4.2/i386/RedHat/RPMS/efax-0.8a-3.i386.rpm

The printcap entry.

The efax documentation is missing the : at the end of the printcap entry.

Solution: Add the following `/etc/printcap` entry:

```
fax:\
        :lp=/dev/null:\
        :sd=/var/spool/fax:\
        :if=/usr/bin/faxlpr:
```

The fax command

Use hard link and not symbolic link to the fax command.

Run the command:

```
ln /usr/bin/fax /usr/bin/faxlpr
```

TOC

The /usr/bin/fax file.

Edit the /usr/bin/fax to your preferences:

choose the right Fax Class for your modem:

```
CLASS=2.0
```

Set the FROM="your international phone number" field

```
NAME="Your Name"
```

Add the line for conversion of international phone number to local for example in Israel you would use:

```
TELCVT='sed -e s/+972/0/ -e s/+/00/'  # Israel
```

There is a problem in the file in line 586, change the `cfile=...` with the following two lines:

```
cfile=`/usr/bin/tail -1 lock`
cfile=`cat $cfile`
```

at lines 586,587 there shouldn't be '-' signs the lines are:

```
0) echo "$l" | mail -s "fax to $num succeeded" $user@$host ;;
*) echo "$l" | mail -s "fax to $num failed   " $user@$host ;;
```

The /var/spool/fax directory.

create the directory /var/spool/fax

```
mkdir /var/spool/fax
chmod 777 /var/spool/fax
```

The lock file.

The lock file (`/var/spool/fax/lock`) is being created with incorrect permissions, use the following command to set it correctly:

```
touch /var/spool/fax/lock ; chmod 644 /var/spool/fax/lock
```

The /dev/modem special file.

The efax program is expecting real device file at the `/dev/modem` and will not work with symbolic link, so create a device file with the same major and minor number as the `/dev/cua?` the is connected to the modem. The file should have the rw-rw-rw- mode to enable any user to use the fax software.

```
ls -lL /dev/modem
rm /dev/modem
mknod /dev/modem c Mj Mi
chmod 666 /dev/modem
```

Mj is 5, and Mi is 64 for the cua0, 65 for cua1 and so on. For example (for cua1):

```
mknod /dev/modem c 5 65
```

The `/var/lock` directory.

change the mode at the `/var/lock` directory

```
chmod 1777 /var/lock
```

2.3 How do I use it, for the server ?

You should use the fax printer using the -P option and the -J option with the fax number.

use one of the following lpr commands:

```
lpr -Pfax -J <Fax-Number> [file-names]
any command | lpr -Pfax -J <Fax-Number>
```

2.4 Where do I specify the target Fax number ?

After the -J option.

2.5 How do I use it from other Unixes on the net ?

Add the following entry into the `/etc/printcap` file and the client Linux systems, create the spool directory, and so on ...

```
fax:\
        :sd=/var/spool/fax:\
        :mx#0:\
        :sh:\
        :rm=host.domain:\
        :rp=fax:
```

add the name of the client hosts into the `/etc/hosts.lpd` on the fax server machine.

use it as before.

Note:

The header string that efax adds to the fax pages is not effected by the user name that sends the fax. (can be updated).

2.6 Caldera, LPRng users

The LPRng printing management software is using a different method to handle the control file.

Thanks to Luca Montecchiani <m.luca@usa.net>, who found the problem and the solution. Here is an update to the `/usr/bin/fax` file. The following line replace the two simple **cfile=...** lines at lines 586,587

```
# Modified to work also with the LPRng package
# Luca Montecchiani (08/11/97 m.luca@usa.net)
if [ !-z "$CONTROL_FILE" ]
    then
        cfile=`cat tail -1 lock`
        cfile=`cat $cfile`
    else
        cfile=$CONTROL_FILE
fi
```

3. Latest version, Contacting the author.

The latest version of this file can be accessed through the world wide web using URLs

```
http://www.newplaces.com/linux/faxsrv/faxsrv-mini-HOWTO.sgml
http://www.newplaces.com/linux/faxsrv/faxsrv-mini-HOWTO.html
http://www.newplaces.com/linux/faxsrv/faxsrv-mini-HOWTO.txt
http://www.newplaces.com/linux/faxsrv/faxsrv-mini-HOWTO.info
```

You can contact me


```
Erez Strauss
erez@newplaces.com
http://www.newplaces.com/linux/
http://www.newplaces.com/
Phone: +972 52 739737
Fax:  +972 9 954 3034
```

Firewall Piercing mini-HOWTO

François-René Rideau, fare@tunes.org

v0.3b, 27 November 1998

Directions for using ppp over telnet to do network stuff transparently through an Internet firewall.

1. Stuff

- 1.1 DISCLAIMER
- 1.2 Legal Blurp
- 1.3 Credits

2. Introduction

- 2.1 Foreword
- 2.2 Security problems
- 2.3 Other requirements
- 2.4 Downloading software

3. Understanding the problem

- 3.1 Giving names to things
- 3.2 The problem
- 3.3 Additional difficulty

4. The solution

- 4.1 Principle
- 4.2 fwprc
- 4.3 .fwprcrc

TOC

5. Reverse piercing

- 5.1 Rationale
- 5.2 Getting the triggering mail

6. Final notes

- 6.1 Other settings
- 6.2 HOWTO maintenance
- 6.3 Extra copy of IMPORTANT DISCLAIMER --- BELIEVE IT!!!

---

1. Stuff

1.1 DISCLAIMER

READ THIS IMPORTANT SECTION !!!

I hereby disclaim all responsibility for this hack. If it backfires on you in any way whatsoever, that's the breaks. Not my fault. If you don't understand the risks inherent in doing this, don't do it. If you use this hack and it allows vicious vandals to break into your company's computers and costs you your job and your company millions of dollars, well that's just tough nuggies. Don't come crying to me.

1.2 Legal Blurp

Copyright © 1998 by François-René Rideau.

This document is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

1.3 Credits

Even though I rewrote most everything but the disclaimers, I'm indebted to Barak Pearlmutter mailto:bap@cs.unm.edu for his Term-Firewall mini-HOWTO: I think there was a necessity for a mini-HOWTO about piercing firewalls, and despite its shortcomings, his mini-HOWTO was a model and an encouragement.

2. Introduction

2.1 Foreword

Because system administrators and users have different constraints and proficiencies, it so happens that a user may find himself behind a firewall, that he may cross, but only in awkward ways. This mini-HOWTO explains a generic and portable way to use standard internet tools seamlessly across such firewalls, by the use of an IP emulator over a telnet session.

It is freely inspired by the Term-Firewall mini-HOWTO by Barak Pearlmutter mailto:bap@cs.unm.edu, which relies on an ancient and no-more-supported program named Term (yet a great program at its time), as well as on peculiarities of a not-so-standard telnet implementation, that is, many obsolete and non-portable facts.

2.2 Security problems

Of course, if your sysadm has setup a firewall, s/he might have a good reason, and you may have signed an agreement to not circumvent it. On the other hand, the fact that you can telnet outside (which is a requisite for the presented hacks to work) means that you are allowed to access external systems, and the fact that you can log into a particular external system somehow means you're allowed to do it, too.

So this is all a matter of *conveniently* using legal holes in a firewall, and allow generic programs to work from there with generic protocols, as opposed to requiring special or modified (and recompiled) programs going through lots of special-purpose proxies that be misconfigured by an uncaring or incompetent sysadm, or to installing lots of special-purpose converters to access each of your usual services (like e-mail) through ways supported by the firewall (like the web).

Moreover, the use of a user-level IP emulator such as SLiRP should still prevent external attackers from piercing the firewall back in the other way, unless explicitly permitted by you (or they are clever and wicked, and root or otherwise able to spy you on the remote host).

All in all, the presented hack should be *relatively* safe. However, it all depends on the particular circumstances in which you set things up, and I can give no guarantee about this hack. Lots of things are intrinsically unsafe about any internet connection, be it with this hack or not, so don't you assume anything is safe unless you have good reasons, and/or use some kind of encryption all the way.

To sum it up, don't use this hack unless you know what you're doing. Re-read the disclaimer above.

TOC

2.3 Other requirements

It is assumed that you know what you're doing; that you know about setting up a network connection; that you have shell accounts on both sides of the firewall; that you can somehow telnet (or ssh, or equivalent) from one account to the other; that you can run an IP emulator on both shell accounts; that you have programs able to use the IP connection emulated on their side. Note that any program can use the connection, in case the local emulator is pppd talking to the Linux kernel; other emulators, like Term, need recompilation and linking to a special library.

Talking about IP emulators, pppd can be found in any good Linux distribution or ftp site; so can SLiRP. If your remote shell account is user-level only, you can use SLiRP to connect.

2.4 Downloading software

Most described software should be available from your standard distribution, possibly among contrib's; at least all but the two small last ones are available in as rpm packages. In case you want to fetch the latest sources or binaries (after all, one of the ends of the connection may not be running linux), use the addresses below:

- `SLiRP` can be found at http://blitzen.canberra.edu.au/slirp and/or ftp://www.ibc.wustl.edu/pub/slirp_bin/.
- `zsh` can be found at http://www.peak.org/zsh/.
- `ppp` can be found at ftp://cs.anu.edu.au/pub/software/ppp/.
- `fwprc` and `cotty` can be found at http://www.tunes.org/~fare/files/fwprc/.

---

3. Understanding the problem

Understanding a problem is the first half of the path to solving it.

3.1 Giving names to things

If you want this hack to work for you, you'll have to get an idea of how it works, so that in case anything breaks, you know where to look for.

The first step toward understanding the problem is to give a name to relevant concepts.

So we'll herein call "local" the machine that initiates the connection, as well as programs and files on that machine; conversely, we'll call "remote" what's on the other side of the connection.

3.2 The problem

The goal is to connect the input and output of a local IP emulator to the output and input respectively of a remote IP emulator.

Only the communication channels with which IP emulators interact are either direct devices (in the usual case of pppd), or the "current tty". The previous case obviously does not happen with telnet sessions. The latter is tricky, because when you launch the local emulator from the command line, the "current tty" is linked to the command-line user, not to a remote session; also, should we open a new session (local or remote) on a new terminal, we must synchronize the launching and connection of IP emulators on both sides, least one session's garbage output is going to be executed as commands on the other session, which would recursively produce more garbage.

3.3 Additional difficulty

To get the best ease of use, the local IP emulator has to provide IP to kernel networking, hence be pppd. However, pppd is dumb enough to only accept having data through /dev or thru the current tty; it must be a tty, not a pair of pipe (which would be the obvious design). This is fine for the remote pppd if any, as it can use the telnet session's tty; but for the local pppd, it sucks, as it can't launch the telnet session to connect to; hence, there must some kind of wrapper around it.

Telnet behaves *almost* correctly with a pair of pipe, except that it will still insist on doing ioctl's to the current tty, with which it will interfere; using telnet without a tty also causes race conditions, so that the whole connection will fail on "slow" computers (fwprc 0.1 worked perfectly on a P/MMX 233, one time out of 6 on a 6x86-P200+, and never on a 486dx2/66).

[Note: if I find the sucker (probably a MULTICS guy, though there must have been UNIX people stupid enough to copy the idea) who invented the principle of "tty" devices by which you read and write from a

"same" pseudo-file, instead of having clean pairs of pipes, I strangle him!]

4. The solution

4.1 Principle

The firewall-piercing program, `fwprc`, will use a "tty proxy", `cotty`, that opens two pseudo-tty devices, launches some command on each of those devices' slaves, and stubbornly copies every character that one outputs to the tty that serves as input of the other command. One command will be telnet connection to remote site, and the other will be the local pppd. pppd can then open and control the telnet session with a chat script as usual.

4.2 fwprc

I wrote a very well self-documented script to pierce firewalls, `fwprc`, available from my site http://www.tunes.org/~fare/files/fwprc/, together with `cotty` (which is required by `fwprc` 0.2 and later). At the time of my writing these lines, latest versions are `fwprc` 0.3a and `cotty` 0.3a.

The name "fwprc" is voluntarily made unreadable and unpronounceable, so that it will confuse the incompetent paranoid sysadm who might be the cause of the firewall that annoys you (of course, there can be legitimate firewalls, too, and even indispensible ones; security is all a matter of *correct* configuration). If you must read it aloud, choose the worst way you can imagine.

CONTEST! CONTEST! Send me a `.au` audio file with a digital audio recording of how you pronounce "fwprc". The worst entry will win a free upgrade and his name on the fwprc 1.0 page!

I tested the program in several settings, by configuring it through resource files. But of course, by Murphy's law, it will break for you. Feel free to contribute enhancements that will make life easier to other people who'll configure it after you.

4.3 .fwprcrc

`fwprc` can be customized through a file `.fwprcrc` meant to be the same on both sides of the firewall. Having several alternate configurations to choose from is sure possible (for instance, *I* do it), and is left as an exercise to the reader.

To begin with, copy the appropriate section of `fwprc` (the previous to last) into a file named `.fwprcrc` in your home directory. Then replace variable values with stuff that fits your configuration. Finally, copy to the other host, and test.

Default behavior is to use pppd locally, and slirp remotely. To modify that, you can redefine the appropriate function in your `.fwprcrc` with such a line as:

```
remote_IP_emu () { remote_pppd }
```

Note that SLiRP is safer than pppd, and easier to have access to, since it does not require being root on the remote machine. Anoter safe feature is that it will drop packets not directly coming from the connected machine (which feature becomes a misfeature if you attempt to route a subnetwork onto it with masquerading). The basic functionality in SLiRP works quite well, but I've found advertised pluses (like run-time controllability) to be deficient; of course, since it is free software, feel free to hack the source so as to actually implement whichever feature you need.

---

5. Reverse piercing

5.1 Rationale

Sometimes, only one side of the firewall can launch telnet sessions into the other side; however, some means of communication is possible (typically, through e-mail). Piercing the firewall is still possible, by triggering with whatever messaging capability is available a telnet connection from the ``right'' side of the firewall to the other.

`fwprc` includes code to trigger such connections from a PGP-authentified e-mail message; all you need is add `fwprc` as a `procmail(1)` filter to messages using the protocol, (instructions included in `fwprc` itself). Note however, that if you are to launch pppd with appropriate priviledges, you might need create your own suid wrapper to become root. Instructions enclosed in `fwprc`.

Also, authentified trigger does not remotely mean secure connection. You should really use ssh (perhaps over telnet) for secure connections. And then, beware of what happens between the triggering of a telnet connection, and ssh taking over that connection. Contribution in that direction welcome.

5.2 Getting the triggering mail

If you are firewalled, your mail may as well be in a central server that doesn't do procmail filtering or allow telnet sessions. No problem! You can use `fetchmail(1)` to run in daemon mode to poll and get mail to your client linux system, and/or add a cron-job to automatically poll for mail every 1-5 minutes. fetchmail will forward mail to a local address through `sendmail(8)`, which itself will have been configured to use `procmail(1)` for delivery. Note that if you run `fetchmail(1)` as a background daemon, it will lock away any other fetchmail that you'd like to run only at other times, like when you open a `fwprc`; of course, if you can also run a fetchmail daemon as a fake user. Too frequent a poll won't be nice to either the server or your host. Too unfrequent a poll means you'll have to wait before the message gets read and the reverse connection gets established. I use two-minute poll frequency.

6. Final notes

6.1 Other settings

There are other kinds of firewalls than those that allow for telnet connections. As long as a continuous flow of packets may go through a firewall, and transmit information both ways, it is possible to pierce it; only the price of writing the piercer may be higher or lower.

In a very easy case, you can just launch `ssh` over a pty, and do some `pppd` in the slave tty. `cotty` 0.3a should be able to do it, but nobody's modified `fwprc` to take it into account yet. May be tonight's exercise for you. You may even want to do it without an adverse firewall, just so as to build a secure ``VPN'' (Virtual Private Network). See the VPN mini-HOWTO about this.

If you need cross a 7-bit line, you'll want to use SLIP instead of PPP. I never tried, because lines are more or less 8-bit clean these days, but it shouldn't be difficult.

Now, if the only way through the firewall is a WWW proxy (usually, a minimum for an internet-connected network), you might want to write a daemon that buffers data in and out, and sends it during in HTTP connections, achieving some telnet-over-HTTP over which to run fwprc. It might be slow and not very responsive, but still good enough to use `fetchmail(1)`, `suck(1)`, and other non-interactive programs.

If you want more performance, or if the only thing that goes through unfiltered is some wierder thing even (DNS queries, ICMP packets, whatever), then you're in the very hard case where you'll have to re-hack a wierd IP stack, using (for instance) the Fox project's packet-protocol functors. You'll then achieve some direct IP-over-HTTP, IP-over-DNS, IP-over-ICMP, or such, which requires not only a complex protocol, but also an interface to an OS kernel, both of which are costly to implement.

By the way, if you use some Firewall-piercing HTTP daemon, don't forget to have it serve fake pages, so as to mislead suspicious adverse firewall administrators.

6.2 HOWTO maintenance

I felt it was necessary to write it, but I don't have that much time for that, so this mini-HOWTO is very rough. So will it stay, until I get enough feedback so as to know what sections to enhance. Feedback welcome. Help welcome. mini-HOWTO maintenance take-over welcome.

In any case, the above sections have shown many problems whose solution is just a matter of someone (you?) spending some time (or money, by hiring someone else) to sit down and write it: nothing conceptually complicated, though the details might be burdensome or tricky.

Do not hesitate to contribute more problems, and hopefully more solutions, to this mini-HOWTO.

6.3 Extra copy of IMPORTANT DISCLAIMER --- BELIEVE IT!!!

TOC

I hereby disclaim all responsibility for this hack. If it backfires on you in any way whatsoever, that's the breaks. Not my fault. If you don't understand the risks inherent in doing this, don't do it. If you use this hack and it allows vicious vandals to break into your company's computers and costs you your job and your company millions of dollars, well that's just tough nuggies. Don't come crying to me.

This document is not subject to copyright. See section 9 below.
Version 1.0: 13 November 1997

TOC

The GIS-GRASS mini-HOWTO

by David A. Hastings
U. S. Department of Commerce
National Oceanic and Atmospheric Administration
National Geophysical Data Center
Boulder CO 80303
dah@ngdc.noaa.gov

Summary: This document describes how to acquire, install and configure a powerful scientific public-domain Geographic Information System (GIS): the Geographic Resources Analysis Support System (GRASS). It provides information on other resources for learning more about GRASS, GIS in general, for acquiring data, etc.

This document also encourages the Linux community to consider enhancing this software as a major application of UNIX/Linux. ("When will Linux become bundled with public domain or Linux Public License 'killer aps'"?) For more on this topic, see Section 8 below.

Contents

1. **What is a GIS?**
2. **What is GRASS?**
3. **A Brief History of GRASS**
    1. **A Re-Invogorated GRASS Management Model**
    2. **Continued Assessment of Future GRASS Management**
4. **System Requirements for GRASS**
5. **How to Acquire GRASS**
6. **How to Get GRASS Running on Your Linux-based Computer.**
7. **Web-based Support for GRASS (and for GIS Matters in General)**
8. **The Future of GRASS?**
9. **Copyright Notice, and Notice on Support of this Document**
10. **References**

Appendix A: Acquisition/Installation of GRASS4.1.3 Binaries
Appendix B: Acquisition/Installation of GRASS4.1.5 Binaries
Appendix C: Acquisition/Compilation of GRASS 4.1.x and 4.2 Source Code
Appendix D: If you plan to enhance any part of GRASS, read this first!
Appendix E: Example Linux versions of some critical GRASS files.

---

1. What is a GIS?

There are many ways to describe a Geographic Information System. Here are three working definitions (from David A. Hastings, 1992, Geographic Information Systems: A Tool for Geoscience Analysis and Interpretation):

1. (The minimal definition): A GIS is a hardware/software system for the storage, management, and (with hardcopy or screen graphic) selective retrieval capabilities of georeferenced data. Definitions like this one are often used by vendors and users of vector-only GIS, whose objective is sophisticated management and output of cartographic data.
2. (A parallel definition): A GIS is a hardware/software system for managing and displaying spatial data. It is similar to a traditional Data Base Management System, where we now think in spatial rather than in tabular terms, and where the "report writer" now allows output of maps as well as of tables and numbers. Thus we can consider a GIS a "spatial DBMS" as opposed to traditional "tabular DBMSs." Few people use this definition, but it might help to explain GIS to a DBMS user.
3. (A more aggressive definition): A GIS is a system of hardware, software, and data that facilitates the development, enhancement, modeling, and display of multivariate (e.g. multilayered) spatially referenced data. It performs some analytical functions itself, and by its analysis, selective retrieval and display capabilities, helps the user to further analyze and interpret the data. Properly configured, the GIS can model (e.g. synthetically recreate) a feature or phenomenon as a function of other features or phenomena which may be related - where all features or phenomena are represented (characterized) by spatial and related tabular data. The analytical objectives described here are sometimes controversial - and often given lip service by cartographic GIS specialists who have not yet seen what can be accomplished scientifically by a select few GISs that go beyond cartographic approaches.
4. Another definition can be found at http://www.geo.ed.ac.uk/home/research/whatisgis.html at the University of Edinburgh

2. What is GRASS?

GRASS (Geographic Resources Analysis Support System) is a public domain raster based GIS, vector GIS, image processing system, and graphics production system. Created by the US Army Corps of Engineers, Constriction Engineering Research Laboratory (USA/CERL) and enhanced by many others, it is used extensively at government offices, universities and commercial organizations throughout the world. It is written mostly in C for various UNIX based machines. Linux is one of its more robust implementations.

GRASS contains over 40 programs to render images on monitor and paper; over 60 raster manipulation programs; over 30 vector manipulation programs; nearly 30 multi-spectral image processing manipulation programs; 16 data management programs; and 6 point file management programs.

GRASS' strengths lie in several fields. The simple user interface makes it an ideal platform for those learning about GIS for the first time. Users wishing to write their own code can do so by examining existing source code, interfacing with the documented GIS libraries, and by using the GRASS Programmers' Manual. This allows more sophisticated functionality to be fully integrated within GRASS.

Other strengths include GRASS' pioneering of mixed resolutions in a data base, mixed geographic coverage areas in a data base, raster image compression techniques via run-length encoding and reclassification lookup tables, GRASS' rescaling of display images on the fly to fill the display screen, plus its fundamental design criterion of powerful computer-assisted scientific analysis of environmental issues (as opposed to merely going for intricate cartographic output of relatively simple processes).

GRASS is usually supplied as free, non-copyright source code to be compiled on host machines. Some compiled binaries are also easily obtainable at no cost via the Internet. It runs on a variety of UNIX platforms.

(Copied from Project Assist Intro to GRASS:http://www.geog.le.ac.uk/assist/grass)

3. A Brief History of GRASS

TOC

In the early 1980s the U. S. Army Corps of Engineers' Construction Engineering Research Laboratory (USA/CERL) in Champaign, Illinois, began to explore the possibilities of using Geographic Information Systems to conduct environmental research, assessments, monitoring and management of lands under the stewardship of the U. S. Department of Defense. Part of the motivation for this action was new responsibility for the environment encoded into the National Environmental Policy Act of the late 1970s.

Bill Goran of USA/CERL conducted a survey of available GISs, assuming that he could find several systems capable of environmental analysis, from which he could select one or more to recommend for use by CERL and perhaps others in the Department of Defense. However, he was surprised to find no GIS that satisfied his needs. What started as a selection process turned into a design exercise for his own GIS development program.

USA/CERL hired several programmers, and began by writing a hybrid raster-vector GIS for the VAX UNIX environment. This made the team one of the first to seriously develop GIS for UNIX. Though they still faced challenges with different versions of UNIX, they developed procedures of coding in ANSI standard UNIX, avoiding "tweaking" the code toward any particular vendor-specific flavor of UNIX.

GRASS developed a programming style characterized by:

- Use of UNIX libraries where possible, plus the creation of GRASS libraries for repeated GIS-specific acts such as opening raster files that might be compressed (by run-length encoding) or not.
- The ability to handle both major GIS data types: raster and vector.
- The favoring of raster data processing, as scientific analysis was easier to encode with raster (than for vector) data models.
- The ability to handle raster grids of mixed grid sizes in the same data base. This was a departure from raster's image processing tradition of requiring identical (and perfectly registered) grid cell arrays in each and every data layer.
- The ability to handle raster grids with different areas of coverage. Again, this was a departure from raster tradition of having all grids be identical in geographic coverage.
- The ability to run-length encode raster data files, in order to greatly reduce file sizes of most files.
- The separate structure of reclassification files. Such files merely contained a look-up table noting the previous and new classes. This is MUCH more compact than replicating the original grid with different numerical values. A reclassified file of a 100x100 km square area of 10 metre grid cells would be a few hundred bytes, rather than 100 megabytes of uncompressed 8-bit raster data.
- The acceptance of de-facto standard data models. While competitors created cumbersome (and in many cases secretive) data formats, GRASS accepted the de-facto standard Digital Line Graph vector format and unheaded binary raster grid format. GRASS later abandoned DLG as its internal vector file format, and let its raster format evolve. However, DLG and the unheaded binary raster grid are still routinely handled formats for GRASS, and its new formats are as open as its previous ones.
- GRASS code was managed in several directories. Initial contributions were placed in the src.contrib directory. More solid code was moved to the src.alpha directory. After remaining in the src.alpha for one full release cycle, the code, with resultant bug fixes, moved to the most honorable level, the src directory.

GRASS was overseen by three levels of oversight committees. USA/CERL kept the ultimate responsibility for GRASS. It implemented most GRASS development, and carried out the day-to-day management of GRASS testing and release. The GRASS Interagency Steering Committee (GIASC), comprised of other Federal agencies, met semi-annually to review development progress, and evaluate future directions for GRASS. (Academic and commercial participants in GRASS also attended GIASC meetings; only part of each meeting was "Federal-Agencies-only." GRASS eventually became nominally and officially a "product" of the GIASC, though everyone recognized USA/CERL's leadership role. The GRASS Military Steering Committee met periodically to review the progress of GRASS in serving its original intent: meeting the Department of Defense's needs to evaluate and manage the environment of military lands.

TOC

The public interacted with CERL and GIASC through USA/CERL's GRASS Information Center. GRASS Beta testi was very widespread, and quite intensive for the leading users of GRASS. Several leading users, such as the Nation Park Service and the Soil Conservation Service, selected GRASS as its prime or only GIS. They made significant commitments to enhance and test GRASS, yet considered this investment well worth their while. They said that they had more influence over the direction of GRASS than they would over any known alternative system. They also felt that, despite their major efforts and expenses in supporting GRASS, they had a bargain in relevant power for the dollar.

Several universities adopted GRASS as an important training and research environment. Many conducted short-courses for the public, in addition to using GRASS in their own curricula. Examples of such leading academic users of GRASS are Central Washington University, The University of Arkansas, Texas A & M University, The University of California at Berkeley, and Rutgers University.

Though GRASS received some criticism (some say) for being so good and so public, it was also reputedly borrowed from liberally by some developers of other systems. Though the first group might have viewed it as unfair competition, the second group may have noted that it was not copyright, and was a valuable testbed for GIS concepts. GRASS received an award from the Urban and Regional Information Systems Association (URISA) for quality software in 1988.

As CERL and GRASS evolved through the late 1980s and early 1990s, CERL attempted to cut overhead costs associated with supporting the public domain version. It created and initially funded the Open GRASS Foundation, in cooperation with several of the leading users of GRASS. The Open GRASS Foundation has since evolved into the Open GIS Consortium, which is aiming for more thorough interoperability at the data and user interface level, but appears not to be taking advantage of the major open GIS testbed (GRASS).

In 1996 USA/CERL, just before beginning the beta testing for GRASS version 5.0, announced that it was formally withdrawing support to the public. USA/CERL announced agreements with several commercial GISs, and agreed to provide encouragement to commercialization of GRASS. One result of this is GRASSLANDS:http://www.las.com/grassland/, a commercial adaptation of much of GRASS. Another result is a migration of several former GRASS users to COTS (Commercial Off-The-Shelf) GISs. However, GRASS' anonymous ftp site contains many enhancements to the last full version 4.1 release of GRASS. Many organizations still use GRASS feeling that, despite the lack of a major release in five years, GRASS still leads the pack in many areas.

3.1 A Re-Invogorated GRASS Management Model

In late 1997, a group at Baylor University took the lead in developing a new Website for GRASS. This quickly developing Website contains GRASS 4.1 source code and Sun Solaris binaries, GRASS 4.1 documentation, and an on-line manual. By November 1997 this site posted the first version of GRASS 4.2 source code and binaries currently for Sun Solaris) with Linux and Windows NT under consideration). GRASS 4.2 incorporates several enhancements from the CERL website, plus some of Baylor's own enhancements. Documentation for GRASS 4.2 is appearing; the group encourages cooperation in further development of GRASS, and is looking for partners. It hopes to use increased use of the World Wide Web in developing and managing GRASS. GRASS 5 development and compilation is underway. The site also links to the Blackland GRASS site at Texas A&M University, for those desiring very inexpensive access to GRASS for Windows 95.

3.2 Continued Assessment of Future GRASS Management

Note: An ad-hoc group (which includes myself) is exploring the basic issue of continued, reconfigured, yea perhaps increased, value of GRASS as a public test-bed for GIS technology. It is exploring shepherding the testing and release of GRASS5.0, and exploring possibilities for a more distributed management model for GRASS design and development. It is exploring the universe of public domain spatial data processing software (including geographic information and image processing systems), and perhaps tabular data base management systems. How can such knowledge be (1) optimized as an open, public test bed for such technology and (2) better used by the public? Might this involve a Linux management model, perhaps? See Section 8 for more discussion on this topic.

TOC

4. System Requirements for GRASS

Minimum requirements include:

- 8 Mbytes of memory (of course, more is better..)
- 100 Mbytes of free disk space
  - ~40 mb for executables,
  - ~40 mb for source code (which you can ignore if you merely install the Linux binaries)
  - ~? for data (the veritable bottomless pit can be filled with data, if you so choose)

GRASS runs on Linux kernel versions as old as 1.2.13 (see more information in the appendices for various specific binaries).
GRASS will run in text mode. However, for displays of data, you will need X. If you are still running a version of X, it will probably work with GRASS.

If you find any other hardware/OS requirements that should be mentioned, please let me know!

5. How to Acquire GRASS

GRASS used to be available on tape from various companies that signed distribution agreements with USA/CERL. These companies usually supported specific platform environments, such as Masscomp, Sun, DEC, Hewlett Packard, IBM (risc), PC (running some flavor of UNIX), and Macintosh (running AUX). Until recently, the flavors of UNIX working on PCs generally were too low-end, or required too much added programming support (e.g. programming drivers for high-end graphics boards like the Number Nine boards of several years back) to be stable or complete. However, with robust systems like Linux, this problem is history. Similarly, few people acquire GRASS on tape, though a few do on CD-ROM.

The main way to acquire GRASS is to get it via anonymous ftp from:

1. The new site at Baylor University:http://www.baylor.edu/~grass

As of the date of this version of the mini-HOWTO, Baylor has source code for GRASS 4.1 and 4.2, as well as Sun Solaris compiled binaries. Blackland GRASS for Windows 95/NT is linked to from this site. Baylor is considering its own Linux and Windows NT binaries, as well. You should be able to compile the Baylor source code under Linux yourself, using information in this mini-HOWTO.

2. The traditional site at USA/CERL:http://www.cecer.army.mil/grass, or from mirrors cited at USA/CERL's website:

The ftp location is:

moon.cecer.army.mil

Appendix A describes how to acquire and install GRASS4.13 compiled binaries from USA/CERL. (See section 6 before installing GRASS!)

Appendix B describes how to acquire and install GRASS4.15 compiled binaries from USA/CERL. (See section 6 before installing GRASS!)

Appendix C describes how to acquire and compile GRASS4.14 and GRASS4.15 source code from USA/CERL, as well as GRASS4.2 source code from Baylor University. (See section 6 before installing GRASS!)

Linux distribution developers! Might you be interested in including GRASS with your distribution? Remember, GRASS source code is in the completely unrestricted, copyright-free, public domain. Your distribution might be more valuable if it contained source code and/or compiled binaries for GRASS.

6. How to Get GRASS Running on Your Linux-based Computer.

TOC

Appendices A, B, and C describe how to acquire and install GRASS. Before actually installing GRASS, you will have to decide where to put three parts of the system:

1. The GRASS binaries, source code (if you install this), man pages, documentation, and the like. Many folks put this stuff off /usr/local (e.g. /usr/local/grass/bin, /usr/local/grass/src).
2. The GRASS executable and gmake utilities. Some folks put this stuff off /usr/local (e.g. /usr/local/grass/grass4.1 and gmake4.1 or /usr/local/bin/grass4.1 and gmake4.1).
3. The GRASS data directories. These can go anywhere, as they are specified in configuration files.

   I have used a different scheme for a decade. As GRASS code, binaries, and the like (except data owned by users) are all owned by the special user "grass" I don't want this stuff to get spread around my system. I create a new directory (usually on a separate file system) called /user, and put all my GRASS stuff below this. For example:

```
/user/grass4.1/bin   (I usually put grass4.1 and gmake4.1 here...)
              /data
              /dev
              /etc
              /man
              /src
              /src.alpha
              /src.contrib
```

   I'm currently building a GRASS5.0 site, which then goes under:

```
/user/grass5/bin
            /data   (some GRASS5 data formats have changed...)
            /dev
            /etc
```

   The GRASS Installation Guide (described in Section 10 and in Appendix C) is useful for getting GRASS running, even if you merely install the binaries as described in Appendices A and B. Please don't overlook one important detail: Most GRASS installations separate user from software manager accounts and UNIX permissions. You should create a "grass" (the quotes here are for emphasis, and should not be part of the actual user userid) user account on your workstation. All installation and configuration of grass should be done by user "grass". Untar (or un"cpio" files, run setup configuration utilities, run Gmakefiles (GRASS versions of makefiles), and edit configuration files as user "grass." Then only RARELY run GRASS as user "grass." (I only run GRASS as user "grass" when I am making archival data files in the PERMANENT mapset.) This is done for much the same reason as not running user software as user "root". YOU CAN DO TOO MUCH DAMAGE AS USER "grass"!

   Beyond the instructions in these appendices, and information in the GRASS Installation Guide, you have some additional housekeeping to do, such as developing a data base. You can acquire sample data bases from USA/CERL (directory pub/grass/grass4.1/data at anonymous "ftp moon.cecer.army.mil"), start from scratch following instructions in the GRASS Programmer's Manual (and, to a lesser degree, buried in the functional descriptions of the GRASS User's Reference Manual).

   I personally recommend that you start with the Spearfish and Global databases available from USA/CERL:

   1. The Spearfish data base covers two 7.5 minute topographic sheets in the northern Black Hills of South Dakota, USA. It is in the Universal Transverse Mercator Projection. It was originally created by Larry Batten (now of the Environmental Systems Research Institute's office in Boulder, Colorado) while he was with the U. S. Geological Survey's EROS Data Center in South Dakota. The data base was enhanced by USA/CERL and cooperators. It is an excellent, and well-used (there are many training materials available for GRASS with this data base) example of a county-scale GIS project in the UTM projection.

TOC

2. The Global data base was developed by Bob Lozar of USA/CERL to prototype a latitude-longitude "projection" data base in GRASS for global environmental study and decision support.

Starting with these two examples, you can build your own data bases in UTM and latitude-longitude projections. (Note, many people don't call latitude-longitude a projection. Others disagree, saying that anything that transfers the Earth's surface to two dimensions is a projection.. We'll stay away from that debate here. Needless to say, lat-lon is treated as other projections are by the computer program.)

7. Web-based Support for GRASS (and for GIS Matters in General)

Support for a public domain program? No way, they say! Actually, as a user of Linux, you probably know better.

GRASS started by having a GRASS Information Office at USA/CERL. There were also very active users outside USA/CERL, who provided valuable user support. GRASS had annual users' meetings, listservers for users and developers, etc. Companies provided value added support services on a contractual or fee basis.

Various people have developed valuable books and training materials on GRASS. Several universities used to conduct training courses in GRASS. I don't know how many of these are continuing. If training courses interest you, try asking on the usenet newsgroup comp.infosystems.gis (see below for more on this newsgroup).

Valuable "books" available on the Internet are noted in the References (Section 10).

World Wide Web-based training materials, including training in GRASS, are highlighted in the CyberInstute Short Course in GIS:http://www.ngdc.noaa.gov/seg/tools/gis/referenc.html (then scan down for the link(s) to Web-based tutorials in GIS).

One of the better GRASS tutorials is Project Assist's - Intro to GRASS:http://www.geog.le.ac.uk/assist/grass

Other good sites:

Central Washington University was an early GRASS user and training facility:http://www.csu.edu/~gishome/grass.htm

"Starting the hunt for mostly free spatial data" by Stephan Pollard:http://cast.uark.edu/local/hunt This is based at the Center for Advanced Spatial Technology of the University of Arkansas, another early educator with GRASS.

Purdue University has several GRASS features:http://pasture.ecn.purdue.edu/~aggrass

USA/CERL's online GRASS manual:http://www.cecer.army.mil/grass/userman/main-alpha.html

Rutgers University's GRASS Information Center:http://deathstar.rutgers.edu/grassinfo.html

The REGIS project:http://www.regis/berkeley.edu at the University of California at Berkeley as a Linux version of GRASS available via ftp, and also has a Web-based version of GRASS called GRASSLINKS.

After getting trained by the books and Web-based tutorials noted just above, where do you turn to for specific advice???

Probably the best source of support these days is usenet newsgroup comp.infosystems.gis If you're not familiar with newsgroups, ask your network administrator or Internet service provider. comp.infosystems.gis contains modestly heavy traffic on such topics as

- "how do I find data on this topic for this area?"
- "how do I convert these data for use in my Aardvark GIS?"
- "how do I get this function to work in my Aardvark GIS?"
- "which GIS can I use to solve my particular problem?"

GRASS used to be one of the top GISs discussed on this group. Traffic in GRASS is dropping slightly, as its community matures. However, there are usually answers to your questions, if you post them. You might also "power search" on subject:GRASS [& your own subject of interest here?] and newsgroup:comp.infosystems.gis in DejaNews:http://www.dejanews.com to see what might appear from the usenet archives.

8. The Future of GRASS?

Excellent question! Several possible answers have been thrown out:

1. USA/CERL's announced intention is to use GRASS and COTS (commercial off-the-shelf software) for internal uses, to leave the GRASS public web- and ftp-site on its system indefinitely, and to sign cooperative research and development agreements with three companies: (1) the Environmental Sciences Research Institute (ESRI), (2) Intergraph, and (3) Logiciels et Applications Scientifiques (L.A.S.) Inc. The first two agreements encouraged the incorporation of GRASS concepts into ESRI's and Intergraph's commercial GISs. The third encouraged the adaptation of GRASS' concepts and code into a new commercial GIS by L.A.S. L.A.S. also offered to encourage the continuation of a public domain GRASS, as a viable stand-alone system and as a potential source of new ideas and code for L.A.S.'s GRASSLAND. One observer noted that the first two agreements might be akin to someone signing Linux over to Microsoft. The same observer considers the experiment by/with L.A.S. to be an interesting possibility - an attempt to keep viable public domain and commercial versions of GRASS.
2. Some people believe that GRASS will wither without USA/CERL's central management. Some believe that the Open GIS Consortium will successfully guide industry into an open architecture that will benefit all developers and users. Others believe that OGIS' effort will lead to a cacophony of almost similar (but not quite interoperable) vendor-specific "standards," so the loss of GRASS as an open development platform will be felt sorely.
3. Some people believe that developments on some campuses and other sites may result in those institutes keeping GRASS for awhile, but in non-standard forms. In short, GRASS will undergo "cell division" and lead to a cacophony of internally valuable, but externally unused, GISs.
4. Others hope that GRASS' previous management model under USA/CERL has left it ready for a new model. Perhaps:
    1. Under a new mentor, such as NASA (which needs an open, powerful and scientific, GIS integrated with image processing system for its Earth Observing System).
    2. Under a distributed management model... perhaps somewhat like Linux?
    3. Perhaps a bit of a hybrid? Perhaps a Web-based effort could spawn a series of usenet discussion groups beginning with
        - comp.infosystems.gis.grass, and evolving to:
        - comp.infosystems.gis.grass.academics
        - comp.infosystems.gis.grass.publicservice
        - comp.infosystems.gis.grass.commercialvalueadded
        - comp.infosystems.gis.grass.commercialdistributors
        - comp.infosystems.gis.grass.programming
        - comp.infosystems.gis.grass.users
        - comp.infosystems.gis.grass.centralcommittee

        Clearly the topics are a bit tongue-in-cheek. However, under this model, a Central Committee (including representation of academic, public service [government and nongovernmental organizations], commercial distributors and value added firms, programmers, and users) would guide overall grass development and testing. The other special interest groups would serve their user communities. Academics, for example, would involve GIS and GRASS education, but would also try to pull GRASS development in its direction. Value added commercial developers would serve their

TOC

own interests, including trying to pull GRASS development in a direction that would help their businesses. Users would help each other learn GRASS, develop workarounds to bugs, etc.

GRASS offers considerable potential for:

- Use as a scientific, as well as a traditional graphically oriented GIS. Many GISs can make pretty maps. Many of those GISs cannot easily perform certain scientific analytical functions as easily or powerfully as GRASS. GRASS was designed and developed in response to a perceived need for scientific GIS, specifically for environmental analysis, and the environmental management/protection of public lands. Incidentally, there is at least one Web-based GRASS version. GRASSLINKS:http://www.regis/berkeley.edu/grasslinks, developed at the University of California at Berkeley, uses Web forms to submit commands to the server, which creates .gif-based display output, places the images into pages, and serves them up to the requester. More on that later.
- Education. GRASS is easier to teach and learn than some other GISs. It is easier to modify (for those that want to learn GIS as computer science, rather than as "geography") than most other GISs that come without source code and treat the program as a magical black box. And, of course, it is more affordable for the student of GIS than many other GISs.
- Applications research and development. Many universities have used GRASS. Its available source code, easy modification, easy scriptability, etc., give it distinct advantages over some more closed systems.
- Public Service. GRASS has been used as a scientific GIS for many public service applications. There is considerable value in continuing a robust GIS that can ba packaged with any UNIX workstation. There is considerably more value if that UNIX workstation universe can include Linux (but is not constrained only to Linux).
- GIS research and development. For example - do you want to experiment with a different data model? Add it to GRASS!
- Commercialization. This document gives contact information for a commercial version of GRASS. That company (and perhaps others?) may welcome your help in enhancing/supporting their product.

Who would be the Linus Torvelds equivalent in this management model? Perhaps no single person. I have been involved in GRASS for about a decade, when GRASS was the only GIS that satisfied my needs in scientific data management and GIS application. Indeed, I had been a dedicated avoider of the user-unfriendly UNIX environment until GRASS forced me to learn it. Several senior GRASS developers are active in GRASS-related activities and would like to see the continued vitality of an open GRASS. It's likely that a reborn GRASS would attract a new crop of friends. Thus the concept of a "Central Committee" to collectively lead GRASS' transition to a more open management and development style.

In short, the Linux community has an opportunity to take under its wing a killer ap. GRASS' current public domain status is slightly different from Linux's. However, that status could be discussed....

Comments would be appreciated!

9. Copyright Notice, and Notice on Support of this Document

Copyright notice:

This document was prepared by a Federal civil servant in support of his work (but mostly on his own time). It is NOT SUBJECT TO COPYRIGHT.

Notice on support of this document:

I believe that the contents of this document are accurate. However, if you use this document, you accept the risks for any errors in this document (and in any documents that are cited here).

I would greatly appreciate help in correcting any errors, or in enhancing this document. However, "my time is limited" in dealing with this issue. Any help that you can provide, that also helps me to more efficiently respond to

your interest, is more likely to be responded to quickly. A complaint might be appreciated, but a suggested improvement that includes draft wording might be REALLY appreciated.

TOC

10. References

For general reference material on GIS, try a very good technical bookstore (in many cases these are campus bookstores at schools with good GIS programs or top-notch technical or general bookstores - you know that ones are near you..), or the following URL for the CyberInstitute Short Course on Geographic Information Systems:hQfttp://www.ngdc.noaa.gov/seg/tools/gis/referenc.html (convened by myself):

Also check
Baylor University's growing GRASS Home Page:http://www.baylor.edu/~grass
USA/CERL's GRASS Home Page:http://www.cecer.army.mil/grass

For a good collection of references on GRASS, try this procedure, to load up on reference goodies from USA/CERL:

```
ftp moon.cecer.army.mil
login: anonymous
password: your email address
cd pub/grass/grass4.1/outgoing
image
get grassman.ps.Z  (or grassman.txt.Z, or grassman.wp.Z)
cd ../documents/programmer/postscript
image
get progman.ps.Z
cd ../../user/postscript
image
get refman.ps.Z
cd ../..
image
get installGuide.ps.Z
bye

uncompress grassman.ps.Z
uncompress progman.ps.Z
uncompress refman.ps.Z
uncompress installGuide.ps.Z

lpr *.ps   (or whatever is appropriate for your environment)
```

installGuide => The GRASS Installation Guide (you need this to compile GRASS source code)
grassman => The GRASS Beginner's Manual (intro to GRASS)
refman => The GRASS User's Reference Manual (function guide)
progman => The GRASS Programmer's Manual (and administrator's guide - this is valuable for info about data formats, etc.)

Browse around the ftp site noted just above, and you may find more stuff of interest. Particularly in the pub/grass/grass4.1/documents directory, there are tutorials on advanced GRASS functions such as r.mapcalc (think of this as math applied to raster arrays), r.combine and r.weight (think of this as how to combine spatial submodels into one type of model), and others.

Incidentally, do you prefer German? Try The University of Hannover's

site:http://www.laum.uni-hannover.de/iln/grass/handbuch

TOC

Appendix A: Acquisition/Installation of GRASS4.13 Binaries

This appendix describes how to acquire and install Linux binaries for GRASS4.13 (the 3rd update to the last full release of GRASS, version 4.1).

How to get these files:

```
  ftp moon.cecer.army.mil
  login: anonymous
  password: your email address
  cd pub/grass/grass4.1/release/binaries/linux
  image
  mget grassa*
  bye

Installation instructions:
*********************************************************************
* GRASS 4.1 Update 3 for Linux
*
* This package contains GRASS programs only, *NO* GIS
* data is included.  You can find example GRASS data at
* moon.cecer.army.mil
*
* Compiled by: Andy Burnett - burnett@zorro.cecer.army.mil
* compiled on: April 7, 1994

*********************************************************************
System Requiremnts:

        35 MB disk space to hold the binary distribution

System library requirements:

        libc4.5.21 or greater

        libX.so.3.1.0 or greater

If you are running libraries that are older than these, this binary
distribution will *NOT* run on your linux system.

-------------------------------------------------------------------------
Files in this release:

        README_4.1.3            what you are currently reading
        ginstall                simple grass installation script
        grassaa --------|
        grassab         |
        grassac         |
        grassad         |
```

```
        grassae         |--     the linux GRASS binaries
        grassaf         |
        grassag         |
        grassah         |
        grassai         |
        grassaj         |
        grassak --------|

INSTALLATION:

        To install this binary distribution of grass for linux, you
can simply run the ginstall script or you can unpack the files by
hand.  I recommend using the ginstall script ... it's very simple and
should be bullet proof.  To run the ginstall script, you will need
gawk (gnu awk) installed on your system and it needs to be in your
PATH.

If, however, you want to do things by hand, here's what you need to
do:

o  make the destination directory (/usr/grass, /usr/local/grass,
   whatever)  This will become your GISBASE for grass.

********************* LOOK HERE ***************************************
from here on, replace $GISBASE with the name of the directory you just
created
********************* LOOK HERE ***************************************

o  cat grassa? | gzip -d | (cd $GISBASE; tar xvf -)
   This will unpack all the grass binaries into the $GISBASE directory

o  copy $GISBASE/etc/moncap.sample to $GISBASE/etc/monitorcap and edit
   it.
o  change all occurrences of GBASE in that file to $GISBASE
o  copy $GISBASE/etc/grass4.1 into a public directory (I suggest
   /usr/bin)
o  edit the copy you just made:
   change all occurrences of GBASE to $GISBASE
```

Appendix B: Acquisition/Installation of GRASS4.1.5 Binaries

This appendix describes how to acquire and install Linux binaries for GRASS4.15 (the 5th and last update to the last full release of GRASS, version 4.1).

How to get these files:

```
ftp moon.cecer.army.mil
login: anonymous
password: your email address
cd pub/grass/grass4.1/release/binaries/linux
image
```

TOC

```
mget linuxa*
bye

Installation instructions:
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Files in this release:
        README_4.1.5            what you are currently reading
        install.sh                simple grass installation script
        linuxaa --------|
        linuxab         |
        linuxac         |
        linuxad         |
        linuxae         |--   the linux GRASS binaries, version 4.1.5
        linuxaf         |
        linuxag         |
        linuxah         |
        linuxai --------|

* * * * * * * * * * *** * * * * * * * * * * * * * * * * * * * * * * * *
*

The GRASS4.15 for Linux was compiled in my Linux box with the
following configuration:
        Slackware 3.0
        kernel 1.2.13
        gcc 2.7.0
        libc 5.0.9
        flex 3.5.2

~ ~ ~ ~ ~ ~ ~
~ IMPORTANT: ~
~ ~ ~ ~ ~ ~ ~
THE LINUX GRASS 4.15 BINARIES ONLY WORK ON ELF-LINUX. THE BINARIES MAY
NOT WORK WITH EARLY VERSION OF KERNEL AND/OR GCC AND FLEX.

The binaries was tared and gziped, then split into 9 (close to 1.3 MB
- 1200 x 1K block) files named from linuxg.aa to linuxg.ai.

You should ftp all the linuxg.a* in binary mode and also get this
readme file and an installation script - install.sh.  Please put all
of these files in the same directory - source directory.

At the source directory under the UNIX prompt, type
        sh ./install.sh full_path_to_the_destination_directory

and it should automatically unzip and untar the linuxg.a* files to the
destination directory and also edit several site-specific files.  The
total space your need is about 26 MB.

At the destination directory, your can find the grass4.1 script.  It
should have been modified to reflect your installation directory.
Now, either move/copy the grass4.1 file to one of your PATH or use the
link command as below:
```

```
        cd /usr/local/bin
        ln -s destination_directory/etc/grass4.1 grass4.1
```

```
Now, your are ready to start GRASS by typing grass4.1 and you should
know how to run GRASS afterward.

There is a readme directory in the destination_directory/etc
directory.  This directory has several readme files that come with
some incoming commands.  You can find all the compiled commands of
this binaries in the commands.readme file.  I can't guarantee that all
of them work but I have tested lots of them.  If you find some
commands that don't work, please post a message on the grass user
group and we can solve it all together.

Yung-Tsung Kang,
Michigan State University
```

Appendix C: Acquisition/Compilation of GRASS Source Code

The GRASS binaries for Linux tend to work. Why would anyone want to mess with the source code?

Let's try to answer this with another question: "Why can't I get the source code to my GIS, so I can see how it works, and maybe fix some things to work the way I like them?" (You probably know the answers to this question, at least for many commercial software packages.)

If you want to

1. Add any of the numerous existing alpha and contributed GRASS functions,
2. Understand how a function works (did any programming shortcuts or performance enhancements affect the accuracy of a function? Can I improve the performance of a function?)
3. Revise or enhance the code (if you do this, please see Appendix D!),
4. Try compiling several tens of megabytes of source code, this appendix is for you. Also check Appendix E.

First, you need to acquire the source code, and the GRASS Installation Guide. You may also want to get the GRASS Programmer's Manual and User's Reference Manual. To do this:

```
ftp moon.cecer.army.mil
login: anonymous
password: your email address
cd pub/grass/grass4.1/release/source
get README.4
get README.5
image
mget s4* (or s5*, your choice)
cd ../../documents
get installGuide.ps.Z
cd /manuals/programmer/postscript
get progman.ps.Z
cd ../../user/postscript
get refman.ps.Z
bye
```

TOC

Don't forget this site. There are several tutorials on some of GRASS' more advanced programs in the pub/grass/grass4.1/document directory. There are two options for source code (I'm only discussing GRASS v 4.14 here, though version 4.15 is also available) The pub/grass/outgoing directory contains many contributed functions (and many other candidates for enhancing your system).

Follow the README.4 file for installing GRASS version 4.14 (which is sometimes called version 4.1.4) source code. Follow the README.5 file for installing GRASS version 4.15 (which is sometimes called version 4.1.5) source code.

After installing the source code, uncompress and print installGuide.ps.Z (or the troff version, if you prefer that and got it from a neighboring directory). You might also want to uncompress and print refman.ps.Z and progman.ps.Z at the same time. Note that progman.ps.Z is called the programmer's manual, but also contains valuable information about data formats and directory structures. Advanced users may also want to know the GRASS system utilities, even if they won't be calling them in code.

Now, use the GRASS Installation Guide (from installGuide.ps.Z) to guide yourself through the installation. The thickness of this document may at first be intimidating. However, if you installed Linux yourself, you should be ready to tackle a GRASS installation. Don't be surprised if a function or two does not compile on your system. I have a couple of uncompiled functions on my own Linux system. Fortunately, these are functions that I don't use... Some day I'll get back to them, fix them, and compile them!?

Here is a late-breaking addition, on how to install the newly released GRASS 4.2 from Baylor University:http://www.baylor.edu/~grass This text is as provided by Baylor, unedited by myself due to its release only a few days ago. Please note the similarity with other installations..

GRASS 4.2 Quick Start
Installtion Instuctions

WARNING: These instructions pertain to the 4.2 release of GRASS Users are urged to consult the complete installation guide for more detailed instructions.

$GIS/src - This directory contains scripts and files used to compile GRASS. By running scripts and changing lists of programs you generate GRASS binaries for your system.

You may mount a disk containing GRASS source on different types of machines and compile without making source code copies. You follow the following instructions for each machine.

WARNING: These instructions presume that you have familiarity with UNIX, C, make, and shell scripting. Murphy's law visits GRASS regularly and your expertise in these matters will be the best defense against Mr. Murphy.

WARNING: These instructions and scripts have been used to compile GRASS on various machines. Please mail results of using this information for compiling GRASS on your platforms and operating system to:

grass@baylor.edu

DIRECTORY CONTENTS

```
GISGEN       script which will compile GRASS

MAKELINKS    script used after GISGEN to establish the user executable
             commands

VERSION      current version number and date of the GRASS release
```

```
generic/    system independent files need by gmake
              gmake     shell script which does compilations
              make.def  make variables
              make.tail some additional make rules

head/       gmake header file(s) for this site.  Header files are
              created by running the utils/setup command.

lists/      lists of programs to be compiled
              GRASS     standard GRASS programs
              local     site specific GRASS programs
              ...       architecture dependent GRASS programs

next_step/  files used by GISGEN to keep track of how far along
            it is in the compilation. Used to restart
            GISGEN (after a failure) where it left off.

utils/      contains the 'setup' script and all support scripts
            and files needed by 'setup'
```

COMPILATION STEPS OVERVIEW

```
(1) Generate files that contain location and machine specific make
   information.

(2) Edit files containing lists of location and machine specific
    programs to be compiled (generally printer, digitizer, and graphics
    drivers).

(3) Run GRASS compilation script

(4) Run GRASS program linking script

(5) Edit device driver configuratin files

(6) Compile GRASS contributed, alpha programs.

(7) Compile GRASS related and hybrid programs.
```

COMPILATION STEPS (DETAILS)

(1) Generate make support files

Each machine and location needs to have GRASS compiled in ways that specify different:

- compilation and load flags
- system libraries
- installation directories
- default data bases and locations

The shell script utils/setup assists you in define many of the make options and definitions that will become part of every compile-time generated makefile (about 350). It also creates your shell script for compiling individual GRASS programs - gmake4.2.

Run "utils/setup" and answer the questions.

The makefile portions are placed in the head/ under a name which you specify/approve in the utils/setup process. The executable shell script which directs compilation is placed in (by default) /usr/local/bin.

Examine the just created file in head/ to make sure things are ok. A brief description for each defined variable follows:

```
ARCH            = Key name identifying the architecture of the machine
                  on which you are compiling GRASS.
GISBASE         = Directory into which compiled GRASS will be contained
UNIX_BIN        = Local directory where the GRASS entry program and gmake
                  will be contained

DEFAULT_DATABASE= Directory where local GRASS data bases are contained
DEFAULT_LOCATION= GRASS data base that users get as the first default

COMPILE_FLAGS   = Compilation flags
LDFLAGS         = Load flags

TERMLIB         = System library containing low-level cursor movement
CURSES          = System library that supports screen cursor control
MATHLIB         = System math library
LIBRULE         = Method for archiving and randomizing libraries

USE_TERMIO      = Flag to use the termio library if available
USE_MTIO        = Flag to use the mtio library if available
CAN_CLEAR       = Flag indicating that the terminal screen can be cleared
DIGITFLAGS      = Flags to set owner and priority of the v.digit program
```

(2) Edit files containing lists of location and machine specified programs

The directory lists/ contains files that list directories that will be compiled. Directory names are relative to the GRASS src directory. The file lists/GRASS lists all basic GRASS programs that get compiled at every site. The file lists/local and lists/$ARCH.

```
-------------------------------------------------------------------
$ARCH is the architecture name you approved while running the
utils/setup script.  You can determine this by running:
    gmake4.2 -sh | grep ARCH
-------------------------------------------------------------------
```

There man not be a lists/$ARCH file, but you are free to create it to add names of programs you want compiled specifically for this architecture. It is an architecture-specific list which allows NFS linked source code to compile one set of programs for one machine, and another set for another machine. All machines that share the same source code via NFS mounts will compile the directories listed in lists/local.

All lists may contain comment lines - indicated by a # as the first character in the line. The lists/local file contains lists of all digitizer, graphics, and paint (hard-copy map) drivers. All machine specific devices are commented out - you must uncomment those that are particular to your site or architecture. You are encouraged to move the graphics driver items to the appropriate lists/$ARCH file.

(3) Run GRASS compilation program

The script GISGEN drives the compilation process. If all goes well you will be able to simply enter the command

GISGEN and wait. The entire compilation process takes from about 1/2 hour on the faster workstations to ab hours on the slower workstations.

GISGEN collects all of the directory names to be compiled from lists/GRASS lists/$ARCH and lists/local and begins running gmake4.2 in each directory. Screen output is a collection of messages from GISGEN and from the UNIX make program. Failure at any step will halt compilation. On failure you might do one of the following things:

```
1 - Fix a compilation problem by modifying code in the directory that
    failed.  After modification, return to this directory and rerun
    GISGEN.  Compilation will pick up at the failed directory and continue
    down the list of directories if successful.

2 - Restart GISGEN.  If the failure requires modifications to code already
    compiled, or the compilation options you set in step 1, you must
    remove next_step/$ARCH (or next_step/next_step if ar architecture name
    was not specified in step 2.  You may then rerun GISGEN.

3 - Skip the failed directory.  In this case you must seek through the
    files list/GRASS lists/$ARCH and lists/local to determine the directory
    name that follows the failed directory name.  The failed name is in
    next_step/$ARCH and must be replaced in that file with the next name.
    After editing, rerun GISGEN
```

When complete GISGEN will put the word DONE into the next_step file and will print the phrase "DONE generating GIS binary code" to the screen.

(4) Run GRASS program linking script

The GISGEN directs a compilation process that stashes the GRASS programs away in directories unavailable to the user community. Most user commands are actually links to a single program called "front.end". Links to this program must be made for every actual GRASS program. This is done AFTER GISGEN is finished. To make (or re-make) links for all user programs run the script MAKELINKS.

(5) Edit device driver configuratin files

Your compiled system may any combination of several graphics, paint, and digitizer drivers. Refer to the GRASS installation instructions for details.

NOTE: If you have trouble compiling your graphics driver, go to the directory $GIS/src/display/devices and compile the proper drivers manually using gmake4.2.

(6) Compile GRASS contributed, alpha programs.

GRASS programs come in three flavors:

MAIN - The programs are those compiled in step 3. They have stood the test of time and are generally reliable programs.

ALPHA - Alpha programs are intended to be included with the MAIN programs in the next release.

CONTRIB - Sites generate lots of special purpose programs in GRASS to get some job done, but do not polish the effort sufficiently to even be considered alpha code can be distributed in this category.

ALPHA programs are found in the directory src.alpha. You, the installer may visit these programs and compile any that you desire. In directories that contain Gmakefile files, simply run: gmake4.2

CONTRIB programs are in the directory src.contrib. The state of these programs are varied. Some programs may

TOC

compile with gmake4.2; others are suitable as a starting point for programmers who will be writing new softw

(7) Compile GRASS related and hybrid programs.

The GRASS user community has discovered that there are several public-domain programs that are very useful in conjunction with GRASS. These are found in the directory src.related. Compile these programs based on instructions (or lack of instructions) in the individual directories.

Hybrid programs are those that mix the capabilities of GRASS with the capabilities of one or more of the "related" programs. These are found in the src.garden directory. They require successful compilation of the "related" programs and generally compile using the gmake4.2 and the included Gmakefile files.

The rest of the compilation should just take some time. If you have already installed GRASS binaries, you should back up your system (or at least get the working binaries out of the way of your compilation!).

Good Luck! And be secure in the likelihood that you can use the compiled binaries if you bail out of a full compilation of the source code.

Appendix D: If you plan to enhance any part of GRASS, read this first!

GRASS has been developed for over a decade as completely unrestricted public domain source code and executables. Though there was initial resistance to the existence of such robust software in the public domain, many competitors learned to take advantage of GRASS. It has reputedly been the intellectual stimulus for several enhancements to other GISs. Several companies conducted business by installing and customizing public domain GRASS for customers, and by providing other value-added services such as data base development.

As USA/CERL no longer supports the public version of GRASS, users are free to use what currently exists. They're also currently completely on their own. At least where the public domain version is concerned.

There is a commercial version of GRASS:http://www.las.com/grassland, adapted from the public domain version by Logiciels et Applications Scientifiques (L.A.S) Inc. of Montreal . In a recent check, LAS sold its GRASSLAND for Sun, Linux and Windows NT. LAS is trying to encourage the continuation of a robust public domain Linux, partly as a source of new ideas and code for their own developments.

Appendix E: Example Linux versions of some critical GRASS files.

This appendix is the home of Linux-specific examples of selected GRASS configuration files. Currently, only several examples of a single file are offered. However, this is the most important file for configuration! Later, examples of database configuration files (e.g. DEFAULT_WIND) and other files may appear.

In the Installation Guide (pp. 10-11) you will see mention of the [header] file in directory $GIS/src/CMD/header (where $GIS is the directory in which you place GRASS - some folks put this in /usr/local - I put everything in a GRASS' own filesystem/directory /user/grass4.1). The installation guide favors Sun systems, as these were the development environment for GRASS4. (In case you cared, Masscomp workstations were earlier development environments.) Below are examples of this file for linux (which you might want to name linux in your $GIS/src/CMD/header directory. You may want to refer to this section when running the setup ($GIS/src/CMD/utils/setup) command.

One version:

```
CC                      = gcc
ARCH                    =

GISBASE                 = /user/grass4.1
```

TOC

```
UNIX_BIN            = /user/grass4.1/bin

DEFAULT_DATABASE    = /user/grass4.1/data
DEFAULT_LOCATION    = china

COMPILE_FLAGS       = -O2
LDFLAGS             = -s

XCFLAGS             = -D_NO_PROTO -DXM_1_1_BC
XLDFLAGS            =
XINCPATH            =
XMINCPATH           =
XLIBPATH            =
XTLIBPATH           = -L/usr/lib
XMLIBPATH           = -L/usr/lib
XLIB                = -lX11
XTLIB               = -lXt
XMLIB               = -lXm
XEXTRALIBS          =

TERMLIB             =
CURSES              = -lcurses $(TERMLIB)
MATHLIB             = -lm

#                   LIBRULE = ar ruv $@ $?
#                   LIBRULE = ar ruv $@ $?; ranlib $@
#                   LIBRULE = ar ruv $@ $?; ar ts $@
#                   LIBRULE = ar rc $@ `lorder $(OBJ) | tsort`
LIBRULE             = ar ruv $@ $?

USE_TERMIO          = -DUSE_TERMIO
USE_MTIO            = -DUSE_MTIO
USE_FTIME           = -DUSE_FTIME
DIGITFLAGS          = -DUSE_SETREUID -DUSE_SETPRIORITY
VECTLIBFLAGS        =
GETHOSTNAME         = -DGETHOSTNAME_OK
```

Another version:

```
#CC                  = gcc
#CC                  = gcc -ggdb -traditional
CC                  = gcc -traditional
#CC                  = gcc -static

ARCH                = linux

GISBASE             = /usr2/local/grass/grass4.1
UNIX_BIN            = /usr/local/bin

DEFAULT_DATABASE    = /usr2/local/grass
DEFAULT_LOCATION    = grass4.1
```

TOC

```
COMPILE_FLAGS        =
#COMPILE_FLAGS        = -O
LDFLAGS              = -s

XCFLAGS              = -D_NO_PROTO
XLDFLAGS             =
XINCPATH             = -I$GISBASE/xgrass
#XINCPATH             =
XMINCPATH            =
XLIBPATH             = -L/usr/lib
XTLIBPATH            = -L/usr/lib
XMLIBPATH            = -L/usr/lib
XLIB                 = -lX11
XTLIB                = -lXt
XMLIB                = -lXm
XEXTRALIBS           =

TERMLIB              =
CURSES               = -lcurses $(TERMLIB)
MATHLIB              = -lm

#                    LIBRULE = ar ruv $@ $?
#                    LIBRULE = ar ruv $@ $?; ranlib $@

#                    LIBRULE = ar ruv $@ $?; ar ts $@
#                    LIBRULE = ar rc $@ `lorder $(OBJ) | tsort`
LIBRULE              = ar ruv $@ $?; ranlib $@

USE_TERMIO           = -DUSE_TERMIO
USE_MTIO             = -DUSE_MTIO
USE_FTIME            = -DUSE_FTIME
DIGITFLAGS           = -DUSE_SETREUID -DUSE_SETPRIORITY
VECTLIBFLAGS         =
GETHOSTNAME          = -DGETHOSTNAME_OK
```

Another version:

```
#CC                   = gcc -traditional -ggdb
CC                   = gcc -traditional -m486
#CC                   = gcc
ARCH                 = linux

GISBASE              = /usr/local/grass/grass4.1
UNIX_BIN             = /usr/local/bin

DEFAULT_DATABASE     = /usr/local/grass
DEFAULT_LOCATION     = grass4.1

COMPILE_FLAGS        = -O2
LDFLAGS              = -s

XCFLAGS              = -D_NO_PROTO -DXM_1_1_BC
```

TOC

```
XLDFLAGS            =
XINCPATH            =
XMINCPATH           =
XLIBPATH            = -L/usr/lib
XTLIBPATH           = -L/usr/lib
XMLIBPATH           = -L/usr/lib
XLIB                = -lX11
XTLIB               = -lXt
XMLIB               = -lXm
XEXTRALIBS          = -lXmu

TERMLIB             =
CURSES              = -lcurses $(TERMLIB)
MATHLIB             = -lm

#                   LIBRULE = ar ruv $@ $?
#                   LIBRULE = ar ruv $@ $?; ranlib $@
#                   LIBRULE = ar ruv $@ $?; ar ts $@
#                   LIBRULE = ar rc $@ `lorder $(OBJ) | tsort`
LIBRULE             = ar ruv $@ $?; ranlib $@

#USE_TERMIO          = #-DUSE_TERMIO
USE_TERMIO          = -DUSE_TERMIO
USE_MTIO            = -DUSE_MTIO
USE_FTIME           = -DUSE_FTIME
DIGITFLAGS          = -DUSE_SETREUID -DUSE_SETPRIORITY
VECTLIBFLAGS        =
GETHOSTNAME         = -DGETHOSTNAME_OK
```

Yet another version:

```
CC                  = cc
ARCH                = linux

GISBASE             = /usr/local/grass4.15/linux
UNIX_BIN            = /usr/local/grass4.15/linux

DEFAULT_DATABASE    = /data/grassdata
DEFAULT_LOCATION    =

# -fwritable-strings - for ps.map only
#COMPILE_FLAGS       = -O -m486 -fwritable-strings
COMPILE_FLAGS       = -O -m486
LDFLAGS             = -s

XCFLAGS             = -D_NO_PROTO
XLDFLAGS            =
XINCPATH            =
XMINCPATH           =
XLIBPATH            = -L/usr/X11R6/lib
XTLIBPATH           = -L/usr/lib
XMLIBPATH           = -L/usr/lib
```

```
XLIB                    = -lX11
XTLIB                   = -lXt
XMLIB                   = -lXm
XEXTRALIBS              =

TERMLIB                 =
CURSES                  = -lcurses $(TERMLIB)
MATHLIB                 = -lm

#                       LIBRULE = ar ruv $@ $?
#                       LIBRULE = ar ruv $@ $?; ranlib $@
#                       LIBRULE = ar ruv $@ $?; ar ts $@
#                       LIBRULE = ar rc $@ `lorder $(OBJ) | tsort`
LIBRULE                 = ar ruv $@ $?

USE_TERMIO              = -DUSE_TERMIO
USE_MTIO                = -DUSE_MTIO
USE_FTIME               = -DUSE_FTIME
DIGITFLAGS              = -DUSE_SETREUID -DUSE_SETPRIORITY
VECTLIBFLAGS            = -DPORTABLE_3
GETHOSTNAME             = -DGETHOSTNAME_OK
```

Intimidating? It probably shouldn't be if you've configured X Windows on your Linux box. These examples should give you patterns to look for when running the setup utility in GRASS (described in the Installation Guide).

GTEK BBS-550 with Linux mini-HOWTO

by Wajihuddin Ahmed <wahmed@sdnp.org.pk>

v1.1, 20 August 1997

TOC

GTEK's BBS-550 is a 8-port serial card with 16C550 UARTS. Only one IRQ can be used for all 8 ports. It does not require any driver on Linux therefore the kernel does not have to be recompiled but the kernel should have serial support.

---

1. What `I/O` addresses to use?
2. The `rc.serial` file and `IRQ` selection
3. Setup of `uugetty`
4. Final checking...

---

1. For detailed information of configuring serial ports on Linux refer to the Serial-HOWTO. Thats where I started from.

I have setup the BBS-550 to use `I/O` address from `0x100` to `0x140` on my Linux box (Slackware 2.3, kernel 1.3.93). The main reason being that the above **address space was vacant** in my computer. Page 4 of the manual gives the jumper settings. Also this range of I/O addresses is seldom used by any other hardware. Just to make sure that this space is available on your machine, use the IOMAP.EXE utility which is on the GTEK floppy disk.

---

2. After you have installed the card on your computer **you need to configure the serial ports**. This is quite easy since the `rc.serial` file already has entries for BOCA 8 port card which can be used for the GTEK. Here is my `rc.serial` file. **Don't forget** to execute it from the `rc.S` file by uncommenting the line that executes it.

--

```
#!/bin/sh
# /etc/rc.serial
# Initializes the serial ports on your system
# Version 2.01
cd /dev
SETSERIAL="/bin/setserial -a"
PORTS=`echo cua? cua??`
echo -n "Configuring serial ports...."
# Do wild interrupt detection
#
#${SETSERIAL} -W ${PORTS}
```

TOC

```
# Do AUTOMATIC_IRQ probing
#
#AUTO_IRQ=auto_irq
${SETSERIAL} /dev/cua16 uart 16550A port 0x100 irq 5
${SETSERIAL} /dev/cua17 uart 16550A port 0x108 irq 5
${SETSERIAL} /dev/cua18 uart 16550A port 0x110 irq 5
${SETSERIAL} /dev/cua19 uart 16550A port 0x118 irq 5
${SETSERIAL} /dev/cua20 uart 16550A port 0x120 irq 5
#${SETSERIAL} /dev/cua21 uart 16550A port 0x128 irq 5
#${SETSERIAL} /dev/cua22 uart 16550A port 0x130 irq 5
#${SETSERIAL} /dev/cua23 uart 16550A port 0x138 irq 5
echo "done."
${SETSERIAL} -bg ${PORTS}
```

--

Notice that i have used IRQ 5. Refer to page 6 of the GTEK manual on howto set the IRQ.

---

3. I am using `uugetty` for dial-in's. Here is an extract from my `/etc/inittab` file (sys V).

```
s1:45:respawn:/sbin/uugetty ttyS16 38400 vt100
s2:45:respawn:/sbin/uugetty ttyS17 38400 vt100
s3:45:respawn:/sbin/uugetty ttyS18 38400 vt100
s4:45:respawn:/sbin/uugetty ttyS19 38400 vt100
s5:45:respawn:/sbin/uugetty ttyS20 57600 vt100
#s6:45:respawn:/sbin/uugetty ttyS21 38400 vt100
#s7:45:respawn:/sbin/uugetty ttyS22 38400 vt100
#s8:45:respawn:/sbin/uugetty ttyS23 38400 vt100
```

Make sure that you have the corresponding entries in the `/etc/gettydefs` file. Also don't forget to make new uugetty configuration files in the `/etc/default` directory.

---

4. At boot up the kernel **will not** report the new tty's.

The serial ports are configured when the `rc.serial` script is run. You can see that on the screen during boot up.

`ps ax` **should show the uugetty connected**. Here is part of my `ps` output.

```
18208 ? S 0:00 /sbin/uugetty ttyS18 38400 vt100
18247 ? S 0:00 /sbin/uugetty ttyS20 57600 vt100
```

TOC

```
18254 ? S 0:00 /sbin/uugetty ttyS19 38400 vt100
18258 ? S 0:00 /sbin/uugetty ttyS17 38400 vt100
18260 ? S 0:00 /sbin/uugetty ttyS16 38400 vt100
```

Finally test the board by dialing out (using `cuaXX`) and then dialing in (`ttySXX`).

Good Luck!

Hard Disk Upgrade Mini How-To

Yves Bellefeuille,
yan@ottawa.com

Version 1.0,
31 January 1998

How to copy a Linux system from one hard disk to another.

---

1. Install both disks on your system
2. Unmount non-Linux partitions
3. Partition the new disk
4. Format the new disk
5. Mount the new disk
6. Copy the files from the old disk to the new disk
7. Modify /etc/fstab as appropriate
8. Prepare LILO to boot the new disk
9. Remove the old disk
10. Reboot the system, install LILO on the new disk

Recently, I replaced my small 249 Mb hard disk with a larger disk. I wanted to transfer my entire Linux system, including LILO, from the old disk to the new disk. This is how I did it.

In the following explanation, I use "`/dev/hda`" to denote the "old" disk, and "`/dev/hda1`" means the old Linux partition. "`/dev/hdb`" means the "new" disk, and "`/dev/hdb1`" means the new Linux partition.

Therefore, I'm assuming that Linux is on the first partition of the first disk. Modify this as appropriate for your set-up.

This document is based on my own system, running Red Hat 4.2, and I've tested all the commands that follow with that distribution. I've also tested them under Debian 1.3.1 and Slackware 3.3, and I indicate a few differences to note if you're using those distributions.

If the commands don't work properly on your system, please let me know, telling me what version of Linux you're using.

---

1. Install both disks on your system

Modern systems can accept four "EIDE" devices on the hard disk controller, so there shouldn't be any problem installing both disks on your system at the same time, even if you also have other EIDE devices. Hard disks and CD-ROM drives are typical EIDE devices. Floppy drives and tape drives are usually connected to the floppy drive controller rather than to the hard disk controller.

TOC

SCSI adapters are even more flexible and can accept seven devices. If you're lucky (and rich) enough to have SCSI adapter, you probably already know this, and you probably know which of your devices are SCSI devi For more information, see the SCSI How-To.

Even the oldest systems can accept two devices on the hard disk controller, so you can still install both hard disks at the same time. However, if you already have another device installed in addition to your hard disk (for example, if you have both a hard disk and a CD-ROM drive), you'll have to remove the other device to be able to install the old hard disk and the new hard disk at the same time.

You must configure the disks as "master" or "slave" by installing the disks' jumpers as appropriate. You'll often find configuration information on the disks themselves; if not, consult the manuals or the disks' manufacturers.

You must also inform the BIOS of the disks' presence and of their "geometry". Usually, you enter the BIOS setup programme by pressing a key during the system boot-up. Here's what to do for some common BIOSes:

American Megatrends (AMI): Del key during Power-On Self-Test (POST)

Award: Ctrl-Alt-Esc

Compaq: F10 key after the square appears in the top right corner of the screen during boot-up

Dell: Ctrl-Alt-Enter

DTK: Esc key during Power-On Self-Test

IBM PS/2: Ctrl-Alt-Del, then Ctrl-Alt-Ins when the cursor is in the top right corner

Phoenix: Ctrl-Alt-Esc, or Ctrl-Alt-S, or Ctrl-Alt-Enter

Many older systems require an Installation or Reference Disk.

(I'm interested in receiving information on other BIOSes to add them to this list.)

Reboot the system and login as `root`.

---

2. Unmount non-Linux partitions

Some people like to mount partitions from other operating systems (DOS, Windows, OS/2, etc.) so they can use them under Linux. These partitions must be created and copied under their own operating system, and you should unmount them before copying your Linux partition. For example, if you have a DOS partition mounted at `/dos`, you must unmount it with this command:

```
umount /dos
```

Note that the command is "`umount`", with the first letter "n" missing from the word "unmount".

---

3. Partition the new disk

Use this command to partition the new disk:

```
fdisk /dev/hdb
```

For more information on partitioning, see the Installation How-To and the Partitioning Mini How-To.

TOC

If your new disk has over 1024 cylinders, see the Large Disk Mini How-To. In brief, you should install all fi required to boot Linux within the first 1024 cylinders. One way to do this is to create a small partition (1 Mb or 2 Mb) just for the `/boot` directory at the beginning of the disk. (**Slackware only:** The kernel is at `/vmlinuz` rather than `/boot/vmlinuz`, so you should put both the `/` directory and the `/boot` directory in this partition.)

Partitions for systems other than Linux should be created using their own `fdisk` or equivalent command rather than with Linux's `fdisk`.

---

4. Format the new disk

Use the following command to format the new disk:

```
mkfs.ext2 /dev/hdb1
```

To check the disk for bad blocks (physical defects), add the `-c` option just before "`/dev/hdb1`".

(Note: Contrary to what the `man` page states, the command "`mkfs -t ext2 -c /dev/hdb1`" doesn't check for bad blocks under any of Red Hat, Debian or Slackware.)

---

5. Mount the new disk

Create a directory where you'll mount the new disk, for example `/new-disk`, and mount it there:

```
mkdir /new-disk
mount -t ext2 /dev/hdb1 /new-disk
```

---

6. Copy the files from the old disk to the new disk

You want to completely reproduce the disk structure, including links.

However, you *don't* want to copy the directory `/new-disk`, since this would copy the new disk to itself!

Furthermore, you want to create the `/proc` directory on the new disk, but you don't want to copy its contents: `/proc` is a "virtual" file system and doesn't have any actual files, but rather contains information on the processes running on the system.

Here are four different ways to copy the old disk to the new one. This may take quite a while, especially if you have a large disk or little memory. You can expect to be able to copy 10 Mb per minute, and possibly much more.

You can follow the copy's progress by using the command "`df`" from another terminal. If you're as easily amused as I am, try "`watch df`" or "`watch ls -l /new-disk`" to see a report updated every two seconds; press `Ctrl-C` to end the display. Be aware that running the "`watch`" programme itself will slow down the copying.

i. `cp -ax / /new-disk`

   This is the simplest method, but will only work if your original Linux system is on a single disk partition. The `-a` option preserves the original system as much as possible. The `-x` option limits `cp` to a single file system; this is necessary to avoid copying the `/new-disk` and `/proc` directories.

TOC

ii. `cd / && cp -a `/bin/ls -1A | egrep -v "^new-disk$|^proc$"` /new-disk`

(Write this all on one line.)

This goes to the root directory and then copies all files and directories except `/new-disk` and `/proc` to `/new-disk`. Note that the first option after `ls` is the number 1, not the letter L!

This command should work in all circumstances.

iii. `(cd / && tar cpf - . --exclude new-disk --exclude proc) | (cd /new-disk && tar xpf -)`

(Write this all on one line.)

This goes to the root directory, "tars" everything except `/new-disk` and `/proc`, switches to `/new-disk` and "untars" everything there. Note that there must not be a slash before or after the names of the directories in the `--exclude` options.

(Note: The option `-l` doesn't work here, since `tar` will still re-create the directories `/new-disk` and `/proc` even though it doesn't copy their contents. Therefore, `tar`'s `-l` option doesn't have the same behaviour as `cp`'s `-x` option.)

This method is somewhat slower than the others.

iv. `cp -a /bin /boot /dev /etc /home /lib /lost+found /mnt /root /sbin /tmp /usr /var /new-disk`

(Write this all on one line.)

The last directory, `/new-disk`, is the destination for the `cp` command. All the other directories are the sources. Therefore, here I'm copying all the directories I'm listing to `/new-disk`.

With this method, you simply list yourself the directories you want to copy. Here I listed all my directories except `/new-disk` and `/proc`. If you can't use the other methods for any reason, you can always use this command to manually specify the directories you want to copy.

With this method only, if there are any files in the root directory itself, you need another command to copy them. In particular, this is required with Debian and Slackware, since these distributions put files in the root directory:

```
cp -dp /* /.* /new-disk
```

After using any of these four methods, you must also create the `/proc` directory on the new disk:

```
mkdir /new-disk/proc
```

At this point, you may verify the file structure on the new disk, if you wish:

```
umount /new-disk
fsck.ext2 -f /dev/hdb1
mount -t ext2 /dev/hdb1 /new-disk
```

You may also use the following script to compare the two disks, to ensure that the files were copied properly:

```
#!/bin/sh
```

TOC

```
cd /
for file in `/bin/ls -1A | egrep -v '^new-disk$|^proc$'`
do
    find $file -xtype f -exec cmp \{\} /new-disk/\{\} \;
done
```

(**Slackware only:** A basic Slackware installation doesn't include the "`cmp`" or "`diff`" commands, so you won't be able to run this script if you have only installed the basic files.)

This will only compare regular files, not character or block special files (in the `/dev` directory), sockets, etc., since the "`cmp`" command doesn't work properly with these. I would welcome suggestions on how to verify these "special" files.

---

7. Modify `/etc/fstab` as appropriate

If your new disk doesn't have the same partitions or organization as the old disk, modify the file `/etc/fstab` on the new disk as appropriate. Remember that this file is currently located at `/new-disk/etc/fstab`.

Make sure that the disk partitions in the first column correspond to the organization you'll have with the new disk, once the old disk has been removed, and that you're only mounting one partition at "`/`" as shown in the second column.

---

8. Prepare LILO to boot the new disk

This is the most complicated step. I'm assuming that LILO is installed on the hard disk's Master Boot Record (MBR); this seems to be the most common configuration.

You want to install LILO on what's presently the second hard disk. It's clear that LILO can't *run* from the second hard disk; however, LILO's documentation does anticipate that you might want to *install* LILO on the second hard disk, for example if the first hard disk will be removed:

> LILO can't be stored at any of the following locations:
>
> - on the second hard disk. (Unless for backup purposes, if the current first disk will be removed or disabled, or if some other boot loader is used, that is capable of loading boot sectors from other drives.)

However, the documentation doesn't explain the proper way to install LILO on the second hard disk if the first hard disk will be removed, and I've concluded after many attempts that it isn't possible to install LILO directly onto the MBR of the second hard disk and have it work correctly the first time.

Instead, I suggest the use of a boot diskette to boot the new hard disk the first time.

Insert an empty diskette, format it, create a file system on it and mount it:

```
fdformat /dev/fd0H1440
mkfs.ext2 /dev/fd0
mount -t ext2 /dev/fd0 /mnt
```

(**Debian only:** The command "`fdformat`" is not included in a basic installation with Debian. If you don't have this command, you may omit it if the floppy is already formatted. In this case, you should check the diskette for

bad blocks by adding "`-c`" after the "`mkfs.ext2`" command.

(**Debian and Slackware only:** Use the command "`fdformat /dev/fd0h1440`", with a lower case "h".)

Copy all files in `/boot` to the diskette:

```
cp -dp /boot/* /mnt
```

(**Slackware only:** Copy the file `/vmlinuz` to the boot diskette; use the command "`cp /vmlinuz /mnt`".)

Create a new file `/mnt/lilo.conf` as follows:

```
boot=/dev/fd0           # Install LILO on floppy disk.
map=/mnt/map            # Location of "map file".
install=/mnt/boot.b     # File to copy to floppy's boot sector.
prompt                  # Have LILO show "LILO boot:" prompt.
timeout=50              # Boot default system after 5 seconds.
                        # (Value is in tenths of seconds.)
image=/mnt/vmlinuz      # Location of Linux kernel on floppy.
    label=linux         # Label for Linux system.
    root=/dev/hda1      # Location of root partition on new hard
                        # disk. Modify this as appropriate for
                        # your system.
                        # Note that you must use the name of the
                        # future location, once the old disk has
                        # been removed.
```

(**Debian only:** In the "`image`" line, use the actual name of the Linux kernel. For example, with Debian 1.3.1, use "`/mnt/vmlinuz-2.0.29`".)

Install LILO on the boot diskette:

```
/sbin/lilo -C /mnt/lilo.conf
```

The `-C` option tells `/sbin/lilo` what configuration file to use.

Unmount the diskette:

```
umount /mnt
```

and shut down the system.

---

9. Remove the old disk

After removing the old disk, remember to modify the disk jumpers and the BIOS information to reflect the changes.

---

10. Reboot the system, install LILO on the new disk

Reboot the system from the boot diskette you just made. To do so, you may have to modify your BIOS's boot-up sequence to "`A:, C:`".

TOC

Make any necessary changes to the `/etc/lilo.conf` file, and run `/sbin/lilo` to install LILO on the r disk. With Debian, make sure that the "`boot`" line says "`/dev/hda`" rather than "`/dev/hda1`" or similar you want to install LILO on the Master Boot Record.

You can then try re-booting your system from your new hard disk to test if everything is working properly. If you run into any problems, you can still use the diskette you just made to boot your system.

Acknowledgements

Special thanks to **Dr Konrad Hinsen** of the Institut de biologie structurale, Grenoble, France, who has been kindly acting as my personal Linux guru. Thanks also to **Frank Damgaard**, **Paul Koning** and **Josh Rabinowitz**, and to **Scott Christensen** for alerting me to some particularities of the Slackware distribution.

Linux I/O port programming mini-HOWTO

Author: Riku Saikkonen <Riku.Saikkonen@hut.fi>

v, 28 December 1997

This HOWTO document describes programming hardware I/O ports and waiting for small periods of time in user-mode Linux programs running on the Intel x86 architecture.

1. Introduction

2. Using I/O ports in C programs

- 2.1 The normal method
- 2.2 An alternate method: `/dev/port`

3. Interrupts (IRQs) and DMA access

4. High-resolution timing

- 4.1 Delays
- 4.2 Measuring time

5. Other programming languages

6. Some useful ports

- 6.1 The parallel port
- 6.2 The game (joystick) port
- 6.3 The serial port

TOC

7. Hints

8. Troubleshooting

9. Example code

10. Credits

1. Introduction

This HOWTO document describes programming hardware I/O ports and waiting for small periods of time in user-mode Linux programs running on the Intel x86 architecture. This document is a descendant of the very small IO-Port mini-HOWTO by the same author.

This document is Copyright 1995-1997 Riku Saikkonen. See the Linux HOWTO copyright for details.

If you have corrections or something to add, feel free to e-mail me (`Riku.Saikkonen@hut.fi`)...

Changes from the previous released version (Mar 30 1997):

- Clarified things regarding `inb_p`/`outb_p` and port 0x80.
- Removed information about `udelay()`, since `nanosleep()` provides a cleaner way of using it.
- Converted to Linuxdoc-SGML, and reorganised somewhat.
- Lots of minor additions and modifications.

2. Using I/O ports in C programs

2.1 The normal method

Routines for accessing I/O ports are in `/usr/include/asm/io.h` (or `linux/include/asm-i386/io.h` in the kernel source distribution). The routines there are inline macros, so it is enough to `#include <asm/io.h>`; you do not need any additional libraries.

Because of a limitation in gcc (present at least in 2.7.2.3 and below) and in egcs (all versions), you *have to* compile any source code that uses these routines with optimisation turned on (`gcc -O1` or higher), or alternatively `#define extern` to be empty before you `#include <asm/io.h>`.

For debugging, you can use `gcc -g -O` (at least with modern versions of gcc), though optimisation can sometimes make the debugger behave a bit strangely. If this bothers you, put the routines that use I/O port access in a separate source file and compile only that with optimisation turned on.

Before you access any ports, you must give your program permission to do so. This is done by calling the `ioperm()` function (declared in `unistd.h`, and defined in the kernel) somewhere near the start of your program (before any I/O port accesses). The syntax is `ioperm(from, num, turn_on)`, where `from` is the first port number to give access to, and `num` the number of consecutive ports to give access to. For example, `ioperm(0x300, 5, 1)` would give access to ports 0x300 through 0x304 (a total of 5 ports). The last argument is a Boolean value specifying whether to give access to the program to the ports (true (1)) or to remove access (false (0)). You can call `ioperm()` multiple times to enable multiple non-consecutive ports. See the `ioperm(2)` manual page for details on the syntax.

The `ioperm()` call requires your program to have root privileges; thus you need to either run it as the root user, or make it setuid root. You can drop the root privileges after you have called `ioperm()` to enable the ports you want to use. You are not required to explicitly drop your port access privileges with `ioperm(..., 0)` at the end of your program; this is done automatically as the process exits.

A `setuid()` to a non-root user does not disable the port access granted by `ioperm()`, but a `fork()` does (the child process does not get access, but the parent retains it).

`ioperm()` can only give access to ports 0x000 through 0x3ff; for higher ports, you need to use `iopl()` (which gives you access to all ports at once). Use the level argument 3 (i.e., `iopl(3)`) to give your program access to *all* I/O ports (so be careful --- accessing the wrong ports can do all sorts of nasty things to your computer). Again, you need root privileges to call `iopl()`. See the `iopl(2)` manual page for details.

Then, to actually accessing the ports... To input a byte (8 bits) from a port, call `inb(port)`, it returns the byte it got. To output a byte, call `outb(value, port)` (please note the order of the parameters). To input a word (16 bits) from ports `x` and `x+1` (one byte from each to form the word, using the assembler instruction `inw`), call `inw(x)`. To output a word to the two ports, use `outw(value, x)`. If you're unsure of which port instructions (byte or word) to use, you probably want `inb()` and `outb()`

TOC

--- most devices are designed for bytewise port access. Note that all port access instructions take at l about a microsecond to execute.

The `inb_p()`, `outb_p()`, `inw_p()`, and `outw_p()` macros work otherwise identically to the ones above, but they do an additional short (about one microsecond) delay after the port access; you can make the delay about four microseconds with `#define REALLY_SLOW_IO` before you `#include <asm/io.h>`. These macros normally (unless you `#define SLOW_IO_BY_JUMPING`, which is probably less accurate) use a port output to port 0x80 for their delay, so you need to give access to port 0x80 with `ioperm()` first (outputs to port 0x80 should not affect any part of the system). For more versatile methods of delaying, read on.

There are man pages for `ioperm(2)`, `iopl(2)`, and the above macros in reasonably recent releases of the Linux manual page collection.

2.2 An alternate method: `/dev/port`

Another way to access I/O ports is to `open()` `/dev/port` (a character device, major number 1, minor 4) for reading and/or writing (the stdio `f*()` functions have internal buffering, so avoid them). Then `lseek()` to the appropriate byte in the file (file position 0 = port 0x00, file position 1 = port 0x01, and so on), and `read()` or `write()` a byte or word from or to it.

Naturally, for this to work your program needs read/write access to `/dev/port`. This method is probably slower than the normal method above, but does not need compiler optimisation nor `ioperm()`. It doesn't need root access either, if you give a non-root user or group access to `/dev/port` --- but this is a very bad thing to do in terms of system security, since it is possible to hurt the system, perhaps even gain root access, by using `/dev/port` to access hard disks, network cards, etc. directly.

---

3. Interrupts (IRQs) and DMA access

You cannot use IRQs or DMA directly from a user-mode process. You need to write a kernel driver; see The Linux Kernel Hacker's Guide for details and the kernel source code for examples.

Also, you cannot disable interrupts from within a user-mode program.

4. High-resolution timing

4.1 Delays

First of all, I should say that you cannot guarantee user-mode processes to have exact control of timing because of the multi-tasking nature of Linux. Your process might be scheduled out at any time for anything from about 10 milliseconds to a few seconds (on a system with very high load). However, for most applications using I/O ports, this does not really matter. To minimise this, you may want to nice your process to a high-priority value (see the `nice(2)` manual page) or use real-time scheduling (see below).

If you want more precise timing than normal user-mode processes give you, there are some provisions for user-mode `real time' support. Linux 2.x kernels have soft real time support; see the manual page for `sched_setscheduler(2)` for details. There is a special kernel that supports hard real time; see http://luz.cs.nmt.edu/~rtlinux/ for more information on this.

Sleeping: `sleep()` and `usleep()`

Now, let me start with the easier timing calls. For delays of multiple seconds, your best bet is probably to use `sleep()`. For delays of at least tens of milliseconds (about 10 ms seems to be the minimum delay), `usleep()` should work. These functions give the CPU to other processes (``sleep''), so CPU time isn't wasted. See the manual pages `sleep(3)` and `usleep(3)` for details.

For delays of under about 50 milliseconds (depending on the speed of your processor and machine, and the system load), giving up the CPU takes too much time, because the Linux scheduler (for the x86 architecture) usually takes at least about 10-30 milliseconds before it returns control to your process. Due to this, in small delays, `usleep(3)` usually delays somewhat more than the amount that you specify in the parameters, and at least about 10 ms.

`nanosleep()`

In the 2.0.x series of Linux kernels, there is a new system call, `nanosleep()` (see the `nanosleep(2)` manual page), that allows you to sleep or delay for short times (a few microseconds or more).

For delays <= 2 ms, if (and only if) your process is set to soft real time scheduling (using `sched_setscheduler()`), `nanosleep()` uses a busy loop; otherwise it sleeps, just like `usleep()`.

The busy loop uses `udelay()` (an internal kernel function used by many kernel drivers), and the length of the loop is calculated using the BogoMips value (the speed of this kind of busy loop is one of the things that BogoMips measures accurately). See `/usr/include/asm/delay.h`) for details on how

it works.

Delaying with port I/O

Another way of delaying small numbers of microseconds is port I/O. Inputting or outputting any byte from/to port 0x80 (see above for how to do it) should wait for almost exactly 1 microsecond independent of your processor type and speed. You can do this multiple times to wait a few microseconds. The port output should have no harmful side effects on any standard machine (and some kernel drivers use it). This is how `{in|out}[bw]_p()` normally do the delay (see `asm/io.h`).

Actually, a port I/O instruction on most ports in the 0-0x3ff range takes almost exactly 1 microsecond, so if you're, for example, using the parallel port directly, just do additional `inb()`s from that port to delay.

Delaying with assembler instructions

If you know the processor type and clock speed of the machine the program will be running on, you can hard-code shorter delays by running certain assembler instructions (but remember, your process might be scheduled out at any time, so the delays might well be longer every now and then). For the table below, the internal processor speed determines the number of clock cycles taken; e.g., for a 50 MHz processor (e.g. 486DX-50 or 486DX2-50), one clock cycle takes 1/50000000 seconds (=200 nanoseconds).

```
Instruction   i386 clock cycles   i486 clock cycles
nop                  3                   1
xchg %ax,%ax         3                   3
or %ax,%ax           2                   1
mov %ax,%ax          2                   1
add %ax,0            2                   1
```

(Sorry, I don't know about Pentiums; probably close to the i486. I cannot find an instruction which would use one clock cycle on an i386. Use the one-clock-cycle instructions if you can, otherwise the pipelining used in modern processors may shorten the times.)

The instructions `nop` and `xchg` in the table should have no side effects. The rest may modify the flags register, but this shouldn't matter since gcc should detect it. `nop` is a good choice.

To use these, call `asm("instruction")` in your program. The syntax of the instructions is as in the table above; if you want multiple instructions in a single `asm()` statement, separate them with semicolons. For example, `asm("nop ; nop ; nop ; nop")` executes four `nop` instructions, delaying for four clock cycles on i486 or Pentium processors (or 12 clock cycles on an i386).

`asm()` is translated into inline assembler code by gcc, so there is no function call overhead.

Shorter delays than one clock cycle are impossible in the Intel x86 architecture.

TOC

`rdtsc` for Pentiums

For Pentiums, you can get the number of clock cycles elapsed since the last reboot with the following C code:

```
extern __inline__ unsigned long long int rdtsc()
{
  unsigned long long int x;
  __asm__ volatile (".byte 0x0f, 0x31" : "=A" (x));
  return x;
}
```

You can poll this value to delay for as many clock cycles as you want.

4.2 Measuring time

For times accurate to one second, it is probably easiest to use `time()`. For more accurate times, `gettimeofday()` is accurate to about a microsecond (but see above about scheduling). For Pentiums, the `rdtsc` code fragment above is accurate to one clock cycle.

If you want your process to get a signal after some amount of time, use `setitimer()` or `alarm()`. See the manual pages of the functions for details.

5. Other programming languages

The description above concentrates on the C programming language. It should apply directly to C++ and Objective C. In assembler, you have to call `ioperm()` or `iopl()` as in C, but after that you can use the I/O port read/write instructions directly.

In other languages, unless you can insert inline assembler or C code into the program or use the system calls mentioned above, it is probably easiest to write a simple C source file with functions for the I/O port accesses or delays that you need, and compile and link it in with the rest of your program. Or use `/dev/port` as described above.

6. Some useful ports

Here is some programming information for common ports that can be directly used for general-purpose TTL (or CMOS) logic I/O.

If you want to use these or other common ports for their intended purpose (e.g., to control a normal printer or modem), you should most likely use existing drivers (which are usually included in the kernel) instead of programming the ports directly as this HOWTO describes. This section is intended for those people who want to connect LCD displays, stepper motors, or other custom electronics to a PC's standard ports.

If you want to control a mass-market device like a scanner (that has been on the market for a while), look for an existing Linux driver for it. The Hardware-HOWTO is a good place to start.

http://www.hut.fi/Misc/Electronics/ is a good source for more information on connecting devices to computers (and on electronics in general).

6.1 The parallel port

The parallel port's base address (called ``BASE'' below) is 0x3bc for `/dev/lp0`, 0x378 for `/dev/lp1`, and 0x278 for `/dev/lp2`. If you only want to control something that acts like a normal printer, see the Printing-HOWTO.

In addition to the standard output-only mode described below, there is an `extended' bidirectional mode in most parallel ports. For information on this and the newer ECP/EPP modes (and the IEEE 1284 standard in general), see http://www.fapo.com/ and http://www.senet.com.au/~cpeacock/parallel.htm. Remember that since you cannot use IRQs or DMA in a user-mode program, you will probably have to write a kernel driver to use ECP/EPP; I think someone is writing such a driver, but I don't know the details.

The port `BASE+0` (Data port) controls the data signals of the port (D0 to D7 for bits 0 to 7, respectively; states: 0 = low (0 V), 1 = high (5 V)). A write to this port latches the data on the pins. A read returns the data last written in standard or extended write mode, or the data in the pins from another device in extended read mode.

The port `BASE+1` (Status port) is read-only, and returns the state of the following input signals:

- Bits 0 and 1 are reserved.
- Bit 2 IRQ status (not a pin, I don't know how this works)
- Bit 3 ERROR (1=high)
- Bit 4 SLCT (1=high)
- Bit 5 PE (1=high)
- Bit 6 ACK (1=high)
- Bit 7 -BUSY (0=high)

(I'm not sure about the high and low states.)

TOC

The port BASE+2 (Control port) is write-only (a read returns the data last written), and controls the following status signals:

- Bit 0 -STROBE (0=high)
- Bit 1 AUTO_FD_XT (1=high)
- Bit 2 -INIT (0=high)
- Bit 3 SLCT_IN (1=high)
- Bit 4 enables the parallel port IRQ (which occurs on the low-to-high transition of ACK) when set to 1.
- Bit 5 controls the extended mode direction (0 = write, 1 = read), and is completely write-only (a read returns nothing useful for this bit).
- Bits 6 and 7 are reserved.

(Again, I am not sure about the high and low states.)

Pinout (a 25-pin female D-shell connector on the port) (i=input, o=output):

```
1io -STROBE, 2io D0, 3io D1, 4io D2, 5io D3, 6io D4, 7io D5, 8io D6,
9io D7, 10i ACK, 11i -BUSY, 12i PE, 13i SLCT, 14o AUTO_FD_XT,
15i ERROR, 16o -INIT, 17o SLCT_IN, 18-25 Ground
```

The IBM specifications say that pins 1, 14, 16, and 17 (the control outputs) have open collector drivers pulled to 5 V through 4.7 kiloohm resistors (sink 20 mA, source 0.55 mA, high-level output 5.0 V minus pullup). The rest of the pins sink 24 mA, source 15 mA, and their high-level output is min. 2.4 V. The low state for both is max. 0.5 V. Non-IBM parallel ports probably deviate from this standard. For more information on this, see http://www.hut.fi/Misc/Electronics/circuits/lptpower.html.

Finally, a warning: Be careful with grounding. I've broken several parallel ports by connecting to them while the computer is turned on. It might be a good thing to use a parallel port not integrated on the motherboard for things like this. (You can usually get a second parallel port for your machine with a cheap standard `multi-I/O' card; just disable the ports that you don't need, and set the parallel port I/O address on the card to a free address. You don't need to care about the parallel port IRQ, since it isn't normally used.)

6.2 The game (joystick) port

The game port is located at port addresses 0x200-0x207. For controlling normal joysticks, there is a kernel-level joystick driver, see ftp://sunsite.unc.edu/pub/Linux/kernel/patches/, filename `joystick-*`.

Pinout (a 15-pin female D-shell connector on the port):

- 1,8,9,15: +5 V (power)
- 4,5,12: Ground
- 2,7,10,14: Digital inputs BA1, BA2, BB1, and BB2, respectively
- 3,6,11,13: ``Analog'' inputs AX, AY, BX, and BY, respectively

The +5 V pins seem to often be connected directly to the power lines in the motherboard, so they should be able to source quite a lot of power, depending on the motherboard, power supply and game port.

The digital inputs are used for the buttons of the two joysticks (joystick A and joystick B, with two buttons each) that you can connect to the port. They should be normal TTL-level inputs, and you can read their status directly from the status port (see below). A real joystick returns a low (0 V) status when the button is pressed and a high (the 5 V from the power pins through an 1 Kohm resistor) status otherwise.

The so-called analog inputs actually measure resistance. The game port has a quad one-shot multivibrator (a 558 chip) connected to the four inputs. In each input, there is a 2.2 Kohm resistor between the input pin and the multivibrator output, and a 0.01 uF timing capacitor between the multivibrator output and the ground. A real joystick has a potentiometer for each axis (X and Y), wired between +5 V and the appropriate input pin (AX or AY for joystick A, or BX or BY for joystick B).

The multivibrator, when activated, sets its output lines high (5 V) and waits for each timing capacitor to reach 3.3 V before lowering the respective output line. Thus the high period duration of the multivibrator is proportional to the resistance of the potentiometer in the joystick (i.e., the position of the joystick in the appropriate axis), as follows:

$$R = (t - 24.2) / 0.011,$$

where R is the resistance (ohms) of the potentiometer and t the high period duration (seconds).

Thus, to read the analog inputs, you first activate the multivibrator (with a port write; see below), then poll the state of the four axes (with repeated port reads) until they drop from high to low state, measuring their high period duration. This polling uses quite a lot of CPU time, and on a non-realtime multitasking system like (normal user-mode) Linux, the result is not very accurate because you cannot poll the port constantly (unless you use a kernel-level driver and disable interrupts while polling, but this wastes even more CPU time). If you know that the signal is going to take a long time (tens of ms) to go down, you can call usleep() before polling to give CPU time to other processes.

The only I/O port you need to access is port 0x201 (the other ports either behave identically or do nothing). Any write to this port (it doesn't matter what you write) activates the multivibrator. A read from this port returns the state of the input signals:

- Bit 0: AX (status (1=high) of the multivibrator output)
- Bit 1: AY (status (1=high) of the multivibrator output)
- Bit 2: BX (status (1=high) of the multivibrator output)
- Bit 3: BY (status (1=high) of the multivibrator output)
- Bit 4: BA1 (digital input, 1=high)
- Bit 5: BA2 (digital input, 1=high)
- Bit 6: BB1 (digital input, 1=high)
- Bit 7: BB2 (digital input, 1=high)

6.3 The serial port

If the device you're talking to supports something resembling RS-232, you should be able to use the serial port to talk to it. The Linux serial driver should be enough for almost all applications (you shouldn't have to program the serial port directly, and you'd probably have to write a kernel driver to do it); it is quite versatile, so using non-standard bps rates and so on shouldn't be a problem.

TOC

See the `termios(3)` manual page, the serial driver source code (`linux/drivers/char/serial.c`), and http://www.easysw.com/~mike/serial/index.html for more information on programming serial ports on Unix systems.

---

7. Hints

If you want good analog I/O, you can wire up ADC and/or DAC chips to the parallel port (hint: for power, use the game port connector or a spare disk drive power connector wired to outside the computer case, unless you have a low-power device and can use the parallel port itself for power, or use an external power supply), or buy an AD/DA card (most of the older/slower ones are controlled by I/O ports). Or, if you're satisfied with 1 or 2 channels, inaccuracy, and (probably) bad zeroing, a cheap sound card supported by the Linux sound driver should do (and it's quite fast).

With accurate analog devices, improper grounding may generate errors in the analog inputs or outputs. If you experience something like this, you could try electrically isolating your device from the computer with optocouplers (on *all* signals between the computer and your device). Try to get power for the optocouplers from the computer (spare signals on the port may give enough power) to achieve better isolation.

If you're looking for printed circuit board design software for Linux, there is a free X11 application called Pcb that should do a nice job, at least if you aren't doing anything very complex. It is included in many Linux distributions, and available in ftp://sunsite.unc.edu/pub/Linux/apps/circuits/ (filename `pcb-*`).

TOC

8. Troubleshooting

Q1.

I get segmentation faults when accessing ports.

A1.

Either your program does not have root privileges, or the `ioperm()` call failed for some other reason. Check the return value of `ioperm()`. Also, check that you're actually accessing the ports that you enabled with `ioperm()` (see Q3). If you're using the delaying macros (`inb_p()`, `outb_p()`, and so on), remember to call `ioperm()` to get access to port 0x80 too.

Q2.

I can't find the `in*()`, `out*()` functions defined anywhere, and gcc complains about undefined references.

A2.

You did not compile with optimisation turned on (`-O`), and thus gcc could not resolve the macros in `asm/io.h`. Or you did not `#include <asm/io.h>` at all.

Q3.

`out*()` doesn't do anything, or does something weird.

A3.

Check the order of the parameters; it should be `outb(value, port)`, not `outportb(port, value)` as is common in MS-DOS.

Q4.

I want to control a standard RS-232 device/parallel printer/joystick...

A4.

You're probably better off using existing drivers (in the Linux kernel or an X server or somewhere else) to do it. The drivers are usually quite versatile, so even slightly non-standard devices usually work with them. See the information on standard ports above for pointers to documentation for them.

9. Example code

Here's a piece of simple example code for I/O port access:

```
/*
 * example.c: very simple example of port I/O
 *
 * This code does nothing useful, just a port write, a pause,
 * and a port read. Compile with `gcc -O2 -o example example.c',
 * and run as root with `./example'.
 */

#include <stdio.h>
#include <unistd.h>
#include <asm/io.h>

#define BASEPORT 0x378 /* lp1 */

int main()
{
  /* Get access to the ports */
  if (ioperm(BASEPORT, 3, 1)) {perror("ioperm"); exit(1);}

  /* Set the data signals (D0-7) of the port to all low (0) */
  outb(0, BASEPORT);

  /* Sleep for a while (100 ms) */
  usleep(100000);

  /* Read from the status port (BASE+1) and display the result */
  printf("status: %d\n", inb(BASEPORT + 1));

  /* We don't need the ports anymore */
  if (ioperm(BASEPORT, 3, 0)) {perror("ioperm"); exit(1);}

  exit(0);
}

/* end of example.c */
```

10. Credits

Too many people have contributed for me to list, but thanks a lot, everyone. I have not replied to all the contributions that I've received; sorry for that, and thanks again for the help.

TOC

This page was last updated January 13th 1997.

Mini How-to on Setting Up IP Aliasing On A Linux Machine

GNU Copyleft 1996/1997 Harish Pillay (h.pillay@ieee.org)

Primary site: http://home.pacific.net.sg/~harish/linuxipalias.html

Synopsis:

This is a cook book recipe on how to set up and run IP aliasing on a Linux box. In addition, there are instructions on how to also set up the machine to receive e-mail on the aliased IP #s.

My setup:

- Latest kernel (2.0.27 - from ftp.funet.fi:/pub/Linux/kernel/src/v2.0) - has worked since 1.3.7x.
- IP Alias compiled as a loadable module. You would have indicated in the "make config" command to make your kernel, that you want the IP Masq to be compiled as a (M)odule. Check the Modules HOW-TO (if that exists) or check the info in /usr/src/linux/Documentation/modules.txt.
- I have to support 2 additional IPs over and above the IP already allocated to me.
- A D-Link DE620 pocket adapter (not important, works with any Linux supported network adapter).

Commands:

- First load the IP Alias module (you can skip this step if you compiled the module into the kernel):

  ```
  /sbin/insmod /lib/modules/`uname -r`/ipv4/ip_alias.o
  ```

- Second, setup the loopback, eth0 and all the IP #s beginning with the main IP # for the eth0 interface:

  ```
  /sbin/ifconfig lo 127.0.0.1
  /sbin/ifconfig eth0 up
  /sbin/ifconfig eth0 172.16.3.1
  /sbin/ifconfig eth0:0 172.16.3.10
  /sbin/ifconfig eth0:1 172.16.3.100
  ```

  172.16.3.1 is the main IP #, while .10 and .100 are the aliases. The magic is the eth0:x where x=0,1,2,...n for the different IP #s. The main IP # does not need to be aliased.

- Third, setup the routes. First route the loopback, then the net and, finally, the various IP #s starting with the default (originally allocated) one:

  ```
  /sbin/route add -net 127.0.0.0
  /sbin/route add -net 172.16.3.0 dev eth0
  /sbin/route add -host 172.16.3.1 dev eth0
  /sbin/route add -host 172.16.3.10 dev eth0:0
  /sbin/route add -host 172.16.3.100 dev eth0:1
  /sbin/route add default gw 172.16.3.200
  ```

That's it.

In the example IP # above, I am using the Private IP #s (RFC 1918) for illustrative purposes. Substitute them with your own official or private IP #s.

TOC

The example shows only 3 IP #s. The max is defined to be 256 in /usr/include/linux/net_alias.h. 256 IP #s on ONE car lot :-)!

Here's what my /sbin/ifconfig looks like:

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
          UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
          RX packets:5088 errors:0 dropped:0 overruns:0
          TX packets:5088 errors:0 dropped:0 overruns:0

eth0      Link encap:10Mbps Ethernet  HWaddr 00:8E:B8:83:19:20
          inet addr:172.16.3.1  Bcast:172.16.3.255  Mask:255.255.255.0
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:334036 errors:0 dropped:0 overruns:0
          TX packets:11605 errors:0 dropped:0 overruns:0
          Interrupt:7 Base address:0x378

eth0:0    Link encap:10Mbps Ethernet  HWaddr 00:8E:B8:83:19:20
          inet addr:172.16.3.10  Bcast:172.16.3.255  Mask:255.255.255.0
          UP BROADCAST RUNNING  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0

eth0:1    Link encap:10Mbps Ethernet  HWaddr 00:8E:B8:83:19:20
          inet addr:172.16.3.100  Bcast:172.16.3.255  Mask:255.255.255.0
          UP BROADCAST RUNNING  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
```

And /proc/net/aliases:

```
device           family address
eth0:0           2      172.16.3.10
eth0:1           2      172.16.3.100
```

And /proc/net/alias_types:

```
type    name            n_attach
2       ip              2
```

Of course, the stuff in /proc/net was created by the ifconfig command and not by hand!

Question: How can I keep the settings through a reboot?

Answer: Whether you are using BSD-style or SysV-style (Redhat for example) init, you can always include it in /etc/rc.d/rc.local. Here's what I have on my SysV init system (Redhat 3.0.3 and 4.0):

- My /etc/rc.d/rc.local: (edited to show the relevant portions)

```
#setting up IP alias interfaces
echo "Setting 172.16.3.1, 172.16.3.10, 172.16.3.100 IP Aliases ..."
/sbin/ifconfig lo 127.0.0.1
/sbin/ifconfig eth0 up
/sbin/ifconfig eth0 172.16.3.1
```

```
/sbin/ifconfig eth0:0 172.16.3.10
/sbin/ifconfig eth0:1 172.16.3.100
#setting up the routes
echo "Setting IP routes ..."
/sbin/route add -net 127.0.0.0
/sbin/route add -net 172.16.3.0 dev eth0
/sbin/route add -host 172.16.3.1 eth0
/sbin/route add -host 172.16.3.10 eth0:0
/sbin/route add -host 172.16.3.100 eth0:1
/sbin/route add default gw 172.16.3.200
#
```

Question: How do I set up the IP aliased machine to receive e-mail on the various aliased IP #s (on a machine using sendmail)?

- ***Answer:*** Create (if not already existing) a file called for example, /etc/mynames.cw. It does not have to be this exact name nor in the /etc directory.
- In that file, place the official domain names of the aliased IP #s. If these aliased IP #s do not have a domain name, then you can place the IP # itself.

```
/etc/mynames.cw:
----------------
# /etc/mynames.cw - include all aliases for your machine here; # is a comment.
domain.one.net
domain.two.com
domain.three.org
4.5.6.7
```

- In your sendmail.cf file, where it defines a file class macro Fw, add the following:

```
.
.
.
##################
#   local info   #
##################
.
.
# file containing names of hosts for which we receive email
Fw/etc/mynames.cw
.
.
.
```

- That should do it. Test out the new setting by invoking sendmail in test mode for example:

```
ganymede$ /usr/lib/sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter < ruleset> < address>
> 0 me@4.5.6.7
rewrite: ruleset  0   input: me @ 4 . 5 . 6 . 7
rewrite: ruleset 98   input: me @ 4 . 5 . 6 . 7
rewrite: ruleset 98 returns: me @ 4 . 5 . 6 . 7
rewrite: ruleset 97   input: me @ 4 . 5 . 6 . 7
rewrite: ruleset  3   input: me @ 4 . 5 . 6 . 7
rewrite: ruleset 96   input: me < @ 4 . 5 . 6 . 7 >
```

```
rewrite: ruleset 96 returns: me < @ 4 . 5 . 6 . 7 . >
rewrite: ruleset  3 returns: me < @ 4 . 5 . 6 . 7 . >
rewrite: ruleset  0   input: me < @ 4 . 5 . 6 . 7 . >
rewrite: ruleset 98   input: me < @ 4 . 5 . 6 . 7 . >
rewrite: ruleset 98 returns: me < @ 4 . 5 . 6 . 7 . >
rewrite: ruleset  0 returns: $# local $: me
rewrite: ruleset 97 returns: $# local $: me
rewrite: ruleset  0 returns: $# local $: me
> 0 me@4.5.6.8
rewrite: ruleset  0   input: me @ 4 . 5 . 6 . 8
rewrite: ruleset 98   input: me @ 4 . 5 . 6 . 8
rewrite: ruleset 98 returns: me @ 4 . 5 . 6 . 8
rewrite: ruleset 97   input: me @ 4 . 5 . 6 . 8
rewrite: ruleset  3   input: me @ 4 . 5 . 6 . 8
rewrite: ruleset 96   input: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 96 returns: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset  3 returns: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset  0   input: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 98   input: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 98 returns: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 95   input: < > me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 95 returns: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset  0 returns: $# smtp $@ 4 . 5 . 6 . 8 $: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 97 returns: $# smtp $@ 4 . 5 . 6 . 8 $: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset  0 returns: $# smtp $@ 4 . 5 . 6 . 8 $: me < @ 4 . 5 . 6 . 8 >
>
```

Notice when I tested me@4.5.6.7, it delivered the mail to the local machine, while me@4.5.6.8 was to be handed off to the smtp mailer. That is the correct response.

- You are all set now.

Hope the preceding is useful to someone.

Thanks to all those who have done this great work on Linux and IP Aliasing. And especially to Juan Jose Ciarlante for clarifying my questions.

Kudos to the ace programmers!

If you do find this document useful or have suggestions on improvements, do send me an e-mail at h.pillay@ieee.org.

Enjoy.

Questions?

Go back to Harish's Home Page

Linux IP Masquerade mini HOWTO

Ambrose Au, ambrose@writeme.com; David Ranch, dranch@trinnet.net

v1.50, 7 February 1999

This document describes how to enable IP masquerade feature on a Linux host, allowing connected computers that do not have registered Internet IP addresses to connect to the Internet through your Linux box.

1. Introduction

- 1.1 Introduction
- 1.2 Foreword, Feedback & Credits
- 1.3 Copyright & Disclaimer

2. Background Knowledge

- 2.1 What is IP Masquerade?
- 2.2 Current Status
- 2.3 Who Can Benefit From IP Masquerade?
- 2.4 Who Doesn't Need IP Masquerade?
- 2.5 How IP Masquerade Works?
- 2.6 Requirements for Using IP Masquerade on Linux 2.2.x
- 2.7 Requirements for Using IP Masquerade on Linux 2.0.x

3. Setting Up IP Masquerade

- 3.1 Compiling the Kernel for IP Masquerade Support
- 3.2 Assigning Private Network IP Address
- 3.3 Configuring the OTHER machines

- 3.4 Configuring IP Forwarding Policies
- 3.5 Testing IP Masquerade

4. Other IP Masquerade Issues and Software Support

- 4.1 Problems with IP Masquerade
- 4.2 Incoming services
- 4.3 Supported Client Software and Other Setup Note
- 4.4 IP Firewall Administration (ipfwadm)
- 4.5 IP Firewalling Chains (ipchains)
- 4.6 IP Masquerade and Demand-Dial-Up
- 4.7 IPautofw Packet Fowarder
- 4.8 CU-SeeMe and Linux IP-Masquerade Teeny How-To
- 4.9 Other Related Tools

5. Frequently Asked Questions

- 5.1 Does IP Masquerade work with dynamically assigned IP?
- 5.2 Can I use cable modem, DSL, satellite link, etc. to connect to the Internet and use IP Masquerade?
- 5.3 What applications are supported with IP Masquerade?
- 5.4 How can I get IP Masquerade running on Redhat, Debian, Slackware, etc.?
- 5.5 I've just upgraded to the 2.2.x kernels, why is IP Masquerade not working?
- 5.6 I've just upgraded to the kernels 2.0.30 or later, why is IP Masquerade not working?
- 5.7 I can't get IP Masquerade to work! What options do I have for Windows Platform?
- 5.8 I've checked all my configurations, I still can't get IP Masquerade to work. What should I do?
- 5.9 How do I join the IP Masquerade Mailing List?
- 5.10 I want to help on IP Masquerade development. What can I do?
- 5.11 Where can I find more information on IP Masquerade?
- 5.12 I want to translate this HOWTO to another language, what should I do?
- 5.13 This HOWTO seems out of date, are you still maintaining it? Can you include more information on ...? Are there any plans for making this better?
- 5.14 I got IP Masquerade working, it" great! I want to thank you guys, what can I do?

6. Miscellaneous

- 6.1 Useful Resources
- 6.2 Linux IP Masquerade Resource
- 6.3 Thanks to
- 6.4 Reference

---

1. Introduction

1.1 Introduction

This document describes how to enable IP masquerade feature on a Linux host, allowing connected computers that do not have registered Internet IP addresses to connect to the Internet through your Linux box. It is possible to connect your machines to the Linux host with ethernet, as well as other kinds of connection such as a dialup ppp link. This document will emphasize on ethernet connection, since it should be the most likely case.

> **This document is intended for users using stable kernels 2.2.x and 2.0.x. Older kernels such as 1.2.x are NOT covered.**

1.2 Foreword, Feedback & Credits

I find it very confusing as a new user setting up IP masquerade on a newer kernel, i.e. 2.x kernel. Although there is a FAQ and a mailing list, there is no document dedicates on that; and there are some requests on the mailing list for such a HOWTO. So, I decided to write this up as a starting point for new users, and possibly a building block for knowledgeable users to build on for documentation. If you think I'm not doing a good job, feel free to tell me so that I can make it better.

This document is heavily based on the original FAQ by Ken Eves , and numerous helpful messages in the IP Masquerade mailing list. And a special thanks to Mr. Matthew Driver whose mailing list message inspired me to set up IP Masquerade and eventually writing this.

Please feel free to send any feedback or comments to ambrose@writeme.com and dranch@trinnet.net if we've mistaken on any information, or if any information is missing. Your invaluable feedback will certainly be influencing the future of this HOWTO!

This HOWTO is meant to be a quick guide to get your IP Masquerade working in the shortest time. As I am not a technical writer, you may find the information in this document not as general and objective as it can be. The latest news and information can be found at the IP Masquerade Resource web page that we maintained. If you have any technical questions on IP Masquerade, please join the IP Masquerade Mailing List instead of sending email to me since I have limited time, and the developers of IP_Masq are more capable of answering your questions.

The latest version of this document can be found at the IP Masquerade Resource, which also contains the HTML and postscript version:

- http://ipmasq.cjb.net/
- http://ipmasq2.cjb.net/
- Please refer to IP Masquerade Resource Mirror Sites Listing for other mirror sites available.

TOC

1.3 Copyright & Disclaimer

This document is `copyright(c) 1999 Ambrose Au`, and it's a free document. You can redistribute it under the terms of the GNU General Public License.

The information and other contents in this document are to the best of my knowledge. However, IP Masquerade is *experimental*, and there is chance that I make mistakes as well; so you should determine if you want to follow the information in this document.

Nobody is responsible for any damage on your computers and any other losses by using the information on this document. i.e.

> **THE AUTHOR AND MAINTAINERS ARE NOT RESPONSIBLE FOR ANY DAMAGES INCURRED DUE TO ACTIONS TAKEN BASED ON THE INFORMATION IN THIS DOCUMENT.**

---

2. Background Knowledge

2.1 What is IP Masquerade?

IP Masquerade is a networking function in Linux. If a Linux host is connected to the Internet with IP Masquerade enabled, then computers connecting to it (either on the same LAN or connected with modems) can reach the Internet as well, even though they have *no official assigned IP addresses*.

This allows a set of machines to *invisibly* access the Internet hidden behind a gateway system, which appears to be the only system using the Internet. Breaking the security of a well set-up masquerading system should be considerably more difficult than breaking a good packet filter based firewall (assuming there are no bugs in either).

2.2 Current Status

IP Masquerade had been out for several years and is maturing as Linux heads into the 2.2.x stage. Kernels since 1.3.x had built-in support already. Many individuals and even busnesses are using it, with satisfactory results.

Browsing web pages and telnet are reported to work well over IP Masquerade. FTP, IRC and listening to Real Audio are working with certain modules loaded. Other network streaming audio such as True Speech and Internet Wave work too. Some fellow users on the mailing list even tried video conferencing software. `Ping` is now working, with the newly available ICMP patch

Please refer to section 4.3 for a more complete listing of software supported.

IP Masquerade works well with 'client machines' on several different OS and platforms. There are successful cases with systems using Unix, Windows 95, Windows NT, Windows for Workgroup(with TCP/IP package), OS/2, Macintosh System's OS with Mac TCP, Mac Open Transport, DOS with NCSA Telnet package, VAX, Alpha with Linux, and even Amiga with AmiTCP or AS225-stack. The list goes on and on, the point is, if your OS platform talks TCP/IP, it should work with IP Masquerade.

2.3 Who Can Benefit From IP Masquerade?

- If you have a Linux host connected to the Internet, and
- if you have some computers running TCP/IP connected to that Linux box on a local subnet, and/or
- if your Linux host has more than one modem and acts as a PPP or SLIP server connecting to others, which
- those **OTHER** machines do not have official assigned IP addresses. (these machines are represented by **OTHER** machines hereby)
- And of course, if you want those **OTHER** machines to make it onto the Internet without spending extra bucks :)

2.4 Who Doesn't Need IP Masquerade?

- If your machine is a stand-alone Linux host connected to the Internet, then it is pointless to have IP Masquerade running, or
- if you already have assigned addresses for your **OTHER** machines, then you don't need IP Masquerade,

- and of course, if you don't like the idea of a 'free ride'.

TOC

2.5 How IP Masquerade Works?

From IP Masquerade FAQ by Ken Eves:

```
  Here is a drawing of the most simple setup:

     SLIP/PPP         +------------+                          +-------------+
     to provider      |  Linux     |       SLIP/PPP           | Anybox      |
    <---------- modem1|            |modem2 ----------- modem  |             |
      111.222.333.444 |            |           192.168.1.100  |             |
                      +------------+                          +-------------+

    In the above drawing a Linux box with ip_masquerading installed and
  running is connected to the Internet via SLIP/or/PPP using modem1.  It has
  an assigned IP address of 111.222.333.444.  It is setup that modem2 allows
  callers to login and start a SLIP/or/PPP connection.

    The second system (which doesn't have to be running Linux) calls into the
  Linux box and starts a SLIP/or/PPP connection.  It does NOT have an assigned
  IP address on the Internet so it uses 192.168.1.100. (see below)

    With ip_masquerade and the routing configured properly the machine
  Anybox can interact with the Internet as if it was really connected (with a
  few exceptions).

Quoting Pauline Middelink:
  Do not forget to mention the ANYBOX should have the Linux box
  as its gateway (whether is be the default route or just a subnet
  is no matter). If the ANYBOX can not do this, the Linux machine
  should do a proxy arp for all routed address, but the setup of
  proxy arp is beyond the scope of the document.

The following is an excerpt from a post on comp.os.linux.networking which
has been edited to match the names used in the above example:
   o I tell machine ANYBOX that my slipped linux box is its gateway.
   o When a packet comes into the linux box from ANYBOX, it will assign it
     new source port number, and slap its own ip address in the packet
     header, saving the originals.  It will then send the modified packet
     out over the SLIP/or/PPP interface to the Internet.
   o When a packet comes from the Internet to the linux box, if the port
     number is one of those assigned above, it will get the original
     port and ip address, put them back in the packet header, and send the
     packet to ANYBOX.
   o The host that sent the packet will never know the difference.
```

An IP Masquerading Example

typical example is given in the diagram below:-

```
+----------+
|          |  Ethernet
| abox     |::::::
|          |2    :192.168.1.x
+----------+     :
                  :   +----------+   PPP
+----------+     :  1|  Linux   |   link
|          |     ::::| masq-gate|::::::::// Internet
| bbox     |::::::   |          |
|          |3    :   +----------+
+----------+     :
                  :
+----------+     :
|          |     :
| cbox     |::::::
|          |4
+----------+


<-Internal Network->
```

In this example there are 4 computer systems that we are concerned about (there is presumably also something on the far right that your IP connection to the internet comes through, and there is something (far off the page) on the internet that you are interested in exchanging information with). The Linux system `masq-gate` is the masquerading gateway for the internal network of machines `abox`, `bbox` and `cbox` to get to the internet. The internal network uses one of the assigned private network addresses, in this case the class C network 192.168.1.0, with the linux box having address 192.168.1.1 and the other systems having addresses on that network.

The three machines `abox`, `bbox` and `cbox` (which can, by the way, be running any operating system as long as they can speak IP - such as **Windows 95**, **Macintosh MacTCP** or even another linux box) can connect to other machines on the internet, however the masquerading system `masq-gate` converts all of their connections so that they appear to originate from `masq-gate`, and arranges that data coming back in to a masqueraded connection is relayed back to the originating system - so the systems on the internal network see a direct route to the internet and are unaware that their data is being masqueraded.

2.6 Requirements for Using IP Masquerade on Linux 2.2.x

**** Please refer to IP Masquerade Resource for the latest information. ****

- Kernel 2.2.x source available from http://www.kernel.org/
  (Most of the modern Linux dributions such as Redhat 5.2 - shipped with 2.0.36 kernel - has modular kernel with all IP Masquerade kernel options compiled. In such cases, there is no need to compile again. If you are upgrading kernel, then you should be aware of what you need, mentioned later in the HOWTO.)
- Loadable kernel modules, preferably 2.1.121 or newer
- A well set up TCP/IP network
  covered in Linux NET-3 HOWTO and the Network Administrator's Guide
  Also check out the Trinity OS Doc, a very compreshensive guide on Linux networking.
- Connectivity to Internet for your Linux host
  covered in Linux ISP Hookup HOWTO, Linux PPP HOWTO, Linux DHCP mini-HOWTO and Linux Cable Modem mini-HOWTO

TOC

- IP Chains 1.3.8 or newer available from http://www.rustcorp.com/linux/ipchains/
  more information on version requirement is on the Linux IP Firewalling Chains page
- For other options, please see Linux IP Masquerade Resource

2.7 Requirements for Using IP Masquerade on Linux 2.0.x

**** Please refer to IP Masquerade Resource for the latest information. ****

- Kernel 2.0.x source available from http://www.kernel.org/
  (Most of the modern Linux dributions such as Redhat 5.2 has modular kernel with all IP Masquerade kernel options compiled. In such cases, there is no need to compile again. If you are upgrading kernel, then you should be aware of what you need, mentioned later in the HOWTO.)
- Loadable kernel modules, preferably 2.0.0 or newer available from
  http://www.pi.se/blox/modules/modules-2.0.0.tar.gz
  (modules-1.3.57 is the minimal requirement)
- A well set up TCP/IP network
  covered in Linux NET-3 HOWTO and the Network Administrator's Guide
  Also check out the Trinity OS Doc, a very compreshensive guide on Linux networking.
- Connectivity to Internet for your Linux host
  covered in Linux ISP Hookup HOWTO, Linux PPP HOWTO, Linux DHCP mini-HOWTO and Linux Cable Modem mini-HOWTO
- Ipfwadm 2.3 or newer available from ftp://ftp.xos.nl/pub/linux/ipfwadm/ipfwadm-2.3.tar.gz
  more information on version requirement is on the Linux IPFWADM page
- You can optionally apply some IP Masquerade patches to enable other functionality. More information availabe on IP Masquerade Resources (these patches apply to all 2.0.x kernels)

---

3. Setting Up IP Masquerade

If your private network contains any vital information, think carefully before using IP Masquerade. This may be a GATEWAY for you to get to the Internet, and vice versa for someone on the other side of the world to get into your network.

3.1 Compiling the Kernel for IP Masquerade Support

If your Linux distribution already has the required features and modules compiled (most modular kernels will have all you need) mentioned below, then you do not have to re-compile the kernel. Reading this section is still highly recommended as it contains other useful informaiton.

Linux 2.2.x Kernels

- First of all, you need the kernel source for 2.2.x
- If this is your first time compiling the kernel, don't be scared. In fact, it's rather easy and it's covered in Linux Kernel HOWTO.
- Unpack the kernel source to `/usr/src/` with a command: `tar xvzf linux-2.2.x.tar.gz -C /usr/src`, where x is the patch level beyond 2.0
  (make sure there is a directory or symbolic link called `linux`)
- Apply appropriate patches. Since new patches are coming out, details will not be included here. Please refer to IP Masquerade Resources for up-to-date information.
- Refer to the Kernel HOWTO and the README file in the kernel source directory for further instructions on compiling a kernel
- Here are the options that you need to compile in:

  Say *YES* to the following,

```
* Prompt for development and/or incomplete code/drivers
  CONFIG_EXPERIMENTAL
  - this will allow you to select experimental IP Masquerade code compiled
    into the kernel

* Enable loadable module support
  CONFIG_MODULES
  - allows you to load ipmasq modules such as ip_masq_ftp.o

* Networking support
  CONFIG_NET

* Network firewalls
  CONFIG_FIREWALL

* TCP/IP networking
  CONFIG_INET

* IP: forwarding/gatewaying
  CONFIG_IP_FORWARD
```

```
* IP: firewalling
  CONFIG_IP_FIREWALL

* IP: masquerading
  CONFIG_IP_MASQUERADE

* IP: ipportfw masq support
  CONFIG_IP_MASQUERADE_IPPORTFW
  - recommended

* IP: ipautofw masquerade support
  CONFIG_IP_MASQUERADE_IPAUTOFW
  - optional

* IP: ICMP masquerading
  CONFIG_IP_MASQUERADE_ICMP
  - support for masquerading ICMP packets, recommended.

* IP: always defragment
  CONFIG_IP_ALWAYS_DEFRAG
  - highly recommended

* Dummy net driver support
  CONFIG_DUMMY
  - recommended

* IP: ip fwmark masq-forwarding support
  CONFIG_IP_MASQUERADE_MFW
  - optional
```

NOTE: These are just the component you need for IP Masquerade, select whatever other options you need for your specific setup.

- After compiling the kernel, you should compile and install the modules:

```
make modules; make modules_install
```

- Then you should add a few lines into your `/etc/rc.d/rc.local` file (or any file you think is appropriate) to load the required modules reside in `/lib/modules/2.2.x/ipv4/` automatically during each reboot:

```
        .
        .
        .
/sbin/depmod -a
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
/sbin/modprobe ip_masq_irc
(and other modules such as ip_masq_cuseeme, ip_masq_vdolive
 if you have applied the patches)
        .
        .
        .
```

IMPORTANT: IP forwarding is disabled by default in 2.2.x kernels, please make sure you enable it by running `echo "1" > /proc/sys/net/ipv4/ip_forwarding` For Redhat users, you may try changing `FORWARD_IPV4=false` to `FORWARD_IPV4=true` in `/etc/sysconfig/network`

- Reboot the Linux box.

TOC

Linux 2.0.x Kernels

- First of all, you need the kernel source (preferably the latest kernel version 2.0.36 or above)
- If this is your first time compiling the kernel, don't be scared. In fact, it's rather easy and it's covered in Linux Kernel HOWTO.
- Unpack the kernel source to `/usr/src/` with a command: `tar xvzf linux-2.0.x.tar.gz -C /usr/src`, where x is the patch level beyond 2.0
  (make sure there is a directory or symbolic link called `linux`)
- Apply appropriate patches. Since new patches are coming out, details will not be included here. Please refer to IP Masquerade Resources for up-to-date information.
- Refer to the Kernel HOWTO and the README file in the kernel source directory for further instructions on compiling a kernel
- Here are the options that you need to compile in:

  Say *YES* to the following,

```
* Prompt for development and/or incomplete code/drivers
  CONFIG_EXPERIMENTAL
  - this will allow you to select experimental IP Masquerade code compiled
    into the kernel

* Enable loadable module support
  CONFIG_MODULES
  - allows you to load modules

* Networking support
  CONFIG_NET

* Network firewalls
  CONFIG_FIREWALL

* TCP/IP networking
  CONFIG_INET

* IP: forwarding/gatewaying
  CONFIG_IP_FORWARD

* IP: firewalling
  CONFIG_IP_FIREWALL

* IP: masquerading (EXPERIMENTAL)
  CONFIG_IP_MASQUERADE
  - although it is experimental, it is a *MUST*

* IP: ipautofw masquerade support (EXPERIMENTAL)
  CONFIG_IP_MASQUERADE_IPAUTOFW
  -recommended

* IP: ICMP masquerading
  CONFIG_IP_MASQUERADE_ICMP
  - support for masquerading ICMP packets, optional.

* IP: always defragment
```

```
          CONFIG_IP_ALWAYS_DEFRAG
          - highly recommended

        * Dummy net driver support
          CONFIG_DUMMY
          - recommended
```

NOTE: These are just the component you need for IP Masquerade, select whatever other options you need for your specific setup.

- After compiling the kernel, you should compile and install the modules:

```
make modules; make modules_install
```

- Then you should add a few lines into your `/etc/rc.d/rc.local` file (or any file you think is appropriate) to load the required modules reside in `/lib/modules/2.0.x/ipv4/` automatically during each reboot:

```
        .
        .
        .
/sbin/depmod -a
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
/sbin/modprobe ip_masq_irc
(and other modules such as ip_masq_cuseeme, ip_masq_vdolive
 if you have applied the patches)
        .
        .
        .
```

IMPORTANT: IP forwarding is disabled by default since 2.0.34 kernels, please make sure you enable it by running
`echo "1" > /proc/sys/net/ipv4/ip_forward` For Redhat users, you may try changing `FORWARD_IPV4=false` to `FORWARD_IPV4=true` in `/etc/sysconfig/network`

- Reboot the Linux box.

3.2 Assigning Private Network IP Address

Since all **OTHER** machines do not have official assigned addressees, there must be a right way to allocate address to those machines.

From IP Masquerade FAQ:

There is an RFC (#1597, probably obsolete by now) on which IP addresses are to be used on a non-connected network. There are 3 blocks of numbers set aside specifically for this purpose. One which I use is 255 Class-C subnets at 192.168.1.n to 192.168.255.n .

```
From RCF 1597:

Section 3: Private Address Space

The Internet Assigned Numbers Authority (IANA) has reserved the
following three blocks of the IP address space for private networks:

              10.0.0.0        -   10.255.255.255
              172.16.0.0      -   172.31.255.255
              192.168.0.0     -   192.168.255.255
```

```
We will refer to the first block as "24-bit block", the second as
"20-bit block", and to the third as "16-bit" block".  Note that the
first block is nothing but a single class A network number, while the
second block is a set of 16 contiguous class B network numbers, and
third block is a set of 255 contiguous class C network numbers.
```

So, if you're using a class C network, you should name your machines as 192.168.1.1, 1.92.168.1.2, 1.92.168.1.3, ..., 192.168.1.x

192.168.1.1 is usually the gateway machine, which is your Linux host connecting to the Internet. Notice that 192.168.1.0 and 192.168.1.255 are the Network and Broadcast address respectively, which are reserved. Avoid using these addresses on your machines.

3.3 Configuring the OTHER machines

Besides setting the appropriate IP address for each machine, you should also set the appropriate gateway. In general, it is rather straight forward. You simply enter the address of your Linux host (usually 192.168.1.1) as the gateway address.

For the Domain Name Service, you can add in any DNS available. The most apparent one should be the one that your Linux is using. You can optionally add any domain search suffix as well.

After you have reconfigured those IP addresses, remember to restart the appropriate services or reboot your systems.

The following configuration instructions assume that you are using a Class C network with 192.168.1.1 as your Linux host's address. Please note that 192.168.1.0 and 192.168.1.255 are reserved.

Configuring Windows 95

1. If you haven't installed your network card and adapter driver, do so now.
2. Go to *'Control Panel'/'Network'*.
3. Add *'TCP/IP protocol'* if you don't already have it.
4. In *'TCP/IP properties'*, goto *'IP Address'* and set IP Address to 192.168.1.x, (1 < x < 255), and then set Subnet Mask to 255.255.255.0
5. Add 192.168.1.1 as your gateway under *'Gateway'*.
6. Under *'DNS Configuration'/'DNS Server search order'* add your the DNS that your Linux host uses (usually find in `/etc/resolv.conf`). Optionally, you can add the appropriate domain search suffix.
7. Leave all the other settings as they are unless you know what you're doing.
8. Click *'OK'* on all dialog boxes and restart system.
9. `Ping` the linux box to test the network connection: *'Start/Run'*, type: `ping 192.168.1.1`
   (This is only a LAN connection testing, you can't `ping` the outside world yet.)
10. You can optionally create a `HOSTS` file in the windows directory so that you can use hostname of the machines on your LAN. There is an example called `HOSTS.SAM` in the windows directory.

Configuring Windows for Workgroup 3.11

1. If you haven't installed your network card and adapter driver, do so now.
2. Install the TCP/IP 32b package if you don't have it already.
3. In *'Main'/'Windows Setup'/'Network Setup'*, click on *'Drivers'*.
4. Highlight *'Microsoft TCP/IP-32 3.11b'* in the *'Network Drivers'* section, click *'Setup'*.
5. Set IP Address to 192.168.1.x (1 < x < 255), then set Subnet Mask to 255.255.255.0 and Default Gateway to 192.168.1.1
6. Do not enable *'Automatic DHCP Configuration'* and put anything in those *'WINS Server'* input areas unless you're in a Windows NT domain and you know what you're doing.

TOC

7. Click *'DNS'*, fill in the appropriate information mentioned in STEP 6 of section 3.3.1, then click *'OK'* when you're done with it.
8. Click *'Advanced'*, check *'Enable DNS for Windows Name Resolution'* and *'Enable LMHOSTS lookup'* if you're using a look up host file, similar to the one mentioned in STEP 10 of section 3.3.1
9. Click *'OK'* on all dialog boxes and restart system.
10. `Ping` the linux box to test the network connection: *'File/Run'*, type: `ping 192.168.1.1`
    (This is only a LAN connection testing, you can't `ping` the outside world yet.)

Configuring Windows NT

1. If you haven't installed your network card and adapter driver, do so now.
2. Go to *'Main'/'Control Panel'/'Network'*
3. Add the TCP/IP Protocol and Related Component from the *'Add Software'* menu if you don't have TCP/IP service installed already.
4. Under *'Network Software and Adapter Cards'* section, highlight *'TCP/IP Protocol'* in the *'Installed Network Software'* selection box.
5. In *'TCP/IP Configuration'*, select the appropriate adapter, e.g. `[1]Novell NE2000 Adapter`. Then set the IP Address to 192.168.1.x (1 < x < 255), then set Subnet Mask to 255.255.255.0 and Default Gateway to 192.168.1.1
6. Do not enable *'Automatic DHCP Configuration'* and put anything in those *'WINS Server'* input areas unless you're in a Windows NT domain and you know what you're doing.
7. Click *'DNS'*, fill in the appropriate information mentioned in STEP 6 of section 3.3.1, then click *'OK'* when you're done with it.
8. Click *'Advanced'*, check *'Enable DNS for Windows Name Resolution'* and *'Enable LMHOSTS lookup'* if you're using a look up host file, similar to the one mentioned in STEP 10 of section 3.3.1
9. Click *'OK'* on all dialog boxes and restart system.
10. `Ping` the linux box to test the network connection: *'File/Run'*, type: `ping 192.168.1.1`
    (This is only a LAN connection testing, you can't `ping` the outside world yet.)

Configuring UNIX Based Systems

1. If you haven't installed your network card and recompile your kernel with the appropriate adapter driver, do so now.
2. Install TCP/IP networking, such as the nettools package, if you don't have it already.
3. Set *IPADDR* to 192.168.1.x (1 < x < 255), then set *NETMASK* to 255.255.255.0, *GATEWAY* to 192.168.1.1, and *BROADCAST* to 192.168.1.255
   For example, you can edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file on a Red Hat Linux system, or simply do it through the Control Panel.
   (it's different in SunOS, BSDi, Slackware Linux, etc...)
4. Add your domain name service (DNS) and domain search suffix in `/etc/resolv.conf`
5. You may want to update your `/etc/networks` file depending on your settings.
6. Restart the appropriate services, or simply restart your system.
7. Issue a `ping` command: `ping 192.168.1.1` to test the connection to your gateway machine.
   (This is only a LAN connection testing, you can't `ping` the outside world yet.)

Configuring DOS using NCSA Telnet package

1. If you haven't installed your network card, do so now.
2. Load the appropriate packet driver. For an NE2000 card, issue `nwpd 0x60 10 0x300`, with your network card set to IRQ 10 and hardware address at 0x300
3. Make a new directory, and then unpack the NCSA Telnet package: `pkunzip tel2308b.zip`
4. Use a text editor to open the `config.tel` file

TOC

5. Set `myip=192.168.1.x` (1 < x < 255), and netmask=255.255.255.0
6. In this example, you should set `hardware=packet, interrupt=10, ioaddr=60`
7. You should have at least one individual machine specification set as the gateway, i.e. the Linux host:

```
name=default
host=yourlinuxhostname
hostip=192.168.1.1
gateway=1
```

8. Have another specification for a domain name service:

```
name=dns.domain.com ; hostip=123.123.123.123; nameserver=1
```

   Note: substitute the appropriate information about the DNS that your Linux host uses
9. Save your `config.tel` file
10. Telnet to the linux box to test the network connection: `telnet 192.168.1.1`

Configuring MacOS Based System Running MacTCP

1. If you haven't installed the appropriate driver software for your Ethernet adapter, now would be a very good time to do so.
2. Open the *MacTCP control panel*. Select the appropriate network driver (Ethernet, NOT EtherTalk) and click on the *'More...'* button.
3. Under *'Obtain Address:'*, click *'Manually'*.
4. Under *'IP Address:'*, select *class C* from the popup menu. Ignore the rest of this section of the dialog box.
5. Fill in the appropriate information under *'Domain Name Server Information:'*.
6. Under *'Gateway Address:'*, enter 192.168.1.1
7. Click *'OK'* to save the settings. In the main window of the *MacTCP control panel*, enter the IP address of your Mac (192.168.1.x, 1 < x < 255) in the *'IP Address:'* box.
8. Close the *MacTCP control panel*. If a dialog box pops up notifying you to do so, restart the system.
9. You may optionally ping the Linux box to test the network connection. If you have the freeware program *MacTCP Watcher*, click on the *'Ping'* button, and enter the address of your Linux box (192.168.1.1) in the dialog box that pops up. (This is only a LAN connection testing, you can't ping the outside world yet.)
10. You can optionally create a `Hosts` file in your System Folder so that you can use the hostnames of the machines on your LAN. The file should already exist in your System Folder, and should contain some (commented-out) sample entries which you can modify according to your needs.

Configuring MacOS Based System Running Open Transport

1. If you haven't installed the appropriate driver software for your Ethernet adapter, now would be a very good time to do so.
2. Open the *TCP/IP Control Panel* and choose *'User Mode ...'* from the *Edit* menu. Make sure the user mode is set to at least *'Advanced'* and click the *'OK'* button.
3. Choose *'Configurations...'* from the *File* menu. Select your *'Default'* configuration and click the *'Duplicate...'* button. Enter 'IP Masq' (or something to let you know that this is a special configuration) in the *'Duplicate Configuration'* dialog, it will probably say something like *'Deafault copy'*. Then click the *'OK'* button, and the *'Make Active'* button
4. Select *'Ethernet'* from the *'Connect via:'* pop-up.
5. Select the appropriate item from the *'Configure:'* pop-up. If you don't know which option to choose, you probably should re-select your *'Default'* configuration and quit. I use *'Manually'*.
6. Enter the IP address of your Mac (192.168.1.x, 1 < x < 255) in the *'IP Address:'* box.
7. Enter 255.255.255.0 in the *'Subnet mask:'* box.

TOC

8. Enter 192.168.1.1 in the *'Router address:'* box.
9. Enter the IP addresses of your domain name servers in the *'Name server addr.:'* box.
10. Enter the name of your Internet domain (e.g. 'microsoft.com') in the *'Starting domain name'* box under *'Implicit Search Path:'*.
11. The following procedures are optional. Incorrect values may cause erratic behavior. If your not sure, it's probably better to leave them blank, unchecked and/or un- selected. Remove any information from those fields, if necessary. As far as I know there is no way through the TCP/IP dialogs, to tell the system not to use a previously select alternate "Hosts" file. If you know, I would be interested.
    Check the *'802.3'* if your network requires 802.3 frame types.
12. Click the *'Options...'* button to make sure that the TCP/IP is active. I use the *'Load only when needed'* option. If you run and quit TCP/IP applications many times without rebooting your machine, you may find that unchecking the *'Load only when needed'* option will prevent/reduce the effects on your machines memory management. With the item unchecked the TCP/IP protocol stacks are always loaded and available for use. If checked, the TCP/IP stacks are automatically loaded when needed and un- loaded when not. It's the loading and unloading process that can cause your machines memory to become fragmented.
13. You may ping the Linux box to test the network connection. If you have the freeware program *MacTCP Watcher*, click on the *'Ping'* button, and enter the address of your Linux box (192.168.1.1) in the dialog box that pops up. (This is only a LAN connection testing, you can't ping the outside world yet.)
14. You can create a `Hosts` file in your System Folder so that you can use the hostnames of the machines on your LAN. The file may or may not already exist in your System Folder. If so, it should contain some (commented-out) sample entries which you can modify according to your needs. If not, you can get a copy of the file from a system running MacTCP, or just create your own (it follows a subset of the Unix `/etc/hosts` file format, described on RFC952). Once you've created the file, open the *TCP/IP control panel*, click on the *'Select Hosts File...'* button, and open the `Hosts` file.
15. Click the close box or choose *'Close'* or *'Quit'* from the *File* menu, and then click the *'Save'* button to save the changes you have made.
16. The changes take effect immediately, but rebooting the system won't hurt.

Configuring Novell network using DNS

1. If you haven't installed the appropriate driver software for your Ethernet adapter, now would be a very good time to do so.
2. Downloaded tcpip16.exe from
3. `edit c:\nwclient\startnet.bat`
   : (here is a copy of mine)

   ```
   SET NWLANGUAGE=ENGLISH
   LH LSL.COM
   LH KTC2000.COM
   LH IPXODI.COM
   LH tcpip
   LH VLM.EXE
   F:
   ```
4. `edit c:\nwclient\net.cfg`
   : (change link driver to yours i.e. NE2000)

   ```
   Link Driver KTC2000
           Protocol IPX 0 ETHERNET_802.3
           Frame ETHERNET_802.3
           Frame Ethernet_II
           FRAME Ethernet_802.2
   ```

```
NetWare DOS Requester
           FIRST NETWORK DRIVE = F
           USE DEFAULTS = OFF
           VLM = CONN.VLM
           VLM = IPXNCP.VLM
           VLM = TRAN.VLM
           VLM = SECURITY.VLM
           VLM = NDS.VLM
           VLM = BIND.VLM
           VLM = NWP.VLM
           VLM = FIO.VLM
           VLM = GENERAL.VLM
           VLM = REDIR.VLM
           VLM = PRINT.VLM
           VLM = NETX.VLM

Link Support
        Buffers 8 1500
        MemPool 4096

Protocol TCPIP
        PATH SCRIPT     C:\NET\SCRIPT
        PATH PROFILE    C:\NET\PROFILE
        PATH LWP_CFG    C:\NET\HSTACC
        PATH TCP_CFG    C:\NET\TCP
        ip_address      xxx.xxx.xxx.xxx
        ip_router       xxx.xxx.xxx.xxx
```

5. and finally created

   ```
   c:\bin\resolv.cfg
   :

   SEARCH DNS HOSTS SEQUENTIAL
   NAMESERVER 207.103.0.2
   NAMESERVER 207.103.11.9
   ```

6. I hope this helps some people get their Novell Nets online, BTW this can be done using Netware 3.1x or 4.x

Configuring OS/2 Warp

1. If you haven't installed the appropriate driver software for your Ethernet adapter, now would be a very good time to do so.
2. Install the TCP/IP protocoll if you don't have it already.
3. Go to *Programms/TCP/IP (LAN) / TCP/IP* Settings
4. In *'Network'* add your TCP/IP Address and set your Netmask (255.255.255.0)
5. Under *'Routing'* press *'Add'*. Set the *Type* to *'default'* and type the IP Address of your Linux Box in the Field *'Router Address'*. (192.168.1.1).
6. Set the same DNS (Nameserver) Address that your Linux host uses in *'Hosts'*.
7. Close the TCP/IP control panel. Say yes to the following question(s).
8. Reboot your system
9. You may ping the Linux box to test the network configuration. Type `'ping 192.168.1.1'` in a 'OS/2 Command prompt Window'. When ping packets are received all is ok.

TOC

Configuring Other Systems

The same logic should apply to setting up other platforms. Consult the sections above. If you're interested in writing about any of systems that have not been covered yet, please send a detail setup instruction to ambrose@writeme.com and dranch@trinnet.net.

3.4 Configuring IP Forwarding Policies

At this point, you should have your kernel and other required packages installed, as well as your modules loaded. Also, the IP addresses, gateway, and DNS should be all set on the **OTHER** machines.

Now, the only thing left to do is to use the IP firewalling tools to forward appropriate packets to the appropriate machine:

> ** This can be accomplished in many different ways. The following suggestions and examples worked for me, but you may have different ideas, please refer to section 4.4 and the ipchains(2.2.x) / ipfwadm(2.0.x) manpages for details. **
>
> **** This section ONLY provides you with the bare minimum rule set to get IP Masquerade working while security issue is not being considered. It is highly recomended that you spend some time to apply appropriate firewall rules to tighten security. ****

Linux 2.2.x Kernels

ipfwadm is no longer the tool for manipulating ipmasq rules for 2.2.x kernels, please use ipchains.

```
ipchains -P forward DENY
ipchains -A forward -s yyy.yyy.yyy.yyy/x -j MASQ
```

where x is one of the following numbers according to the class of your subnet, and yyy.yyy.yyy.yyy is your network address.

```
netmask         | x   | Subnet
~~~~~~~~~~~~~~~~|~~~~|~~~~~~~~~~~~~~~
255.0.0.0       | 8   | Class A
255.255.0.0     | 16  | Class B
255.255.255.0   | 24  | Class C
255.255.255.255 | 32  | Point-to-point
```

You may also use the format yyy.yyy.yyy.yyy/xxx.xxx.xxx.xxx, where xxx.xxx.xxx.xxx specfies your subnet mask such as 255.255.255.0

For example, if I'm on a class C subnet, I would have entered:

```
ipchains -P forward DENY
ipchains -A forward -s 192.168.1.0/24 -j MASQ

or

ipchains -P forward DENY
ipchains -A forward -s 192.168.1.0/255.255.255.0 -j MASQ
```

You can also do it on a per machine basis. For example, if I want 192.168.1.2 and 192.168.1.8 to have access to the Internet, but not the other machines, I would have entered:

```
ipchains -P forward DENY
ipchains -A forward -s 192.168.1.2/32 -j MASQ
ipchains -A forward -s 192.168.1.8/32 -j MASQ
```

Do **not** make your default policy be masquerading - otherwise someone who can manipulate their routing will be able to t straight back through your gateway, using it to masquerade their identity!

Again, you can add these lines to the `/etc/rc.local` files, one of the rc files you prefer, or do it manually every time you need IP Masquerade.

For detail ipchains usage, please refer to the Linux IPCHAINS HOWTO

Linux 2.0.x Kernels

```
ipfwadm -F -p deny
ipfwadm -F -a m -S yyy.yyy.yyy.yyy/x -D 0.0.0.0/0

or

ipfwadm -F -p deny
ipfwadm -F -a masquerade -S yyy.yyy.yyy.yyy/x -D 0.0.0.0/0
```

where x is one of the following numbers according to the class of your subnet, and yyy.yyy.yyy.yyy is your network address.

```
netmask          | x   | Subnet
~~~~~~~~~~~~~~~~|~~~~|~~~~~~~~~~~~~~~
255.0.0.0        | 8   | Class A
255.255.0.0      | 16  | Class B
255.255.255.0    | 24  | Class C
255.255.255.255  | 32  | Point-to-point
```

You may also use the format yyy.yyy.yyy.yyy/xxx.xxx.xxx.xxx, where xxx.xxx.xxx.xxx specfies your subnet mask such as 255.255.255.0

For example, if I'm on a class C subnet, I would have entered:

```
ipfwadm -F -p deny
ipfwadm -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0
```

Since bootp request packets comes without valid IP's once the client knows nothing about it, for people with a bootp server in the masquerade/firewall machine it is necessary to use the following before the deny command:

```
ipfwadm -I -a accept -S 0/0 68 -D 0/0 67 -W bootp_clients_net_if_name -P udp
```

You can also do it on a per machine basis. For example, if I want 192.168.1.2 and 192.168.1.8 to have access to the Internet, but not the other machines, I would have entered:

```
ipfwadm -F -p deny
ipfwadm -F -a m -S 192.168.1.2/32 -D 0.0.0.0/0
ipfwadm -F -a m -S 192.168.1.8/32 -D 0.0.0.0/0
```

What appears to be a common mistake is to make the first command be this

```
ipfwadm -F -p masquerade
```

Do **not** make your default policy be masquerading - otherwise someone who can manipulate their routing will be able to tunnel straight back through your gateway, using it to masquerade their identity!

Again, you can add these lines to the `/etc/rc.local` files, one of the rc files you prefer, or do it manually every time you need IP Masquerade.

Please read section 4.4 for a detail guide on Ipfwadm

TOC

3.5 Testing IP Masquerade

It's time to give it a try, after all these hard work. Make sure the connection of your Linux hosts to the Internet is okay.

You can try browsing some *'INTERNET!!!'* web sites on your **OTHER** machines, and see if you get it. I recommend using an IP address rather than a hostname on your first try, because your DNS setup may not be correct.

For example, you can access the Linux Documentation Project site http://metalab.unc.edu/mdw/linux.html with an entry of http://152.19.254.81/mdw/linux.html

If you see The Linux Documentation Project homepage, then congratulations! It's working! You may then try one with hostname entry, and then ping, telnet, ssh, ftp, Real Audio, True Speech, whatever supported by IP Masquerade.....

So far, I have no trouble with the above settings, and it's full credit to the people who spend their time making this wonderful feature working.

---

4. Other IP Masquerade Issues and Software Support

4.1 Problems with IP Masquerade

Some protocols will not currently work with masquerading because they either assume things about port numbers, or encode data in their data stream about addresses and ports - these latter protocols need specific proxies built into the masquerading code to make them work.

4.2 Incoming services

Masquerading cannot handle incoming services at all. There are a few ways of allowing them, but they are completely separate from masquerading, and are really part of standard firewall practice.

If you do not require high levels of security then you can simply redirect ports. There are various ways of doing this - I use a modified redir program (which I hope will be available from sunsite and mirrors soon). If you wish to have some level of authorisation on incoming connections then you can either use TCP wrappers or Xinetd on top of redir (0.7 or above) to allow only specific IP addresses through, or use some other tools. The TIS Firewall Toolkit is a good place to look for tools and information.

More details can be found at IP Masquerade Resource.

A section on more about forwarding services will be added soon.

4.3 Supported Client Software and Other Setup Note

> **** The following list is not being maintained anymore. Please refer to this page on applications that work thru Linux IP masquerading and IP Masquerade Resource for more detail. ****

Generally, application that uses TCP and UDP should work. If you have any suggestion, hints, or questions about applications with IP Masquerade, please visit this page on applications that work thru Linux IP masquerading by Lee Nevo.

Clients that Work

General Clients

HTTP

all supported platforms, surfing the web

POP & SMTP

all supported platforms, email client

Telnet

all supported platforms, remote session

FTP

all supported platforms, with ip_masq_ftp.o module (not all sites work with certain clients; e.g. some sites cannot

be reached using ws_ftp32 but works with netscape)

Archie

all supported platforms, file searching client (not all archie clients are supported)

NNTP (USENET)

all supported platforms, USENET news client

VRML

Windows(possibly all supported platforms), virtual reality surfing

traceroute

mainly UNIX based platforms, some variations may not work

ping

all platforms, with ICMP patch

anything based on IRC

all supported platforms, with ip_masq_irc.o modules

Gopher client

all supported platforms

WAIS client

all supported platforms

Multimedia Clients

Real Audio Player

Windows, network streaming audio, with ip_masq_raudio module loaded

True Speech Player 1.1b

Windows, network streaming audio

Internet Wave Player

Windows, network streaming audio

Worlds Chat 0.9a

Windows, Client-Server 3D chat program

Alpha Worlds

Windows, Client-Server 3D chat program

Internet Phone 3.2

Windows, Peer-to-peer audio communications, people can reach you only if you initiate the call, but people cannot call you

Powwow

Windows, Peer-to-peer Text audio whiteboard communications, people can reach you only if you initiate the call, but people cannot call you

CU-SeeMe

all supported platforms, with cuseeme modules loaded, please see IP Masquerade Resource for detail

VDOLive

Windows, with vdolive patch

Note: Some clients such as IPhone and Powwow may work even if you're not the one who initiate the call by using *ipautofw package* (refer to section 4.6)

Other Clients

NCSA Telnet 2.3.08

DOS, a suite containing telnet, ftp, ping, etc.

PC-anywhere for windows 2.0

MS-Windows, Remotely controls a PC over TCP/IP, only work if it is a client but not a host

Socket Watch

uses ntp - network time protocol

Linux net-acct package

Linux, network administration-account package

Clients that do not Work

Intel Internet Phone Beta 2

Connects but voice travels one way (out) Traffic only

Intel Streaming Media Viewer Beta 1

Cannot connect to server

Netscape CoolTalk

Cannot connect to opposite side

talk,ntalk

will not work - requires a kernel proxy to be written.

WebPhone

Cannot work at present (it makes invalid assumptions about addresses).

X

Untested, but I think it cannot work unless someone builds an X proxy, which is probably an external program to the masquerading code. One way of making this work is to use **ssh** as the link and use the internal X proxy of that to make things work!

Platforms/OS Tested as on OTHER machines

- Linux
- Solaris
- Windows 95
- Windows NT (both workstation and server)
- Windows For Workgroup 3.11 (with TCP/IP package)
- Windows 3.1 (with Chameleon package)
- Novel 4.01 Server
- OS/2 (including Warp v3)

TOC

- Macintosh OS (with MacTCP or Open Transport)
- DOS (with NCSA Telnet package, DOS Trumpet works partially)
- Amiga (with AmiTCP or AS225-stack)
- VAX Stations 3520 and 3100 with UCX (TCP/IP stack for VMS)
- Alpha/AXP with Linux/Redhat
- SCO Openserver (v3.2.4.2 and 5)
- IBM RS/6000 running AIX

Basically all OS platforms support TCP/IP and give you the option to specify the gateway/router should work with IP Masquerade.

4.4 IP Firewall Administration (ipfwadm)

This section provides a more in-depth guide on using ipfwadm.

This is a setup for a firewall/masquerade system behind a PPP link with a static PPP address follows. Trusted interface is 192.168.255.1, PPP interface has been changed to protect the guilty :). I listed each incoming and outgoing interface individually to catch IP spoofing as well as stuffed routing and/or masquerading. Also anything not explicitly allowed is forbidden!

```
#!/bin/sh
#
# /etc/rc.d/rc.firewall, define the firewall configuration, invoked from
# rc.local.
#

PATH=/sbin:/bin:/usr/sbin:/usr/bin

# testing, wait a bit then clear all firewall rules.
# uncomment following lines if you want the firewall to automatically
# disable after 10 minutes.
# (sleep 600; \
# ipfwadm -I -f; \
# ipfwadm -I -p accept; \
# ipfwadm -O -f; \
# ipfwadm -O -p accept; \
# ipfwadm -F -f; \
# ipfwadm -F -p accept; \
# ) &

# Incoming, flush and set default policy of deny. Actually the default policy
# is irrelevant because there is a catch all rule with deny and log.
ipfwadm -I -f
ipfwadm -I -p deny
# local interface, local machines, going anywhere is valid
ipfwadm -I -a accept -V 192.168.255.1 -S 192.168.0.0/16 -D 0.0.0.0/0
# remote interface, claiming to be local machines, IP spoofing, get lost
ipfwadm -I -a deny -V your.static.PPP.address -S 192.168.0.0/16 -D 0.0.0.0/0 -o
# remote interface, any source, going to permanent PPP address is valid
ipfwadm -I -a accept -V your.static.PPP.address -S 0.0.0.0/0 -D
```

```
your.static.PPP.address/32
# loopback interface is valid.
ipfwadm -I -a accept -V 127.0.0.1 -S 0.0.0.0/0 -D 0.0.0.0/0
# catch all rule, all other incoming is denied and logged. pity there is no
# log option on the policy but this does the job instead.
ipfwadm -I -a deny -S 0.0.0.0/0 -D 0.0.0.0/0 -o

# Outgoing, flush and set default policy of deny. Actually the default policy
# is irrelevant because there is a catch all rule with deny and log.
ipfwadm -O -f
ipfwadm -O -p deny
# local interface, any source going to local net is valid
ipfwadm -O -a accept -V 192.168.255.1 -S 0.0.0.0/0 -D 192.168.0.0/16
# outgoing to local net on remote interface, stuffed routing, deny
ipfwadm -O -a deny -V your.static.PPP.address -S 0.0.0.0/0 -D 192.168.0.0/16 -o
# outgoing from local net on remote interface, stuffed masquerading, deny
ipfwadm -O -a deny -V your.static.PPP.address -S 192.168.0.0/16 -D 0.0.0.0/0 -o
# outgoing from local net on remote interface, stuffed masquerading, deny
ipfwadm -O -a deny -V your.static.PPP.address -S 0.0.0.0/0 -D 192.168.0.0/16 -o
# anything else outgoing on remote interface is valid
ipfwadm -O -a accept -V your.static.PPP.address -S your.static.PPP.address/32 -D
0.0.0.0/0
# loopback interface is valid.
ipfwadm -O -a accept -V 127.0.0.1 -S 0.0.0.0/0 -D 0.0.0.0/0
# catch all rule, all other outgoing is denied and logged. pity there is no
# log option on the policy but this does the job instead.
ipfwadm -O -a deny -S 0.0.0.0/0 -D 0.0.0.0/0 -o

# Forwarding, flush and set default policy of deny. Actually the default policy
# is irrelevant because there is a catch all rule with deny and log.
ipfwadm -F -f
ipfwadm -F -p deny
# Masquerade from local net on local interface to anywhere.
ipfwadm -F -a masquerade -W ppp0 -S 192.168.0.0/16 -D 0.0.0.0/0
# catch all rule, all other forwarding is denied and logged. pity there is no
# log option on the policy but this does the job instead.
ipfwadm -F -a deny -S 0.0.0.0/0 -D 0.0.0.0/0 -o
```

You can block traffic to a particular site using the -I, -O or -F. Remember that the set of rules are scanned top to bottom and -a means "append" to the existing set of rules so any restrictions need to come before global rules. For example (and untested) :-

Using -I rules. Probably the fastest but it only stops the local machines, the firewall itself can still access the "forbidden" site. Of course you might want to allow that combination.

```
... start of -I rules ...
# reject and log local interface, local machines going to 204.50.10.13
ipfwadm -I -a reject -V 192.168.255.1 -S 192.168.0.0/16 -D 204.50.10.13/32 -o
# local interface, local machines, going anywhere is valid
ipfwadm -I -a accept -V 192.168.255.1 -S 192.168.0.0/16 -D 0.0.0.0/0
... end of -I rules ...
```

Using -O rules. Slowest because the packets go through masquerading first but this rule even stops the firewall

accessing the forbidden site.

```
... start of -O rules ...
# reject and log outgoing to 204.50.10.13
ipfwadm -O -a reject -V your.static.PPP.address -S your.static.PPP.address/32 -D
204.50.10.13/32 -o
# anything else outgoing on remote interface is valid
ipfwadm -O -a accept -V your.static.PPP.address -S your.static.PPP.address/32 -D
0.0.0.0/0
... end of -O rules ...
```

Using -F rules. Probably slower than -I and this still only stops masqueraded machines (i.e. internal), firewall can still get to forbidden site.

```
... start of -F rules ...
# Reject and log from local net on PPP interface to 204.50.10.13.
ipfwadm -F -a reject -W ppp0 -S 192.168.0.0/16 -D 204.50.10.13/32 -o
# Masquerade from local net on local interface to anywhere.
ipfwadm -F -a masquerade -W ppp0 -S 192.168.0.0/16 -D 0.0.0.0/0
... end of -F rules ...
```

No need for a special rule to allow 192.168.0.0/16 to go to 204.50.11.0, it is covered by the global rules.

There is more than one way of coding the interfaces in the above rules. For example instead of -V 192.168.255.1 you can code -W eth0, instead of -V your.static.PPP.address you can use -W ppp0. Personal choice and documentation more than anything.

4.5 IP Firewalling Chains (ipchains)

This is the firewall ruleset manipulation tool primarily intended for 2.2.x kernels (there is a patch for this to work on 2.0.x).

We will update this section to give several examples on ipchains usage soon.

See the Linux IP Firewalling Chains page and Linux IPCHAINS HOWTO for detail.

4.6 IP Masquerade and Demand-Dial-Up

1. If you would like to setup your network to automatically dial up the Internet, the *diald* demand dial-up package will be of great utility.
2. To setup the diald, please check out the Setting Up Diald for Linux Page
3. Once diald and IP masq have been setup, you can go to any of the client machines and initiate a web, telnet or ftp session.
4. Diald will detect the incoming request, then dial up your ISP and establish the connection.
5. There is a timeout that will occur with the first connection. This is inevitable if you are using analog modems. The time taken to establish the modem link and the PPP connections will cause your client program to timeout. This can be avoided if you are using an ISDN connection. All you need to do is to terminate the current process on the client and restart it.

TOC

4.7 IPautofw Packet Fowarder

IPautofw is a generic forwarder of TCP and UDP for Linux masquerading. Generally to utilize a package which requires UDP, a specific ip_masq module needs to be loaded; ip_masq_raudio, ip_masq_cuseeme, ... Ipautofw acts in a more generic manner, it will forward any type of traffic including those which the application specific modules will not forward. This may create a security hole if not administered correctly.

4.8 CU-SeeMe and Linux IP-Masquerade Teeny How-To

Provided by Michael Owings.

Introduction

> This section will explain the necessary steps to get CU-SeeMe (both the Cornell and White Pine versions) working together with Linux's IP-Masquerade.

CU-SeeMe is a desktop video conferencing package available for both Windows and Macintosh clients. A free version is available from Cornell University. A significantly enhanced commercial version can be obtained from White Pine Software.

IP Masquerading allows one or more workstations on a LAN to "masquerade" behind a single Linux machine connected to the Internet. The workstations on the LAN can access the Internet almost transparently even without valid IP addresses. The Linux box rewrites outgoing packets from the LAN to the Internet in such a way that they they appear to originate from the Linux machine. Response packets coming back in are re-written and routed back to the correct workstations on the LAN. This arrangement allows many Internet applications to run transparently from the lan workstations. For some other applications (such as CU-SeeMe), however, the Linux masquerade code needs a little help to route packets properly. This help usually comes in the form of special kernel loadable modules. For more information on IP-Masquerading, see The Linux IP Masquerading Website.

Getting It Running

First you will need a properly configured kernel. You should have full support compiled in for both IP-Masquerading and IP AutoForwarding. IP Autoforwarding is available as a config option on kernels 2.0.30 and later -- you will need to patch earlier kernels. See the Linux IP Masquerade Resource for pointers to the IP-Autoforwarding material.

Next, you will need to get the latest version of ip_masq_cuseeme.c. The latest version is available via anon ftp from ftp://ftp.swampgas.com/pub/cuseeme/ip_masq_cuseeme.c. This new module will also be rolled up into the kernel 2.0.31 distribution. You should replace the version in your kernel distribution with this new version. ip_masq_cuseeme.c normally resides in net/ipv4 off of the Linux source tree. You should compile and install this module.

Now, you should set up ip autoforwarding for udp ports 7648-7649 as follows:

```
ipautofw -A -r udp 7648 7649 -c udp 7648 -u
```

OR

```
ipautofw -A -r udp 7648 7649 -h www.xxx.yyy.zzz
```

The first form will allow calls to/from the last workstation to use port 7648 (the primary cu-seeme port) . The second invocation of ipautofw will allow cu-seeme calls only to/from www.xxx.yyy.zzz. I prefer the former invocation, as it is

more flexible because there is no need to specify a fixed workstation IP. However, this invocation also requires a workstation to have previously placed an outgoing call in order to receive incoming calls.

Note that both invocations leave UDP ports 7648-7649 on the client machines open to the outside world -- and while this does not pose an enormous security hazard, you should use appropriate caution.

Finally, load up the new ip_masq_cuseeme module as follows:

```
modprobe ip_masq_cuseeme
```

You should now be able to fire up CU-SeeMe from a masqueraded machine on your lan and connect to a remote reflector, or another CU-Seeme user. You should also be able to get incoming calls. Note that outside callers should call using the ip of your linux gateway, NOT the masqueraded workstation.

Restrictions/Caveats

Password Protected Reflectors

No way, no how. Uh-uh. Negatory. White Pine uses the source IP (as computed by the client program) to encrypt the password prior to transmission. Since we have to rewrite this address, the reflector ends up using the wrong source IP to decrypt it, which yields an invalid password. This will only be fixed if White Pine changes their password encryption scheme (which I have suggested), or if they would be willing to make their password encryption routines public so I could add in a fix to ip_masq_cuseeme. While chances for the latter solution are vanishingly small, I would encourage anyone reading this to contact White Pine and suggest the former approach. As the traffic on this page is relatively high, I suspect we could generate enough email to get this problem moved up on White Pine's list of priorities.

Thanx to Thomas Griwenka for bringing this to my attention.

Running a Reflector

You should not attempt to run a reflector on the same machine where you have ip_masq_cuseeme and ipautoforwarding for port 7648 loaded. It simply won't work, as both setups require port 7648. Either run the reflector on another Internet-reachable host, or unload CU-SeeMe client support prior to running the reflector.

Multiple CU-SeeMe Users

You cannnot have multiple simultaneous CU-SeeMe users on the LAN at this time. This is due largely to CU-SeeMe's stubborn insistence on always sending to port 7648, which can only be redirected (easily) to one LAN workstation at a time.

Using the -c (control port) invocation of ipautofw above, you can avoid to having to specify a fixed workstation address allowed to use CU-SeeMe -- the first workstation to send anything out on control port 7648 will be designated to receive traffic on 7648-7649. 5 minutes or so after this workstation has been inactive on port 7648, another workstation can come along and use CU-SeeMe.

Help on Setting up CU-SeeMe

Feel free to email any comments or questions to mikey@swampgas.com. Or if you wish, you can call me up via CU-SeeMe.

4.9 Other Related Tools

We will be updating this section soon to cover more ipmasq related tools such as ipportfw and masqadmin.

5. Frequently Asked Questions

If you can think of any useful FAQ, please send it to ambrose@writeme.com and dranch@trinnet.net. Please clearly state the question and an appropriate answer. Thank you!

5.1 Does IP Masquerade work with dynamically assigned IP?

Yes, it works with dynamic IP assigned by your ISP, usually by a DHCP server. As long as you have an valid Internet IP address, it should work. Of course, static IP works too.

5.2 Can I use cable modem, DSL, satellite link, etc. to connect to the Internet and use IP Masquerade?

Sure, as long as Linux supports that network interface, it should work.

5.3 What applications are supported with IP Masquerade?

It is very difficult to keep track of a list of "working applications". However, most of the normal Internet applications are supported, such as browsing the Internet (Netscape, MSIE, etc.), ftp (such as WS_FTP), Real Audio, telnet, SSH, POP3 (incoming email - Pine, Outlook), SMTP (outgoing email), etc.

Applications involving more complicated protocols or special connection methods such as video conferencing software need special helper tools.

For more detail, please see this page about applications that work thru Linux IP masquerading by Lee Nevo.

5.4 How can I get IP Masquerade running on Redhat, Debian, Slackware, etc.?

No matter what Linux distribution you got, the procedures for setting up IP Masquerade mentioned in this HOWTO should apply. Some distributions may have GUI or special configuration files that make the setup easier. We try our best to write the HOWTO as general as possible.

TOC

5.5 I've just upgraded to the 2.2.x kernels, why is Masquerade not working?

There are several things you should check assuming your Linux ipmasq box already have proper connection to the Internet and your LAN:

- Make sure you have the necessary features and modules are compiled and loaded. See earlier sections for detail.
- Check `/usr/src/linux/Documentation/Changes` and make sure you have the minimal requirement for the network tools installed.
- Make sure you have enabled IP forwarding. Try running `echo "1" > /proc/sys/net/ipv4/ip_forwarding`.
- You should use ipchains to manipulate ipmasq and firewalling rules.
- Go through all setup and configuration again! A lot of time it's just a typo or a stupid mistake you oversee.

5.6 I've just upgraded to the kernels 2.0.30 or later, why is IP Masquerade not working?

There are several things you should check assuming your Linux ipmasq box already have proper connection to the Internet and your LAN:

- Make sure you have the necessary features and modules are compiled and loaded. See earlier sections for detail.
- Check `/usr/src/linux/Documentation/Changes` and make sure you have the minimal requirement for the network tools installed.
- Make sure you have enabled IP forwarding. Try running `echo "1" > /proc/sys/net/ipv4/ip_forward`.
- You should use ipfwadm to manipulate ipmasq and firewalling rules. You need to patch the 2.0.x kernels to use ipchains.
- Go through all setup and configuration again! A lot of time it's just a typo or a stupid mistake you oversee.

5.7 I can't get IP Masquerade to work! What options do I have for Windows Platform?

Giving up a free, reliable, high performance solution that works on minimal hardware and pay a fortune for something that needs more hardware, lower performance and less reliable? (IMHO. And yes, I have real life experience with these ;-)

Okay, it's your call. Do a web search on MS Proxy Server, Wingate, or see www.winfiles.com. Don't tell

TOC

anyone I sent you.

5.8 I've checked all my configurations, I still can't get IP Masquerade to work. What should I do?

- Stay calm. Get yourself a cup of tea and have a rest, then try the suggestions mentioned below.
- Check the IP Masquerade Mailing List Archive, most likely your answer is there waiting for you.
- Post your question to the IP Masquerade Mailing List, see next the FAQ for deatil. Please only try this if you cannot find the answer from the mailing list archive.
- Post your question to related Linux networking newsgroup.
- Send email to ambrose@writeme.com and dranch@trinnet.net. You have a better chance of getting a reply if you send to both of us. David is usually pretty good on replying, and I do not want to comment on my response time.
- Check your configurations again :-)

5.9 How do I join the IP Masquerade Mailing List?

Join the Linux IP Masquerading mailing list by sending an email to masq-subscribe@indyramp.com.

Subject and body of the message are IGNORED. This gives you every message on the list as it comes out. You are welcome to use this form if you need it, but if you can stand the digest, please choose it instead. The digest puts less of a load on the list servers. Note that you can only post from an account/address you are subscribed from.

For more commands, email masq-help@tori.indyramp.com.

5.10 I want to help on IP Masquerade development. What can I do?

Join the Linux IP Masquerading DEVELOPERS list and ask the great developers there, by sending an email to masq-dev-subscribe@tori.indyramp.com (or for a digest format, use masq-dev-digest-subscribe@tori.indyramp.com).

DON'T ask non IP Masquerade development related questions there!!!!

5.11 Where can I find more information on IP Masquerade?

You can find more information on IP Masquerade at the Linux IP Masquerade Resource that David and I also maintained. See section 6.2 for availability.

You may also find more information at The Semi-Original Linux IP Masquerading Web Site maintai by Indyramp Consulting, who also provided the ipmasq mailing lists.

5.12 I want to translate this HOWTO to another language, what should I do?

Make sure the lanaguage you want to translate to is not already covered by someone else, a list of available HOWTO translations is available at the Linux IP Masquerade Resource.

Send an email to ambrose@writeme.com and I will send you the SGML source of the latest version of the HOWTO.

5.13 This HOWTO seems out of date, are you still maintaining it? Can you include more information on ...? Are there any plans for making this better?

Yes, this HOWTO is still being maintained. I'm guilty of being too busy working on two jobs and don't have much time to work on this, my apology. However, with the addition of David Ranch as the HOWTO maintainer, things should improve.

If you think of a topic that could be included in the HOWTO, please send email to me and David. It will be even better if you can provide that information. I and David will include the information into the HOWTO if it is appropriate. And many thanks for your contributions.

We have a lot of new ideas and plans for improving the HOWTO, such as case studies that will cover different network setup involving IP Masquerade, more on security, ipchains usage, ipfwadm/ipchains ruleset examples, more FAQs, more coverage on protocol and port forwarding utilities like masqadmin, etc. If you think you can help, please do. Thanks.

5.14 I got IP Masquerade working, it" great! I want to thank you guys, what can I do?

Thank the developers and appreciate the time and effort they spent on this. Send an email to us and let us know how happy you are. Introduce other people to Linux and help them when they have problems.

---

6. Miscellaneous

6.1 Useful Resources

- IP Masquerade Resource page should have enough information for setting up IP Masquerade
- IP masquerade mailing list archive contains some of the recent messages sent to the mailing list.
- This Linux IP Masquerade mini HOWTO for kernel 2.2.x and 2.0.x
- IP Masquerade HOWTO for kernel 1.2.x if you're using an older kernel
- IP masquerade FAQ has some general information
- Linux IPCHAINS HOWTO and http://www.rustcorp.com/linux/ipchains/ has lots of information for ipchains usage, as well as source and binaries for the ipchains.
- X/OS Ipfwadm page contains sources, binaries, documentation, and other information about the `ipfwadm` package
- A page on applications that work thru Linux IP masquerading by Lee Nevo provides tips and tricks on getting applications to work with IP Masquerade.
- The LDP Network Administrator's Guide is a must for beginners trying to set up a network.
- Trinity OS Doc, a very compreshensive guide on Linux networking.
- Linux NET-3 HOWTO also has lots of useful information about Linux networking
- Linux ISP Hookup HOWTO and Linux PPP HOWTO gives you information on how to connect your Linux host to the Internet
- Linux Ethernet-Howto is a good source of information about setting up a LAN running ethernet
- You may also be interested in Linux Firewalling and Proxy Server HOWTO
- Linux Kernel HOWTO will guide you through the kernel compilation process
- Other Linux HOWTOs such as Kernel HOWTO
- Posting to the USENET newsgroup: comp.os.linux.networking

6.2 Linux IP Masquerade Resource

The Linux IP Masquerade Resource is a website dedicated to Linux IP Masquerade information also maintained by David Ranch and I. It usually has the latest information related to IP Masquerade and may have information that is not being included in the HOWTO.

You may find the Linux IP Masquerade Resource at the following locations:

- http://ipmasq.cjb.net/, Primary Site, redirected to http://www.tor.shaw.wave.ca/~ambrose/

- http://ipmasq2.cjb.net/, Secondary Site, redirected to http://www.geocities.com/SiliconValley/Heights/2288/

TOC

6.3 Thanks to

- David Ranch, dranch@trinnet.net
  help maintaining this HOWTO and the Linux IP Masquerade Resource Page, ..., too many to list here :-)
- Michael Owings, mikey@swampgas.com
  on providing section for CU-SeeMe and Linux IP-Masquerade Teeny How-To
- Gabriel Beitler, gbeitler@aciscorp.com
  on providing section 3.3.8 (setting up Novel)
- Ed Doolittle, dolittle@math.toronto.edu
  on suggestion to `-V` option in `ipfwadm` command for improved security
- Matthew Driver, mdriver@cfmeu.asn.au
  on helping extensively on this HOWTO, and providing section 3.3.1 (setting up Windows 95)
- Ken Eves, ken@eves.com
  on the FAQ that provides invaluable information for this HOWTO
- Ed. Lott, edlott@neosoft.com
  for a long list of tested system and software
- Nigel Metheringham, Nigel.Metheringham@theplanet.net
  on contributing his version of IP Packet Filtering and IP Masquerading HOWTO, which make this HOWTO a better and technical in-depth document
  section 4.1, 4.2, and others
- Keith Owens, kaos@ocs.com.au
  on providing an excellent guide on ipfwadm section 4.2
  on correction to `ipfwadm -deny` option which avoids a security hole, and clarified the status of `ping` over IP Masquerade
- Rob Pelkey, rpelkey@abacus.bates.edu
  on providing section 3.3.6 and 3.3.7 (setting up MacTCP and Open Transport)
- Harish Pillay, h.pillay@ieee.org
  on providing section 4.5 (dial-on-demand using diald)
- Mark Purcell, purcell@rmcs.cranfield.ac.uk
  on providing section 4.6 (IPautofw)
- Ueli Rutishauser, rutish@ibm.net
  on providing section 3.3.9 (setting up OS/2 Warp)
- John B. (Brent) Williams, forerunner@mercury.net
  on providing section 3.3.7 (setting up Open Transport)
- Enrique Pessoa Xavier, enrique@labma.ufrj.br
  on the bootp setup suggestion
- developers of IP Masquerade for this great feature

    - Delian Delchev, delian@wfpa.acad.bg
    - Nigel Metheringham, Nigel.Metheringham@theplanet.net
    - Keith Owens, kaos@ocs.com.au
    - Jeanette Pauline Middelink, middelin@polyware.iaf.nl
    - David A. Ranch, trinity@value.net
    - Miquel van Smoorenburg, miquels@q.cistron.nl
    - Jos Vos, jos@xos.nl
    - Paul Russell, Paul.Russell@rustcorp.com.au
    - And more who I may have failed to mention here (please let me know)
- all users sending feedback and suggestion to the mailing list, especially the ones who reported errors in the document and the clients that are supported and not supported
- I appologize if I have not included information that some fellow users sent me. There are many suggestions and ideas sent to me, but I just do not have enough time to verify or I lost track of them. I am trying my best to incorporate all the information sent to me into the HOWTO. I thank you for the effort, and I hope you understand my situation.

6.4 Reference

- IP masquerade FAQ by Ken Eves
- IP masquerade mailing list archive by Indyramp Consulting
- Ipfwadm page by X/OS
- Various networking related Linux HOWTOs

IP Sub-Networking Mini-Howto

Robert Hart, hartr@interweft.com.au

v1.0, 31 March 1997

This document describes why and how to subnetwork an IP network - that is using a single A, B or C Class network number to function correctly on several interconnected networks.

1. Copyright

2. Introduction

- 2.1 Other sources of information

3. The Anatomy of IP numbers

- 3.1 IP numbers belong to Interfaces - **NOT** hosts!
- 3.2 IP Numbers as "Dotted Quads"
- 3.3 Classes of IP Networks
- 3.4 Network numbers, interface addresses and broadcast addresses
- 3.5 The network mask

4. What are subnets?

5. Why subnetwork?

6. How to subnetwork a IP network number

- 6.1 Setting up the physical connectivity
- 6.2 Subnetwork sizing
- 6.3 Calculating the subnetwork mask and network numbers

7. Routing

- 7.1 The routing tables

1. Copyright

This document is distributed under the terms of the GNU Public License (GPL).

This document is directly supported by InterWeft IT Consultants (Melbourne, Australia).

The latest version of this document is available at the InterWeft WWW site at InterWeft IT Consultants and from The Linux Documentation Project.

2. Introduction

With available IP network numbers rapidly becoming an endangered species, efficient use of this increasingly scarce resource is important.

This document describes how to split a single IP network number up so that it can be used on several different networks.

This document concentrates on C Class IP network numbers - but the principles apply to A and B class networks as well.

2.1 Other sources of information

There are a number of other sources of information that are of relevance for both detailed and background information on IP numbers. Those recommended by the author are:-

- The Linux Network Administrators Guide.
- The Linux System Administration Guide.
- TCP/IP Network Administration by Craig Hunt, published by O'Reilly and Associates.

3. The Anatomy of IP numbers

Before diving into the delight of sub-networking, we need to establish some IP number basics.

3.1 IP numbers belong to Interfaces - NOT hosts!

First of all, let's clear up a basic cause of misunderstanding - IP numbers are **not** assigned to hosts. IP numbers are assigned to network interfaces on hosts.

Eh - what's that?

Whilst many (if not most) computers on an IP network will possess a single network interface (and have a single IP number as a consequence), this is not the only way things happen. Computers and other devices can have several (if not many) network interfaces - and each interface has its own IP number.

So a device with 6 active interfaces (such as a router) will have 6 IP numbers - one for each interface to each network to which it is connected. The reason for this becomes clear when we look at an IP network!

Despite this, most people refer to *host addresses* when referring to an IP number. Just remember, this is simply shorthand for *the IP number of this particular interface on this host*. Many (if not the majority) of devices on the Internet have only a single interface and thus a single IP number.

3.2 IP Numbers as "Dotted Quads"

In the current (IPv4) implementation of IP numbers, IP numbers consist of 4 (8 bit) bytes - giving a total of 32 bits of available information. This results in numbers that are rather large (even when written in decimal notation). So for readability (and organisational reasons) IP numbers are usually written in the 'dotted quad' format. The IP number

```
192.168.1.24
```

is an example of this - 4 (decimal) numbers separated by (.) dots.

As each one of the four numbers is the decimal representation of an 8 bit byte, each of the 4 numbers can range from 0 to 255 (that is take on 256 unique values - remember, zero is a value too).

In addition, part of the IP number of a host identifies the network on which the host resides, the remaining 'bits' of the IP number identify the host (oops - network interface) itself. Exactly how many bits are used by the network ID and how many are available to identify hosts (interfaces) on that network is determined by the network 'class'.

3.3 Classes of IP Networks

TOC

There are three classes of IP numbers

- Class A IP network numbers use the leftmost 8 bits (the leftmost of the dotted quads) to identify the network, leaving 24 bits (the remaining three dotted quads) to identify host interfaces on that network.
  Class A addresses **always** have the leftmost bit of the leftmost byte a zero - that is a decimal value of 0 to 127 for the first dotted quad. So there are a maximum of 128 class A network numbers available, with each one containing up to 33,554,430 possible interfaces.

  However, the networks 0.0.0.0 (known as the default route) and 127.0.0.0 (the loop back network) have special meanings and are not available for use to identify networks. So there are only 126 *available* A class network numbers.
- Class B IP network numbers use the leftmost 16 bits (the leftmost two dotted quads) to identify the network, leaving 16 bits (the last two dotted quads) to identify host interfaces. Class B addresses always have the leftmost 2 bits of the leftmost byte set to 1 0. This leaves 14 bits left to specify the network address giving 32767 available B class networks. B Class networks thus have a range of 128 to 191 for the first of the dotted quads, with each network containing up to 32,766 possible interfaces.
- Class C IP network numbers use the leftmost 24 bits (the leftmost three bytes) to identify the network, leaving 8 bits (the rightmost byte) to identify host interfaces. Class C addresses always start with the leftmost 3 bits set to 1 1 0 or a range of 192 to 255 for the leftmost dotted quad. There are thus 4,194,303 available C class network numbers, each containing 254 interfaces. (C Class networks with the first byte greater than 223 are however reserved and unavailable for use).

In summary:

```
Network class   Usable range of first byte values (decimal)
        A                 1 to 126
        B               128 to 191
        C               192 to 254
```

There are also special addresses that are reserved for 'unconnected' networks - that is networks that use IP but are not connected to the Internet, These addresses are:-

- One A Class Network
  10.0.0.0
- 16 B Class Networks
  172.16.0.0 - 172.31.0.0
- 256 C Class Networks 192.168.0.0 - 192.168.255.0

You will note that this document uses these sequences throughout to avoid confusion with 'real' networks and hosts.

3.4 Network numbers, interface addresses and broadcast addresses

TOC

IP numbers can have three possible meanings:-

- the address of an IP network (a group of IP devices sharing common access to a transmission medium - such as all being on the same Ethernet segment). A network number will always have the interface (host) bits of the address space set to 0 (unless the network is sub-networked - as we shall see);
- the broadcast address of an IP network (the address used to 'talk', simultaneously, to all devices in an IP network). Broadcast addresses for a network always have the interface (host) bits of the the address space set to 1 (unless the network is sub-networked - again, as we shall see).
- the address of an interface (such as an Ethernet card or PPP interface on a host, router, print server etc). These addresses can have any value in the host bits **except** all zero or all 1 - because with the host bits all 0, the address is a network address and with the host bits all 1 the address is the broadcast address.

In summary and to clarify things

```
For an A class network...
(one byte of network address space followed by three bytes of host
address space)

        10.0.0.0 is an A Class  network number  because all the host
                bits of the address space are 0
        10.0.1.0 is a host address on this network
        10.255.255.255 is the broadcast address of this network
                because all the host bits of the address space are 1

For a B class network...
(two bytes of network address space followed by two bytes of host
address space)

        172.17.0.0 is a B Class network number
        172.17.0.1 is a host address on this network
        172.17.255.255 is the network broadcast address

For a C Class network...
(three bytes of network address space followed by one byte of host
address space)

        192.168.3.0 is a C Class network number
        192.168.3.42 is a host address on this network
        192.168.3.255 is the network broadcast address
```

Almost all IP network numbers remaining available for allocation at present are C Class addresses.

3.5 The network mask

The network mask is more properly called the subnetwork mask. However, it is generally referred to as the network mask.

It is the network mask and its implications on how IP addresses are interpreted *locally* on an IP network segment that concerns us most here, as this determines what (if any) sub-networking occurs.

The standard (sub-) network mask is all the network bits in an address set to '1' and all the host bits set to '0'. This means that the standard network masks for the three classes of networks are:-

- A Class network mask: 255.0.0.0
- B Class network mask: 255.255.0.0
- C Class network mask: 255.255.255.0

There are two important things to remember about the network mask:-

- The network mask affects only the **local** interpretation of **local** IP numbers (where local means on this particular network segment);
- The network mask is **not** an IP number - it is used to modify how local IP numbers are interpreted locally.

---

4. What are subnets?

A subnet is a way of taking a single IP network address and **locally** splitting it up so that this single network IP address can actually be used on several interconnected local networks. Remember, a single IP network number can only be used on a single network.

The important word here is **locally**: as far as the world outside the machines and physical networks covered by the sub-netted IP network are concerned, nothing whatsoever has changed - it is still just a single IP network. This is important - sub-networking is a **local** configuration and is invisible to the rest of the world.

5. Why subnetwork?

The reasons behind sub-networking date back to the early specification of IP - where just a few sites were running on Class A network numbers, which allow for millions of connected hosts.

It is obviously a huge traffic and administration problem if all IP computers at a large site need to be connected to the same network: trying to manage such a huge beast would be a nightmare and the network would (almost certainly) collapse under the load of its own traffic (saturate).

Enter sub-networking: the A class IP network address can be split up to allow its distribution across several (if not many) separate networks. The management of each separate network can easily be delegated as well.

This allows small, manageable networks to be established - quite possibly using different networking technologies. Remember, you cannot mix Ethernet, Token Ring, FDDI, ATM etc on the same physical network - they can be interconnected, however!

Other reasons for sub-networking are:-

- Physical site layout can create restrictions (cable run lengths) in terms of the how the physical infrastructure can be connected, requiring multiple networks. Sub-networking allows this to be done in an IP environment using a single IP network number.

  This is in fact now very commonly done by ISPs who wish to give their permanently connected clients with local networks static IP numbers.
- Network traffic is sufficiently high to be causing significant slow downs. By splitting the network up using subnetworks, traffic that is local to a network segment can be kept local - reducing overall traffic and speeding up network connectivity without requiring more actual network bandwidth;
- Security requirements may well dictate that different classes of users do not share the same network - as traffic on a network can always be intercepted by a knowledgeable user. Sub-networking provides a way to keep the marketing department from snooping on the R & D network traffic (or students from snooping on the administration network)!
- You have equipment which uses incompatible networking technologies and need to interconnect them (as mentioned above).

6. How to subnetwork a IP network number

Having decided that you need to subnetwork your IP network number, how do you go about it? The following is an overview of the steps which will then be explained in detail:-

- Set up the physical connectivity (network wiring and network interconnections - such as routers;
- Decide how big/small each subnetwork needs to be in terms of the number of devices that will connect to it - ie how many usable IP numbers are required for each individual segment.
- Calculate the appropriate network mask and network addresses;
- Give each interface on each network its own IP address and the appropriate network mask;
- Set up the routes on the routers and the appropriate gateways, routes and/or default routes on the networked devices;
- Test the system, fix problems and then relax!

For the purpose of this example, we will assume we are sub-networking a single C class network number: 192.168.1.0

This provides for a maximum of 254 connected interfaces (hosts), plus the obligatory network number (192.168.1.0) and broadcast address (192.168.1.255).

6.1 Setting up the physical connectivity

You will need to install the correct cabling infrastructure for all the devices you wish to interconnect designed to meet your physical layout.

You will also need a mechanism to interconnect the various segments together (routers, media converters etc.).

A detailed discussion of this is obviously impossible here. Should you need help, there are network design/installation consultants around who provide this sort of service. Free advice is also available on a number of Usenet news groups (such as comp.os.linux.networking).

6.2 Subnetwork sizing

There is a play off between the number of subnetworks you create and 'wasted' IP numbers.

Every individual IP network has two addresses unusable as interface (host) addresses - the network IP number itself and the broadcast address. When you subnetwork, each subnetwork requires its own, unique IP network number and broadcast address - and these have to be valid addresses from within the range provided by the IP network that you are sub-networking.

So, by sub-networking an IP network into two separate subnetworks, there are now **two** network addresses and **two** broadcast addresses - increasing the 'unusable' interface (host) addresses; creating 4 subnetworks creates **eight** unusable interface (host) addresses and so on.

In fact the smallest usable subnetwork consists of 4 IP numbers:-

- Two usable IP interface numbers - one for the router interface on that network and one for the single host on that network.

TOC

- One network number.
- One broadcast address.

Quite why one would want to create such a small network is another question! With only a single host on the network, any network communication must go out to another network. However, the example does serve to show the law of diminishing returns that applies to sub-networking.

In principle, you can only divide your IP network number into 2^n (where n is one less that the number of host bits in your IP network number) equally sized subnetworks (you can subnetwork a subnetwork and combine subnetworks however).

So be realistic about designing your network design - you want the **minimum** number of separate local networks that is consistent with management, physical, equipment and security constraints!

6.3 Calculating the subnetwork mask and network numbers

The network mask is what performs all the **local** magic of dividing an IP network into subnetworks.

The network mask for an un-sub-networked IP network number is simply a dotted quad which has all the 'network bits' of the network number set to '1' and all the host bits set to '0'.

So, for the three classes of IP networks, the standard network masks are:-

- Class A (8 network bits) : 255.0.0.0
- Class B (16 network bits): 255.255.0.0
- Class C (24 network bits): 255.255.255.0

The way sub-networking operates is to *borrow* one or more of the available host bits and make then make interfaces **locally** interpret these borrowed bits as part of the network bits. So to divide a network number into two subnetworks, we would borrow one host bit by setting the appropriate bit in the network mask of the first (normal) host bit to '1'.

For a C Class address, this would result in a netmask of
11111111.11111111.11111111.10000000
or 255.255.255.128

For our C Class network number of 192.168.1.0, these are some of the sub-networking options you have:-

---

```
No of       No of
subnets     Hosts/net    netmask
2             126        255.255.255.128 (11111111.11111111.11111111.10000000)
4              62        255.255.255.192 (11111111.11111111.11111111.11000000)
8              30        255.255.255.224 (11111111.11111111.11111111.11100000)
16             14        255.255.255.240 (11111111.11111111.11111111.11110000)
32              6        255.255.255.248 (11111111.11111111.11111111.11111000)
64              2        255.255.255.252 (11111111.11111111.11111111.11111100)
```

In principle, there is absolutely no reason to follow the above way of subnetworking where network mask bits are

added from the most significant host bit to the least significant host bit. However, if you do not do it this way, resulting IP numbers will be in a *very* odd sequence! This makes it extremely difficult for us humans to decide which subnetwork an IP number belongs as we are not too good at thinking in binary (computers on the other hand are and will use whatever scheme you tell them with equal equanimity).

Having decided on the appropriate netmask, you then need to work out what the various Network and broadcast addresses are - and the IP number range for each of these networks. Again, considering only a C Class IP Network number and listing only the *final* (host part) we have:-

Netmask	Subnets	Network	B'cast	MinIP	MaxIP	Hosts	Total Hosts
128	2	0	127	1	126	126	
		128	255	129	254	126	252
192	4	0	63	1	62	62	
		64	127	65	126	62	
		128	191	129	190	62	
		192	255	193	254	62	248
224	8	0	31	1	30	30	
		32	63	33	62	30	
		64	95	65	94	30	
		96	127	97	126	30	
		128	159	129	158	30	
		160	191	161	190	30	
		192	223	193	222	30	
		224	255	225	254	30	240

As can be seen, there is a very definite sequence to these numbers, which make them fairly easy to check. The 'downside' of sub-networking is also visible in terms of the reducing total number of available host addresses as the number of subnetworks increases.

With this information, you are now in a position to assign host and network IP numbers and netmasks.

7. Routing

If you are using a Linux PC with two network interfaces to route between two (or more) subnets, you need to have IP Forwarding enabled in your kernel. Do a

```
cat /proc/ksyms | grep ip_forward
```

You should get back something like...

```
00141364 ip_forward_Rf71ac834
```

If you do not, then you do not have IP-Forwarding enabled in your kernel and you need to recompile and install a new kernel.

For the sake of this example, let us assume that you have decided to subnetwork you C class IP network number 192.168.1.0 into 4 subnets (each of 62 usable interface/host IP numbers). However, two of these subnets are being combined into a larger single network, giving three physical networks.

These are :-

Network	Broadcast	Netmask	Hosts
192.168.1.0	192.168.1.63	255.255.255.192	62
192.168.1.64	192.168.1.127	255.255.255.192	62
182.168.1.128	192.168.1.255	255.255.255.126	124 (see note)

Note: the reason the last network has only 124 usable network addresses (not 126 as would be expected from the network mask) is that it is really a 'super net' of two subnetworks. Hosts on the other two networks will interpret 192.168.1.192 as the *network* address of the 'non-existent' subnetwork. Similarly, they will interpret 192.168.1.191 as the broadcast address of the 'non-existent' subnetwork.

So, if you use 192.168.1.191 or 192 as host addresses on the third network, then machines on the two smaller networks will not be able to communicate with them.

This illustrates an important point with subnetworks - the usable addresses are determined by the **SMALLEST** subnetwork in that address space.

7.1 The routing tables

TOC

Let us assume that a computer running Linux is acting as a router for this network. It will have three network interfaces to the local LANs and possibly a fourth interface to the Internet (which would be its default route.

Let us assume that the Linux computer uses the lowest available IP address in each subnetwork on its interface to that network. It would configure its network interfaces as

```
Interface        IP Address       Netmask
eth0             192.168.1.1      255.255.255.192
eth1             192.168.1.65     255.255.255.192
eth2             192.168.1.129    255.255.255.128
```

The routing it would establish would be

```
Destination      Gateway          Genmask          Iface
192.168.1.0      0.0.0.0          255.255.255.192 eth0
192.168.1.64     0.0.0.0          255.255.255.192 eth1
192.168.1.128    0.0.0.0          255.255.255.128 eth2
```

On each of the subnetworks, the hosts would be configured with their own IP number and net mask (appropriate for the particular network). Each host would declare the Linux PC as its gateway/router, specifying the Linux PCs IP address for its interface on to that particular network.

Robert Hart Melbourne, Australia March 1997.

ISP-Connectivity-mini-HOWTO

Michael Strates, mstrates@croftj.net

v2.0, 6 November 1997

This document describes how to setup PPP, connect up to your ISP, configure mail and news, get a permanent IP (if available), get a domain name, and have a bonda fide system running in a little over thirty minutes.

1. Introduction

- 1.1 New versions of this document
- 1.2 Feedback
- 1.3 Standard Disclaimer
- 1.4 Copyright Information

2. Connecting to the Outside World

- 2.1 Talking and Communicating with pppd and chat
- 2.2 IP's, Domain Names and Subnets

3. Electronic Mail on your Linux Box

1. Introduction

The main goal of this document obviously is to make the new user friendly with the many terms of connecting your Linux PC up to the Internet, obtaining IP addresses, domain names, and setting things up. This guide is intended for the intermediate user in mind, although intelligent newbies shouldn't have any problems.

1.1 New versions of this document

New versions of this document will be periodically posted to *comp.os.linux.answers*. They will also be added to the various anonymous FTP sites who archive such information, including:

ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO

In addition, you should generally be able to find this document on the Linux Documentation Project page via:

http://sunsite.unc.edu/LDP/

1.2 Feedback

I certaintly welcome any feedback about this HOWTO, spelling mistakes, how it all worked out, thankyou notes and critisisms. I hope I helped a few people with this HOWTO, and if I did, I'd be really happy to hear from you.

mstrates@croftj.net

http://linloft.home.ml.org/

1.3 Standard Disclaimer

No liability for the contents of this documents can be accepted. Use the concepts, examples and other content at your own risk. As this is a new edition of this document, there may be errors and inaccuracies, that may of course be damaging to your system. Proceed with caution, and although this is highly unlikely, I don't take any responsibility for that.

Naturally, there are probably better and easier ways to do things in this document. There will always be another way in the Linux World. This is the way I've done things, and that's the way I'll be presenting them in this HOWTO.

1.4 Copyright Information

TOC

This document is copyrighted (c)1997 Michael Strates and distributed under the following terms:

- Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.
- All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.
- If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at

linux-howto@sunsite.unc.edu Finger for phone number and snail mail address.

---

2. Connecting to the Outside World

In this document, we'll explain how to do this using PPP (Point to Point Protocol), a popular protocol nearly always used over the Internet. It allows your modem to `speak` to the outside world. This is what applications like Trumpet Winsock in Windows 3.x did, and many other programs that you've probably have never seen.

In Linux, we use a thing called chat to do the dialing up to the ISP and then use a utility called pppd to 'use' the connection. In a sense, chat is your dialer, and pppd is your protocol. We'll describe how to setup both below.

2.1 Talking and Communicating with pppd and chat

Probably the easiest way to go about things is to make a shell script in root's home directory called `ppp-connect` and involke the script whenever you wish to make your connection. We'll discuss this method.

Open up your favourite editor as root on /ppp-connect. You'll then have to decide on your parameters.

pppd connect 'chat -v "" "your_init_string" "" ATDTisp_number CONNECT "" ogin: your_username word: your_passwd' /dev/tty(0/1/2) speed modem

pppd involkes /usr/sbin/pppd on my system, then loads up chat to do the dialing. Chat sends *your_init_string* to the modem, then dials *isp_number*. It then waits for CONNECT, then waits for ogin: (l removed as the first character is sometimes lost), sends *your_passwd*, chat then terminates and hands the show over to pppd.

The last of the command specifies your modem port (mine's /dev/ttyS1). In most cases it will be ttyS1 (COM2: in DOS), ttyS0 (COM1: in DOS), or if your using Slackware, cua1 or cua0. The speed is the speed of the modem. I use 115200 for my modem (a 33.6k). If you have got a fairly recent computer (one with a 16550 UART), then I wouldn't go any lower than 57600. Otherwise, for 14.4k 38400. Modem just tells pppd that it's a serial/modem based connection. Remove the -v option if you don't want verbose logging to your logfiles.

The scenario below is one of a person who dials up an ISP that automatically starts PPP for them, ie; they don't have a shell that actually starts. This is his command in his /ppp-connect:

pppd connect 'chat "" "ATZ" "" ATDT555-1800 CONNECT "" ogin: johnny word: blackjak' /dev/ttyS1 115200 modem

But for some people, they're ISP starts up a shell and doesn't automatically start PPP this may be a problem. Luckily, chat can deal with that too. You just add another command to your chat script. For example, below this johnny character is using an ISP that just dumps him to a shell, requiring him to type ppp to get a ppp connection. His shell prompt ends with a $. *pppd connect 'chat "" "ATZ" ""*

ATDT555-1800 CONNECT "" ogin: johnny word: blackjak $ ppp' /dev/ttyS1 115200 modem

If it's more than one word, ensure you quote it. I hope you can see the drift of this, and are able to create your own script up to suit your connection. Simply modify either the first johnny or the second johnny script to suit your taste, port, server, etc and save the file.

Now you've made your file, ensure that only root can execute, read or write to it. This is extreemly important. Also make sure nobody can read your logfiles, if you decide to leave the -v option in, as your password is seen in cleartext in the logs (I don't see much need for -v, if you don't know what I'm talking about, leave -v out).

2.2 IP's, Domain Names and Subnets

For most people using the options above, a changing IP address won't bother them. These people include basic, easy going users, that just have dialup accounts, and aren't very technically minded. For those people, skim read this section, I'll come to important things you need to do to setup your system properly. Newbies, skip the sections dealing with permanent IP, Domain Names, Subnets, and just read the last bit of this section.

Getting a permanent IP address might be free for your ISP, so if in doubt ask them. Personally, I'd pay for a permanent IP address. It lets you send e-mail to and from using a unique IP or domain, etc. If you want to get yourself a permanent IP, write an e-mail to root@yourisp.com, and ask him nicely if he can arrange a permanent IP for you.

When you get your permanent IP address, grep through your /etc directory to find where your old IP addresses are. I had to change files in my sendmail directory and /etc/hosts. There are some other key files that you will only discover with grepping. Open up /etc/hosts, and add your new IP address in the standard format. Reboot your computer, and you should be ready to go.

You'll now need to change your chat script to reflect your new settings. If you are forced into PPP as soon as you start your connection, you'll need to tell your System Administrator of your ISP to ensure their PPP system recognises that you have a permanent IP address and allocates you that instead of a changing one. If you get dumped at a shell prompt, and you need to type ppp or something to start the connection, instead of typing that, change your /ppp-connect script to send this instead of just ppp or whatever when it sees $ or whatever your shell prompt is.

/usr/sbin/pppd :Your_IP_Address

Substitute your IP address for the IP address your ISP gave to you. Be sure you encapsulate the thing in " " marks when you put it into your chat script. If this doesn't work, consult your ISP where your PPP daemon is located, and ask him for the command to give. You could just try leaving it as is and seeing if the server will recognise you and give you your rightful address.

The next thing probably to do is to get yourself a domain name. I know that in Australia, .asn.au and .org.au are free. In the United States, you can get a .us domain for free, but they tend to be long. If your in Australia, you must go to http://www.aunic.net/ to register your domains. In the United States, it is http://www.internic.net/ .

TOC

To register domains you need to be able to provide DNS services, and gorey stuff like that. If your IS can't provide these, throw out an official .asn.au or whatever domain out the window, and get a Mon Internet Domain.

Monolith offer free domains to anybody and anyone all around the world. Everything is done without human interaction, via a web forms interface with your browser. Your domain comes in the form of Your_Choice.ml.org. Monolith will then host the DNS locally for you. If you want to send and receive mail from that domain, ask your ISP to become a mail exchanger for you.

Go to http://www.ml.org/ and fill out an application, enter the NIC with your username and password, and make a FREED domain. You'll need to enter your IP address, so have that ready. Your domain will be in the DNS in a couple of days.

Okay now, we'll move onto the newbies section, or for those people who can't get a permanent IP address or a domain name. All you have to do is edit /etc/hosts as root, call your site something that won't clash, give it a 10.10.10 or something for an IP address and reboot your computer.

There you go, you've just setup your computer with pppd and chat in just ten minutes. Now let's move onto the next section, which deals with Electronic Mail.

---

3. Electronic Mail on your Linux Box

One of the most important aspects of the Internet, is it's fasinating capaiblity to transfer mail to and from countries, or more locally perhaps. Linux is extreemly strong in easy mail packages for the console. The one we're going to document today is called Pine (Program for Internet Mail and News), made by the University of Washington, and to download the mail, a program called Fetchmail, made by Eric S. Raymond. Both should be included in your Linux distribution.

Fetchmail is a program that downloads your e-mail from your server using POP, transfers the mail onto your computer and then deletes it off the server, much like programs like Eudora or Microsoft Internet Mail/Exchange do. To configure and automate fetchmail, you use a file in your home directory called .fetchmailrc. Simply open up /.fetchmailrc (Remember: your doing this bit as yourself, not as root) with your favourite editor and observe the command lin eoptions below:

poll mail.yourisp.com proto pop3 user login_name password your_passwd

user login_name with pass your_passwd is login_name here

All you have to do is replace *mail.yourisp.com* with the name of the mail server of your ISP, *your_passwd* with your password, and *login_name* with your login name.

An important thing to note. For Pine and this procedure to work correctly, your login name must corrospond with the login name you use on your ISP. That is your local login name must match the one you use on your server, and your e-mail address.

Next, ensure that .fetchmailrc has the correct permissions (user read/write only) and your laughing. Fetchmail can be started in two ways, in standard mode (where it'll fetch messages from the server and terminate), or in daemon mode (where it will stay active, and check/download mail every X seconds). To use daemon mode, type *fetchmail -a -d(Seconds between Polls)*. -a ensures it downloads all mail. To use the standard mode, just type *fetchmail -a*.

Next, you need to setup Pine. Open up Pine, by typing pine at your prompt, choose Setup - Configuration. Setup your userdomain as the domain in your e-mail address, for example jack@linux.org, would be linux.org. Next, setup smtp-server as your POP mail server (the same you used in the fetchmail setup). So we enter www.linux.org. If you want news, setup your nntp server to your ISP's news server.

So there you have it folks, everything should be working now. To connect up to your ISP, just run /ppp-connect as root. Then, to get your e-mail run fetchmail -a as yourself. To browse your e-mail and news, use Pine. Install a text-based browser such as Lynx to browse the web if you like.

Send any comments questions and suggestions to mstrates@croftj.net

Linux Install From PPA-Zip drive mini-HOWTO

by Kevin Snively, k.snively@seaslug.org

v01.04, 29 April 1998

This document describes how to install Linux from a parallel port zip drive using the Slackware Linux distribution.

1. Prelude

- 1.1 Intro:
- 1.2 Background:
- 1.3 Advantages:
- 1.4 Material:
- 1.5 Conventions:
- 1.6 Other reading:

2. Slackware

- 2.1 Making the first zip disk.
- 2.2 Making the second zip disk.
- 2.3 Making the boot floppy.

3. The install:

- 3.1 Round 1.
- 3.2 Round 2.
- 3.3 Library Trouble

4. Thank you

1. Prelude

1.1 Intro:

The following method has been written for use with the Slackware Linux distribution. It assumes some familiarity with it's installation. The installer will also need access to a working Linux system of any distribution. It also assumes the destination system has 4 to 6 megs ram. More powerful systems can use this method also but you will probably be able to find a faster easier way. It is not intended to be used by first time installers or by people new to Linux. Only issues specific to the use of a zip drive as source media will be covered here.

1.2 Background:

I have an OLD 486/20 slc with a 120 meg hd, 4 megs ram and a 2400 baud modem that I wanted to run Linux on. It has no cdrom drive or pcmcia connectivity, and will never see a network card (parallel port style). There is nothing left that can be upgraded on this unit.

Linux has been at the cutting edge from the start but also breathes new life into some older 386 and 486 hardware. The documentation says "386 with 2 megs ram no math coprocessor" but try to find a distribution that will do it today. Slackware can be placed on a 4 meg machine with some work and a prayer. The following system greatly reduces the work. Prayer always helps, even with the best of systems.

I've used the following system with 3 other laptops, and one 386 desktop machine. The lamest being a 386 with 4 megs ram & 60 meg HD no math coprocessor. While this system works well I'd like to be able to do similar tricks with Debian, RedHat, Caldera Lite, and some of the other popular distributions offering more choice to Linux users I help this way.

1.3 Advantages:

Installing from cdrom or Ethernet are the methods of choice when available. In cases where floppy disk install appears to be the only choice a parallel port zip drive offers numerous advantages. These include better speed than a floppy disk, and fewer disk changes. In addition on systems with minimal ram "6 megs. or less" the ability to install a swap file on the root disk uncomplicates the install greatly by freeing you from the need to creat one on the hard disk before running setup. When installing Slackware the swap file allows installers to choose the **color.gz** root disk or **text.gz**. The **color.gz** is not recommended for floppy based installs on machines with such limited resources. With a zip disk you may also produce an emergency boot disk during the initial install, by freeing the first floppy drive. Something not possible during a normal floppy based install on systems of this type.

TOC

1.4 Material:

This method has been used with Slackware Ver's. 3.1, through 3.4 A full Slackware install requires 2 zip disk's not counting source packages if they are desired. You will need one parallel port zip drive & cables, and a 1.44 meg boot disk. Pencil and paper are optional for making a few notes during install.

At least two systems are required to use this install method. The first (host) system is used to prepare the installation media and must contain a working Linux system. The second will be the target system. Once the media has been prepared it can be used to install to as many target systems as you like. I work from cdrom but you can down load your distribution from the Internet if you have a fast enough connection. For me this is not practical.

1.5 Conventions:

```
let (floppy)    represent the path to and directory you
                mount your 1.44 meg floppy disk on.

let    (zip)    represent the path to and directory you
                mount your zip disk on.

let  (cdrom)    represent the path to and directory you
                mount your cdrom on.

let (verx.x)    represent the distribution version you
                are installing.
```

If you down load your disk images please let (cdrom)/slakware the path and directory to those images. My parallel port zip drive shows up on my system as

```
/dev/sda4
```

I've never seen it come up as any thing else unless I run fdisk on it and change the partition table, creating more than one partition on the disk. Then I always get corrupt disk error messages when I try to use it. You may however mark the partition type 83 Linux etc... if you wish with out trouble. You also need to know the device names of your cdrom and 1.44 meg floppy drives. Mine are

```
/dev/hdc     cdrom
/dev/fd0     1.44 meg floppy
```

I'll use my device names, you use yours ;-|)

About Slackware and slakware (note the missing c in slakware). Slackware is the name of the overall

distribution and slakware is the name of the directory where Patrick Volkerding places the stable "no experimental or developmental" packages for installation. Whenever I'm writing about slakware I'm writing about the directory.

1.6 Other reading:

Linux Installation and Getting Started by Matt Welsh

If you are having trouble accessing your zip drive please see the Linux ZIP drive mini-HOWTO by Grant Guenther

When you have completed your install I can not recommend strongly enough that you work through the Small-Memory mini-HOWTO by Todd Burgess

Slackware

I also recommend the LOWMEM.TXT document from the Slackware documentation on your cdrom or Slackware ftp site. This document discribes a system that eleminates many of the difficulties discribed in that document.

---

2. Slackware

2.1 Making the first zip disk.

1) Attach zip drive to host system, insert blank zip disk into the drive and boot the system to Linux

2) Login. You probably need to login as root or the user you log in as will need to be able to write file systems, mount and unmount disks etc....

3) Install an ext2fs file system on the zip disk. with the command:

```
/sbin/mke2fs -b 1024 /dev/sda4
```

4) Mount the zip disk with:

```
mount /dev/sda4 /(zip)
```

5) Insert your cdrom into its drive and mount it:

```
mount /dev/hdc /(cdrom)
```

6) Make a directory to install the slakware disk images in:

```
mkdir /(zip)/slakware
```

7) Make a writable (non zip disk) temporary directory on your main disk to work in. May I suggest:

```
mkdir /root/slakware/(verx.x)
```

8) Copy the root disk image to the above named directory:

```
cp /(cdrom)/rootdsks/text.gz /root/slakware/(verx.x)/
```

9) Change directories to the one the root disk image was copied to and decompress it with the commands:

```
cd /root/slakware/(verx.x)
gunzip text.gz
```

TOC

10) Install a 1.44 meg floppy with a dos or ext2fs file system on it in its drive and copy the root disk image to it this time with the dd command:

```
dd if=text of=/dev/fd0
```

11) Mount the 1.44 meg floppy with the command:

```
mount /dev/fd0 /(floppy)
```

12) Copy the root disk image to the zip disk:

```
cp -dpR /(floppy)/* /(zip)/
```

13) In order to ease the install process later we will want to create a swap file on the zip disk of about 8 megs. with:

```
dd if=/dev/zero of=/(zip)/swap bs=1024 count=8208
```

14) Initialize the swap file with:

```
/sbin/mkswap /(zip)/swap 8208
```

15) Next with your favorite text editor you will need to edit the file **/(zip)/etc/rc** and after the line **/bin/mount -av -t nonfs** insert a new line

```
/sbin/swapon /swap
```

16) Copy the disk images to the zip disk:

```
cp -r /(cdrom)/slakware/[a-ty]* /(zip)/slakware/ &
```

include the square brackets [a-ty]* exactly as shown above. the & sign at the end of the line will make this command execute in the background. It takes about 20 min to complete. Use this time for some coffee and reading more HOWTO's.

17) External zip drives have two lights--mine are green and yellow; the internal drives appear to have only one. When my zip disk has finished receiving information the yellow light goes out and I'm left with just the green power light. on internal drives the single "status?" light will go out.

It is now time to prepare the floppy and zip disks for removal from their drives by unmounting them:

```
cd
umount /dev/fd0
umount /dev/sda4
```

18) Life will be easier later if we check the root disk's file system before we try to use it with the command:

```
/sbin/e2fsck -f /dev/sda4
```

When disk activity ends if there are no errors you may remove the zip disk from the drive and label it Slackware Root disk. I also include the version number in pencil.

2.2 Making the second zip disk.

If the X window system is to be installed or you will be needing one of the other prebuilt kernels it is now time to build the 2nd. zip disk.

x1) Insert disk 2 in the zip drive. I like to write an ext2fs file system on this disk also, but if it has a M$ Dos file system that will work also. I use the up arrow key on my keyboard to recall previously used commands from my history file because I'm a bit lazy, but if you want to retype the command repeat instructions 3), 4), and 6) above.

x2) Copy the X11R6 "X Window" packages to zip

```
cp -r /(cdrom)/slakware/x* /(zip)/slakware/ &
```

again in the background. This will take about 1/2 as long as step 9) did. If your system was slow then it will be again. More coffee and HOWTO reading.

x3) There is also room on this disk for the custom kernels directory. It is classy to install one that meets the system's needs as closely as possible to go with the one needed to access the zip drive. Make this kernel the system default. So now we will make a directory for that and then copy the information to it:

```
mkdir /(zip)/kernels
cp -r /(cdrom)/kernels/* /(zip)/kernels/ &
```

This will take another 10 or 15 min. Dispose of the used coffee? Read more Linux Documentation?

x4) When disk activity is at an end unmount the disk and run fsck on it:

```
umount /dev/sda4
/sbin/e2fsck -f /dev/sda4
```

x5) When e2fsck is done with the disk and the command prompt returns to the screen eject the disk and label it.

2.3 Making the boot floppy.

b1) I now make a 1.44 meg boot disk. Use the **/(cdrom)/bootdsks.144/iomega.s** image. Install a new 1.44 meg floppy disk in the proper drive. The command to make the disk is:

```
dd if=/(cdrom)/bootdsks.144/iomega.s of=/dev/fd0
```

b2) Check the file systems of the boot disk:

TOC

```
/sbin/e2fsck -f /dev/fd0
```

When disk activity ends if there are no errors you may remove all your disks, write protect your boot floppy and label it including version number.

---

3. The install:

After connecting everything up to the target machine and putting the zip disk in place I boot the machine with the appropriate boot disk. Before logging in it is a good idea to use the scroll back feature built into Linux and check to see all your hardware was detected properly. To do this hold the shift key and use the page up and page down keys as appropriate. Getting a login prompt at all usually means the boot time errors were not too bad.

3.1 Round 1.

Slackware will present some instructions. Please read them Carefully. Login as root and fdisk root and swap partitions on the target drive if needed. Reboot the machine if you ran fdisk and **Carefully** reread the instructions on the screen if there are any, then login as root.

Even on machines with only 4 megs. of ram you will not need to create a swap partition until prompted to do so in the setup program. We already have swap running on the zip disk. Run the Slackware setup program:

```
setup
```

CAREFULLY read and follow the instructions the menus provide. The first thing setup will want to do is create and initialize your swap partition on your hard drive this is ok despite the dire warnings and will not affect the swap file on the zip drive.

You do not want to allow /dev/sda4 to be automatically mounted and when questions come up about this in the menus. Do not add it unless this zip drive is dedicated to the target machine full time.

When you get the menu that asks where the source of the installation media is select item number 4. "Install from a premounted directory". The next screen will ask for its name, it is /slakware.

I recommend installing only the "a" disk set at this time. It will crash the install if the person selecting packages gets greedy and fills the disk before lilo is written and the kernel is installed. During basic install I install the kernel from the boot disk only. There are relatively few choices to be made from here and the Slackware documentation from the cdrom or your ftp site will do you much better than I can from here.

Leave the zip drive and disk in place after the initial install is complete. Remove the floppy when prompted to do so and reboot. If all is right in the world Bill, er, your friend/customer will have a working Linux system come up.

3.2 Round 2.

After rebooting the system login as root and mount the zip disk again:

```
mount /dev/sda4 /mnt
```

Check available drive space:

```
df
```

Run setup again from your new system:

```
setup
```

Choose "s" SOURCE from the menu and press the enter key. Select item number 4. "Install from a premounted directory" and press the enter key again.

Enter the directory name. This time use:

```
/mnt/slakware
```

Choose the packages you want or need. All are available except those which start with "x". When you're finished installing packages from this disk exit setup and run:

```
umount /dev/sda4
```

You may now remove the disk and if X will be installed insert the 2nd zip disk and work through setup again this time only packages beginning with X will be available. If you will be installing a custom kernel from the prebuilt kernels you may also do this while the 2nd. zip disk is installed.

3.3 Library Trouble

Some times I have trouble with the links for the libraries either on the root zip disk or on the final destination machine. There are two possible fixes for this.

Fix 1) If the system boots to a command prompt and you can login as root in single user mode try the command:

```
ldconfig
```

If things are not broken so badly that this command will not run then reboot and watch for library related errors. If there are none your in business.

Fix 2) If **ldconfig** can not repair things you will need to re master the root floppy disk and recopy the

files from it as in instruction 8) through 12) or reinstall Linux to the destination system as appropriat

4. Thank you

This mini-HOWTO brought to you by the letters G. N. and U. and the Linux documentation project.

Special thanks to Dr. Randolph Bentson whose devotion to Linux inspired me to give something back to the people who gave us the **ONE TRUE OPERATING SYSTEM.**

```
I hope this helps someone.

Sincerely

Kevin Snively
P.O. Box 1013
Everett Wa. 98206-1013
USA
```

k.snively@seaslug.org

```
root@aldrovanda
```

The LBX Mini-HOWTO

Paul D. Smith, psmith@baynetworks.com

v1.04, 11 December 1997

LBX (Low Bandwidth X) is an X server extension which performs compression on the X protocol. It is meant to be used in conjunction with X applications and an X server which are separated by a slow network connection, to improve display and response time.

1. Introduction

2. What's The Status Of LBX?

3. Who Can Benefit From LBX?

4. Who Doesn't Need LBX?

5. How Does LBX Work?

6. What Do I Need To Use LBX?

7. What Don't I Need To Use LBX?

8. How Do I Start LBX?

9. Problems

10. Documentation

11. Alternatives

- 11.1 dxpc - The Differential X Protocol Compressor
- 11.2 Ssh (Secure Shell)
- 11.3 Which Is Better?

---

1. Introduction

Low-Bandwidth X (LBX) attempts to recognize that in this day and age, not everyone will be a fast LAN hop or two away from the system that they are running their applications on.

The X protocol can generate an extraordinary amount of traffic, especially for simple-seeming things such as creating new windows. As anyone who has tried to use X over a dial-in modem at 28.8 or even higher can attest, creating new X windows can involve an excruciating wait.

LBX is fundamentally a compression and caching scheme designed to minimize the amount of X traffic generated between two systems.

2. What's The Status Of LBX?

As of the X Consortium's release of X11R6.3 in December, 1996, LBX is a full extension to the X protocol. For XFree86 folks, that's XFree86 version 3.3.

3. Who Can Benefit From LBX?

If you use a modem to dial into a service provider, then run X applications on remote machines with their DISPLAYs set to your local machine (or vice versa), LBX will speed up that connection. Also if you set DISPLAYs from systems across WANs (other countries, for example) or other slow links, LBX can help.

4. Who Doesn't Need LBX?

LBX is useless, of course, if you're only running applications locally, or if you're not running X at all.

Also, if you're running on a fast LAN, LBX won't be much help. Some people say "if LBX cuts down on network traffic, wouldn't it be good to use even on fast LANs?" It might be, if your goal is to reduce network traffic. But if your goal is to get better response time LBX probably isn't what you want. Although it does introduce caching and compression, that comes at a cost on both ends (extra memory for caching, and extra CPU for decompression). If your link is fairly speedy LBX will probably result in an overall slowdown.

5. How Does LBX Work?

LBX works by introducing a *proxy server* at the client side, which performs caching and compression. The X server knows that the client is using a proxy server, and decompresses accordingly.

Here's a normal setup for remote X clients. In our discussion, LOCAL is always the workstation sitting in front of you, whose monitor you're looking at, and REMOTE is the remote workstation, where the actual application is running.

```
    REMOTE                                  LOCAL
+-----+                                                +-----+
| APP |-\          Network            +----------+  |     |\
+-----+  \---------------------------->| X SERVER |=>|     ||
+-----+  /       (X Protocol)          +----------+  +-----+\
| APP |-/                                            /_____//
+-----+
```

When using LBX, a proxy server (`lbxproxy`) is introduced on the remote side, and the applications talk to that process instead of directly to the LOCAL server. That process then performs the caching and compression of X requests and forwards them. It looks like this:

```
    REMOTE                                          LOCAL
                                                              +-----+
+-----+  +-------+           Network            +----------+  |     |\
| APP |->| PROXY |------------------------------->| X SERVER |=>|     ||
+-----+  +-------+       (LBX/X Protocol)       +----------+  +-----+\
+-----+   /                                                  /_____//
| APP |--/
+-----+
```

Details on exactly what caching and compression LBX does is beyond the scope of this document.

6. What Do I Need To Use LBX?

You need an X server on your LOCAL system which has the LBX extension compiled in. Unless you explicitly told it not to when building it, X11R6.3 servers automatically enable LBX. Also, all XFree86 3.3 servers have LBX enabled by default.

You can use the `xdpyinfo` command to see if your server has the LBX extension: run `xdpyinfo` and look at the list just under "number of extensions"; you should see "LBX" listed there.

Next, you need to get an `lbxproxy` program compiled for the REMOTE system. This is the tricky part. If the remote system is not the same type as your local system, the `lbxproxy` on your local system will do you no good, of course.

There is unfortunately no "broken out" distribution of `lbxproxy`, so you will have to either (a) get and build most, if not all, of X11R6.3 for the remote system, or (b) find someplace to get a pre-compiled `lbxproxy` binary for your system. The latter is much simpler of course.

The `lbxproxy` is simply a single executable. There are no configuration files, resource files, etc. associated with it.

7. What Don't I Need To Use LBX?

The REMOTE system **does not** need a new X server (as always, the REMOTE system doesn't need *any* X server running).

The application you want to run **does not** need to be linked with any special version of X, or any special libraries; I regularly use commercial X11R5 apps over LBX with no trouble.

You **do not** need root or other privileged access on the REMOTE system; the `lbxproxy` process runs under your normal access permissions. Further, you can run it right from your home directory: it does not have to be installed anywhere.

8. How Do I Start LBX?

OK, here it is... after all that it's actually quite simple. Replace LOCAL and REMOTE below with the hostnames of your local workstation and remote system, respectively (don't get them mixed up!)

On LOCAL:

1. Start your X server.
2. Tell your X server that the remote system is allowed access. Using the host-list method, type `xhost +REMOTE`. If you use `xauth` you may need to do more than this; see the *xauth(1)* man page for more information. You should consult the Remote X Apps Mini-HOWTO if you're not familiar with remote X access permission setup.

On REMOTE:

1. Start `lbxproxy` and tell it to forward to the LOCAL X server, like this:

   ```
   $ lbxproxy -display LOCAL:0 :1 &
   ```

   This tells `lbxproxy` to use display `:1` on the REMOTE system; if that system has >1 display already you can use `:2` or whatever instead.
2. Set your DISPLAY environment variable to point to the display that `lbxproxy` is providing, instead of the normal display:

   ```
   $ DISPLAY=:1
   $ export DISPLAY
   ```

   Or, if you use csh or clones:

   ```
   % setenv DISPLAY :1
   ```
3. If you're using `xauth` you will need to ensure that your cookie is available locally. See the Remote X Apps Mini-HOWTO for more information on this.
4. Start your X applications!

That's it; all X apps that are started up pointing to `:1` will use LBX. Of course, there's no reason you couldn't also start X apps pointing to `LOCAL:0` and have both running at the same time.

9. Problems

Here are some common problems:

Q)

`lbxproxy` exits with an "access denied" error.

A)

This means the LOCAL system isn't accepting connections from the REMOTE system due to permissions errors. See the Remote X Apps Mini-HOWTO for details on these issues.

As a simple trouble-shooting measure, try running a simple X app like `xclock` on REMOTE and have it display on the local system without using `lbxproxy`:

```
$ xclock -display LOCAL:0
```

If that doesn't work, it's `xhost` or some other basic X problem, not LBX.

10. Documentation

The only documentation available in a standard X distribution may be the *lbxproxy(1)* man page.

If you have access to the X source tree, then very interesting information on LBX is available there:

- `xc/doc/specs/Xext/lbx.mif` (Framemaker MIF)
- `xc/doc/hardcopy/Xext/lbx.PS.Z` (Compressed Postscript)
- `xc/doc/hardcopy/Xext/lbxTOC.html` (HTML)

More detailed discussion of specific LBX algorithms is available here:

- `xc/doc/specs/Xext/lbxalg.mif` (Framemaker MIF)
- `xc/doc/specs/Xext/lbxalg.PS.Z` (Compressed Postscript)

If you don't have access to the X11 source, you can obtain these files from the X Consortium's FTP site.

11. Alternatives

If you don't like `lbxproxy` for some reason: you're not satisfied with the performance, it doesn't work for you, you don't want to hassle with creating an lbxproxy for the remote host, or you simply are interested in trying other options, there is at least one other package for X protocol compression (anyone have others?)

11.1 dxpc - The Differential X Protocol Compressor

- Original Author: Brian Pane <brianp@cnet.com>
- Current Maintainer: Zachary Vonler <lightborn@mail.utexas.edu>

`dxpc` works in essentially the same way as LBX. However, to avoid having to implement an X extension and modify the X server code, `dxpc` uses **two** proxies: one that runs on the REMOTE host, like `lbxproxy`, and one that runs on the LOCAL host.

The REMOTE host proxy communicates between the X clients and the LOCAL host proxy, and the LOCAL host proxy communicates between the X server and the REMOTE host proxy.

So, to *both* the X clients and the X server, it looks like X protocol as usual.

Advantages

- Since it's a completely separate application that does not require any X internals, it's *much* simpler to compile and install.
- It's maintained separately, so you don't have to wait for the OSF to release new X versions for enhancements or fixes.
- It provides more and better compression information and statistics than `lbxproxy`.

Disadvantages

- It is not a standard part of X; you must obtain and build it separately.
- It is slightly more complex to set up, since it requires a LOCAL-side proxy as well as the REMOTE proxy.

Where Can I Get dxpc?

The source for dxpc is available at ftp.x.org.

There is a WWW homepage for dxpc that gives a lot of good information, including pointers to the dxpc mailing list, access to the source code, and a number of pre-built binaries for various platforms:

http://ccwf.cc.utexas.edu/~zvonler/dxpc/

11.2 Ssh (Secure Shell)

Ken Chase <lbxhowto@sizone.org> notes that `ssh` can be used for compression. Although its main purpose is to provide security, it also compresses the data it sends.

Thus, if you run X over a `ssh` link you will automatically obtain some amount of compression.

11.3 Which Is Better?

I don't know. Both LBX and `dxpc` are certainly better at raw compression than `ssh`. Of course, `ssh` provides the added advantage of security. And of course, there's no reason you can't use both `ssh` and one of the other two, to get good compression and security.

It shouldn't be hard to run some benchmarking against these options and get both subjective and statistical measurings of performance. But I haven't done this, and I don't know of anyone who has.

Lilo mini-Howto

Cameron Spitzer (`cls@truffula.sj.ca.us`), Alessandro Rubini (`rubini@linux.it`).

v2.02, 16 August 1998

*LILO is the most used **Li**nux **Lo**ader for the x86 flavour of Linux; I'll call it Lilo rather than LILO here because I don't appreciate uppercase. This file describes some typical Lilo installations. It's intended as a supplement to the Lilo User's Guide. I think examples are informative even if your setup isn't much like mine. I hope this saves you trouble. Since Lilo's own documentation is very good, who's interested in the details is referred to /usr/doc/lilo**

1. Introduction

2. Background Information and Standard Installation

- 2.1 Where Should I Install Lilo?
- 2.2 How Should I Configure my IDE Hard Drives?
- 2.3 How Can I Interact at Boot Time?
- 2.4 How Can I Uninstall Lilo?

3. The Simple Configuration

- 3.1 How to Deal with Big Kernels
- 3.2 Other Sources of Information

4. Installing hdc to Boot as hda and Using `bios=`

5. Using Lilo When the BIOS Can't See the Root Partition

6. Accessing Huge Disks When the BIOS Can't

7. Booting from a Rescue Floppy

---

1. Introduction

Although the documentation found in Lilo's sources (the one installed in `/usr/doc/lilo`-version) is very comprehensive, most Linux users experience some trouble in building their own `/etc/lilo.conf` file. This document is meant to support them by giving the minimal information and by showing five sample installations:

- The first example is the classical ``Linux and other'' installation.
- The next one shows how to install Lilo on a hard drive connected as `/dev/hdc` that will boot as `/dev/hda`. This is usually needed when you install a new Linux drive from your own running system. This also tells how to boot from SCSI disks when your BIOS is modern enough.
- The third example shows how to boot a Linux system whose root partition can't be accessed by the BIOS.
- The next sample file is used to access huge disks, that neither the BIOS nor DOS can access easily (this one is somehow outdated).
- The last example shows how to restore a damaged disk, if the damage resulted from installing another operating system).

The last three examples are by Cameron, `cls@truffula.sj.ca.us`, who wrote the original document. Alessandro (tt/rubini@linux.it/, the current maintainer doesn't run anything but Linux, so he can't check nor update them by himself. Needless to say, any feedback is welcome.

2. Background Information and Standard Installation

When Lilo boots the system, it uses BIOS calls to load the Linux kernel off the disk (IDE drive, floppy or whatever). Therefore, the kernel must live in some place that can be accessed by the bios.

At boot time, Lilo is not able to read filesystem data, and any pathname you put in `/etc/lilo.conf` is resolved at installation time (when you invoke */sbin/lilo*). Installation time is when the program builds the tables that list which sectors are used by the files used to load the operating system. As a consequence, all of these files must live in a partition that can be accessed by the BIOS (the files are usually located in the `/boot` directory, this means that only the root partition of your Linux system needs to be accessed via the BIOS).

Another consequence of being BIOS-based is that you must reinstall the loader (i.e., you must reinvoke */sbin/lilo*) any time you modify the Lilo setup. Whenever you recompile your kernel and overwrite your old image you must reinstall Lilo.

2.1 Where Should I Install Lilo?

The `boot=` directive in `/etc/lilo.conf` tells Lilo where it should place its primary boot loader. In general, you can either specify the master boot record (`/dev/hda`) or the root partition of your Linux installation (is usually is `/dev/hda1` or `/dev/hda2`).

If you have another operating system installed in your hard drive, you'd better install Lilo to the root partition instead of the MBR. In this case, you must mark the partition as ``bootable'' using the ``a'' command of *fdisk* or the ``b'' command of *cfdisk*. If you don't overwrite the master boot sector you'll find it easier to uninstall Linux and Lilo if needed.

2.2 How Should I Configure my IDE Hard Drives?

I personally don't use LBA or LARGE settings in the BIOS (but I only run Linux); they are horrible kludges forced on by design deficiencies in the PC world. This requires that the kernel lives in the first 1024 cylinders, but this is not a problem as long as you partition your hard drives and keep root small (as you should do anyways).

If your hard disk already carries another operating system, you won't be able to modify the BIOS settings, or the old system won't work any more. All recent Lilo distribution are able to deal with LBA and LARGE disk settings.

Note that the "`linear`" keyword in `/etc/lilo.conf` can help in dealing with geometry problems. The keyword instructs Lilo to use linear sector addresses instead of sector/head/cylinder tuples. Conversion to 3D addresses is delayed to run-time, therefore making the setup more immune to geometry

problems.

If you have more than one hard disk and some of them are only used by Linux and are not involved in the boot process, you can tell your BIOS that they are not installed. Your system will boot more quickly and Linux will autodetect all the disks in no time. I often switch disks in my computers, but I never touch the BIOS configuration.

2.3 How Can I Interact at Boot Time?

When you see the Lilo prompt, you can hit the <Tab> key to show the list of possible choices. If Lilo is not configured to be interactive, press and hold the <Alt> or <Shift> key before the ``LILO'' message appears.

If you choose to boot a Linux kernel, you can add command-line arguments after the name of the system you choose. The kernel accepts many command-line arguments. All the arguments are listed in the ``BootPrompt-HOWTO'' by Paul Gortmaker, and I won't replicate it here. A few command line arguments, however, are particularly important and worth describing here:

- `root=`: you can tell the Linux kernel to mount as root a different partition than the one appearing in `lilo.conf`. For example, my system has a tiny partition hosting a minimal Linux installation, and I've been able to boot the system after destroying my root partition by mistake.
- `init=`: verson 1.3.43 and newer of the Linux kernel can execute another command instead of `/sbin/init`, as specified on the command line. If you experience bad problems during the boot process, you can access the bare system by specifying `init=/bin/sh` (when you are at the shell prompt you most likely will need to mount your disks: try ``mount -w -n -o remount /; mount -a'', and remember to ``umount -a'' before turning off the computer).
- A number: by specifying a number on the kernel command line, you instruct *init* to enter a specific run-level (the default is usually 3 or 2, according to the distribution you chose). Refer to the *init* documentation, to `/etc/inittab` and to `/etc/rc*.d` to probe further.

2.4 How Can I Uninstall Lilo?

When Lilo overwrites a boot sector, it saves a backup copy in `/boot/boot.`*xxyy*, where *xxyy* are the major and minor numbers of the device, in hex. You can see the major and minor numbers of your disk or partition by running ``ls -l /dev/*device*''. For example, the first sector of `/dev/hda` (major 3, minor 0) will be saved in `/boot/boot.0300`, installing Lilo on `/dev/fd0` creates `/boot/boot.0200` and installing on `/dev/sdb3` (major 8, minor 19) creates `/boot/boot.0813`. Note that Lilo won't create the file if there is already one so you don't need to care about the backup copy whenever you reinstall Lilo (for example, after recompiling your kernel). The backup copies found in `/boot/` are always the snapshot of the situation before installing any Lilo.

If you ever need to uninstall Lilo (for example, in the unfortunate case you need to uninstall Linux), you just need to restore the original boot sector. If Lilo is installed in `/dev/hda`, just do ``dd if=/boot/boot.0300 of=/dev/hda bs=446 count=1'' (I personally just do ``cat /boot/boot.0300 > /dev/hda'', but this is not safe, as this will restore the original partition table as well, which you might have modified in the meanwhile). This command is much easier to run than

TOC

trying ```fdisk /mbr`'' from a DOS shell: it allows you to cleanly remove Linux from a disk without ever booting anything but Linux. After removing Lilo remember to run Linux' *fdisk* to destroy any Linux partition (DOS' *fdisk* is unable to remove non-dos partitions).

If you installed Lilo on your root partition (e.g., `/dev/hda2`), nothing special needs to be done to uninstall Lilo. Just run Linux' *fdisk* to remove Linux partitions from the partition table. You must also mark the DOS partition as bootable.

---

3. The Simple Configuration

Most Lilo installations use a configuration file like the following one:

```
boot = /dev/hda   # or your root partition
delay = 10        # delay, in tenth of a second (so you can interact)
vga = 0           # optional. Use "vga=1" to get 80x50
#linear           # try "linear" in case of geometry problems.

image = /boot/vmlinux  # your zImage file
  root = /dev/hda1     # your root partition
  label = Linux        # or any fancy name
  read-only            # mount root read-only

other = /dev/hda4   # your dos partition, if any
  table = /dev/hda  # the current partition table
  label = dos       # or any non-fancy name
```

You can have multiple ``image'' and ``other'' sections if you want. It's not uncommon to have several kernel images configured in your *lilo.conf*, at least if you keep up to date with kernel development.

3.1 How to Deal with Big Kernels

If you compile a ``zImage'' kernel and it is too big to fit in half a megabyte (this is commong with new 2.1 kernels), you should build a ``big zImage'' instead: ```make bzImage`''. To boot a big kernel image nothing special is needed, but you need version 18 or newer of Lilo. If your installation is older, you should upgrade your Lilo package.

3.2 Other Sources of Information

In addition to the Lilo docs, there are a number of mini-howto's that can be useful for your needs. All of them are called ``Linux+*foobarOS*'', for some *foobarOS*, they deal with coexistence of Linux and other operationg system(s). Also, ``Multiboot-with-LILO'' describes how the various Windows flavours can be made to coexist with Linux.

4. Installing hdc to Boot as hda and Using `bios=`

Lilo allows to map the kernel image from one disk and instruct the BIOS to retrieve it from another disk. For example, it's common for me to install Linux on a disk I connect to `hdc` (master disk of secondary controller) and boot it as a standalong system on the primary IDE controller of another computer. I copied the installation floppy to a tiny partition, so I can run *chroot* in a virtual console to install `hdc` while I use the system to do something else.

The *lilo.conf* file I use to install Lilo looks like:

```
# This file must be used from a system running off /dev/hdc
boot = /dev/hdc   # overwrite MBR of hdc
disk = /dev/hdc   # tell how hdc will look like:
   bios = 0x80    #  the bios will see it as first drive
delay = 0
vga = 0

image = /boot/vmlinux  # this is on /dev/hdc1
  root = /dev/hda1     # but at boot it will be hda1
  label = Linux
  read-only
```

This configuration file must be read by a Lilo running **off /dev/hdc1**. The Lilo maps that get written the boot sector (`/dev/hdc`) must refer to the files in `/boot` (currently installed as hdc); such files will be accessed under hda when this disk will be booted as a standalone system.

I call this configuration file `/mnt/etc/lilo.conf.hdc` (`/mnt` is where hdc is mounted during the installation. I install Lilo by invoking ``cd /mnt; chroot . sbin/lilo -C /etc/lilo.conf.hdc''. Refer to the manual page for *chroot* if this looks magic.

The ``bios=" directive in `lilo.conf` is used to tell Lilo what the BIOS thinks of your devices. BIOS calls identify floppy disks and hard drives with a number: 0x00 and 0x01 select the floppy drives, 0x80 and the following numbers select hard disks (old BIOSes can only access two disks). The meaning of ``bios = 0x80` in the previous sample file is therefore ``use 0x80 in your BIOS calls for `/dev/hdc`''.

This Lilo directive can be handy in other situations, for example when your BIOS is able to boot from SCSI disks instead of IDE ones. When both IDE and SCSI devices are there, Lilo can't tell whether 0x80 will refer to one or the other because the user is able to choose it in the BIOS configuration menus, and the BIOS can't be accessed while Linux is running.

By default, Lilo assumes that IDE drives are mapped first by the BIOS, but this can be overridden by using instructions like these in `/etc/lilo.conf`:

```
disk = /dev/sda
  bios = 0x80
```

5. Using Lilo When the BIOS Can't See the Root Partition

I have two IDE drives, and a SCSI drive. The SCSI drive can't be seen from BIOS. The Linux Loader, Lilo, uses BIOS calls and can only see drives that BIOS can see. My stupid AMI BIOS will only boot from "A:" or "C:" My root file system is on a partition on the SCSI drive.

The solution consists in storing the kernel, map file, and chain loader in a Linux partition on the first IDE. Notice that it is not necessary to keep your kernel on your root partition.

The second partition on my first IDE (`/dev/hda2`, the Linux partition used to boot the system) is mounted on `/u2`. Here is the `/etc/lilo.conf` file I used.

```
#  Install Lilo on the Master Boot Record
#  on the first IDE.
#
boot = /dev/hda
#  /sbin/lilo (the installer) copies the Lilo boot record
#  from the following file to the MBR location.
install = /u2/etc/lilo/boot.b
#
#  I wrote a verbose boot menu.  Lilo finds it here.
message = /u2/etc/lilo/message
#  The installer will build the following file. It tells
#  the boot-loader where the blocks of the kernels are.
map = /u2/etc/lilo/map
compact
prompt
#  Wait 10 seconds, then boot the 1.2.1 kernel by default.
timeout = 100
#  The kernel is stored where BIOS can see it by doing this:
#      cp -p /usr/src/linux/arch/i386/boot/zImage /u2/z1.2.1
image = /u2/z1.2.1
        label = 1.2.1
#  Lilo tells the kernel to mount the first SCSI partition
#  as root.  BIOS does not have to be able to see it.
        root = /dev/sda1
#  This partition will be checked and remounted by /etc/rc.d/rc.S
        read-only
#  I kept an old Slackware kernel lying around in case I built a
#  kernel that doesn't work.  I actually needed this once.
```

```
image = /u2/z1.0.9
        label = 1.0.9
        root = /dev/sda1
        read-only
#  My DR-DOS 6 partition.
other = /dev/hda1
        loader=/u2/etc/lilo/chain.b
        label = dos
        alias = m
```

---

6. Accessing Huge Disks When the BIOS Can't

The system in my office has a 1GB IDE drive. The BIOS can only see the first 504 MB of the IDE. (Where MB means 2**10 bytes, not 10**6 bytes.) So I have MS-DOS on a 350 MB partition `/dev/hda1` and my Linux root on a 120 MB partition `/dev/hda2`.

MS-DOS was unable to install itself correctly when the drive was fresh. Novell DOS 7 had the same problem. Luckily for me, "Options by IBM" forgot to put the "OnTrack" diskette in the box with the drive. The drive was supposed to come with a product called "OnTrack Disk Manager." If you only have MSDOS, I guess you have to use it.

So I made a partition table with Linux' fdisk. MSDOS-6.2 refused to install itself in `/dev/hda1`. It said something like ``this release of MS-DOS is for new installations. Your computer already has MS-DOS so you need to get an upgrade release from your dealer." Actually, the disk was brand new.

What a crock! So I ran Linux' fdisk again and deleted partition 1 from the table. This satisfied MS-DOS 6.2 which proceeded to create the exact same partition 1 I had just deleted and installed itself. MS-DOS 6.2 wrote its Master Boot Record on the drive, but it couldn't boot.

Luckily I had a Slackware kernel on floppy (made by the Slackware installation program "setup"), so I booted Linux and wrote Lilo over MS-DOS' broken MBR. This works. Here is the `/etc/lilo.conf` file I used:

```
boot = /dev/hda
map = /lilo-map
delay = 100
ramdisk = 0             # Turns off ramdisk in Slackware kernel
timeout = 100
prompt
disk = /dev/hda         # BIOS only sees first 500 MB.
   bios = 0x80          # specifies the first IDE.
   sectors = 63         # get the numbers from your drive's docs.
   heads = 16
   cylinders = 2100
image = /vmlinuz
  append = "hd=2100,16,63"
  root = /dev/hda2
  label = linux
  read-only
  vga = extended
other = /dev/hda1
  label = msdos
  table = /dev/hda
```

```
loader = /boot/chain.b
```

After I installed these systems, I verified that the partition containing the zImage, boot.b, map, chain.b and message files can use an msdos file system, as long as it is not "stackered" or "doublespaced." So I could have made the DOS partition on `/dev/hda1` 500 MB.

I have also learned that "OnTrack" would have written a partition table starting a few dozen bytes into the drive, instead of at the beginning, and it is possible to hack the Linux IDE driver to work around this problem. But installing would have been impossible with the precompiled Slackware kernel. Eventually, IBM sent me an "OnTrack" diskette. I called OnTrack's technical support. They told me Linux is broken because Linux doesn't use BIOS. I gave their diskette away.

---

7. Booting from a Rescue Floppy

Next, I installed Windows-95 on my office system. It blew away my nice Lilo MBR, but it left my Linux partitions alone. Kernels take a long time to load from floppy, so I made a floppy with a working Lilo setup on it, which could boot my kernel from the IDE.

I made the lilo floppy like so:

```
  fdformat /dev/fd0H1440      #  lay tracks on virgin diskette
  mkfs -t minix /dev/fd0 1440 #  make file system of type minix
  mount /dev/fd0 /mnt         #  mount in the standard tmp mount point
  cp -p /boot/chain.b /mnt    #  copy the chain loader over
  lilo -C /etc/lilo.flop      #  install Lilo and the map on the diskette.
  umount /mnt
```

Notice that the diskette **must be mounted when you run the installer** so that Lilo can write its map file properly.

This file is /etc/lilo.flop. It's almost the same as the last one:

```
#  Makes a floppy that can boot kernels from HD.
boot = /dev/fd0
map = /mnt/lilo-map
delay = 100
ramdisk = 0
timeout = 100
prompt
disk = /dev/hda     # 1 GB IDE, BIOS only sees first 500 MB.
   bios=0x80
   sectors = 63
   heads = 16
   cylinders = 2100
image = /vmlinuz
  append = "hd=2100,16,63"
  root = /dev/hda2
  label = linux
  read-only
  vga = extended
other = /dev/hda1
  label = msdos
  table = /dev/hda
  loader = /mnt/chain.b
```

Finally, I needed MS-DOS 6.2 on my office system, but I didn't want to touch the first drive. I added a SCSI controller and drive, made an msdos file system on it with Linux' mkdosfs, and Windows-95 sees it as "D:".

TOC

But of course MSDOS will not boot off of D:. This is not a problem when you have Lilo. I added the following to the `lilo.conf` in Example 2.

```
other = /dev/sda1
  label = d6.2
  table = /dev/sda
  loader = /boot/any_d.b
```

With this modification MSDOS-6.2 runs, and it thinks it is on C: and Windows-95 is on D:.

Large Disk HOWTO

Andries Brouwer, aeb@cwi.nl

v2.2d, 12 September 1999

All about disk geometry and the 1024 cylinder limit for disks.

1. The problem

2. Summary

3. Units and Sizes

- 3.1 Sectorsize
- 3.2 Disksize

4. Disk Access

- 4.1 BIOS Disk Access and the 1024 cylinder limit
- 4.2 History of BIOS and IDE limits

5. Booting

- 5.1 LILO and the `linear' option
- 5.2 1024 cylinders is not 1024 cylinders

6. Disk geometry, partitions and `overlap'

TOC

7. Translation and Disk Managers

8. Kernel disk translation for IDE disks

- 8.1 EZD
- 8.2 DM6:DDO
- 8.3 DM6:AUX
- 8.4 DM6:MBR
- 8.5 PTBL

9. Consequences

10. Details

- 10.1 IDE details - the seven geometries
- 10.2 SCSI details

11. The Linux IDE 8 GiB limit

- 11.1 BIOS complications
- 11.2 Jumpers that select the number of heads
- 11.3 Jumpers that clip total capacity

12. The Linux 65535 cylinder limit

- 12.1 IDE problems with 34+ GB disks

13. Extended and logical partitions

14. Problem solving

- 14.1 Problem: Linux invents the wrong geometry for my disk.
- 14.2 Nonproblem: Identical disks have different geometry?
- 14.3 Nonproblem: fdisk sees much more room than df?

---

1. The problem

Suppose you have a disk with more than 1024 cylinders. Suppose moreover that you have an operating system that uses the old INT13 BIOS interface to disk I/O. Then you have a problem, because this interface uses a 10-bit field for the cylinder on which the I/O is done, so that cylinders 1024 and past are inaccessible.

Fortunately, Linux does not use the BIOS, so there is no problem.

Well, except for two things:

(1) When you boot your system, Linux isn't running yet and cannot save you from BIOS problems. This has some consequences for LILO and similar boot loaders.

(2) It is necessary for all operating systems that use one disk to agree on where the partitions are. In other words, if you use both Linux and, say, DOS on one disk, then both must interpret the partition table in the same way. This has some consequences for the Linux kernel and for `fdisk`.

Below a rather detailed description of all relevant details. Note that I used kernel version 2.0.8 source as a reference. Other versions may differ a bit.

2. Summary

You got a new large disk. What to do? Well, on the software side: use `fdisk` (or, better, `cfdisk`) to create partitions, and then `mke2fs` to create a filesystem, and then `mount` to attach the new filesystem to the big file hierarchy.

You need not read this HOWTO since there are *no* problems with large hard disks these days. The great majority of apparent problems is caused by people who think there might be a problem and install a disk manager, or go into `fdisk` expert mode, or specify explicit disk geometries to LILO or on the kernel command line.

However, typical problem areas are: (i) ancient hardware, (ii) several operating systems on the same disk, and sometimes (iii) booting.

Advice:

For large SCSI disks: Linux has supported them from very early on. No action required.

For large IDE disks: get a recent stable kernel (2.0.34 or later). Usually, all will be fine now, especially if you were wise enough not to ask the BIOS for disk translations like LBA and the like.

If LILO hangs at boot time, also specify `linear` in the configuration file `/etc/lilo.conf`.

There may be geometry problems that can be solved by giving an explicit geometry to kernel/LILO/fdisk.

If you have an old `fdisk` and it warns about overlapping partitions: ignore the warnings, or check using `cfdisk` that really all is well.

If you think something is wrong with the size of your disk, make sure that you are not confusing binary and decimal units , and realize that the free space that `df` reports on an empty disk is a few percent smaller than the partition size, because there is administrative overhead.

Now, if you still think there are problems, or just are curious, read on.

3. Units and Sizes

A kilobyte (kB) is 1000 bytes. A megabyte (MB) is 1000 kB. A gigabyte (GB) is 1000 MB. A terabyte (TB) is 1000 GB. This is the SI norm. However, there are people that use 1 MB=1024000 bytes and talk about 1.44 MB floppies, and people who think that 1 MB=1048576 bytes. Here I follow the proposed standard and write Ki, Mi, Gi, Ti for the binary units, so that these floppies are 1440 KiB (1.47 MB, 1.41 MiB), 1 MiB is 1048576 bytes (1.05 MB), 1 GiB is 1073741824 bytes (1.07 GB) and 1 TiB is 1099511627776 bytes (1.1 TB).

Quite correctly, the disk drive manufacturers follow the SI norm and use the decimal units. However, Linux boot messages and some fdisk-type programs use the symbols MB and GB for binary, or mixed binary-decimal units. So, before you think your disk is smaller than was promised when you bought it, compute first the actual size in decimal units (or just in bytes).

3.1 Sectorsize

In the present text a sector has 512 bytes. This is almost always true, but for example certain MO disks use a sectorsize of 2048 bytes, and all capacities given below must be multiplied by four. (When using `fdisk` on such disks, make sure you have version 2.9i or later, and give the `-b 2048' option.)

3.2 Disksize

A disk with C cylinders, H heads and S sectors per track has C*H*S sectors in all, and can store C*H*S*512 bytes. For example, if the disk label says C/H/S=4092/16/63 then the disk has 4092*16*63=4124736 sectors, and can hold 4124736*512=2111864832 bytes (2.11 GB). There is an industry convention to give C/H/S=16383/16/63 for disks larger than 8.4 GB, and the disk size can no longer be read off from the C/H/S values reported by the disk.

4. Disk Access

In order to read or write something from or to the disk, we have to specify a position on the disk, for example by giving a sector or block number. If the disk is a SCSI disk, then this sector number goes directly into the SCSI command and is understood by the disk. If the disk is an IDE disk using LBA, then precisely the same holds. But if the disk is old, RLL or MFM or IDE from before the LBA times, then the disk hardware expects a triple (cylinder,head,sector) to designate the desired spot on the disk.

The correspondence between the linear numbering and this 3D notation is as follows: for a disk with C cylinders, H heads and S sectors/track position (c,h,s) in 3D or CHS notation is the same as position c*H*S + h*S + (s-1) in linear or LBA notation. (The minus one is because traditionally sectors are counted from 1, not 0, in this 3D notation.)

Consequently, in order to access a very old non-SCSI disk, we need to know its *geometry*, that is, the values of C, H and S.

4.1 BIOS Disk Access and the 1024 cylinder limit

Linux does not use the BIOS, but some other systems do. The BIOS, which predates LBA times, offers with INT13 disk I/O routines that have (c,h,s) as input. (More precisely: `AH` selects the function to perform, `CH` is the low 8 bits of the cylinder number, `CL` has in bits 7-6 the high two bits of the cylinder number and in bits 5-0 the sector number, `DH` is the head number, and `DL` is the drive number (80h or 81h). This explains part of the layout of the partition table.)

Thus, we have CHS encoded in three bytes, with 10 bits for the cylinder number, 8 bits for the head number, and 6 bits for the track sector number (numbered 1-63). It follows that cylinder numbers can range from 0 to 1023 and that no more than 1024 cylinders are BIOS addressable.

DOS and Windows software did not change when IDE disks with LBA support were introduced, so DOS and Windows continued needing a disk geometry, even when this was no longer needed for the actual disk I/O, but only for talking to the BIOS. This again means that Linux needs the geometry in those places where communication with the BIOS or with other operating systems is required, even on a modern disk.

This state of affairs lasted for four years or so, and then disks appeared on the market that could not be addressed with the INT13 functions (because the 10+8+6=24 bits for (c,h,s) can address not more than 8.5 GB) and a new BIOS interface was designed: the so-called Extended INT13 functions, where DS:SI points at a 16-byte Disk Address Packet that contains an 8-byte starting absolute block number.

Very slowly the Microsoft world is moving towards using these Extended INT13 functions. Probably a few years from now no modern system on modern hardware will need the concept of `disk geometry' anymore.

4.2 History of BIOS and IDE limits

TOC

ATA Specification (for IDE disks) - the 137 GB limit

At most 65536 cylinders (numbered 0-65535), 16 heads (numbered 0-15), 255 sectors/track (numbered 1-255), for a maximum total capacity of 267386880 sectors (of 512 bytes each), that is, 136902082560 bytes (137 GB). This is not yet a problem (in 1999), but will be a few years from now.

BIOS Int 13 - the 8.5 GB limit

At most 1024 cylinders (numbered 0-1023), 256 heads (numbered 0-255), 63 sectors/track (numbered 1-63) for a maximum total capacity of 8455716864 bytes (8.5 GB). This is a serious limitation today. It means that DOS cannot use present day large disks.

The 528 MB limit

If the same values for c,h,s are used for the BIOS Int 13 call and for the IDE disk I/O, then both limitations combine, and one can use at most 1024 cylinders, 16 heads, 63 sectors/track, for a maximum total capacity of 528482304 bytes (528MB), the infamous 504 MiB limit for DOS with an old BIOS. This started being a problem around 1993, and people resorted to all kinds of trickery, both in hardware (LBA), in firmware (translating BIOS), and in software (disk managers). The concept of `translation' was invented (1994): a BIOS could use one geometry while talking to the drive, and another, fake, geometry while talking to DOS, and translate between the two.

The 2.1 GB limit (April 1996)

Some older BIOSes only allocate 12 bits for the field in CMOS RAM that gives the number of cylinders. Consequently, this number can be at most 4095, and only 4095*16*63*512=2113413120 bytes are accessible. The effect of having a larger disk would be a hang at boot time. This made disks with geometry 4092/16/63 rather popular. And still today many large disk drives come with a jumper to make them appear 4092/16/63. See also over2gb.htm.

The 3.2 GB limit

There was a bug in the Phoenix 4.03 and 4.04 BIOS firmware that would cause the system to lock up in the CMOS setup for drives with a capacity over 3277 MB. See over3gb.htm.

The 4.2 GB limit (Feb 1997)

Simple BIOS translation (ECHS=Extended CHS, sometimes called `Large disk support' or just `Large') works by repeatedly doubling the number of heads and halving the number of cylinders shown to DOS, until the number of cylinders is at most 1024. Now DOS and Windows 95 cannot handle 256 heads, and in the common case that the disk reports 16 heads, this means that this simple mechanism only works up to 8192*16*63*512=4227858432 bytes (with a fake geometry with 1024 cylinders, 128 heads, 63 sectors/track). Note that ECHS does not change the number of sectors per track, so if that is not 63, the limit will be lower. See over4gb.htm.

The 7.9 GB limit

Slightly smarter BIOSes avoid the previous problem by first adjusting the number of heads to 15

(`revised ECHS'), so that a fake geometry with 240 heads can be obtained, good for 1024*240*63*512=7927234560 bytes.

The 8.4 GB limit

Finally, if the BIOS does all it can to make this translation a success, and uses 255 heads and 63 sectors/track (`assisted LBA' or just `LBA') it may reach 1024*255*63*512=8422686720 bytes, slightly less than the earlier 8.5 GB limit because the geometries with 256 heads must be avoided. (This translation will use for the number of heads the first value H in the sequence 16, 32, 64, 128, 255 for which the total disk capacity fits in 1024*H*63*512, and then computes the number of cylinders C as total capacity divided by (H*63*512).)

The 33.8 GB limit (August 1999)

The next hurdle comes with a size over 33.8 GB. The problem is that with the default 16 heads and 63 sectors/track this corresponds to a number of cylinders of more than 65535, which does not fit into a short. Maybe no BIOS in existence today can handle such disks. Linux still needs a patch.

For another discussion of this topic, see Breaking the Barriers, and, with more details, IDE Hard Drive Capacity Barriers.

Hard drives over 8.4 GB are supposed to report their geometry as 16383/16/63. This in effect means that the `geometry' is obsolete, and the total disk size can no longer be computed from the geometry.

---

5. Booting

When the system is booted, the BIOS reads sector 0 (known as the MBR - the Master Boot Record) from the first disk (or from floppy or CDROM), and jumps to the code found there - usually some bootstrap loader. These small bootstrap programs found there typically have no own disk drivers and use BIOS services. This means that a Linux kernel can only be booted when it is entirely located within the first 1024 cylinders.

This problem is very easily solved: make sure that the kernel (and perhaps other files used during bootup, such as LILO map files) are located on a partition that is entirely contained in the first 1024 cylinders of a disk that the BIOS can access - probably this means the first or second disk.

Thus: create a small partition, say 10 MB large, so that there is room for a handful of kernels, making sure that it is entirely contained within the first 1024 cylinders of the first or second disk. Mount it on `/boot` so that LILO will put its stuff there.

5.1 LILO and the `linear' option

Another point is that the boot loader and the BIOS must agree as to the disk geometry. LILO asks the kernel for the geometry, but more and more authors of disk drivers follow the bad habit of deriving a geometry from the partition table, instead of telling LILO what the BIOS will use. Thus, often the geometry supplied by the kernel is worthless. In such cases it helps to give LILO the `` `linear' `` option. The effect of this is that LILO does not need geometry information at boot loader install time (it stores linear addresses in the maps) but does the conversion of linear addresses at boot time. Why is this not the default? Well, there is one disadvantage: with the `linear' option, LILO no longer knows about cylinder numbers, and hence cannot warn you when part of the kernel was stored above the 1024 cylinder limit, and you may end up with a system that does not boot.

5.2 1024 cylinders is not 1024 cylinders

Tim Williams writes: `I had my Linux partition within the first 1024 cylinders and still it wouldnt boot. First when I moved it below 1 GB did things work.' How can that be? Well, this was a SCSI disk with AHA2940UW controller which uses either H=64, S=32 (that is, cylinders of 1 MiB = 1.05 MB), or H=255, S=63 (that is, cylinders of 8.2 MB), depending on setup options in firmware and BIOS. No doubt the BIOS assumed the former, so that the 1024 cylinder limit was found at 1 GiB, while Linux used the latter and LILO thought that this limit was at 8.4 GB.

6. Disk geometry, partitions and `overlap'

If you have several operating systems on your disks, then each uses one or more disk partitions. A disagreement on where these partitions are may have catastrophic consequences.

The MBR contains a *partition table* describing where the (primary) partitions are. There are 4 table entries, for 4 primary partitions, and each looks like

```
struct partition {
        char active;    /* 0x80: bootable, 0: not bootable */
        char begin[3];  /* CHS for first sector */
        char type;
        char end[3];    /* CHS for last sector */
        int start;      /* 32 bit sector number (counting from 0) */
        int length;     /* 32 bit number of sectors */
};
```

(where CHS stands for Cylinder/Head/Sector).

This information is redundant: the location of a partition is given both by the 24-bit `begin` and `end` fields, and by the 32-bit `start` and `length` fields.

Linux only uses the `start` and `length` fields, and can therefore handle partitions of not more than 2^32 sectors, that is, partitions of at most 2 TiB. That is a hundred times larger than the disks available today, so maybe it will be enough for the next eight years or so. (So, partitions can be very large, but there is a serious restriction in that a file in an ext2 filesystem on hardware with 32-bit integers cannot be larger than 2 GiB.)

DOS uses the `begin` and `end` fields, and uses the BIOS INT13 call to access the disk, and can therefore only handle disks of not more than 8.4 GB, even with a translating BIOS. (Partitions cannot be larger than 2.1 GB because of restrictions of the FAT16 file system.) The same holds for Windows 3.11 and WfWG and Windows NT 3.* and Novell NetWare.

Windows 95 has support for the Extended INT13 interface, and uses special partition types (c, e, f instead of b, 6, 5) to indicate that a partition should be accessed in this way. When these partition types are used, the `begin` and `end` fields contain dummy information (1023/255/63). Windows 95 OSR2 introduces the FAT32 file system (partition type b or c), that allows partitions of size at most 2 TiB.

What is this nonsense you get from `fdisk` about `overlapping' partitions, when in fact nothing is wrong? Well - there is something `wrong': if you look at the `begin` and `end` fields of such partitions, as DOS does, they overlap. (And that cannot be corrected, because these fields cannot store cylinder numbers above 1024 - there will always be `overlap' as soon as you have more than 1024 cylinders.) However, if you look at the `start` and `length` fields, as Linux does, and as Windows 95 does in the case of partitions with partition type c, e or f, then all is well. So, ignore these warnings when `cfdisk` is satisfied and you have a Linux-only disk. Be careful when the disk is shared with DOS. Use the commands `cfdisk -Ps /dev/hdx` and `cfdisk -Pt /dev/hdx` to look at the partition table of `/dev/hdx`.

7. Translation and Disk Managers

Disk geometry (with heads, cylinders and tracks) is something from the age of MFM and RLL. In those days it corresponded to a physical reality. Nowadays, with IDE or SCSI, nobody is interested in what the `real' geometry of a disk is. Indeed, the number of sectors per track is variable - there are more sectors per track close to the outer rim of the disk - so there is no `real' number of sectors per track. Quite the contrary: the IDE command INITIALIZE DRIVE PARAMETERS (91h) serves to tell the disk how many heads and sectors per track it is supposed to have today. It is quite normal to see a large modern disk that has 2 heads report 15 or 16 heads to the BIOS, while the BIOS may again report 255 heads to user software.

For the user it is best to regard a disk as just a linear array of sectors numbered 0, 1, ..., and leave it to the firmware to find out where a given sector lives on the disk. This linear numbering is called LBA.

So now the conceptual picture is the following. DOS, or some boot loader, talks to the BIOS, using (c,h,s) notation. The BIOS converts (c,h,s) to LBA notation using the fake geometry that the user is using. If the disk accepts LBA then this value is used for disk I/O. Otherwise, it is converted back to (c',h',s') using the geometry that the disk uses today, and that is used for disk I/O.

Note that there is a bit of confusion in the use of the expression `LBA': As a term describing disk capabilities it means `Linear Block Addressing' (as opposed to CHS Addressing). As a term in the BIOS Setup, it describes a translation scheme sometimes called `assisted LBA' - see above under ` The 8.4 GB limit '.

Something similar works when the firmware doesn't speak LBA but the BIOS knows about translation. (In the setup this is often indicated as `Large'.) Now the BIOS will present a geometry (C,H,S) to the operating system, and use (C',H',S') while talking to the disk controller. Usually S = S', C = C'/N and H = H'*N, where N is the smallest power of two that will ensure C' <= 1024 (so that least capacity is wasted by the rounding down in C' = C/N). Again, this allows access of up to 8.4 GB (7.8 GiB).

(The third setup option usually is `Normal', where no translation is involved.)

If a BIOS does not know about `Large' or `LBA', then there are software solutions around. Disk Managers like OnTrack or EZ-Drive replace the BIOS disk handling routines by their own. Often this is accomplished by having the disk manager code live in the MBR and subsequent sectors (OnTrack calls this code DDO: Dynamic Drive Overlay), so that it is booted before any other operating system. That is why one may have problems when booting from a floppy when a Disk Manager has been installed.

The effect is more or less the same as with a translating BIOS - but especially when running several different operating systems on the same disk, disk managers can cause a lot of trouble.

Linux does support OnTrack Disk Manager since version 1.3.14, and EZ-Drive since version 1.3.29. Some more details are given below.

8. Kernel disk translation for IDE disks

If the Linux kernel detects the presence of some disk manager on an IDE disk, it will try to remap the disk in the same way this disk manager would have done, so that Linux sees the same disk partitioning as for example DOS with OnTrack or EZ-Drive. However, NO remapping is done when a geometry was specified on the command line - so a `hd=*cyls*,*heads*,*secs*' command line option might well kill compatibility with a disk manager.

The remapping is done by trying 4, 8, 16, 32, 64, 128, 255 heads (keeping H*C constant) until either C <= 1024 or H = 255.

The details are as follows - subsection headers are the strings appearing in the corresponding boot messages. Here and everywhere else in this text partition types are given in hexadecimal.

8.1 EZD

EZ-Drive is detected by the fact that the first primary partition has type 55. The geometry is remapped as described above, and the partition table from sector 0 is discarded - instead the partition table is read from sector 1. Disk block numbers are not changed, but writes to sector 0 are redirected to sector 1. This behaviour can be changed by recompiling the kernel with `#define FAKE_FDISK_FOR_EZDRIVE 0` in `ide.c`.

8.2 DM6:DDO

OnTrack DiskManager (on the first disk) is detected by the fact that the first primary partition has type 54. The geometry is remapped as described above and the entire disk is shifted by 63 sectors (so that the old sector 63 becomes sector 0). Afterwards a new MBR (with partition table) is read from the new sector 0. Of course this shift is to make room for the DDO - that is why there is no shift on other disks.

8.3 DM6:AUX

OnTrack DiskManager (on other disks) is detected by the fact that the first primary partition has type 51 or 53. The geometry is remapped as described above.

8.4 DM6:MBR

An older version of OnTrack DiskManager is detected not by partition type, but by signature. (Test whether the offset found in bytes 2 and 3 of the MBR is not more than 430, and the short found at this offset equals 0x55AA, and is followed by an odd byte.) Again the geometry is remapped as above.

8.5 PTBL

TOC

Finally, there is a test that tries to deduce a translation from the `start` and `end` values of the primary partitions: If some partition has start and end sector number 1 and 63, respectively, and end heads 31, 63, 127 or 254, then, since it is customary to end partitions on a cylinder boundary, and since moreover the IDE interface uses at most 16 heads, it is conjectured that a BIOS translation is active, and the geometry is remapped to use 32, 64, 128 or 255 heads, respectively. However, no remapping is done when the current idea of the geometry already has 63 sectors per track and at least as many heads (since this probably means that a remapping was done already).

---

9. Consequences

What does all of this mean? For Linux users only one thing: that they must make sure that LILO and `fdisk` use the right geometry where `right' is defined for `fdisk` as the geometry used by the other operating systems on the same disk, and for LILO as the geometry that will enable successful interaction with the BIOS at boot time. (Usually these two coincide.)

How does `fdisk` know about the geometry? It asks the kernel, using the `HDIO_GETGEO` ioctl. But the user can override the geometry interactively or on the command line.

How does LILO know about the geometry? It asks the kernel, using the `HDIO_GETGEO` ioctl. But the user can override the geometry using the `disk=' option in `/etc/lilo.conf` (see lilo.conf(5)). One may also give the `linear` option to LILO, and it will store LBA addresses instead of CHS addresses in its map file, and find out of the geometry to use at boot time (by using INT 13 Function 8 to ask for the drive geometry).

How does the kernel know what to answer? Well, first of all, the user may have specified an explicit geometry with a `hda=*cyls*,*heads*,*secs*' kernel command line option (see bootparam(7)). And otherwise the kernel will guess, possibly using values obtained from the BIOS or the hardware.

10. Details

10.1 IDE details - the seven geometries

The IDE driver has five sources of information about the geometry. The first (G_user) is the one specified by the user on the command line. The second (G_bios) is the BIOS Fixed Disk Parameter Table (for first and second disk only) that is read on system startup, before the switch to 32-bit mode. The third (G_phys) and fourth (G_log) are returned by the IDE controller as a response to the IDENTIFY command - they are the `physical' and `current logical' geometries.

On the other hand, the driver needs two values for the geometry: on the one hand G_fdisk, returned by a `HDIO_GETGEO` ioctl, and on the other hand G_used, which is actually used for doing I/O. Both G_fdisk and G_used are initialized to G_user if given, to G_bios when this information is present according to CMOS, and to G_phys otherwise. If G_log looks reasonable then G_used is set to that. Otherwise, if G_used is unreasonable and G_phys looks reasonable then G_used is set to G_phys. Here `reasonable' means that the number of heads is in the range 1-16.

To say this in other words: the command line overrides the BIOS, and will determine what `fdisk` sees, but if it specifies a translated geometry (with more than 16 heads), then for kernel I/O it will be overridden by output of the IDENTIFY command.

Note that G_bios is rather unreliable: for systems booting from SCSI the first and second disk may well be SCSI disks, and the geometry that the BIOS reported for sda is used by the kernel for hda. Moreover, disks that are not mentioned in the BIOS Setup are not seen by the BIOS. This means that, e.g., in an IDE-only system where hdb is not given in the Setup, the geometries reported by the BIOS for the first and second disk will apply to hda and hdc.

10.2 SCSI details

The situation for SCSI is slightly different, as the SCSI commands already use logical block numbers, so a `geometry' is entirely irrelevant for actual I/O. However, the format of the partition table is still the same, so `fdisk` has to invent some geometry, and also uses `HDIO_GETGEO` here - indeed, `fdisk` does not distinguish between IDE and SCSI disks. As one can see from the detailed description below, the various drivers each invent a somewhat different geometry. Indeed, one big mess.

If you are not using DOS or so, then avoid all extended translation settings, and just use 64 heads, 32 sectors per track (for a nice, convenient 1 MiB per cylinder), if possible, so that no problems arise when you move the disk from one controller to another. Some SCSI disk drivers (aha152x, pas16, ppa, qlogicfas, qlogicisp) are so nervous about DOS compatibility that they will not allow a Linux-only system to use more than about 8 GiB. This is a bug.

What is the real geometry? The easiest answer is that there is no such thing. And if there were, you

wouldn't want to know, and certainly NEVER, EVER tell `fdisk` or LILO or the kernel about it. It is strictly a business between the SCSI controller and the disk. Let me repeat that: only silly people tell `fdisk`/LILO/kernel about the true SCSI disk geometry.

But if you are curious and insist, you might ask the disk itself. There is the important command READ CAPACITY that will give the total size of the disk, and there is the MODE SENSE command, that in the Rigid Disk Drive Geometry Page (page 04) gives the number of cylinders and heads (this is information that cannot be changed), and in the Format Page (page 03) gives the number of bytes per sector, and sectors per track. This latter number is typically dependent upon the notch, and the number of sectors per track varies - the outer tracks have more sectors than the inner tracks. The Linux program `scsiinfo` will give this information. There are many details and complications, and it is clear that nobody (probably not even the operating system) wants to use this information. Moreover, as long as we are only concerned about `fdisk` and LILO, one typically gets answers like C/H/S=4476/27/171 - values that cannot be used by `fdisk` because the partition table reserves only 10 resp. 8 resp. 6 bits for C/H/S.

Then where does the kernel `HDIO_GETGEO` get its information from? Well, either from the SCSI controller, or by making an educated guess. Some drivers seem to think that we want to know `reality', but of course we only want to know what the DOS or OS/2 FDISK (or Adaptec AFDISK, etc) will use.

Note that Linux `fdisk` needs the numbers H and S of heads and sectors per track to convert LBA sector numbers into c/h/s addresses, but the number C of cylinders does not play a role in this conversion. Some drivers use (C,H,S) = (1023,255,63) to signal that the drive capacity is at least 1023*255*63 sectors. This is unfortunate, since it does not reveal the actual size, and will limit the users of most `fdisk` versions to about 8 GiB of their disks - a real limitation in these days.

In the description below, M denotes the total disk capacity, and C, H, S the number of cylinders, heads and sectors per track. It suffices to give H, S if we regard C as defined by M / (H*S).

By default, H=64, S=32.

aha1740, dtc, g_NCR5380, t128, wd7000:

> H=64, S=32.

aha152x, pas16, ppa, qlogicfas, qlogicisp:

> H=64, S=32 unless C > 1024, in which case H=255, S=63, C = min(1023, M/(H*S)). (Thus C is truncated, and H*S*C is not an approximation to the disk capacity M. This will confuse most versions of `fdisk`.) The `ppa.c` code uses M+1 instead of M and says that due to a bug in `sd.c` M is off by 1.

advansys:

> H=64, S=32 unless C > 1024 and moreover the `> 1 GB' option in the BIOS is enabled, in which case H=255, S=63.

aha1542:

> Ask the controller which of two possible translation schemes is in use, and use either H=255, S=63 or H=64, S=32. In the former case there is a boot message "aha1542.c: Using extended bios translation".

aic7xxx:

H=64, S=32 unless C > 1024, and moreover either the "extended" boot parameter was given, or the `extended' bit was set in the SEEPROM or BIOS, in which case H=255, S=63. In Linux 2.0.36 this extended translation would always be set in case no SEEPROM was found, but in Linux 2.2.6 if no SEEPROM is found extended translation is set only when the user asked for it using this boot parameter (while when a SEEPROM is found, the boot parameter is ignored). This means that a setup that works under 2.0.36 may fail to boot with 2.2.6 (and require the `linear' keyword for LILO, or the `aic7xxx=extended' kernel boot parameter).

buslogic:

H=64, S=32 unless C >= 1024, and moreover extended translation was enabled on the controller, in which case if M < 2^22 then H=128, S=32; otherwise H=255, S=63. However, after making this choice for (C,H,S), the partition table is read, and if for one of the three possibilities (H,S) = (64,32), (128,32), (255,63) the value endH=H-1 is seen somewhere then that pair (H,S) is used, and a boot message is printed "Adopting Geometry from Partition Table".

fdomain:

Find the geometry information in the BIOS Drive Parameter Table, or read the partition table and use H=endH+1, S=endS for the first partition, provided it is nonempty, or use H=64, S=32 for M < 2^21 (1 GiB), H=128, S=63 for M < 63*2^17 (3.9 GiB) and H=255, S=63 otherwise.

in2000:

Use the first of (H,S) = (64,32), (64,63), (128,63), (255,63) that will make C <= 1024. In the last case, truncate C at 1023.

seagate:

Read C,H,S from the disk. (Horrors!) If C or S is too large, then put S=17, H=2 and double H until C <= 1024. This means that H will be set to 0 if M > 128*1024*17 (1.1 GiB). This is a bug.

ultrastor and u14_34f:

One of three mappings ((H,S) = (16,63), (64,32), (64,63)) is used depending on the controller mapping mode.

If the driver does not specify the geometry, we fall back on an educated guess using the partition table, or using the total disk capacity.

Look at the partition table. Since by convention partitions end on a cylinder boundary, we can, given `end = (endC,endH,endS)` for any partition, just put H = `endH+1` and S = `endS`. (Recall that sectors are counted from 1.) More precisely, the following is done. If there is a nonempty partition, pick the partition with the largest `beginC`. For that partition, look at `end+1`, computed both by adding `start` and `length` and by assuming that this partition ends on a cylinder boundary. If both values agree, or if `endC` = 1023 and `start+length` is an integral multiple of `(endH+1)*endS`, then assume that this partition really was aligned on a cylinder boundary, and put H = `endH+1` and S = `endS`. If this fails, either because there are no partitions, or because they have strange sizes, then look only at the disk capacity M. Algorithm: put H = M/(62*1024) (rounded up), S = M/(1024*H) (rounded up), C = M/(H*S) (rounded down). This has the effect of producing a (C,H,S) with C at most 1024 and S at most 62.

11. The Linux IDE 8 GiB limit

The Linux IDE driver gets the geometry and capacity of a disk (and lots of other stuff) by using an ATA IDENTIFY request. Until recently the driver would not believe the returned value of lba_capacity if it was more than 10% larger than the capacity computed by C*H*S. However, by industry agreement large IDE disks (with more than 16514064 sectors) return C=16383, H=16, S=63, for a total of 16514064 sectors (7.8 GB) independent of their actual size, but give their actual size in lba_capacity.

Recent Linux kernels (2.0.34, 2.1.90) know about this and do the right thing. If you have an older Linux kernel and do not want to upgrade, and this kernel only sees 8 GiB of a much larger disk, then try changing the routine `lba_capacity_is_ok` in `/usr/src/linux/drivers/block/ide.c` into something like

```
static int lba_capacity_is_ok (struct hd_driveid *id) {
        id->cyls = id->lba_capacity / (id->heads * id->sectors);
        return 1;
}
```

For a more cautious patch, see 2.1.90.

11.1 BIOS complications

As just mentioned, large disks return the geometry C=16383, H=16, S=63 independent of the actual size, while the actual size is returned in the value of LBAcapacity. Some BIOSes do not recognize this, and translate this 16383/16/63 into something with fewer cylinders and more heads, for example 1024/255/63 or 1027/255/63. So, the kernel must not only recognize the single geometry 16383/16/63, but also all BIOS-mangled versions of it. Since 2.2.2 this is done correctly (by taking the BIOS idea of H and S, and computing C = capacity/(H*S)). Usually this problem is solved by setting the disk to Normal in the BIOS setup (or, even better, to None, not mentioning it at all to the BIOS). If that is impossible because you have to boot from it or use it also with DOS/Windows, and upgrading to 2.2.2 or later is not an option, use kernel boot parameters.

If a BIOS reports 16320/16/63, then this is usually done in order to get 1024/255/63 after translation.

11.2 Jumpers that select the number of heads

Many IBM disks have jumpers that allow you to choose between a 15-head an a 16-head geometry. The default settings will give you a 16-head disk. Sometimes both geometries address the same number of sectors, sometimes the 15-head version is smaller. There may be a good reason for this setup: Petri Kaukasoina writes: `A 10.1 Gig IBM Deskstar 16 GP (model IBM-DTTA-351010) was jumpered for 16 heads as default but this old PC (with AMI BIOS) didn't boot and I had to jumper it for 15 heads. hdparm -i tells RawCHS=16383/15/63 and LBAsects=19807200. I use 20960/15/63 to get the full capacity.' The

geometry 16383/15/63 is not yet recognized by the kernel, so explicit boot parameters are required h
For the jumper settings, see http://www.storage.ibm.com/techsup/hddtech/hddtech.htm.

TOC

11.3 Jumpers that clip total capacity

Many IBM disks have jumpers that allow you to make the disk appear smaller than it is. A silly thing to do, and probably no Linux user ever wants to use this, but some BIOSes crash on big disks. The usual solution is to keep the disk entirely out of the BIOS setup. But this may be feasible only if the disk is not your boot disk.

For example, a Deskstar 25 GB disk (DJNA-352500) can be jumpered to appear as a 2 GB disk. What a waste!

A more recent limit is the 33.8 GB limit. Also Linux still needs a patch to be able to cope with IDE disks larger than this. Disks larger than this limit, like a Deskstar 37.5 GB disk (DPTA-353750) can be jumpered to appear as a 33.8 GB disk. See also the BIOS 33.8 GB limit.

---

12. The Linux 65535 cylinder limit

The `HDIO_GETGEO` ioctl returns the number of cylinders in a short. This means that if you have more than 65535 cylinders, the number is truncated, and (for a typical SCSI setup with 1 MiB cylinders) a 80 GiB disk may appear as a 16 GiB one. Once one recognizes what the problem is, it is easily avoided.

12.1 IDE problems with 34+ GB disks

Drives larger than 33.8 GB will not work with recent kernels. The details are as follows. Suppose you bought a new IBM-DPTA-373420 disk with a capacity of 66835440 sectors (34.2 GB). Recent kernels will tell you that the size is 769*16*63 = 775152 sectors (0.4 GB), which is a bit disappointing. And giving command line parameters hdc=4160,255,63 doesn't help at all - these are just ignored. What happens? The routine idedisk_setup() retrieves the geometry reported by the disk (which is 16383/16/63) and overwrites what the user specified on the command line, so that the user data is used only for the BIOS geometry. The routine current_capacity() or idedisk_capacity() recomputes the cylinder number as 66835440/(16*63)=66305, but since this is stored in a short, it becomes 769. Since lba_capacity_is_ok() destroyed id->cyls, every following call to it will return false, so that the disk capacity becomes 769*16*63. A patch is available - probably it will soon get into some official kernel.

13. Extended and logical partitions

Above, we saw the structure of the MBR (sector 0): boot loader code followed by 4 partition table entries of 16 bytes each, followed by an AA55 signature. Partition table entries of type 5 or F or 85 (hex) have a special significance: they describe *extended* partitions: blobs of space that are further partitioned into *logical* partitions. (So, an extended partition is only a box, it cannot be used itself, one uses the logical partitions inside.) Only the location of the first sector of an extended partition is important. This first sector contains a partition table with four entries: one a logical partition, one an extended partition, and two unused. In this way one gets a chain of partition table sectors, scattered over the disk, where the first one describes three primary partitions and the extended partition, and each following partition table sector describes one logical partition and the location of the next partition table sector.

It is important to understand this: When people do something stupid while partitioning a disk, they want to know: Is my data still there? And the answer is usually: Yes. But if logical partitions were created then the partition table sectors describing them are written at the beginning of these logical partitions, and data that was there before is lost.

The program sfdisk will show the full chain. E.g.,

```
# sfdisk -l -x /dev/hda

Disk /dev/hda: 16 heads, 63 sectors, 33483 cylinders
Units = cylinders of 516096 bytes, blocks of 1024 bytes, counting from 0

   Device Boot Start     End   #cyls   #blocks   Id  System
/dev/hda1          0+    101     102-    51376+  83  Linux
/dev/hda2        102    2133    2032   1024128   83  Linux
/dev/hda3       2134   33482   31349  15799896    5  Extended
/dev/hda4          0       -       0         0    0  Empty

/dev/hda5       2134+   6197    4064-  2048224+  83  Linux
    -           6198   10261    4064   2048256    5  Extended
    -           2134    2133       0         0    0  Empty
    -           2134    2133       0         0    0  Empty

/dev/hda6       6198+  10261    4064-  2048224+  83  Linux
    -          10262   16357    6096   3072384    5  Extended
    -           6198    6197       0         0    0  Empty
    -           6198    6197       0         0    0  Empty
...
/dev/hda10     30581+  33482    2902-  1462576+  83  Linux
    -          30581   30580       0         0    0  Empty
    -          30581   30580       0         0    0  Empty
    -          30581   30580       0         0    0  Empty

#
```

It is possible to construct bad partition tables. Many kernels get into a loop if some extended partition points to itself or to an earlier partition in the chain. It is possible to have two extended partitions in one of these partition table sectors so that the partition table chain forks. (This can happen for example with an fdisk that does not recognize each of 5, F, 85 as an extended partition, and creates a 5 next to an F.) No standard fdisk type program can handle such situations, and some handwork is required to repair them. The Linux kernel will accept a fork at the outermost level. That is, you can have two chains of logical partitions. Sometimes this is useful - for example, one can use type 5 and be seen by DOS, and the other type 85, invisible for DOS, so that DOS FDISK will not crash because of logical partitions past cylinder 1024.

---

14. Problem solving

Many people think they have problems, while in fact nothing is wrong. Or, they think that the problems they have are due to disk geometry, while in fact disk geometry has nothing to do with the matter. All of the above may have sounded complicated, but disk geometry handling is extremely easy: do nothing at all, and all is fine; or perhaps give LILO the keyword `linear' if it doesn't get past `LI' when booting. Watch the kernel boot messages, and remember: the more you fiddle with geometries (specifying heads and cylinders to LILO and fdisk and on the kernel command line) the less likely it is that things will work. Roughly speaking, all is fine by default.

And remember: nowhere in Linux is disk geometry used, so no problem you have while running Linux can be caused by disk geometry. Indeed, disk geometry is used only by LILO and by fdisk. So, if LILO fails to boot the kernel, that may be a geometry problem. If different operating systems do not understand the partition table, that may be a geometry problem. Nothing else. In particular, if mount doesnt seem to work, never worry about disk geometry - the problem is elsewhere.

14.1 Problem: Linux invents the wrong geometry for my disk.

It is quite possible that a disk gets the wrong geometry. The Linux kernel asks the BIOS about hd0 and hd1 (the BIOS drives numbered 80H and 81H) and assumes that this data is for hda and hdb. But on a system that boots from SCSI, the first two disks may well be SCSI disks, and thus it may happen that the fifth disk, which is the first IDE disk hda, gets assigned a geometry belonging to sda. Such things are easily solved by giving boot parameters `hda=C,H,S' for the appropriate numbers C, H and S, either at boot time or in /etc/lilo.conf.

14.2 Nonproblem: Identical disks have different geometry?

`I have two identical 10 GB IBM disks. However, fdisk gives different sizes for them. Look:

```
# fdisk /dev/hdb
Disk /dev/hdb: 255 heads, 63 sectors, 1232 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot  Start      End   Blocks   Id  System
/dev/hdb1           1     1232  9896008+  83  Linux native
# fdisk /dev/hdd
Disk /dev/hdd: 16 heads, 63 sectors, 19650 cylinders
```

```
Units = cylinders of 1008 * 512 bytes

   Device Boot   Start      End   Blocks   Id  System
/dev/hdd1            1    19650  9903568+  83  Linux native
```

How come?'

What is happening here? Well, first of all these drives really are 10gig: hdb has size 255*63*1232*512 = 10133544960, and hdd has size 16*63*19650*512 = 10141286400, so, nothing is wrong and the kernel sees both as 10.1 GB. Why the difference in size? That is because the kernel gets data for the first two IDE disks from the BIOS, and the BIOS has remapped hdb to have 255 heads (and 16*19650/255=1232 cylinders). The rounding down here costs almost 8 MB.

If you would like to remap hdd in the same way, give the kernel boot parameters `hdd=1232,255,63'.

14.3 Nonproblem: fdisk sees much more room than df?

fdisk will tell you how many blocks there are on the disk. If you make a filesystem on the disk, say with mke2fs, then this filesystem needs some space for bookkeeping - typically something like 4% of the filesystem size, more if you ask for a lot of inodes during mke2fs. For example:

```
# sfdisk -s /dev/hda9
4095976
# mke2fs -i 1024 /dev/hda9
mke2fs 1.12, 9-Jul-98 for EXT2 FS 0.5b, 95/08/09
...
204798 blocks (5.00%) reserved for the super user
...
# mount /dev/hda9 /somewhere
# df /somewhere
Filesystem         1024-blocks  Used Available Capacity Mounted on
/dev/hda9            3574475      13  3369664      0%   /mnt
# df -i /somewhere
Filesystem           Inodes   IUsed   IFree  %IUsed Mounted on
/dev/hda9            4096000      11 4095989     0%  /mnt
#
```

We have a partition with 4095976 blocks, make an ext2 filesystem on it, mount it somewhere and find that it only has 3574475 blocks - 521501 blocks (12%) was lost to inodes and other bookkeeping. Note that the difference between the total 3574475 and the 3369664 available to the user are the 13 blocks in use plus the 204798 blocks reserved for root. This latter number can be changed by tune2fs. This `-i 1024' is only reasonable for news spools and the like, with lots and lots of small files. The default would be:

```
# mke2fs /dev/hda9
```

```
# mount /dev/hda9 /somewhere
# df /somewhere
Filesystem          1024-blocks  Used Available Capacity Mounted on
/dev/hda9             3958475      13  3753664      0%   /mnt
# df -i /somewhere
Filesystem            Inodes   IUsed   IFree  %IUsed Mounted on
/dev/hda9             1024000      11 1023989     0%  /mnt
#
```

Now only 137501 blocks (3.3%) are used for inodes, so that we have 384 MB more than before. (Apparently, each inode takes 128 bytes.) On the other hand, this filesystem can have at most 1024000 files (more than enough), against 4096000 (too much) earlier.

Leased line Mini HOWTO

TOC

Rob van der Putten, rob@sput.webster.nl
v1.4, 5 July 1998
The most recent (beta) version of this HOWTO can be found at: www.sput.webster.nl

What is a leased line

Any fixed, that is permanent, point to point data communications link, which is leased from a telco or similar organisation.
The leased line involves cables, such as twisted pair, coax or fiber optic, and may involve all sorts of other hardware such as (pupin) coils, transformers, amplifiers and regenerators.

This document deals with:

> Configuring your modem and pppd to use a 2 wire twisted pair leased line.

This document does **NOT** deal with:

> SLIP, getting or installing pppd, synchronous data communication, baseband modems.

Assumptions

You should already have a working pppd on your system.
You also need Minicom or a similar program to configure your modems.

Modem

A leased line is not connected to a telephone exchange and does not provide DC power, dial tone, busy tone or ring signal. This means that your modems are on their own and have to be able to deal with this situation.

You should have 2 identical **external** modems supporting both leased line and dumb mode. Make sure your modems can actually do this! Also make sure your modem is properly documented.
You also need:

- 2 fully wired shielded RS232 cables. The shield should be connected to the connector shell (not pin 1) at both ends (not at one end).
- A RS232 test plug may be handy for test purposes.
- 2 RJ11 cords, one for each end of the leased line.
- A basic understanding of 'AT' commands.

Modem Configuration

A note on modem configuration and init strings in general:
Configure your modem software such as minicom or (m)getty to use the highest possible speed; 57600 bps for 14k4 and 115200 bps for 28k8 or faster modems.
Lots of people use very long and complicated init strings, often starting with AT&F and containing lots of modem brand and -type specific commands. This however is needlessly complicated.
Most programs feel happy with the same modem settings, so why not write these settings in the non volatile

memory of all your modems, and only use 'ATZ' as an init string in all your programs. This way you can sv or upgrade your modems without ever having to reconfigure any of your software.

Most programs require you to use the following settings;

- Fixed baud rate (no auto baud)
- Hardware bidirectional RTS-CTS flow control (no x-on/x-off)
- 8 Bits, no parity, 1 stopbit
- The modem should produce the **TRUE** DCD status (&C1)
- The modem should **NOT** ignore the DTR status (&D2 or &D3)

Check this with AT&V or AT&Ix (consult your modem documentation)

These settings are not necessarily the same as the default factory profile (&F), so starting an init string with AT&F is probably not a good idea in the first place. The smart thing to do is probably to use AT&F only when you have reason to believe that the modem setup stored in the non volatile memory is really screwed up.
If you think you have found the right setup for your modems, write it to non volatile memory with AT&W and test it thoroughly with Z-modem file transfers of both ASCII text and binary files.
Only if all of this works perfectly should you configure your modems for leased line.

Find out how to put your modem into dumb mode and, more importantly, how to get it out of dumb mode; The modem can only be reconfigured when it is not in dumb mode.
Make sure you actually configure your modems at the highest possible speed. Once in dumb mode it will ignore all 'AT' commands and consequently will not adjust its speed to that of the com port, but will use the speed at which it was configured instead (this speed is stored in a S-register by the AT&W command).

Now configure your modem as follows;

- Reset on DTR toggle (&D3, this is sometimes a S register). This setting is required by some ISP's!
- Leased line mode (&L1 or &L2, consult your modem documentation)
- The remote modem auto answer (S0=1), the local originate (S0=0)
- Disable result codes (Q1, sometimes the dumb mode does this for you)
- Dumb mode (\D1 or %D1, this is sometimes a jumper)
  In dumb mode the modem will ignore all AT commands (sometimes you need to disable the ESC char as well).

Write the configuration to non-volatile memory (&W).

Test

Now connect the modems to 2 computers using the RS232 cables and connect the modems to each other using a RJ11 lead. Use a modem program such as Minicom (Linux), procom or telix (DOS) on both computers to test the modems.
You should be able to type text from one computer to the other and vice versa. If the screen produces garbage check your com port speed and other settings.
Now disconnect and reconnect the RJ11 cord. Wait for the connection to reestablish itself. Disconnect and reconnect the RS232 cables, switch the modems on and off, stop and restart minicom.
The modems should always reconnect at the highest possible speed (some modems have speed indicator leds).
Check whether the modems actually ignores the ESC (+++) character. If necessary disable the ESC character.

If all of this works you may want to reconfigure your modems;

Switch off the sound at the remote modem (M0) and put the local modem at low volume (L1).

Examples

Hi-Tech

Originate (local):

ATL1&C1&D3&L2%D1&W&W1

Answer (remote):

ATM0&C1&D3&L2%D1S0=1&W&W1

Tron DF

The ESC char should be disabled by setting S2 > 127;

Originate:

ATL1&L1Q1&C1&D3S2=171\D1&W

Answer:

ATM0&L2Q1&C1&D3S0=1S2=171\D1&W

US Robotics Courier V-Everything

The following is based on information supplied by Rolf Raar.
The USR Sportster and USR Courier-I do not support leased line. You need the Courier V-everything version for this job.
There is a webpage on the USR site 'explaining' how to set-up your Courier for leased line. However, if you follow these instructions you will end up with a completely brain dead modem, which can not be controlled or monitored by your pppd.

The USR Courier can be configured with dip switches, however you need to feed it the config string first.
First make sure it uses the right factory profile. Unlike most other modems it has three; &F0, &F1 and &F2. The default, which is also the one you should use, is &F1. If you send it an AT&F, however it will load the factory profile &F0!
For the reset on DTR toggle you set bit 0 of S register 13. This means you have to set S13 to 1. Furthermore you need set it to leased line mode with &L1;
ATS13=1&L1&W
The dip switches are all default except for the following:

3

OFF Disable result codes

4

ON Disable offline commands

5

ON For originate, OFF For answer

8

OFF Dumb mode

PPPD

TOC

You need a pppd (Point to Point Protocol Daemon) and a reasonable knowledge of how it works. Consult the relevant RFC's or the Linux PPP HOWTO if necessary.

Since you are not going to use a login procedure, you don't use (m)getty and you do not need a (fake) user associated with the pppd controlling your link. You are not going to dial so you don't need any chat scripts either.

In fact, the modem circuit and configuration you have just build, are rather like a fully wired null modem cable. This means you have to configure your pppd the same way as you would with a null modem cable.

For a reliable link, your setup should meet the following criteria;

- Shortly after booting your system, pppd should raise the DTR signal in your RS232 port, wait for DCD to go up, and negotiate the link.
- If the remote system is down, pppd should wait until it is up again.
- If the link is up and then goes down, pppd should reset the modem (it does this by dropping and then raising DTR), and then try to reconnect
- If the quality of the link deteriorates too much, pppd should reset the modem and then reestablish the link.
- If the process controlling the link, that is the pppd, dies, a watchdog should restart the pppd.

Configuration

Suppose the modem is connected to COM2, the local IP address is 'Loc_Ip' and the remote IP address is 'Rem_Ip'. We want to use 576 as our MTU.

The /etc/ppp/options.ttyS1 would now be:

```
crtscts
mru 576
mtu 576
passive
Loc_Ip:Rem_Ip
-chap
modem
-pap
persist
```

So, if the local system is 192.168.1.1 and the remote system is 10.1.1.1, then /etc/options.ttyS1 on the local system would be:

```
crtscts
mru 576
mtu 576
passive
192.168.1.1:10.1.1.1
-chap
modem
-pap
```

```
persist
```

The options.ttyS1 on the remote system would be:

```
crtscts
mru 576
mtu 576
passive
10.1.1.1:192.168.1.1
-chap
modem
-pap
persist
```

The passive option limits the number of (re)connection attempts.
The persist option will keep pppd alive in case of a disconnect or when it can't connect in the first place.
If you telnet a lot while doing filetransfers (FTP or webbrowsing) at the same time, you might want to use a smaller MTU and MRU such as 296. This will make the remote system more responsive.
If you don't care much about telnetting during FTP, you could set the MTU and MRU to 1500.

Scripts

Starting the pppd and keeping it alive

The script /usr/local/sbin/PRem_Host.sh shown below starts the pppd. You have to start it **after** the com ports are configured (subsitute Rem_Host with the remote host's name).

```
#!/bin/bash
# Optional sleep
#/usr/bin/sleep 30
( /usr/sbin/pppd /dev/ttyS1 115200 ) &
```

It could be called from the script which configures your networkcard (/etc/init.d/network on Debian systems), in which case you have to uncomment the sleep, to make sure the comports are configured first.
If you use setserial, you could call it from there. An other way is to call it from rc.local.
On Debian systems, there is a directory /etc/rc.boot/. According to Rolf Raar, the scripts in this Directory are executed in alphabetical order. The setserial script is called 0setserial. If you call your script from a script called 2leased-lines, it will be executed after 0setserial.

Though the persist option should make this superfluous, the pppd can also be restarted using ip-down;

```
#!/bin/bash
case $s in
     /dev/ttyS1)
          /usr/bin/sleep 30
          /usr/local/sbin/PRem_Host.sh &
          ;;
esac
```

If the pppd dies it should be restarted by the persist option or the ip-up script (optional). If this doesn't happen the script /usr/local/sbin/test-Rem_Host-ppp restarts it for you. This script checks whether the remote interface

TOC

exists and will try to start the pppd if it doesn't;

```
#!/bin/bash
if ! ( /sbin/ifconfig | grep Rem_Ip > /dev/null )
then
     # PPP gone
     logger "Rem_Host PPP gone ; restarted"
     /usr/local/sbin/PRem_Host.sh &
fi
```

The following line in crontab runs the above script every 5 minutes. You could run the script more often if you like but I would not run it more often then once every two minutes.

```
*/5 * * * *     /usr/local/sbin/test-Rem_Host-ppp
```

You can edit the crontab with 'crontab -e'. This will restart cron after editing. It uses the editor specified by the 'EDITOR' environment variable, which may be set by /etc/profile or ~/.profile (export EDITOR=Your_Favourite_Editor).

Some people run pppd from /etc/inittab, but I never tested this.

Setting the routes

The default route can be set with the defaultroute option or with the /etc/ppp/ip-up script;

```
#!/bin/bash
case $2 in
     /dev/ttyS1)
          /sbin/route add -net 0.0.0.0 gw Rem_Ip netmask 0.0.0.0
          ;;
esac
```

Ip-up can also be used to sync your clock using netdate.

Of course the route set in ip-up is not necessarily the default route.
Your ip-up sets the route to the remote network while the ip-up script on the remote system sets the route to your network. If your network is 192.168.1.0 and your ppp interface 192.168.1.1, the ip-up script on the remote machine looks like this;

```
#!/bin/bash
case $2 in
   /dev/ttyS1)
      /sbin/route add -net 192.168.1.0 gw 192.168.1.1 netmask 255.255.255.0
      ;;
esac
```

The 'case $2' and '/dev/ttyS1)' bits are there in case you use more than one ppp link. Ip-up will run each time a link comes up, but only the part between '/dev/ttySx)' and ';;' will be executed, setting the right route for the right ttyS.
You can find more about routing in the NET-3-HOWTO section on routing

TOC

Test

Test the whole thing just like the modem test.
If it works, get on your bike and bring the remote modem to the remote side of your link.

The Linux ``Linux-DOS-Win95-OS2'' mini-HOWTO

Mike Harlan, r3mdh@raex.com

v1.3.1, 11 November 1997

This document presents a procedure to make 4 operating systems co-exist on a single hard disk.

Disclaimer:

Any damages inflicted on any machine by you as a result of your reading of this HOWTO is still YOUR FAULT. When you read about deleting partitions and formatting disks, be smart. Realize that doing any of the above will result in loss of data. So, BACKUP SOON, BACKUP OFTEN. This is your second-to-the-last warning.

Stuff to make the lawyers happy:

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have any questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu via email.

Updates from v1.3:

- Updated my e-mail address. My address has changed from r3mdh@imperium.net to r3mdh@raex.com.

Updates from v1.0:

- Updated my e-mail address. My address has changed from r3mdh@dax.cc.uakron.edu t r3mdh@imperium.net.

TOC

Now.....on with the show!

1. Introduction

2. The Procedure

---

1. Introduction

After many days of struggle and frustration, I finally figured out how to accomplish what I wanted. I have a 1.2GB HD and 16MB RAM PC. I wanted to have 4 operating systems on my system: MSDOS v6.22, Windows 95, OS/2, and Linux. Until now, I have found no Linux HOWTO to perform the task of getting each and every one of these operating systems on one machine and still have the ability to boot each (it is possible to write the OSs to different partitions, but getting them to boot and not hang at the ``Starting MSDOS'' message, for example, is something that I had to figure out. Well, after much trial and error, I have come up with the following recipe to perform this feat:

Before I begin going through the procedure step-by-step, let me first clue you in on what I eventually wish to accomplish:

```
<NAME>                        <SIZE>   <LABEL>  <PARTITION NUMBER>
-------------------------------------------------------------------
MSDOS v6.22                     11MB   P1       Primary Partition 1
-------------------------------------------------------------------
Windows '95                    350MB   P2       Primary Partition 2
-------------------------------------------------------------------
OS/2 Boot Manager                2MB   P3       Primary Partition 3
-------------------------------------------------------------------
DOS/Win Data                   511MB   E1       Primary Partition 4
OS/2 Warp 3.0                  127MB   E2       Extended into 4 logical
Linux Slakware 3.2 Swap          XMB   E3       drives (sub-partitions)
Linux Slakware 3.2 Native  219-XMB   E4       labeled E1-E4
-------------------------------------------------------------------
```

What does all of this mean? Well, let me talk you through it. First, we need to use up all 4 partitions on our 1 harddrive. Partitions 1 through 3 are PRIMARY partitions. We will store MSDOS, Win95 and the OS/2 Boot Manager (which inefficiently requires its OWN partition) on these partitions, respectively. We then have 1 partition left. But, we want to have a drive just for DOS/Win95 data (this gives us a large place to dump all of those ZIP files we so often download from the Internet as well as a place to store Win95 programs and data that we don't have enough room for on the Win95 partition), OS/2, and Linux (both a Linux native as well as Linux swap partition).

Now, I've been getting a lot of grief through e-mail lately about why I have a separate partition for Win95 and one for DOS and one for Win95/DOS data. Here's my answer: having separate Win95 and DOS partitions isn't for everyone. Perhaps you don't even use DOS anymore. Perhaps Windows '95 is your answer for any software written for a Microsoft platform. Well, there are some people out there who still use DOS and can't live without it. Whether it's because DOS can run 16-bit applications faster and more efficient, or because the certain DOS program that you may use won't run under a Windows environment (the Gravis Ultrasound soundcard setup program comes to mind), you simply MUST be able to boot into DOS from time to time. This HOWTO was written with those people in mind. If you still have difficulty swallowing the fact that the word ``MSDOS'' is included in this HOWTO, then feel free to sit down and write a Linux-Win95-OS/2-only HOWTO. :)

Now, back to the diagram above. Like I said, we have 1 partition to cram 4 things onto: DOS/Win data, OS/2 and Linux native & swap. This can be done by creating what are called logical drives (or logical partitions depending on which book you read) within the 1 primary partition. When we create these logical drives within a primary partition, we refer to this primary partition as an EXTENDED PARTITION (because it is extended beyond the scope of a single, primary partition and instead contains up to 3 subpartitions (logical drives)). Confused? If so, you might want to read the OS/2 manual about this topic. It will explain it better than I have here.

Now, you might be scratching your head saying ``he wants to cram 4 things onto that extended partition, but he just got done saying we can only have 3 subpartitions to put them on!'' This is true, at least in MS-DOS's and OS/2's reasoning. But, here one of the many powerful advantages to Linux comes to save the day. Linux can create more than 3 subpartitions on an extended drive. Just how many, I don't know. But, I know it can create at LEAST 4 (what we need). So, when we go through the steps of the installation procedure below, keep in mind that when we create partitions using an MSDOS or OS/2 program, we will only create 3 logical drives. Then, when we go to Linux, we will split one of them into two. Essentially, you can think of it as ``tricking'' MSDOS and OS/2 into seeing only 3 logical drives, but in reality (and to Linux), there will be 4.

---

2. The Procedure

NOTE:

Numbers in parentheses were the number of megabytes that I used on my 1.2GB harddrive.

Now, let's move on to the step-by-step procedure:

1. Run `view.exe` on the Linux Slakware CD and create the Linux Boot and Linux Root floppies. For some reason, the Linux bootstrap program calls the Root disk the Ramdisk floppy. I'll refer to this disk from now on as the Ramdisk (Root) floppy.
2. Backup any information that you wish to keep to tape (or whatever media you have available to you).
3. Boot your original MSDOS installation floppy, disk 1.
4. When ```Starting MSDOS`'' appears, press F5 to bypass `config.sys` and `autoexec.bat`.
5. Run `a:\fdisk.exe`.
6. Delete all partitions (you have been warned: DELETING / MODIFYING OF ANY PARTITION WILL RESULT IN THE LOSS OF ALL DATA ON THE DISK!).
7. Add an MSDOS primary partition. (11MB)
8. `Format` this partition.
9. Run `a:\setup.exe` to install MSDOS on this partition.
10. Boot OS/2.
11. Select Advanced installation.
12. Run `FDISK`. (this will eventually popup for you if you run through the OS/2 installation)
13. Add 1 primary partition after the MSDOS one. This will become our Windows 95 partition. (349MB)
14. Add the Boot Manager to the next primary partition. (2MB)
15. Add an extended partition.
16. Add a logical drive to the extended partition. This will become the data portion of our MSDOS system. (511MB)
17. Add another logical drive to the extended partition. This will become our OS/2 HPFS (``High Performance File System'') partition. (127MB)
18. Add 1 last logical drive to the extended partition using the remaining space on the drive. This will later become 2 partitions under Linux — our swap partition and our native Linux partition. But, since OS/2 (and DOS as well) can only write up to 6 partitions per drive (3 primary and 3 logical drives housed within 1 extended partition), we have to create only one at this time. And we DO have to create this partition. Don't leave this as free space and expect Linux to be able to create the two partitions. Due to the way that OS/2's FDISK works, where you add your last logical drive to the extended partition marks the END of the extended partition. You can**not** add partitions beyond this point. So, in other words, creating this one last logical drive serves as a space-filler for Linux. Later we will delete this partition and add 2 new ones in the space that it once took up.
19. Add partitions 1, 2, and 5 to the Boot Manager.
20. Make partition 5 installable. Your `FDISK` screen should now look like this (or something like this):

```
                                FDISK

Disk 1
```

```
Partition Information
Name            Status          Access          FS Type         MBytes
---------------------------------------------------------------------------

MSDOS           Bootable        C: Primary      FAT              11
WIN 95          Bootable         : Primary      FAT             350
                Startable        : Primary      BOOT MANAGER      2
                None            D: Logical      Unformatted     511
OS/2            Installable     E: Logical      FAT             127
                None            F: Logical      Unformatted     219

---------------------------------------------------------------------------
```

21. Continue on with the OS/2 installation process.
22. Reboot and select partition 2 (Windows 95) from the Boot Manager.
23. When the missing operating system error pops up, boot your MSDOS installation disk. We selected this partition in order to ``hide'' the MSDOS partition. OS/2's Boot Manager is a bit strange. For every bootable partition you have (in our case, MSDOS and OS/2), you can have only one of them visible at a time. What this means is that if you boot into one partition, MSDOS for example, the other partition (Win95) is invisible. The MSDOS partition and Win95 essentially SHARE a drive letter. That's why under ``Access'' above, MSDOS has drive letter C and Win95 has no drive letter. Under the current circumstances, MSDOS is visible with drive letter C and Win95 is invisible with no drive letter. If we were to boot Win95, the opposite would be true: MSDOS would be invisible with no drive letter and Win95 would be visible with drive letter C. **An invisible drive cannot be accessed AT ALL.** If you wish to copy files between two drives in which only one can be visible at one time, you must use a common (non-bootable) drive to swap files. In our case, the DOS/Win95 Data drive (drive D 511MB above) will serve as this common drive. You might be asking ``Won't we eventually be booting Linux also?''. The answer is yes, we will. But let's not get into that just yet.
24. Install MSDOS to Partition 2 (we'll need this in order to install Windows 95).
25. Boot Partition 2.
26. Install Windows 95 to this partition (if you are running the upgrade version of Win95, you may need to have your Win3.1 installation disk 1 ready to insert).
27. Boot Partition 1.
28. Format Partition 4.
29. Restore DOS data from tape (if any) to partitions 1 and 4.
30. Boot the Linux Boot Floppy.
31. Follow up with the Linux Ramdisk (Root) floppy.
32. When you log in as root and get to the # prompt, type ```fdisk`'' and press enter.
33. Delete the last partition (the one we created in step 18).
34. Add 1 16MB partition and tag it as filesystem type Linux Swap. (17MB)
35. Add 1 last partition with the remaining cylinders on the disk and tag this as filesystem type Linux native. (198MB)
36. Write the changes to the boot sector and reboot.
37. When you get to the # prompt again, run `setup`.
38. Install Linux to the last partition.
39. When you install LILO, be sure to install it to the root of the last partition (NOT to the MBR, as you will destroy all of your previous work in this HOWTO if you do so). Add only the last partition to LILO and set the timer to zero. By doing this, when you select Linux from the OS/2 Boot Manager, LILO will activate and will then boot Linux from the logical drive on the extended partition. Since Linux is the only partition that we wish to activate from LILO, we don't need a timer on it (unless you have more than one Kernel that you wish to load. In this case, you may want to set the timer to something more than 0 seconds).
40. Activate the Linux swap partition (refer to the Linux Installation and Getting Started Manual by Matt Welsh for this).

41. Boot OS/2.
42. Run FDISK.
43. Add Linux to the Boot Manager using the Linux NATIVE partition (type `83` **not** `82`!).

Your `FDISK` screen should now look like this (or something like this):

```
                                FDISK

Disk 1
------------------------------------------------------------------------------

Partition Information
Name            Status          Access          FS Type         MBytes
------------------------------------------------------------------------------

MSDOS           Bootable        C: Primary      FAT              11
WIN 95          Bootable         : Primary      FAT             350
                Startable        : Primary      BOOT MANAGER      2
                None            D: Logical      FAT             511
OS/2            Bootable        E: Logical      HPFS            127
                None             : Logical      Type 82          17
Linux           Bootable         : Logical      Type 83         198
------------------------------------------------------------------------------
```

...And you're done!

Send any comments/suggestions/problems (as a last resort, please!) to me at r3mdh@raex.com.

Mike Harlan, 11 NOV 1997

The Linux+FreeBSD mini-HOWTO

Niels Kristian Bech Jensen, nkbj@image.dk

v1.10, 15 January 1999

This document describes how to use Linux and FreeBSD on the same system. It introduces FreeBSD and discusses how the two operating systems can cooperate, e.g. by sharing swap space. You should probably have some experience with Linux or FreeBSD and hard drive partitioning (`fdisk`) before you read this document. The tips herein are tested using FreeBSD 2.2.2, but they should be valid for newer versions as well. Do not hesitate to mail me if you have comments, questions or suggestions about this document. I would also like to hear from people who have experience using Linux together with NetBSD or OpenBSD.

1. What is FreeBSD?

2. The FreeBSD way of labelling hard drives

- 2.1 FreeBSD ``slices'' and ``partitions''
- 2.2 Drive and partition labelling in Linux and FreeBSD

3. Sharing swap space between Linux and FreeBSD

- 3.1 Installing and preparing Linux
- 3.2 Installing FreeBSD
- 3.3 Setting up the FreeBSD swap partition in Linux

4. Booting FreeBSD using LILO

5. Mounting filesystems

- 5.1 Mounting UFS filesystems under Linux
- 5.2 Mounting ext2fs filesystems under FreeBSD

TOC

6. Running foreign binaries

- 6.1 Running FreeBSD binaries under Linux
- 6.2 Running Linux binaries under FreeBSD

7. Information resources

8. Credits and legal stuff

- 8.1 Legal stuff

---

1. What is FreeBSD?

FreeBSD is a free Unix-like operating system much like Linux. The main difference is that, while the Linux kernel has been written from scratch, FreeBSD is based on the freely redistributable parts of 4.4BSD (Berkeley Software Distribution) known as 4.4BSD-lite. This fact might lead some people to suggest that FreeBSD is closer to being ``real'' UNIX® than Linux. FreeBSD runs only on the Intel PC platform (i386 and higher); ports to the DEC Alpha and Sun Sparc platforms are being worked on at the moment. NetBSD and OpenBSD are similar to FreeBSD, and both run on several platforms. Hardware requirements for all these *BSD systems are similar to those for Linux.

The development of FreeBSD is managed in a different way than the Linux development. A core team of developers serve as arbitrators and provide leadership for the project. Big changes are discussed in advance on the mailing lists. The FreeBSD project has two development trees (just like Linux): **``-CURRENT''** and **``-STABLE''**. The ``-CURRENT'' development tree is where the development of new features happens. Development to the ``-STABLE'' tree are restricted to bug fixes and some thoroughly tested new features.

FreeBSD can be used and (re-)distributed freely just as Linux. Most parts of the system are released under the BSD copyright; the rest is under the GNU GPL or other open-source licences.

2. The FreeBSD way of labelling hard drives

Linux and FreeBSD label hard drives and partitions after two differents schemes. This section explains the main differences between the two schemes. In fact the FreeBSD labelling scheme is an adaption of the traditional BSD labelling style ported to live within the PC's fdisk partitions. Thus it is very similar to other BSD-based Unix systems such as NetBSD, OpenBSD, Ultrix, Digital Unix, SunOS, and Solaris.

2.1 FreeBSD ``slices'' and ``partitions''

FreeBSD needs one of the four entries in the partition table on your PC's hard drive. This primary partition is called a ``**slice**'' in FreeBSD terminology. It then uses the `disklabel` program to make up to eight partitions in this primary partition. These logical partitions are called ``**partitions**'' in FreeBSD terminology. This concept is similar to the way Linux (and DOS) handles logical partitions in an extended partition. You cannot install FreeBSD in an extended partition made by Linux (or DOS). Note that the Linux `fdisk` program doesn't display the BSD partitions in a FreeBSD slice from the main menu, but it can display BSD disklabel information if you give the command `b'. The output is something like this (`/dev/hda4` is the FreeBSD slice):

```
bash# fdisk /dev/hda

Command (m for help): p

Disk /dev/hda: 64 heads, 63 sectors, 621 cylinders
Units = cylinders of 4032 * 512 bytes

   Device Boot    Begin    Start      End   Blocks   Id  System
/dev/hda1   *         1        1       27    54400+  83  Linux native
/dev/hda2            28       28       55    56448   83  Linux native
/dev/hda3            56       56      403   701568   83  Linux native
/dev/hda4           404      404      621   439488   a5  BSD/386

Command (m for help): b
Reading disklabel of /dev/hda4 at sector 1624897.

BSD disklabel command (m for help): p

8 partitions:
#        size   offset    fstype   [fsize bsize   cpg]
  a:    64512  1624896    4.2BSD        0     0     0   # (Cyl.  404 - 419)
  b:   104832  1689408      swap                         # (Cyl.  420 - 445)
  c:   878976  1624896    unused        0     0         # (Cyl.  404 - 621)
  e:    64512  1794240    4.2BSD        0     0     0   # (Cyl.  446 - 461)
  f:   645120  1858752    4.2BSD        0     0     0   # (Cyl.  462 - 621)

BSD disklabel command (m for help): q
```

```
bash#
```

The letters `a'...`f' in the first column are the same labels as shown below in the example for a FreeBSD slice. There are three special partitions in BSD parlace. The letter `a' designates the root partition, `b' designates the swap partition, while `c' designates the whole slice. See the FreeBSD documentation for more information on the ``standard'' way of assigning these letters to different partition types.

2.2 Drive and partition labelling in Linux and FreeBSD

The hard drives are labelled in the following way in Linux and FreeBSD:

```
                        Linux           FreeBSD
First IDE drive         /dev/hda        /dev/wd0
Second IDE drive        /dev/hdb        /dev/wd1
First SCSI drive        /dev/sda        /dev/sd0
Second SCSI drive       /dev/sdb        /dev/sd1
```

The partitions (FreeBSD slices) on an IDE drive are labelled in the following way (`/dev/hda` is used as an example):

```
                                Linux           FreeBSD
First primary partition         /dev/hda1       /dev/wd0s1
Second primary partition        /dev/hda2       /dev/wd0s2
Third primary partition         /dev/hda3       /dev/wd0s3
Fourth primary partition        /dev/hda4       /dev/wd0s4
```

The partitions in my FreeBSD slice is labelled in the following way. It is the labelling you get by default. It is possible to change the labelling if you do a custom installation of FreeBSD (`/dev/hda4` is the FreeBSD slice in the example):

```
Linux label     FreeBSD label   FreeBSD mount point
/dev/hda5       /dev/wd0s4a     /
/dev/hda6       /dev/wd0s4b     swap
/dev/hda7       /dev/wd0s4e     /var
/dev/hda8       /dev/wd0s4f     /usr
```

If you run `dmesg` in Linux you will see this as (The linux kernel must be build with **UFS filesystem support** for this to work. See section Installing and preparing Linux):

```
Partition check:
 hda: hda1 hda2 hda3 hda4 < hda5 hda6 hda7 hda8 >
```

If you have installed FreeBSD in the `/dev/sd1s3` slice (`/dev/sdb3` in Linux parlace), and `/dev/sdb2` is a Linux extended partition containing two logical partitions (`/dev/sdb5` and `/dev/sdb6`), the previous example would look like this:

```
Linux label     FreeBSD label   FreeBSD mount point
/dev/sdb7       /dev/sd1s3a     /
```

```
/dev/sdb8        /dev/sd1s3b     swap
/dev/sdb9        /dev/sd1s3e     /var
/dev/sdb10       /dev/sd1s3f     /usr
```

This will be shown as

```
Partition check:
 sdb: sdb1 sdb2 < sdb5 sdb6 > sdb3 < sdb7 sdb8 sdb9 sdb10 >
```

in the output from `dmesg`.

If you have a Linux extended partition *after* your FreeBSD slice you're in for trouble, because most Linux kernels installation floppies are build without UFS support, they will not recognise the FreeBSD partitions inside the slice. What should have have been seen as (`/dev/hda3` is the FreeBSD slice and `/dev/hda4` is the Linux extended partition)

```
Partition check:
 hda: hda1 hda2 hda3 < hda5 hda6 hda7 hda8 > hda4 < hda9 hda10 >
```

is seen as:

```
Partition check:
 hda: hda1 hda2 hda3 hda4 < hda5 hda6 >
```

This can give you the wrong device assignment and cause the loss of data. My advice is to *always put your FreeBSD slice after any Linux extended partitions, and do not change any logical partitions in your Linux extended partitions after installing FreeBSD!*

---

3. Sharing swap space between Linux and FreeBSD

This section describes how I got Linux and FreeBSD to share a swap partition. There may be other ways to get the same result. You can install FreeBSD before Linux if you want to, just pay attention to the order of the partitions in the FreeBSD slice.

3.1 Installing and preparing Linux

The first step is to install Linux as normal. You have to leave space for the FreeBSD slice at your hard drive. You don't have to make a Linux swap partition, but if you want one, put it in the space you want to allocate for FreeBSD. That way you can delete the Linux swap partition later and use the space for FreeBSD.

When you have installed Linux you have to build a new kernel. Read **The Linux Kernel HOWTO** if this is new to you. You *have* to include both **UFS filesystem support (read only)** and **BSD disklabel (FreeBSD partition tables) support**:

```
UFS filesystem support (read only) (CONFIG_UFS_FS) [N/y/m/?] y
BSD disklabel (FreeBSD partition tables) support (CONFIG_BSD_DISKLABEL) [N/y/?]
(NEW) y
```

Install the new kernel and reboot. Remove any line including the word *swap* from your `/etc/fstab` file if you have made a Linux swap partition. *Make sure you have a working Linux boot floppy with the new kernel.* Now you are ready to install FreeBSD.

3.2 Installing FreeBSD

Install FreeBSD as described in the FreeBSD documentation. Remove the Linux swap partition if you have made one (you can use the FreeBSD `fdisk` program.) Pay attention to the order of the partitions in the FreeBSD slice. If you use the default labelling the second partition will be the swap partition. Complete the installation of FreeBSD and reboot into Linux *using the new Linux boot floppy*.

3.3 Setting up the FreeBSD swap partition in Linux

Run `dmesg` when you have booted into Linux. In the output you should see something like this:

```
Partition check:
 hda: hda1 hda2 hda3 hda4 < hda5 hda6 hda7 hda8 >
```

This means that `/dev/hda4` is your FreeBSD slice, while `/dev/hda5`, `/dev/hda6`, `/dev/hda7` and `/dev/hda8` are the FreeBSD partitions. If your swap partition is the second partition in the slice, it will be `/dev/hda6`.

You have to put the following line into your Linux `/etc/fstab` file to enable the swap partition:

```
/dev/hda6       none            swap            sw              0       0
```

While FreeBSD can use any type of partition as swap space, Linux needs a special signature in the swap partition. This signature is made by `mkswap`. FreeBSD ruins this signature when it uses the shared swap partition, so you have to run `mkswap` each time you boot into Linux. To do this automagically you have to find the script that runs `swapon` at boot time. In Red Hat Linux it is `/etc/rc.d/rc.sysinit`. Put the following line into that file just *before* `swapon -a`:

```
awk -- '/swap/ && ($1 !~ /#/) { system("mkswap "$1"") }' /etc/fstab
```

This will run `mkswap` on any swap partitions in `/etc/fstab` every time you boot except if they are commented out (having ``#'' as the first character in the line.)

Run `free` to check out the size of the swap space when you have rebooted into Linux. You should also reboot into FreeBSD to make sure everything works as expected. If it does not, you have probably used the wrong partition as swap partition. The only solution to that problem is to reinstall FreeBSD and try again. Experience is a great teacher. :-)

---

4. Booting FreeBSD using LILO

You can easily boot FreeBSD with LILO. Do not install the FreeBSD boot selector (`Booteasy`) if you want to use LILO. Append the following lines to your `/etc/lilo.conf` file and run `lilo` (assuming the FreeBSD slice is `/dev/hda4`):

```
other=/dev/hda4
        table=/dev/hda
        label=FreeBSD
```

If you have installed FreeBSD on the second SCSI drive, use something like this (the FreeBSD slice being `/dev/sdb2`):

```
other=/dev/sdb2
        table=/dev/sdb
        loader=/boot/chain.b
        label=FreeBSD
```

5. Mounting filesystems

5.1 Mounting UFS filesystems under Linux

Unfortunately the UFS driver in the Linux 2.0.xx kernels do not include support for FreeBSD. When you try to mount a FreeBSD filesystem, you just get some error messages (the filesystem actually gets mounted, but you cannot do anything with it.) This problem has been solved in the newer Linux kernels (version 2.1.87 and higher.)

There is another version of the UFS driver for Linux 2.0.xx kernels (xx <= 30) on metalab.unc.edu (the former sunsite.unc.edu). It is called **U2FS** and the current version is `u2fs-0.4.3.tar.gz`. A version of U2FS (`ufs-0.4.4.tar.gz`) for Linux 2.0.31 and higher (2.0.xx; not 2.1.xx) can be found at this site along with further information about U2FS (and UFS.)

Now you have to build a new kernel with support for the U2FS filesystem and BSD disklabels. See section Installing and preparing Linux for more information on this. You can leave out **UFS filesystem support** from the kernel when you use U2FS.

When you have installed the new kernel, you can mount your UFS filesystems (all the partitions in the FreeBSD slice except the swap partition) with a command like this:

```
mount -t u2fs /dev/hda8 /mnt
```

You should use a command like

```
mount -t ufs /dev/hda8 /mnt
```

if you use a Linux kernel version 2.1.87 or higher. From Linux kernel version 2.1.112 you must add `-o ufstype=44bsd` to the command like this:

```
mount -t ufs -o ufstype=44bsd /dev/hda8 /mnt
```

The UFS (and U2FS) driver is read-only. That is; you can read from the UFS filesystems but you cannot write to them. An experimental read-write UFS driver has replaced the read-only driver in Linux kernels version 2.1.112 and higher; writing to FreeBSD partitions is supported from version 2.1.127.

5.2 Mounting ext2fs filesystems under FreeBSD

To mount ext2fs filesystems under FreeBSD, you first have to build a new kernel with ext2fs support. Read the FreeBSD handbook to learn how to do that. Put the line

```
options         "EXT2FS"
```

in your kernel configuration file for the new kernel.

When you have booted with the new kernel, you can mount an ext2fs filesystem by giving a command like:

```
mount -t ext2fs /dev/wd0s3 /mnt
```

Note that you cannot mount ext2fs filesystems in extended partitions from FreeBSD.

Due to a bug in FreeBSD 2.2.8 and earlier you will have to unmount all ext2fs filesystems *before* you shut down FreeBSD if you are using these versions. If you shut down FreeBSD with an ext2fs filesystem mounted, FreeBSD cannot sync the UFS filesystems. This results in `fsck` being run the next time FreeBSD is booted. You can work around this bug by putting the line:

```
umount -a -t ext2fs
```

in the `/etc/rc.shutdown` file. The bug should have been fixed in FreeBSD-3.x (can anybody confirm this?)

---

6. Running foreign binaries

6.1 Running FreeBSD binaries under Linux

The `iBCS` package has support for running FreeBSD binaries under Linux; but it's old and unmaintained. I cannot get it to work. Please let me know if you have had better luck with this.

6.2 Running Linux binaries under FreeBSD

FreeBSD has the ability to run Linux binaries, both in a.out and ELF formats. To do this you have to take the following three steps:

1. You have to enable Linux compatibility. To do this (in FreeBSD 2.2.2 --- details may vary in other versions) you have to edit your `/etc/rc.conf` file and change

   ```
   linux_enable="NO"
   ```

   to

   ```
   linux_enable="YES"
   ```

   and reboot. Another way to load the Linux binary support is to execute the command `/usr/bin/linux`. This way you don't have to reboot, and you don't always have the Linux binary support loaded (i.e. you save memory.) Remember to add the line

   ```
   options         COMPAT_LINUX
   ```

   to the FreeBSD kernel config file if you build a new FreeBSD kernel.
2. You have to install the Linux shared libraries if your Linux binaries are dynamically linked. The libraries are included in FreeBSD 2.2.{2,5,6} as the package `linux_lib-2.4.tgz` (newer versions might be available.) Run the following command to install the package:

   ```
   pkg_add <path_to_package>/linux_lib-2.4.tgz
   ```

   <path_to_package> is the directory where the package is stored. You may also load it off the net by:

   ```
   pkg_add ftp://ftp.freebsd.org/pub/FreeBSD/packages-stable/All/linux_lib-2.4.tgz
   ```

   or by re-running `/stand/sysinstall`. Enter ``Configure'', ``Packages'' and use the menus. You should execute the following command if you are running statically linked Linux binaries:

   ```
   brandelf -t Linux <name_of_statically_linked_linux_binary>
   ```
3. Install the Linux program(s) you want to run. The program(s) can be installed on either UFS or ext2fs filesystems. See section Mounting ext2fs filesystems under FreeBSD for more information about using ext2fs filesystems under FreeBSD.

I have successfully run the Linux versions of Applixware 4.3 and Netscape 3.01 (both ELF format) under FreeBSD 2.2.2 using this method (yes, I know there is a native FreeBSD version of Netscape 4.) The Linux versions of acroread and StarOffice 3 and 4 also work well under FreeBSD. StarOffice 5 depends on native Linux threads and currently do not work under FreeBSD. Read the FreeBSD documentation for more information on this topic.

7. Information resources

The latest version of this mini-HOWTO can be downloaded from this site in several formats (including SGML and PostScript.) The document has been translated into Japanese by Mr. Teruyoshi Fujiwara as part of the JF project.

Gregor Hoffleit maintains a site with information about the developement of U2FS and UFS.

Annelise Anderson is working on a Linux+FreeBSD FAQ, which will cover the same subjects as this mini-HOWTO (and more.) The FAQ is intended to be part of the freebsd-doc project.

You can find some articles about the difference between Linux and FreeBSD here.

You can find more informations about FreeBSD (and download the whole system) at this site. You can also buy the system on CDROMs from Walnut Creek CDROM (their servers are running FreeBSD.)

The Linux Kernel HOWTO (and this mini-HOWTO) is released as part of The Linux Documentation Project.

8. Credits and legal stuff

Thanks to the members of the *BSD user group in Denmark for answering the questions of a FreeBSD newbie, to Mr. Takeshi Okazaki for bringing the existence of U2FS to my attention, and to Mr. David O'Brien for valuable suggestions.

8.1 Legal stuff

Trademarks are owned by their owners.

Although the information given in this document is believed to be correct, the author will accept no liability for the content of this document. Use the tips and examples given herein at your own risk.

Copyright © 1997, 1998, 1999 by Niels Kristian Bech Jensen. This document may be distributed only subject to the terms and conditions set forth in the LDP license.

NT OS Loader + Linux mini-HOWTO

Bernd Reichert, <reichert@dial.eunet.ch>

v1.11, 2 September 1997

1. Abstract

2. How does the NT OS Loader work

3. Tips on how to install Windows NT and Linux on the same System

- 3.1 Windows NT installation
- 3.2 Partitioning
- 3.3 Linux installation
- 3.4 Lilo

4. Bootpart

5. The Linux part of the work

6. The Windows NT part of the work

7. Play it again Sam

8. Troubleshooting

TOC

9. References

10. Acknowledgements

11. Feedback

---

1. Abstract

This document describes the use of the Windows NT boot loader to start Linux. This procedures have been tested with Windows NT 4.0 WS and Linux 2.0.

2. How does the NT OS Loader work

The NT OS loader likes to have the boot sector from the other operating systems available as a file. It reads this file and starts the operating system selected, i.e. either Windows NT in different Modes or any other OS.

3. Tips on how to install Windows NT and Linux on the same System

3.1 Windows NT installation

Try to install Windows NT first. If you want to use NTFS for your Windows NT-partition, keep in mind, that the todays production version of Linux cannot access NTFS partitions. An alpha driver that can read NTFS-Partitions is available at `http://www.informatik.hu-berlin.de/~loewis/ntfs`. You may create a separate FAT-Partition for data exchange or you have to use DOS-formatted floppies.

3.2 Partitioning

Another Mini-howto recommends not to use NTs "Disk Administartor" to create the Linux swap- and root-partitions. It is sufficient to see the free space there. I also recommend to use linux's fdisk later.

3.3 Linux installation

Now boot linux from diskettes, create the swap and root-partition. Fdisk assumes the ntfs-partition to be a HPFS-partition. This is normal. Boot again from diskettes and install Linux as you like.

Just in case the installation procedure suggests that you could mount the HPFS partition which it has found: Ignore it.

3.4 Lilo

When you come to the Lilo-Section, specify your Linux-root-partition as your boot device because the Master Boot Record (MBR) of your harddisk is owned by Windows NT. This means that the root-entry and the boot-entry in your `/etc/lilo.conf` have the same value. If you have a IDE-harddisk and your Linux-partition is is the second partition, your boot-entry in `/etc/lilo.conf` looks like:

```
boot=/dev/hda2
```

If you have two disks and your Linux resides on the first partition of your second disk, your boot-entry in `/etc/lilo.conf` looks like:

```
boot=/dev/hdb1
```

Run lilo with a kernel that matches your system. Check the kernel by booting from diskette first if you are not sure.

If you cannot boot Windows NT now, you have a problem. I hope you have created a repair-disk recently.

4. Bootpart

There is a NT-programm called bootpart written by G. Vollant that can do the jobs from the next two points for you. Bootpart is available at `http://ourworld.compuserve.com/homepages/gvollant/bootpart.htm`.

If you want to know how things are working together, use the procedures described in point 5 and 6.

5. The Linux part of the work

You have to boot from diskettes until the NT-part is fixed.

Now you have to peel the bootsector from your Linux-root-Partition. With /dev/hda2 as your linux-partition, the dd-command is:

```
# dd if=/dev/hda2 of=/bootsect.lnx bs=512 count=1
```

There is something wrong if your bootsect.lnx has more than 512 bytes.

Now copy the file bootsect.lnx to a DOS-formated floppy if this is your way to transfer files to the NTFS-Windows-partition.

You can copy it with

```
# mcopy /bootsect.lnx a:
```

or with

```
# mount -t msdos /dev/fd0 /mnt
# copy /bootsect.lnx /mnt
# umount /mnt
```

6. The Windows NT part of the work

Copy the file from the diskette to `C:\\bootsect.lnx`. I don't tell you how to do that.

What lilo.conf is for linux is `c:\\boot.ini` for Windows NT. Remove the, system- and the read-only-attribute before you can modify it with:

```
C:\attrib -s -r c:\boot.ini
```

Now change the file boot.ini with an editor, notepad for example, as follows:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows NT Workstation ...
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows NT Workstation ...
C:\BOOTSECT.LNX="Linux"
```

Only the last line has been added in this example. Restore the attributes after you have saved `boot.ini` with:

```
C:\attrib +s +r c:\boot.ini
```

After a shutdown of your Windows NT and a restart your should see the following:

```
  OS Loader V4.00

  Please select the operating system to start:

  Windows NT Workstation Version 4.0
  Windows NT Workstation Version 4.0 [VGA mode]
  Linux

Select Linux and see

  LILO loading zImage ....
```

7. Play it again Sam

A new copy of bootsect.lnx must be transfererd to `C:\BOOTSECT.LNX` evry time the bootsector of your linux-partition has been modified. This happens for example when you install a new kernel with lilo. As you can see such a system in not ideal for testing experimental kernels.

8. Troubleshooting

If things do not work as expected, check bootability with a floppy disk. With `/dev/hdb1` as your Linux-partition, your `/etc/lilo.conf` has the following entries:

```
root=/dev/hdb1
boot=/dev/fd0
```

Run lilo with a diskette inserted. Now try to boot from the diskette. If your Linux on `/dev/hdb1` can't be started, the NT OS loader will also fail to start it. If you see a lot of `01 01 01 01`, your root-disk is not accessible. Check whether all your disks are known by the bios.

If the floppy boots your Linux partition, you can peel of your boot-sector for the NT OS loader with:

```
# dd if=/dev/fd0 of=/bootsect.lnx bs=512 count=1
```

You can keep this diskette as your rescue-diskette just in case your Windows NT installation breaks.

9. References

- The Linux+WindowsNT mini-HOWTO
- The FAQ for FreeBSD 2.X

10. Acknowledgements

- Thanks to Xiaoming Yi <z3c20@ttacs.ttu.edu> for the tip that the procedures work also with other disks than the first one.
- Thanks to Frank Dennler <Frank.Dennler@zkb.ch> for the diskette-trick.

11. Feedback

Any comments are welcome.

Authored 11/95 by Jonathan Katz. Fixed/revised April 10, 1996, not to mention slightly updated Jun 1996 and "completed" October 26, 1996, all by Jonathan Katz, jkatz@in.net

TOC

1 Introduction

Many people have recently bought new machines, which (unfortunatly) come pre-loaded with Windows 95. After exploring the Internet, many users find the likes of Linux, and want to run a better OS at a much better price. There are many different issues to address, because new computers usually come with large EIDE hard drives, and Windows 95 has extentions on the old FAT filesystem. Linux understands these issues and can handle these challenges in hardware and software.

2 Let's get Ready to Rumble!

Linux and Windows 95 can get along quite well on the same hard disk. You can also install Linux onto a separate hard disk on the same machine. If you have the money to spare to get a second hard disk, go ahead and do that. Although it is safe and reliable to run Windows 95 and Linux on the same hard disk, it is safer to have a second disk. But, since I am a poor student, (and so are most of the people I know) we are stuck with one large disk.

I assume your hard disk looks like this:

```
        +---------------------------------
C:      |  800M or so, maybe bigger?
        |
        +---------------------------------
```

and Windows 95 and MS-Office and Qmodem Pro, and whatever DOS/Windows software you have only takes up around 400M + Swap (this is a big, sarcastic assumption)! I take it you don't want to lose all this software you've spent a lot of time configuring and installing, so you don't want to delete this partition and restart all over again.

3 Ok, I have this partition I want to spare!

Don't lose hope. There is a program called FIPS, which can re-partition your Hard Disk without destroying data. HOWEVER, make sure before you use it, you defrag your hard disk (with the optimum defrag method). Use the defrag that came with Win95 and use it in the GUI-- otherwise you will loose your long file names. After you are all defraged, run FIPS and make your disk look something like the following:

```
        +-------------------------------------
C:      | This is your FAT/VFAT/Win95 partition
 450M   |
        +-------------------------------------
???     | This is empty space that starts BELOW the 1024th
        |       cylander
        +-------------------------------------
```

FIPS can be found at your favorite Linux FTP sites (sunsite.unc.edu, tsx-11.mit.edu, ftp.redhat.com)

usually in the /pub/utils/msdos directory. If you have a CD for Linux, there is usually a \utils\msdos or \utils directory that has FIPS in it as well.

It is **VERY IMPORTANT** that your Linux partition start before (below) the 1024th cylander, otherwise you will be unable to boot it (and that's not a good thing).

What exactly is the 1024th cylander? That's where IDE ends and EIDE begins-- that's the 528M "mark" on your hard disk. Start your Linux partition at around 520M, so the entire kernel and other boot/loader files will completely reside below that cylander.

4 What Next?

Go ahead and install Linux to that new free space. If you are paranoid, when the Linux install proceedure goes to boot, you can boot into Windows 95 and mak sure it still runs OK. PLEASE, PLEASE, PLEASE listen to the advice given in the tail of section number three. You will be unable to use Linux otherwise.

When it comes to piecing out the disk, if you are like most people, you'll make a 400M Linux partition (which is pretty comfortable) as well as 30M of SWAP (which is more than enough).

At this point, you should be able to install whatever distribution of Linux you have without any troubles.

5 Using your new system!

Linux can mount, read, and write to Win95's VFAT partitions serveral ways. You can use the stock msdos filesystem support that has been included in all kernels greater than 1.0. However, using commands like:

```
litterbox~#: mount -t msdos /dev/hda1 /mnt
```

will only get you as far as filenames with the 8.3 standard. Yick! You have Windows 95 so you can use those nifty long file names.

Some Linux users still run kernels that are anchient (by Linux standards). In other words, 1.2.xx kernels. Someone coded a module for this series of kernels so a user can read files that do not conform to the old 8.3 standard. FTP to ftp://mm-ftp.cs.berkeley.edu/pub/multimedia/linux/xmsdos/ and pick up a copy of README before doing anything.

Finally, newer kernels (1.3.4x) as well as the long awaited 2.0 RELEASE have internal vfat support that you can compile in. These kernels allow safe read/writes to your VFAT partitions.

NOTE: A word of caution!

IF you are running a "newer" version of Windows 95-- i.e., possibly one that ships on computers made after June 1996 as well as the P5-MMX series please listen to the following.

Check to see what version of Win95 you are really running. To do this, open up a DOS box and type: "ver /r". Here's some sample output:

```
C:\> ver /r

Mircosoft Windows 95 [4.00.1034]
```

```
(C) 1981-1996 Microsoft Corporation.
```

Note the minor revision number. If the number is GREATER than 950 you may be running a version of the FAT partition type known as FAT32. If that is the case, you can still use Linux and Win95, HOWEVER, there is NO support to read/write a FAT32 partition. You are SOL if that is the case.

Just because you are running a newer version of Win95 doesn't mean that you are running FAT32. Load up "fdisk" and use the "display partition information" option to show indeed if you are using FAT32.

5.1 HELP!, I'm stuck with FAT32!

5.1.2.1 I have this brand new box that I got from (unamed chain/retail store)

If you are installing Linux onto a drive with FAT32 follow the steps from above about installing Linux onto a Win95 system, but do NOT install LILO. Instead, skip to section 5.1.3 entitled "linload, what's that?" and read the step-by-step workaround for this unfortunate situation.

5.1.2.2 I messed up-- big time!

This happened to me-- I was running a Win95 beta with FAT32 installed, I upgrade Win95, and my system died. I didn't have a Win95 boot disk for the version of Win95 I just upgraded to. Here's a little work around.

Find ANY MS boot disk made using DOS version 5.0 or greater. Under Linux mount the Win95 Beta CD. Mount the floppy as well.

```
litterbox~# mount -t iso9660 /dev/cdrom /cdrom
litterbox~# mount -t msdos /dev/fd0 /mnt
```

On the CD find the "beta stuff" directory. The MS people move this location around... try /cdrom/admin/beta/floppy, /cdrom/betaonly/floppy. cp the files io.sys, msdos.sys, and command.com to the floppy. "sync" and unmount the disks you mounted. Do a reboot and that boot disk *should* work.

5.1.3 Linload, What's That?

Linload is a DOS execuatble which loads a kernel image from a DOS HD (or floppy) and then boots the rest of the Linux Operating System from an appropriate root partition. If you're really a good hacker, you can do a floppyless Linux install this way, but today we are going to do more traditional uses. When booting into Windows 95 hit the F8 key (when:

```
Starting Windows 95...
```

is displayed) and select "Safe mode, command prompt only." Go to your directory where you put your kernel and run

```
C:\LINUX> linload.exe zimage root=/dev/hda2 ro"
```

from there. If you are any good with MS-DOS setups, you can code multiple AUTOEXEC.BAT and CONFIG.SYS files, so that you have a menu setup and can choose which OS to boot.

6 Problems?

TOC

In the reverse of the begining scenario (you have a working Linux box and are forced by some power install Windows 95) Windows 95 WILL overwrite your MBR, and thus crush LILO. You have two options: use loadlin.exe to load your kernel, and use that to run Linux, or boot Linux using a floppy and/or loadlin, then re-install LILO.

Best of luck with your new system!


```
-Jon
jkatz@mac.edu
+1 217 479-7309
Box 4454 MacMurray College
Jacksonville, Illinois 62650
Personal: http://www.in.net/~jkatz
Corinne:  http://corinne.mac.edu
Resume:   http://www.in.net/~jkatz/resume.html
Room-101: http://room-101.1984.org
HOWTO:    http://www.in.net/~jkatz/win95/Linux-HOWTO.html
```

The Loadlin+Win95 mini-HOWTO

Authored by: Chris Fischer, protek@brigadoon.com

v1.4.6, 13 March 1999

This document describes how to use Loadlin with Windows 95 to boot to Linux.

1. Introduction

- 1.1 Copyright
- 1.2 New Versions Of This Document and Windows 98
- 1.3 Feedback
- 1.4 Disclaimer

2. IMPORTANT: Preliminary Information and Setup

- 2.1 WARNING! for Windows95 OSR2 (version 4.00.950b) users:
- 2.2 Things that are assumed:
- 2.3 Items you need before you start:

3. Getting started:

- 3.1 Determine what version of Windows 95 you have.
- 3.2 Determine if you are using the FAT32 file system.
- 3.3 Make sure filename extensions are visible in Windows Explorer.
- 3.4 Rename your system files.
- 3.5 Important Warning.
- 3.6 Edit your system files.

TOC

4. *** Use this section ONLY if you are using a DOS 6.x boot disk.***

- 4.1 Edit and Rename your DOS 6.x system files.

5. Final Steps

- 5.1 Almost Finished
- 5.2 You are finished!
- 5.3 How it works.

6. Alternate Methods for Auto-Booting Linux, Especially for Users With a FAT32 File System.

- 6.1 Method 1.
- 6.2 Method 2.

7. Frequently Asked Questions:

- 7.1 Where can I find more information on using Loadlin?
- 7.2 I am not sure what partition Linux is installed on. How do I find out?
- 7.3 Where do I find the kernel image file and how do I copy it to my MS-DOS partition?
- 7.4 Does it matter if I install Linux or Windows 95 first?
- 7.5 I currently have LILO installed, but I want to use Loadlin instead. How do I remove LILO?
- 7.6 What if I already dual boot between Windows 95 and Windows 3.x, and I want to boot to Linux using this method?
- 7.7 Is it possible to initiate Linux from the Windows 95 desktop?
- 7.8 Where can I get a plain text version of this document?

8. Credits

---

1. Introduction

This document describes how to easily use a combination of Windows 95's Boot Menu and Loadlin.exe to autoboot Linux using the F4 Function Key. This procedure should work with Windows 95 versions 4.00.950, 4.00.950a, and 4.00.95b with FAT16, and all versions of Linux that support Loadlin. If you prefer to use the LILO boot manager, please consult the Linux+Win95 Mini-HOWTO.

There are also two alternate methods for dual booting between Linux and Windows 95 using Loadlin for users using FAT32 with Windows 95 version 4.00.950b.

1.1 Copyright

The Loadlin+Win95 mini-HOWTO is copyright (C) 1997- 1998 by Protek Computer Solutions. This document may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would *like* to be notified of any such distributions.

1.2 New Versions Of This Document and Windows 98

This mini-HOWTO is posted first at The Linux+Windows 95 Reference Page, so check there to make sure you have the most recent revision of this mini-HOWTO. You will also find a link there for dual booting with Windows 98. The next revision of this mini-HOWTO will be a major revision and will include information on dual booting with Windows 98.

1.3 Feedback

Questions, comments, and suggestions are welcome. I am always looking for ways to improve and expand this mini-HOWTO. I rely heavily on feedback to make improvements, and will do my best to be prompt with a helpful response. I can be reached at protek@brigadoon.com.

1.4 Disclaimer

Every attempt has been made to ensure that the information presented in this mini-HOWTO is safe and accurate. However, this information is given without any warranty, either expressed or implied, as to its suitability for a particular use. It is generally considered a GoodThing(TM) to make backups of your system files before changing system configurations and/or files. I suggest you take this precaution "just in case".

2. IMPORTANT: Preliminary Information and Setup

2.1 WARNING! for Windows95 OSR2 (version 4.00.950b) users:

Do **NOT** attempt dual booting with Windows 95 without first consulting the Windows 95 OSR2 FAQ. The FAQ also contains a great deal of information regarding FAT32. In a nutshell, if you have OSR2 and are using a FAT16 file system, you can still dual boot using the F4 Function Key.

NOTE: If you have Windows 95 OSR2, and are using a FAT32 file system, then the dual booting methods in this mini-HOWTO using the F4 Function Key *will not work and should not be attempted.*

However, there are a couple of alternate methods you can use to automate booting Linux at startup, which are discussed in Section 6.

2.2 Things that are assumed:

- You have successfully installed Windows 95.
- Windows 95 is installed on Drive C.
- You have successfully installed Linux.
- You know what partition Linux is installed on.
- LILO is NOT installed on your hard drive.
- If you are using Windows 95 OSR2 (version 4.00.950b), you have carefully read the Windows95 OSR2 FAQ.

2.3 Items you need before you start:

- A bootable floppy disk with DOS 5.0 or DOS 6.x system files (io.sys, msdos.sys, command.com) and a config.sys file. If you are using a DOS 5.0 boot disk, you will also need an autoexec.bat file. If you are using Windows 95 OSR2 (version 4.00.950b), you will also need to create a file called Winboot.sys with a text editor and place it in the root directory of Drive C. Winboot.sys can be an empty file, just so long as you have the file.

NOTE: If your hard drive already contains the autoexec.dos, config.dos, io.dos, msdos.dos, and command.dos files, then you will not need the floppy. You can check by typing: `dir` **and** `dir /ah` at the root of Drive C. Also, if you installed the upgrade version of Windows 95, then it is likely that you have them.

TOC

- The Loadlin.exe program file
- Your kernel image file,usually zImage or vmlinuz. (See the FAQ for info).
- A simple text editor, such as Notepad.

---

3. Getting started:

3.1 Determine what version of Windows 95 you have.

Boot your computer to Windows 95 and right click on the My computer icon. Now left click on Properties. On the General Tab, look under the "System:" line. There you will see the version number.

3.2 Determine if you are using the FAT32 file system.

NOTE: If you have Windows 95 version 4.00.950b, *make certain you are **not** using the FAT32 file system*. Double click the My Computer icon, then right click the Drive C icon. Left click on Properties. On the General tab, the "Type:" line will indicate whether or not you have a FAT32 file system. **If you are using the FAT32 file system, then go directly to Section 6** . Otherwise, continue with **Section 3.3**.

3.3 Make sure filename extensions are visible in Windows Explorer.

- Run the Windows Explorer file manager program.
- On the Menu Bar, click on View, then click on Options. On the View Tab, make sure that the "Show all files" radio button is marked. Also, make sure that the line with "Hide MS-DOS file extensions for file types that are registered" is unchecked. Click on the OK button.

NOTE: If your hard drive already contains the *.dos files mentioned in the NOTE: of Section 2.3, then skip to the next NOTE:. Otherwise, continue with **Section 3.4**.

3.4 Rename your system files.

- Insert the DOS 5.0 or DOS 6.x bootable floppy into Drive A. Click on the Drive A icon so you can see the contents on the floppy disk.
- Rename each file to a .dos file extension. You can do this by single clicking on a file, pressing the F2 function key, and typing in the new name. Do not worry if you need the bootable floppy to work properly, just rename the files back after you are finished with this procedure.

3.5 Important Warning.

- **EXTREMELY IMPORTANT:** Make sure you rename the files on the floppy **BEFORE** you copy them to Drive C. Otherwise you will overwrite your Windows 95 system files. *You have been warned*. Now, copy all files to the root directory of Drive C. One method is to click on the Drive A icon, press CTRL+A, press CTRL+C, click on the Drive C icon, and press CTRL+V.

NOTE: If you **do not** have Windows 95 OSR2 (version 4.00.950b), please skip to Section 3.6.

- If (and only if) you have Windows95 version 4.00.950b, make sure you have a file named Winboot.sys in the root directory of Drive C, as stated in Section 2.3 above. **Failure to have Winboot.sys in the root directory of Drive C will result in a system lockup should you attempt to boot to an earlier version of DOS.**

NOTE: If you are using a DOS 6.x boot disk skip to Section 4, otherwise continue with the next step.

3.6 Edit your system files.

- Open Notepad. On the Menu Bar, click on Open. Change "Files of type" to "All files (*.*)". Change "Look in:" from Desktop to Drive C. Now open Config.dos by double clicking on the filename. Delete all the contents. This will create a 0K file...I will explain later. Save the file.
- Use Notepad to open and edit Autoexec.dos and type in the Loadlin command. The Loadlin command must be in the following syntax:

```
Loadlin DriveLetter:\LinuxKernelFile root=/dev/LinuxBootPartition ro
```

Here is my Autoexec.dos file as an example:

```
Loadlin f:\vmlinuz root=/dev/hdc2 ro
```

NOTE: The kernel, as well as the Loadlin.exe, file may be anywhere on your hard drive as long as you reference the correct path. For example, if your kernel file was located in the directory e:\linux\kernels and Loadlin.exe was located in the directory c:\utils, you would use:

```
c:\utils\loadlin e:\linux\kernels\vmlinuz root=/dev/hdc2 ro
```

- Save the file.

4. *** Use this section ONLY if you are using a DOS 6.x boot disk.***

4.1 Edit and Rename your DOS 6.x system files.

- Open Notepad. On the Menu Bar, click on Open. Change "Files of type" to "All files (*.*)". Change "Look in:" from Desktop to Drive C. Now open Config.dos by double clicking on the filename. Delete all the contents. Now, you will need to enter the shell command with the following syntax:

```
shell=loadlin.exe LinuxKernelFile root=/dev/LinuxBootPartition ro
```

Here is my Config.dos file as an example:

```
shell=f:\loadlin.exe f:\vmlinuz root=/dev/hdc2 ro
```

NOTE: The Loadlin and kernel file may be anywhere on your hard drive as long as you reference the correct path. For example, if your Loadlin file was in the directory f:\utils and the kernel file was located in the directory e:\linux\kernels you would use:

```
shell=f:\utils\loadlin.exe e:\linux\kernels\vmlinuz root=/dev/hdc2 ro
```

- Save the file.

5. Final Steps

5.1 Almost Finished

- One last file to edit and then you are finished. Use Notepad to open and edit Msdos.sys. Goto the "Options" section. Look for a line that has BootMulti=. If you do not find one then add it and set the value to 1. It should look something like:

```
[Options]
BootMulti=1
```

- There may be other items listed. Just leave them. Now save the file.

5.2 You are finished!

Now all you have to do is reboot (Don't forget to remove the floppy disk). When you see "Starting Windows 95...", press the F4 Function Key and Linux should boot right up.

5.3 How it works.

Windows 95 has a built in Boot Menu. By default you do not see it at startup (although you can change that if you wish), but you can get to it by pressing the F8 Function Key when you see "Starting Windows 95...". On the menu you will see several options. The last option is "Previous version of MS-DOS". Instead of pressing F8 and then selecting "Previous version of MS-DOS", you can use the F4 shortcut by pressing F4 when "Starting Windows 95..." appears on the screen. When Windows 95 boots to the previous version of MS-DOS, it uses the files that end in the .dos file extension. This is why it is important to have a 0K Config.dos file if you are initiating Linux from the Autoexec.dos file. If Windows 95 does not find the files with a .dos extension while trying to boot the previous operating system, it will load the Windows 95 startup files. Therefore, if you do not have a Config.dos file, Windows 95 will load all the drivers in your Config.sys file. While this will not prevent Linux from loading, it will take longer. Having a 0K Config.dos file prevents Windows 95 from reading your Config.sys file while booting to the previous version of MS-DOS.

6. Alternate Methods for Auto-Booting Linux, Especially for Users With a FAT32 File System.

This section will work for all versions of Windows 95 to date, regardless of whether you have a FAT32 file system. There are two methods I will discuss in this section. First, I'll start with the simplest.

6.1 Method 1.

This method uses the Autoexec.bat file to call (or execute) another batch file named Linux.bat, during boot up.

- Creat a Linux.bat file using a simple text editor such as EDIT or Notepad. The contents should be similar to the following:

```
@echo off
cls
echo.
echo.
echo.
echo.
choice /t:y,5 "Do you wish to boot Linux? "
if errorlevel 2 goto End
c:\loadlin c:\vmlinuz root=/dev/hdc2 ro
:End
```

This batch file script clears the screen, adds four blank lines, displays the text in quotes + [Y,N], then waits 5 seconds for you to press a key. If you do not press a key within 5 seconds, it defaults to Y and runs Linux. If you select Y or N, the batch file determines what option you selected, then executes the selected option. For example, if you select 'N' then the batch file terminates and continues to process your Autoexec.bat file, which loads Windows 95. If you select Y, then of course Linux will load.

If you do not want four blank lines before the text, modify the number of lines with `echo`. If you do not want the screen to clear, then remove the line with `cls`. The `/t` switch tells the choice command to wait 5 seconds and, if no key is pressed, to default to Y. Change the `y` to an `n` if you want Windows 95 to boot after the 5 second timeout: `choice /t:n,5`. Also, you can change the amount of time to wait from 0 to 99 seconds. For more information on the *choice* command, change to the c:\windows\command directory and type: `choice /?` at the command prompt.

NOTE: You will have to modify the line that boots Linux to match your configuration. Please see Section 3.6 for examples of how to configure Loadlin to boot to Linux.

- Next, create an Autoexec.bat file (if you don't already have one) with a simple text editor. Then, on the **first** line, add the following:

```
call c:\linux
```

If Linux.bat is in another directory you will have to reference the correct path. For example if Linux.bat is in a directory called C:\batch, you would add:

```
call c:\batch\linux
```

to the first line of your Autoexec.bat file. Save and exit the file. Now all you need to do is reboot. You should be prompted on whether you want to boot Linux.

TOC

6.2 Method 2.

This method is a bit more involved, but offers the greatest flexibility. What this method does it create your own custom boot menu, though it does not replace the Windows 95 boot menu. You will need a Config.sys and an Autoexec.bat file.

- First you will need to define a Startup Menu in Config.sys (mycomments to you are in parenthesis):

```
 [menu]
 menuitem=Linux, Boot to Linux  (This defines a Menu Block and gives it a
description).
 menuitem=Win95, Boot to Windows 95
 menucolor=15,1  (This gives a blue background with bright white text)
 menudefault=Linux, 15  (This sets the default menuitem and waits up to 15 seconds
for input).

 [linux]
 shell=f:\loadlin.exe f:\vmlinuz root=/dev/hdc2 ro (Please see Section 4 for examples
and syntax).

 [win95]
 (Include the normal contents of your config.sys file here. If you did
 not have a config.sys file before now, then leave this section blank).
```

- Save and exit your Config.sys file. If you want, you can use this Config.sys template to get started.
- Next, edit your Autoexec.bat file (my comments to you are in parenthesis):

```
 goto %config%

 :win95
 (Include the normal contents of your autoexec.bat file here. If you did
 not have an autoexec.bat file before now, then leave this section blank).
```

- Save and exit your Autoexec.bat file. If you want, you can use this Autoexec.bat template to get started. That should do it. The next time you reboot, you should get a Start Up menu with the option to boot to Linux or Windows 95.

7. Frequently Asked Questions:

7.1 Where can I find more information on using Loadlin?

- Slackware versions 3.2 and 3.5 contain a file named "loadlin.tgz" in the \slakware\A5 directory, Slackware versions 3.3 and 3.4 have loadlin.tgz in the \slakware\A4 directory. Slackware version 3.6 has loadlin.tgz in the \slackware\A8 directory. Unarchive the file and look in the \docs directory for a file named manual.txt. I do not know about earlier versions of Slackware.
- RedHat versions 4.2 through 5.2 contain a file named "loadlin16.tgz" in the \dosutils directory. Unarchive the file and look in the \docs directory for a file named manual.txt. RedHat 4.1 does not seem to have this archive. The manual.txt is also available for download at:
  `ftp://ftp.eskimo.com/u/p/praxis/manual.txt`
- Additional information is available at:
  `http://metalab.unc.edu/LDP/HOWTO/BootPrompt-HOWTO-2.html#ss2.2`

7.2 I am not sure what partition Linux is installed on. How do I find out?

- At a Linux shell prompt, run the `fdisk` utility and press `"p"`. From there you should be able to tell. If you need additional help using fdisk, please consult the man pages.

7.3 Where do I find the kernel image file and how do I copy it to my MS-DOS partition?

- To find your linux kernel file type:

```
find / -name vmlinuz
```

  at a Linux shell prompt. This will search all Linux partitions for the vmlinuz file. If you have multiple vmlinuz files, then make sure you use the correct one. If you are not sure, then the safest bet would be to use the most recent one.

  To copy your linux kernel file to your DOS partition, you need to make your DOS partition visible to Linux, then mount the partition if it is not already. Generally, this should have been set up when

TOC

you installed Linux. All you need to do next is change to the directory the vmlinuz file is in and copy it over to DOS using the cp command.

However, if Linux was not set up to recognize your DOS partition, then copy vmlinuz to a floppy. Take any DOS formatted floppy (with enough disk space to hold your kernel image file) and insert it into your floppy drive. Type:

```
mount /dev/fd0 /tmp
```

at a Linux shell prompt. Then change to the directory your kernel image file is in and type:

```
cp vmlinuz /tmp
```

This will copy vmlinuz to your floppy disk and it will be readable by DOS. Shutdown Linux, boot to DOS, then copy vmlinuz to whichever directory you choose. If you recompile your kernel, do not forget to copy the new kernel image file to your DOS partition. This will overwrite your old file, so it might be a good idea to rename the old file first just in case the new one does not work properly.

7.4 Does it matter if I install Linux or Windows 95 first?

- Technically no. However, it would be *much* simpler to install Win95 first. That way it is easier to setup Linux to recognize your DOS partition(s) as you can usually do this during the Linux installation.

7.5 I currently have LILO installed, but I want to use Loadlin instead. How do I remove LILO?

- You will need to restore the Win95 MBR (Master Boot Record) that LILO overwrote when you installed LILO. At a DOS prompt type:

```
fdisk /mbr
```

7.6 What if I already dual boot between Windows 95 and Windows 3.x, and I want to boot to Linux using this method?

- Frankly, if you want to do this I would recommend using LILO. However, if you really do not want to use LILO, you will have to boot to Win 3.x first, then issue the Loadlin command. (Make sure Windows 3.x is NOT running, but that you are in DOS 5.0 or 6.x).

7.7 Is it possible to initiate Linux from the Windows 95 desktop?

- Yes. First, create a batch file called "linux.bat", for example. Edit the file to contain a Loadlin command such as:

```
loadlin f:\vmlinuz root=/dev/hdc2 ro
```

  Now save the file in the \windows\desktop directory. Next, right click on the Linux.bat icon, then left click on Properties. Now click on the Program tab, then click on the Advanced button. Click on the box next to "MS-DOS mode" and make sure the box next to "Warn before entering MS-DOS mode is checked". Click OK, then click on OK again. Now when you double click on the Linux icon, a warning box will appear before going into MS-DOS mode. If you click on "Yes" then Windows enters MS-DOS mode and executes the Linux.bat file.

  NOTE: You must be in MS-DOS mode in order to use Loadlin. Please see the manual.txt file mentioned in Section 7.1 of this FAQ for more information.

7.8 Where can I get a plain text version of this document?

- You can get a plain text version of this document at:

  ftp://ftp.eskimo.com/u/p/praxis/loadlin.txt

8. Credits

A special thanks to Norm Jacobowitz and Fred Harris for helping me sort out the Windows 95 OSR2 issues and for helping me improve this mini-HOWTO.

The MacTerminal MINI-HOWTO

Robert Kiesling

v1.4, 9 November 1997

This mini HOWTO describes the 1,002nd use for a dead Macintosh (grin): how to configure the Mac for use as a Linux terminal. Configurations using getty and the terminal program kermit are described, as well as using kermit peer-to-peer networking between between Linux and a Macintosh. This document may be reproduced freely, in whole or in part, provided that any usage conforms to the general copyright notice of the HOWTO series of the Linux Documentation Project. See the file COPYRIGHT for details. Send all complaints, suggestions, errata, and any miscellany to kiesling@terracom.net, so I can keep this document as complete and up to date as possible.

1. Introduction.

2. Setting up a serial link.

3. Client-server connection with `kermit`.

- 3.1 Macintosh resources.

4. Logging in via `kermit`.

- 4.1 Other Mac terminal programs.

5. Conclusion.

1. Introduction.

This mini-HOWTO should give you some Insanely Great ideas for how to make your Macintosh work with Linux. Unfortunately, I have been very busy, and so I haven't been able to include even half of what I wanted to include, like using MacTCP and Open Transport to connect to your Linux box via a PPP line. That will need to wait for future versions.

This mini-HOWTO doesn't cover networking with LocalTalk and AppleTalk, either. I might explore these avenues if there's enough interest in, say, printing to a LaserWriter printer from Linux. Otherwise, it seems to me that such applications, being more trouble than they're worth (not to mention pricey), are beyond the scope of this document.

I don't plan to cover MkLinux in this document, either. It's more than adequately documented elsewhere.

So if you have ideas for this document, drop me a line at the e-mail above. Both systems embody a lot of the beginner's mindset as well as technical prowess, and in my opinion they don't talk to each other nearly enough.

2. Setting up a serial link.

To set up a serial link between a Mac and a Linux machine, you will need, on the Linux side, either a DB9 Female-to-DB25 Male serial cable or a DB25 Female-to-DB25 Male serial cable, depending on your serial port. On the Macintosh side, you will need a DIN9-to-DB25 Male high-speed modem cable.

Make sure that the cable is labeled a "high speed" cable, because some older Macintosh cables are configured with their handshaking lines tied high, which makes them useless for high-speed serial connections.

You will also need a null modem adapter, available at Comp USA, Radio Shack, and similar outlets, and a DB25 Female-to-DB25 Female serial gender changer to connect the two serial cables.

I have heard that Mac printer cables are really null modem cables in disguise, but I can't confirm this. Some of them are DIN9-to-DIN9 anyway, and wiring one into a serial link would be more trouble than it's worth.

If this sounds like Greek to you, read the Serial-HOWTO for details of RS-232 cable configurations and data transmission protocols.

Before connecting the Mac and the Linux machines, you should determine that you have a working serial port on both machines, either by connecting a modem and dialing out to another computer with `minicom` (Linux), ZTerm (Mac), `kermit` (either), or the communications program of your choice.

The latest version of minicom is available from sunsite.unc.edu/pub/Linux/apps/serialcomm/dialout and mirror sites.

ZTerm is a complete, easy to use comm program. Unfortunately, it's shareware. A current version is available from mac.archive.umich.edu and outlets like it.

The `kermit` program has been ported to every computer and operating system in existence. The archives are located at ftp.columbia.edu/kermit.

You should strongly consider using `kermit` on both machines at this stage at least, because 1) it's free (although it's not covered by the Free Software Foundation's General Public License); and 2) it's a lot less confusing to have `kermit` on both machines than two completely different communications programs.

If you have another way to determine that the serial ports of the two machines are operational, feel free to use that. The point is to ensure that both machines have working serial ports.

Making the actual serial connection should be easy, given the directions above. In case it isn't, the connection looks like this:

```
Linux PC    DB9- or DB25-  Null     Gender   DIN9-to-       MacBox
```

```
---------   to-DB25 male   Modem    Changer  DB25 Mac      -------
|       |   serial cable. |     |  |     |   Serial Cable  |      |
|       |-----------------|     |--|     |-----------------|      |
|       |                 |     |  |     |                 |      |
---------                 Adapter                        --------
```

3. Client-server connection with `kermit`.

This is the most transient of all the configurations described here. It requires the least amount of system configuration, although in operation, it is the more difficult to use of the systems described here.

In brief, you start `kermit` on both the Linux machine and the Mac, and place one of them in server mode. It doesn't matter which machine is the client ant which is the server, because this is a peer-to-peer connection. However, the Linux `kermit` can take advantage of Linux's superior scripting abilities, so it seems logical (to me at least) to designate the Linux-side `kermit` as the server, because this is the more readily automated task.

You should ensure that `kermit` is installed correctly on both the Mac and the Linux PC. Follow the instructions in the respective kermit distributions. On the Linux machine type `kermit` at the shell prompt to start it. You may need root permissions in order to set the port and baud rate.

`kermit`, the recent POSIX versions for Unices, supports baud rates up to 115 Kbps. The more recent Macintosh versions support serial port speeds up to 57.6 Kbps. This should be more than sufficient for any dumb tty-type application, but if you need a higher-speed connection, you're s.o.l, as far as `kermit` and serial lines are concerned. However, `kermit` provides facilities for communication over a TCP/IP link, but I haven't been able to test it. See the alternative in the following sections. Just remember, especially on the Mac side, to use a different port for `kermit` serial connections than your TCP/IP connections, because Mac `kermit` will rudely hose a serial port that is already in use.

With that in mind, your `.kermrc` file would contain something like this:

```
echo Executing site initialization file /usr/local/bin/ckermit.local.ini....
set prompt Chanel3 >
set line /dev/ttyS0
set baud 38400
set send packet-length 2000
set receive packet-length 2000
set block 3
set file type binary
```

Then, in your `~/.kermrc` file, you would have a line like

```
take /usr/local/bin/ckermit.local.ini
```

On the Macintosh side, set the same communication parameters for bps, stop bits, parity, and word length. Some older versions of Mac Kermit do not support 2k packets, so you might need to set a smaller packet size. Howerver, `kermit` sets the communication packet length based on the receive packet-length setting, so you need to set a shorter packet size on the Linux end, too.

To actually communicate over the link, you need to enter server mode on either the Mac or Linux side. It doesn't matter which. See the `kermit` docs for details of server mode.

3.1 Macintosh resources.

This is one of the very few `kermit` applications where setting a `text` file type for transfers is useful. This is because Macintosh files have two parts: the **data fork** and the **resource fork**. The data fork corresponds to what we in the Linux world think of as a file: it's the actual data. The resource fork contains bitmaps for the icons, keymaps, font specifications, and the like. If you transfer a file from Linux to the Mac, the file won't be recognized as a text file by the Mac, if you use binary mode.

When transferring binary files between the two systems, you should use the Macintosh `.hqx` BinHex format, which is a 7-bit encoding of an 8-bit data file. Mac utilities like BinHexer or StuffIt will covert the file to its binary form.

If you have a text file which inadvertently ends up as a data-only file on the Mac, it's likely that it won't even appear in an Open dialog list box. What you need to do is open the file with ResEdit, which is available from `mac.archive.umich.edu`. ResEdit will tell you that the file you're opening has no resource fork and then asks if you would like to add one. You should answer "Yes" to this question. You can then edit the file's Type and Creator by selecting the Open Special option of the File menu. All Macintosh text files are type `TEXT`, so replace the question marks in the Text box with that. The Creator code depends on your text editor or word processor. Each one is unique, incidentally, and is how the Mac identifies different apps. The Creator code for GNU Emacs on the Mac is `EMAC`, for example. If in doubt what the creator code of your text editor or word processor is, use `ttxt`, which is the creator code for TeachText (which is the Mac equivalent of `EDLIN.EXE`.) Then your real word processor or text editor can translate the file from TeachText to its native type.

There are many other neato things which TeachText can do, so it's worthwhile to keep it permanently on your Mac. The book *Voodoo Mac*, by Kay Yarborough Nelson, is a good source of tried-and-true Macintosh tricks that use ResEdit, TeachText, the Finder, and other overlooked programs.

---

4. Logging in via `kermit`.

Configuring Linux to use the Mac as a `login:` terminal is even easier. `kermit` is ideal for this purpose, because it is one of the few free communication programs which provides credible VT100/120/220 emulation.

Essentially, what you want to do is start `kermit` on the Macintosh side as in the previous section, but rather than issue server commands, you enter `connect` mode. This is the normal terminal emulation mode that most people use, anyway.

On the Linux side, the serial line must be configured with a `getty` on it to start a `login:` shell. To do this, you need to tell `init` that the serial line has a terminal on it. In your `/etc/inittab` file you will need a line something like this:

```
T1:23:respawn:/sbin/getty -L ttyS0 9600 vt100
```

Be sure to substitute the appropriate serial device for `/dev/ttyS0` and the correct baud rate for `9600` in the command line above.

This command tells `getty` to start `login` (the `-L` switch) on the terminal display, and, when the login times out, to re-start (`respawn`) the login program until someone logs in. If no device is connected to the serial line, or if the connection is defective, you may see a message on the system console like: `/dev/ttyS0 respawning too fast: disabling for 5 minutes`. If this happens, you can return things to normal by (as root) `kill`ing the `getty` process, or using the `init q` command. Both of them have the effect of re-spawning the `getty` processe(s). If everything is in order, you should see the Linux banner and login prompt on the Mac's `kermit` window. That's all there is to it.

Also, if you use something besides vanilla `getty`, like `getty_ps`, the command above will look somewhat different. The important thing to remember is that everything to the right of `/sbin/getty` is an argument for `getty` itself; not `init`. You should look at the manual pages for `getty`, `init`,and `inittab` if you have questions concerning the setup of `init` and `getty`.

The Serial HOWTO provides helpful details on how to configure `/etc/inittab` for `getty_ps`, if that's what your system uses.

To transfer files back and forth between the Macintosh and the Linux machine, you can (via the Mac's Kermit) issue the `kermit -x` command to start the Linux `kermit` in server mode. You can then use the normal file transfer commands to send files across the serial line. It's useful to set a prompt in your `~/.kermrc` with a line like

```
set prompt Linux-kermit >
```

Otherwise, remembering which machine you're on can quickly become confusing.

4.1 Other Mac terminal programs.

TOC

This method should work equally well for any other Mac terminal program. If you have ZTerm, you can use `rz` and `sz` on the Linux machine to transfer files via the ZModem protocol. If Microphone Lite came bundled with your fax modem, that works equally well, albeit without `kermit's` superior scripting and configuration facilities.

---

5. Conclusion.

If you have questions about any of this material, or suggestions for future directions of Mac-Linux serial-line connectivity, don't hesitate to drop me a line at `kiesling@terracom.net`.

Linux Mail-Queue mini-HOWTO

Leif Erlingsson, leif@lege.com, Jan P Tietze, jptietze@mail.hh.provi.de

v2.02, 3 September 1997, sendmail 8.8.7

*Queue Remote Mail + Deliver Local Mail The Configuration Changes Neccessary to Make Sendmail Deliver Local Mail ***Now*** While Stashing Remote Mail in The Queue Until "I Say So".*

1. Introduction

2. NON dial-on-demand solutions PART

- 2.1 Starting sendmail
- 2.2 Configuring sendmail
- 2.3 Menu support suggestions

3. Dial-on-demand solutions PART

- 3.1 Configuring sendmail.cf
- 3.2 Adding dial delay

4. Delivering e-mail

- 4.1 How to have e-mail delivered at special times.

1. Introduction

The document is written by two authors. The NON dial-on-demand solutions part (oldest part) is written by Leif Erlingsson <leif@lege.com>, and the newer dial-on-demand solutions part is written by Jan P Tietze <jptietze@mail.hh.provi.de>.

2. NON dial-on-demand solutions PART

Written by Leif Erlingsson <leif@lege.com>.

The original version of this part contained a lot of unnecessary stuff. This is all it takes, really...

2.1 Starting sendmail

Slackware et al: /etc/rc.d/rc.M:

```
echo "Starting sendmail daemon (/usr/sbin/sendmail -bd -os) [queue only mode]..."
/usr/sbin/sendmail -bd -os      # NOT "-bd -q 15m", the "standard" flags!
```

RedHat et al: /etc/rc.d/init.d/sendmail.init:

```
echo -n "Starting sendmail: [queue only mode]"
daemon sendmail -bd -os         # NOT daemon sendmail -bd -q1h
```

The -os is not really essential, all it does is this:

```
SuperSafe [s] Be super-safe when running things, i.e.,
        always instantiate the queue file,  even  if
        you are going to attempt immediate delivery.
        Sendmail always instantiates the queue  file
        before  returning  control  the client under
        any  circumstances.   This   should   really
        always be set.
```

This should already be configured in the default sendmail.cf anyway.

2.2 Configuring sendmail

Serious sendmail users use the m4 source for this. I recommend this solution if you ever plan on upgrading sendmail and also make anything but trivial changes to sendmail.cf.

If you never intend to fix sendmail so envelope return headers et al works even though you might be on dynamic dial in IP or something, then you may not need to get the m4 source.

Configuring sendmail.cf directly, for trivial configurations

This way of doing things is extremeley version dependent vs. sendmail.cf versions. The following solution is *only* valid for sendmail-8.8.x.

Edit /etc/sendmail.cf:

```
# avoid connecting to "expensive" mailers on initial submission?
O HoldExpensive=True
```

... later ...

```
#####  @(#)smtp.m4      8.33 (Berkeley) 7/9/96  #####
```

TOC

```
Msmtp,          P=[IPC], F=mDFMuXe, S=11/31, R=21, E=\r\n, L=990,
                T=DNS/RFC822/SMTP,
                A=IPC $h
Mesmtp,         P=[IPC], F=mDFMuXae, S=11/31, R=21, E=\r\n, L=990,
                T=DNS/RFC822/SMTP,
                A=IPC $h
Msmtp8,         P=[IPC], F=mDFMuX8e, S=11/31, R=21, E=\r\n, L=990,
                T=DNS/RFC822/SMTP,
                A=IPC $h
Mrelay,         P=[IPC], F=mDFMuXa8e, S=11/31, R=61, E=\r\n, L=2040,
                T=DNS/RFC822/SMTP,
                A=IPC $h
```

The important flag above is ``e''. Don't fuss if the other flags look different in your file. Keep your flags as-is, only add ``e'' to your flags according to the above examples, unless it's there already. ``e'' marks the mailers as ``expensive''.

Configuring sendmail.cf using m4 source

In the following I will, for simplicity, assume that the sendmail version is 8.8.7. If you have a different version, replace 8.8.7 with that version number below! Also, the instructions will not work for older versions of sendmail. Get the latest sendmail!

Download the sendmail source. Try ``http://WWW.Sendmail.ORG'' or possibly ``ftp.sendmail.org''.

I also recommend that you obtain my patch for allowing envelope sender reverse aliasing and other nice stuff to really make you take control over your mail environment.

Write to ``Sendmail Patch <sendmail@lege.com>'', Subject: ``sendmail-8.8.7'', if 8.8.7 is your sendmail version.

They are also available from ``http://www.lege.com'', as is the sgml source of this mini-HOWTO!

You don't have to get my patches in order to get ``Queue Remote Mail + Deliver Local Mail'' to work. My patches solve other things. But I just thought this would be a nice place to mention them, as many Linux users will find them extremely useful. (They will even give you properly working virtual domains, if you like. The virtual domains don't have to be ``local''. They will give you ``xaliases'', or in other words ``reverse aliasing''.)

Unpack the sendmail source. You may get /usr/src/sendmail-8.8.7/. cd /usr/src/sendmail-8.8.7/cf

Now overlay my patch, if you want it, otherwise skip this step: If you don't want to use procmail as Local Delivery Agent, save away your /usr/src/sendmail-8.8.7/cf/ostype/linux.m4 before doing this... Save my patch to ``/tmp/sendmail-8.8.7-cf-cpio-idcmu.gz'', then...

```
cd /usr/src/sendmail-8.8.7/cf
gzip -dc < /tmp/sendmail-8.8.7-cf-cpio-idcmu.gz | cpio -idcmu
```

If you didn't want to use procmail, write back the saved copy of /usr/src/sendmail-8.8.7/cf/ostype/linux.m4 again.

And regardless of if you applied my patch or not, you must make sure these lines or very similar ones are added to /usr/src/sendmail-8.8.7/cf/cf/yourhostname.smtp.mc (but if you applied my patch you may want to investigate filenames containing the word ``elijah'', under /usr/src/sendmail-8.8.7/cf):

```
dnl # Defer Delivery to "expensive" mailers until next time the
dnl # queue is processed using "O HoldExpensive=True" and make
dnl # sure smtp mailers are "expensive".
dnl # (See original "sendmail" book Chapter 30: Options,
dnl # "Oc - Don't connect to expensive mailers", or
dnl # 2nd Edition "sendmail" book Chapter 34.8.29,
dnl # "HoldExpensive (c), Queue for expensive mailers".)
dnl #                          / Leif Erlingsson <leif@lege.com>
```

```
define(`confCON_EXPENSIVE', `True')
define(SMTP_MAILER_FLAGS, e)
MAILER(local)dnl
MAILER(smtp)dnl
```

2.3 Menu support suggestions

The 1.x versions of this document contained Menu support suggestions for /var/X11R6/lib/fvwm/system.fvwmrc. I have dropped those in the current version, but they are available on request:

Write to ``Menu support suggestions <fvwmrc@lege.com>", Subject: ``Menu support suggestions"

---

3. Dial-on-demand solutions PART

Written by Jan P Tietze <jptietze@mail.hh.provi.de>.

Many Linux users access the Internet through a dialup line, and many have decided to implement dial-on-demand facilities on their system. That is, whenever an IP packet of some sort has to leave the local network or the local host, the link to an Internet Service Provider (ISP) will automatically be established. The link will be dropped after some period of time that no packet has travelled across.

Although this is very comfortable and cost effective, there is one special case in which this is neither comfortable (as the time to bring up a "traditional" modem dialup is very noticeable) nor cost effective, and this is sending e-mail. E-Mail is commonly sent by SMTP, either delivered by your own system or through a SMTP host on the Internet that usually resides in your ISP's network.

With dialup lines, every time you send a message the link will have to be brought up. This is quite okay if you send only one message, but if you happen to create and send multiple messages, bringing up the line more than once can be tedious and cost ineffective. Also, if your ISP imposes limits as to what times you are allowed to login, this would also restrict you to postpone messages at certain times of the day, and you would have to manually send them later.

Section 1 of this document will solve the situation, however in situations where an external DNS lookup would cause the link up, the link will still be established even if e-mail is just being queued. The reason is that sendmail wishes to "canonify" host names.

The solution to this problem is twofold: First, we'll have to moderately change sendmail.cf. And then we have to define the process of actual mail delivery. Personally, I prefer to have cron do the job for me and describe the necessary changes below.

3.1 Configuring sendmail.cf

For the reasons stated in [1.2], I recommend modifying the m4 sources instead of editing sendmail.cf directly. It will actually save you a lot of hassle and make configuration changes more verbose.

First, perform all the changes described in the first part of this document. Then go through the dial-on-demand specific stuff.

Configuring sendmail.cf directly, for trivial configurations

Configuring directly is highly impractical and anything but verbose, but obviously, this is your decision.

Close to very bottom of your sendmail.cf should be a line that reads:

```
R$* < @ $* $~P > $*                    $: $1 < @ $[ $2 $3 $] > $4
```

TOC

Precede that line with a "#" so that it reads

```
#R$* < @ $* $~P > $*                    $: $1 < @ $[ $2 $3 $] > $4
```

Configuring sendmail.cf using the m4 source.

Add the following line to /usr/src/sendmail-8.8.7/cf/cf/yourhostname.smtp.mc:

```
FEATURE(nocanonify)dnl
```

Your final sendmail.cf can then be built by issuing the following commands. Remember to always back up your old /etc/sendmail.cf before installing the new one:

```
cp /etc/sendmail.cf /etc/sendmail.cf.bak
cd /usr/src/sendmail-8.8.7/cf/cf
m4 yourhostname.smtp.mc > /etc/sendmail.cf
```

3.2 Adding dial delay

It is oftentimes useful, especially when using modem lines, to have a dial delay installed. This means that if sendmail tries to initiate a connection in an attempt to send an e-mail (and this causes the line to go up) but the link actually takes more time to get established than what sendmail thinks should be a reasonable timeout, sendmail will simply wait some seconds and then retry.

Configuring sendmail.cf directly

Somewhere in your sendmail.cf could be a line that would read:

```
#O DialDelay=10s
```

(or very similar). Delete the ``#''. If there's no ``#'' at the beginning of the line, things should be considered okay (it just means this had already been enabled before).

If there is no such line in your sendmail.cf, add one (it is a wise thing to do to add this in the "options" part of the file):

```
O DialDelay=10s
```

Now change the ``10s'' part to the number of seconds you deem suitable.

Configuring sendmail.cf using m4 source

Add the following line to /usr/src/sendmail-8.8.7/cf/cf/yourhostname.smtp.mc:

```
define(`confDIAL_DELAY',`10s')
```

Now change the ``10s'' part to the number of seconds you deem suitable.

Your final sendmail.cf can then be built by issuing the following commands. Remember to always back up your old /etc/sendmail.cf before installing the new one:

```
cp /etc/sendmail.cf /etc/sendmail.cf.bak
cd /usr/src/sendmail-8.8.7/cf/cf
m4 yourhostname.smtp.mc > /etc/sendmail.cf
```

---

4. Delivering e-mail

E-Mail delivery can be invoked by issuing the command "sendmail -q". For those who are interested in what sendmail actually does, "sendmail -q -v" will give a more verbose version of the delivery process.

It is very convenient to automate the process of e-mail delivery. A tool commonly used for this process is cron.

4.1 How to have e-mail delivered at special times.

Edit your crontab:

```
crontab -e
```

Add lines of the form:

```
05 18-23,0-7    * * Mon,Tue,Wed,Thu,Fri /usr/sbin/sendmail -q
05 *            * * Sat,Sun             /usr/sbin/sendmail -q
```

Please refer to the crontab man page (available through "man 5 \ crontab") for further information. I think the format is pretty obvious. The example crontab entries shown above send e-mail (if, and only if, e-mail is available from the queue) 5 minutes after an hour on weekdays, starting at 6:05 pm, and stopping at 7:05 am. On weekends, e-mail is delivered 5 minutes after an hour, starting at 12:05 pm on Saturday, and stopping 11:05 pm on Sunday.

As a dial-on-demand user, it is sometimes desirable to have your system collect your e-mail via the POP3 protocol at certain times of the day. You could therefore add an entry similar to the following to your crontab:

```
0 21            * * * popclient -3 -u <your pop3 user name
goes here> -p <put your password here> -o /var/spool/mail/<the user
on your system that should receive the collected e-mail>
<mailhost.somedomain.com>
```

Of course, this should all go on a single line.

Then, save the file and leave the editor. The crontab should now be installed.

Mail2News Mini-Howto

Rick Dean <howto@fdd.com>

v2.0, 1 July 1999

This document describes how to feed a mailing list to a news server.

1. Copyright, Distribution, etc.

- 1.1 What is this?
- 1.2 Copyright and such
- 1.3 Where can I get this HOW-TO?
- 1.4 Document history

2. Overview

- 2.1 What? Why? and some definitions
- 2.2 Assumed environment
- 2.3 Methodology
- 2.4 Not covered

3. The news reader

- 3.1 Netscape
- 3.2 Free Agent
- 3.3 trn and friends

4. The news server

- 4.1 Installing the news server
- 4.2 Running the newsserver

- 4.3 Creating the newsgroup
- 4.4 Unrestricting access
- 4.5 Changing permissions of /usr/bin/rnews
- 4.6 Testing article posting
- 4.7 Did it work?
- 4.8 Increasing expiration times

5. The posting script & perl

- 5.1 What is Perl?
- 5.2 Location of perl
- 5.3 The mail2news.pl script
- 5.4 How do I know if the script is running?
- 5.5 What is with the Aussie?
- 5.6 What do you mean $PATH?

6. The local mailer daemon

- 6.1 Enabling alias files
- 6.2 Documentation
- 6.3 Recompiling your sendmail.cf file
- 6.4 Creating a mail alias
- 6.5 Did it work?
- 6.6 Unprivledged

7. The mailing list remailer

- 7.1 Subscribing
- 7.2 Forging mail

1. Copyright, Distribution, etc.

1.1 What is this?

This document describes how to feed a mailing list to a news server using a Linux box. It is called a mini-HOWTO, specifically the Mail2news mini-HOWTO.

1.2 Copyright and such

Copyright (c) 1999, Rick Dean.
Copyright (c) 1996, Robert Hart.

The authors retain their copyright of this document. You are hereby granted permission to redistribute this document in whole or in part as long as it includes this copyright notice. Commercial redistribution is allowed and encouraged. All translations or derivative works of this document must be covered under this copyright notice, and without additional restrictions on distribution. This arrangement is also known as a copyleft.

This copyright notice, itself, is hereby placed in the public domain. You may copy it without atribution.

1.3 Where can I get this HOW-TO?

The latest version of this document is available at http://fdd.com/howto/.

Many Linux distributions include mini-HOWTOs in the `/usr/doc/HOWTO` directory.

This mini-HOWTO like most Linux mini-HOWTO's was written initially is SGML. This allows the text to be automatically translated to many formats including text, HTML, PostScript, etc. Those other formats are available somewhere.

More information about Linux documentation can be found at the Linux Documentation Project, and so many other places.

1.4 Document history

- Version 1.0 was written by Robert Hart `<hartr@redhat.com>` in 1996
- Version 2.0 was written by Rick Dean `<howto@fdd.com>` in 1999

2. Overview

2.1 What? Why? and some definitions

This document describes how to feed a mailing list to a news server using a Linux box. It is called a mini-HOWTO, specifically the Mail2news mini-HOWTO.

A mailing list (also known as a remailer), is an address where e-mail will be resent to a list of other addresses. This is useful for colaboration of geographically disperse groups. Many standards bodies like the working groups of the IETF use mailing lists.

Unfortunately, if one is subscribed to several mailing lists, one's inbox may be routinely flooded. Furthermore, some companies (such as 3Com) specify which e-mail client (such as Lotus Notes) their employees must suffer with. Redirecting these e-mails to a news server frees people to choose a news reader and utilize refined features specifically designed for the task (of deriving signal from noise).

News servers started on the Internet long ago, many years before the WWW. They (and the news reader clients) have features such as...

- threading - responses are grouped with the e-mails they refer to
- read indication - the computer keeps track of what you have seen (even across newsgroups!) so you don't have to.
- subject kill - mark threads as read by thread.
- archiving
- segregation by newsgroup (i.e. mailing list)
- a quick standardized method subscribing/unsubscribing
- and more..

A big focus of news servers is sharing news between servers. The largest of these groups became known as USENET. This mini-HOWTO does not address that. You could share the newsgroups created with this mini-HOWTO on your own, but you will live just fine without it. Like a web server, ubiquitous Internet connectivity has made centralized news servers acceptable. Furthermore, recent benchmarks have shown single-processor Linux boxes can handle 1300+ HTTP hits per second, so scalability is a minor issue.

Athough you do not need to own the mailing list to use mail2news, it is a good idea to own the news server.

TOC

2.2 Assumed environment

This document assumes you are using Linux, but other Unices are nice. Currently only sendmail (for a mail delivery agent) is described, but as `qmail` grows in popularity (in part because it is easier to configure). Hopefully someone with submit configuration notes for it too. For a news server, this document describes `innd`. It is pretty dominant as news servers go, but any NNTP compliant one should work. A bit of glue called mail2news.pl is a perl script, thus you need the Perl interpreter, but it is very common and probably already installed. Finally, I assume you are running all this (except the mailing list remailer) on one machine. Dividing it up is left as an exercise for the reader. :-)

At the time of this writing this mini-HOWTO was only tested against a RedHat-6.0 distribution. As a good computer scientist, you should not believe anything works until it has been specifically tested. Any feedback or notes relating to other distributions would be welcomed by the author.

For most of this mini-HOWTO you will need root access unless otherwise specified.

2.3 Methodology

This mini-HOWTO is presented backwards, as this is the easiest way to build and debug it. Backwards means we start with the newsreader and work upstream to the mailing list remailer, the opposite direction of normal data flow. This systems uses several hairy pieces (like sendmail and innd) which are sizeable mini-HOWTO's in thier own right.

2.4 Not covered

This mini-HOWTO does not cover...

- setting up a mailing list remailer.
- exchanging news between news servers.
- a complete list of things not covered.

Please do not e-mail me about these subjects (or SPAM).

---

3. The news reader

3.1 Netscape

Netscape comes with an integrated news reader. The easy way to subscribe to a group is type in (or click on) a link like

```
news://fuji.sfour.com/ietf.confctrl
```

Once you have subscribed, you only need to go to the message center. This can be done by clicking on the small talk balloons icon in the bottom right of a browser window.

3.2 Free Agent

Free Agent is a wonderful news reader by Forte for Windoze. See http://www.forteinc.com/agent/freagent.htm

3.3 trn and friends

The old command line newsgreaders like `trn` and `tin` are a good standby. Be sure to set the environment variable NNTPSERVER first. For example...

```
export NNTPSERVER=fuji.sfour.com
trn
```

4. The news server

4.1 Installing the news server

You need to install a news server. I used `inn` but others are available (somewhere).

Preferably just check the "News Server" box during your initial install, but alternatively if your distribution uses RPM (RedHat package manager), then use something like...

```
rpm -i inn-2.2.9.i386.rpm
```

4.2 Running the newsserver

To manually start or stop the news server, use a command like

```
/etc/rc.d/init.d/innd start
```

or

```
/etc/rc.d/init.d/innd stop
```

or

```
/etc/rc.d/init.d/innd restart
```

To have the news server start at boot you could add a command like this to end of `/etc/rc.d/rc.local`, but that is not the best. Many distributions have a graphical tool for choosing which daemons run. You can also try the command line program

```
setup
```

or

```
chkconfig --add innd
```

4.3 Creating the newsgroup

`innd` is pretty picky about permissions and ownership. For much of the news config you will need to be the user `news`. To become this user from root...

```
su - news
```

TOC

Using `ctlinnd`, create the newsgroup on your news server. Remember, the newsgroup will be **local**. I start it with a distinctive name so you can filter it out from your news distributions if you do that stuff. I shamelessly named my newsgroup `ietf.confctrl`. The words from left to right go from less to more specific.

You also need to tell `innd` that the group is moderated (by using ctlinnd). Indicating a moderated group is done by specifying `m` to the `newgroup` command. For example...

```
ctlinnd newgroup ietf.confctrl m confctrl@isi.edu
```

The newsgroup is set up as a **moderated** group, as this allows us to take advantage of the email capabilities of innd. Any messages posted to a moderated group are **not** immediately submitted to the group. Instead, messages are emailed to the moderator of the group. In our example confctrl@isi.edu is the address which is resent by the remailer.

If you are sharing news with other servers, remember to edit your newsfeeds so that this group is not **not** distributed (unless you specificaly wish this to occur).

4.4 Unrestricting access

By default the news server, doesn't let any clients read news, so I needed to disable the user authentication of innd. This was done in the /etc/news/nnrp.access. Check out the nnrp.access man page to learn the syntax of this file. I changed the first non-comment line to ...

```
*:Read Post:::*
```

If you want a username/password, fill in the 3rd and 4th (colin separated) fields. For more information on the syntax, check the man page...

```
man nnrp.access
```

4.5 Changing permissions of /usr/bin/rnews

I had to change the permissions of /usr/bin/rnews. It was not world read/executable, but `sendmail` runs scripts as `nobody`.

```
chmod a+rx /usr/bin/rnews
```

4.6 Testing article posting

If you copy the following article to a file named `rick.article`...

TOC

```
Path: rick
From: rick@fdd.com
Message-ID: <199907120548.AAA05475@fdd.com>
Subject: test
Date: Mon, 12 Jul 1999 00:48:49 -0500 (CDT)
Newsgroups: ietf.confctrl
Approved: ietf-confctrl@kepler.hedland.edu.au
NNTP-Posting-Host: localhost
Organisation: (mail2news gateway)


test
```

Then you should be able to post a file with...

```
/usr/bin/rnews -r localhost <rick.article
```

4.7 Did it work?

You don't need to wait for the article to show up as unread, just look at the `/var/spool/news/articles/` subdirectories for files being created.

4.8 Increasing expiration times

You may want to increase the expiration time for articles of your new newsgroup. In my case I wanted them never to expire, so I added the following line....

```
ietf*:A:never:never:never
```

....to the `/etc/news/expire.ctl` file.

To learn more about the syntax of this file type...

```
man expire.ctl
```

---

5. The posting script & perl

5.1 What is Perl?

Perl stands for Practical Extension and Report Language. It is very popular for small scripts which manipulate text which is exactly what we need.

Perl is installed by default on almost every Unix system.

5.2 Location of perl

If you perl intepreter is in an unusual place (not `/usr/bin/`) then you will have to modify the first line of the script. If this line is wrong, on my 2.2 kernel system I get "bash: /usr/local/bin/mail2news.pl: No such file or directory" Can we please change this to "bash: /usr/local/bin/mail2news.pl: Interpreter not found. Check first line of script." ?

5.3 The mail2news.pl script

```
#!/usr/bin/perl

($program = $0) =~ s%.*/%%;

#( $version  ) = $] =~ /(\d+\.\d+).*\nPatch level/;
#die "$program: requires at least version 3 of perl\n"
#        if $version < 3;

# $news_poster_program = "/usr/bin/inews";
# $news_poster_options = "-h -o \"mail2news gateway\"";
$news_poster_program = "/usr/bin/rnews";
$news_poster_options = "-r localhost";
$postinghost = "localhost";

if ($#ARGV < 0) {
    # $newsgroup = "test";
    # we'll expect the newsgroup line in the body
} elsif ($#ARGV == 0) {
    $newsgroup = $ARGV[0];
} else {
    die "usage: $program [newsgroup]\n";
}

# in case inews dumps core or something crazy
$SIG{'PIPE'} = "plumber";
sub plumber { die "$program: \"$news_poster_program\" died prematurely!\n"; }
```

TOC

```
open (INEWS, "| $news_poster_program $news_poster_options") ||
    die "$program: can't run $news_poster_program\n";

# header munging loop
while (<STDIN>) {
   last if /^$/;

   # transform real from: line back to icky style
   s/^From:\s+(.*) <(.*)>/From: $2 ($1)/;

   s/Message-Id/Message-ID/;

   # transform from_ line to path header; also works locally
   s/^From\s+(\S+)@(\S+).*/Path: $2!$1/
     || s/^From\s+(\S+)[^@]*$/Path: $1\n/;

   print INEWS
#       if /^(Date|From|Subject|Path|Newsgroups|Organization|Message-ID):/i;
   if /^(Date|From|Subject|Path|Newsgroups|Message-ID):/i;
   $saw_subject |= ( $+ eq 'Subject' );

   $saw_msgid |= ( $+ eq 'Message-ID' );

#   $saw_newsgroup |= ( $+ eq 'Newsgroups' );
}

warn "$program: didn't expect newsgroup in both headers and ARGV\n"
    if $newsgroup && $saw_newsgroup;

die "$program: didn't get newsgroup from either headers or ARGV\n"
    unless $newsgroup || $saw_newsgroup;

$approved = $newsgroup;
$approved =~ s/\./'-'/eg;

($sec,$min,$hour,$mday,$mon,$year)=localtime(time);
$madeupid = "\<$year$mon$mday.$hour$min$sec.$$\@kepler.hedland.edu.au\>";

printf INEWS "Newsgroups: %s\n", $newsgroup if $newsgroup;
printf INEWS "Approved: %s\@kepler.hedland.edu.au\n", $approved;
print  INEWS "Subject: Untitled\n" unless $saw_subject;
printf INEWS "Message-ID: %s\n", $madeupid unless $saw_msgid;
printf INEWS "NNTP-Posting-Host: %s\n", $postinghost;
print  INEWS "Organisation: (mail2news gateway)\n";
print  INEWS "\n";

print INEWS while <STDIN>;   # gobble rest of message

close INEWS;
exit $?;
```

TOC

I saved the script in /usr/local/bin (and will use this path throughout the HOWTO).

Be sure to make the script executable by all, but not writable by group or other. Sendmail is picky.

```
chmod a+x /usr/local/bin/mail2news.pl
chmod go-w /usr/local/bin/mail2news.pl
```

or

```
chmod 555 /usr/local/bin/mail2news.pl
```

for short.

5.4 How do I know if the script is running?

I tested this script by changed my news poster from /usr/bin/rnews to /bin/cat. I then saved an e-mail send to myself in a file. Finally I ran the mail2news.pl on the saved mail and captured the output to a file.

```
/usr/local/bin/mail2news.pl ietf.confctrl </tmp/savedMailFile >/tmp/article
```

5.5 What is with the Aussie?

Yes, the output of your posting script should contain the e-mail address of an austrailian. My guess is that the Austrailian's address is trusted address in your news configuration (althogh I could not find it in mine) for approving moderated postings.

5.6 What do you mean $PATH?

If the mail2news.pl script is not in my path, I get the error `bash: mail2news.pl: command not found`. You will need to either add this directory to your path

```
PATH=$PATH:/usr/local/bin
```

(which only works for the current login) or give an absolute path when you run the script

```
/usr/local/bin/mail2news.pl
```

To view your current path type

```
echo $PATH
```

---

6. The local mailer daemon

6.1 Enabling alias files

This document only describes modifying sendmail. Many people prefer qmail.

Out of the box, sendmail does not come with the "pipe to program" feature enabled. Rightly so, this feature is a security risk. It enambles users to have their mail fed to a program instead of appended to a file. (Users can configure this in thier .forward file.)

Please don't believe me. Test your own system first. Much of this chapter might not be needed by you.

6.2 Documentation

IMHO, sendmail installs with woefully insufficient documentation.

http://www.sendmail.org/ ca/email/doc/op-sh-5.html describes the syntax of sendmail.cf. I hope you don't have to use it.

This is a description of the features you are trashing in sendmail.mc
http://www.sendmail.org/m4/features.html

Sendmail has a man page...

```
man sendmail
```

6.3 Recompiling your sendmail.cf file

To get the "pipe to program" stuff in the aliases file to work you need to modify the default sendmail.cf (by modifying the sendmail.mc) so that the restricted shell is not used. I suppose the proper solution would be to add the one program to the restricted shell list, but their was no man page on smrsh. Strangely, uncommenting the smrsh feature didn't work, I needed to change the shell from `/usr/sbin/smrsh` to `/bin/bash`. Yeah, this is slightly risky, but it was not an issue on my machine. Without this change I kept getting a `"Service unavailable"` error message in the `/var/log/maillog` file.

The header of `/etc/senmail.mc` of RedHat-6 has a bug. The proper command line is...

```
m4 /etc/sendmail.mc >/etc/sendmail.cf
```

You need to do this when you change `sendmail.mc`. Hopefully, RedHat will extend the super cool Makefile idea in `/etc/mail` You will need to install `sendmail-cf.`something`.rpm` first. e.g. ...

TOC

```
rpm -i sendmail-cf-8.9.3-10.i386.rpm
```

Whenever you modify the sendmail.cf file, you should restart sendmail...

```
/etc/rc.d/init.d/sendmail restart
```

6.4 Creating a mail alias

Instead of creating a new user account, we will only create an alias.

When modifying the `/etc/aliases` file, the double quotes are required. There cannot be a space between the first double quotes and the | (pipe) character, or sendmail will complain "User unknown"

Add a line like ...

```
confctrl:   "| /usr/local/bin/mail2news.pl ietf.confctrl "
```

Whenever you modify the `/etc/aliases` file you need to notify sendmail.

```
sendmail -bi
```

6.5 Did it work?

Check the /var/log/maillog to see if it worked, or for error messages. I found it useful to open up another terminal (ssh) window to watch the log with

```
tail -f /var/log/maillog
```

6.6 Unprivledged

If you are having trouble, and create an e-mail alias with a different (simpler) target program to test it, remember that sendmail runs the program as an unprivledged user, who probably doesn't have privledges to write anywhere except globally writable directories such as `/tmp`.

---

7. The mailing list remailer

7.1 Subscribing

The method of subscribing to each e-mail list is different, although most involve sending some kind of e-mail message. Please be very careful not to send to the replicating address of the mailgroup. This is a sure way to annoy the members of the list. Most remailers (but not all) have a separate address for subscribing. In confctrl, the administration address is confctrl-request@isi.edu

Some mailing lists do not let you subscribe an address you are not mailing from. In this case you will need to forge the return address of an e-mail.

When you successfully subscribe you should (hopefully) see a welcome message in the newsgroup.

7.2 Forging mail

Forgeing mail is easiest to do with an old Netscape client. It would accept anything for a return address. Fortunately, modern browsers require you to retreive mail successfully for an address before they will let you send. Thus you are required to fall back the tried and true method....telnet.

Looking up MX hosts

When you speficy a web page, your browser does a DNS lookup to convert the domain name into an IP address. This is a lookup of an "A" record. (Also "CNAME" records are considered.) When sending mail a "MX" record is looked up. If this is missing a "CNAME" or "A" one is used. Thus, to forge mail you need to do an MX lookup.

```
dig mx isi.edu
```

will give...

```
; <<>> DiG 8.2 <<>> mx isi.edu
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 4
;; QUERY SECTION:
;;      isi.edu, type = MX, class = IN

;; ANSWER SECTION:
isi.edu.                1D IN MX        0 tnt.isi.edu.
isi.edu.                1D IN MX        10 venera.isi.edu.

;; AUTHORITY SECTION:
isi.edu.                1D IN NS        venera.isi.edu.
isi.edu.                1D IN NS        ns.isi.edu.
isi.edu.                1D IN NS        east.isi.edu.

;; ADDITIONAL SECTION:
tnt.isi.edu.            1D IN A         128.9.128.128
venera.isi.edu.         1D IN A         128.9.176.32
ns.isi.edu.             1D IN A         128.9.128.127
east.isi.edu.           1D IN A         38.245.76.2
```

```
;; Total query time: 448 msec
;; FROM: fdd.com to SERVER: default -- 127.0.0.1
;; WHEN: Sun Jul 25 15:49:32 1999
;; MSG SIZE  sent: 25  rcvd: 182
```

thus you would use tnt.isi.edu.

SMTP

Mail is delivered using the Simple Mail Transport Protocol (SMTP). Like most good Internet protocols, it is ASCII based to make troubleshooting and development easier. I will not explain everything, but simply give an example. Hopefully, this is enough.

The protocol is line oriented. Each email as specified in RFC822, is composed of headers and body which are separated by the first blank line (no characters, not even spaces). SMTP specifies the end of an e-mail with a line containing only a period.

```
halyard$ telnet tnt.isi.edu 25
Trying 128.9.128.128...
Connected to tnt.isi.edu.
Escape character is '^]'.
220 tnt.isi.edu ESMTP Sendmail 8.8.7/8.8.6; Sun, 25 Jul 1999 14:01:25 -0700 (PDT)
helo isi.edu
250 tnt.isi.edu Hello rick@node-d8e9822 [216.233.8.34] (may be forged), pleased to
meet you
mail from:<confctrl@fuji.sfour.com>
250 <confctrl@fuji.sfour.com>... Sender ok
rcpt to:<confctrl-request@isi.edu>
250 <confctrl-request@isi.edu>... Recipient ok
data
354 Enter mail, end with "." on a line by itself
From:<confctrl@fuji.sfour.com>
Subject: help

help
```

THE LINUX MAN-PAGE-HOWTO

Copyright 1995-1998 by Jens Schweikhardt, email: <schweikh@noc.dfn.de>

http://www.shuttle.de/schweikh/home.html

See further information on copying conditions below.

Last update: July 1998. Click here to browse the author's latest version of this document. Corrections and suggestions welcome!

This HOWTO explains what you should bear in mind when you are going to write on-line documentation -- a so called man page -- that you want to make accessible via the man(1) command. Throughout this HOWTO, a manual entry is simply referred to as a man page, regardless of actual length and without sexist intention.

Table of contents

- 0) A few thoughts on documentation
- 1) How are man pages accessed?
- 2) How should a formatted man page look like?
- 3) How do I document several programs/functions in a single man page?
- 4) Which macro package should I use?
- 5) What preprocessors may I use?
- 6) Should I distribute source and/or already formatted documentation?
- 7) What are the font conventions?
- 8) How do I polish my man page?
- 9) How do I get a plain text man page without all that ^H^_ stuff?
- 10) How do I get a high quality PostScript man page?
- 11) How do I get apropos and whatis to work?
- A) Copying conditions

0) A few thoughts on documentation

Why do we write documentation? Silly question. Because we want others to be able to use our program, library function or whatever we have written and made available. But writing documentation is not all there is to it:

- Documentation must be accessible. If it's hidden in some non-standard place where the documentation related tools won't find it -- how can it serve its purpose?
- documentation must be reliable and accurate. There's nothing more annoying than having program behaviour and documentation disagree. Users will curse you, send you hate mail and throw your work into the bit bucket, with the firm intent to never install anything written by that jerk again.

The historical and well known way documentation is accessed on UNIX is via the man(1) command. This HOWTO describes what you have to do to write a man page that will be correctly processed by the documentation related tools. The most important of these tools are man(1), xman(1x), apropos(1), makewhatis(8) and catman(8). Reliability and accuracy of the information are, of course, up to you. But even in

this respect you will find some ideas below that help you avoid some common glitches.

TOC

1) How are man pages accessed?

You need to know the precise mechanism how man pages are accessed in order to give your man page the right name and install it in the right place. Any man page belongs to a specific section, which is denoted by a single character. The most common sections under Linux and their human readable names are

```
Section The human readable name
   1    User commands that may be started by everyone.
   2    System calls, that is, functions provided by the kernel.
   3    Subroutines, that is, library functions.
   4    Devices, that is, special files in the /dev directory.
   5    File format descriptions, e.g. /etc/passwd.
   6    Games, self-explanatory.
   7    Miscellaneous, e.g. macro packages, conventions.
   8    System administration tools that only root can execute.
   9    Another (Linux specific) place for kernel routine documentation.
   n    New documentation, that may be moved to a more appropriate section.
   o    Old documentation, that may be kept for a grace period.
   l    Local documentation referring to this particular system.
```

The name of the source file for a man page (the input to the formatting system) is the name of the command, function or file name, followed by a dot, followed by the section. If you write the documentation on the format of the `passwd' file you have to name the source file `passwd.5'. Here we also have an example of a file name that is the same as a command name. There might be even a library subroutine named passwd. Sectioning is the usual way to resolve these ambiguities: The command description is found in the file `passwd.1' and the hypothetical library subroutine in `passwd.3'.

> Sometimes additional characters are appended and the file name looks for example like `xterm.1x' or `wish.1tk'. The intent is to indicate that this is documentation for an X Window program or a Tk application, respectively. Some manual browsers can make use of this additional information. For example xman will use `xterm(x)' and `wish(tk)' in the list of available documentation.

Please don't use the n, o and l sections; according to the File System Standard these sections are deprecated. Stick to the numeric sections. Beware of name clashes with existing programs, functions or file names. It is certainly a bad idea to write yet another editor and call it ed, sed (for smart ed) or red (for Rocky's ed). By making sure your program's name is unique you avoid that someone executes your program and reads someone else's man page or vice versa. Checking out the lsm database on a program name is a place to start doing so.

Now we know the name to give our file. The next decision is which directory it will finally get installed (say, when the user runs `make install' for your package.) On Linux, all man pages are below directories mentioned in the environment variable MANPATH. The doc related tools use it quite similar like the shell uses PATH to locate executables. In fact, MANPATH has the same format as PATH. Both hold a colon separated list of directories (with the exception that MANPATH does not allow empty fields and relative pathnames but has absolute names only.) If MANPATH is not set or not exported, a default will be used that contains at least the /usr/man directory. To speed up the search and to keep directories small, the directories specified by MANPATH (the so called base directories) contain a bunch of subdirectories named `man<s>' where <s> stands for the one character section introduced in the table above. Not all of the sections may be represented by a subdirectory because there simply is no reason to keep an empty `mano' subdirectory. However, there may be directories named `cat<s>', `dvi<s>' and `ps<s>' which hold documentation that is ready to display or print.

More on this later. The only other file in any base directory should be a file named `whatis'. The purpose a creation of this file will also be described under paragraph 11). The safest way to have a man page for sec <s> installed in the right place is to put it in the directory /usr/man/man<s>. A good Makefile, however, wil allow the user to chose a base directory, by means of a make variable, MANDIR, say. Most of the GNU packages can be configured with the --prefix=/what/ever option. The manuals will then be installed under the base directory /what/ever/man. I suggest you also provide a way to do something similar.

With the advent of the Linux File System Standard (FS-Stnd), things became more complicated. The FS-Stnd 1.2 states that

> "Provisions must be made in the structure of /usr/man to support manual pages which are written in different (or multiple) languages."

This is achieved by introducing another directory level that distinguishes between different languages. Quoting again from FS-Stnd 1.2:

> "This naming of language subdirectories of /usr/man is based on Appendix E of the POSIX 1003.1 standard which describes the locale identification string -- the most well accepted method to describe a cultural environment. The <locale> string is: <language>[_<territory>][.<character-set>][,<version>]"

(See the FS-Stnd for a few common <locale> strings.) According to these guidelines, we have our man pages in /usr/man/<locale>/man[1-9lno]. The formatted versions should then be in /usr/man/<locale>/cat[1-9lno] of course, otherwise we could only provide them for a single locale. HOWEVER, I can not recommend switching to that structure at this time. The FS-Stnd 1.2 also allows that

> "Systems which use a unique language and code set for all manual pages may omit the <locale> substring and store all manual pages in <mandir>. For example, systems which only have English manual pages coded with ASCII, may store manual pages (the man[1-9] directories) directly in /usr/man. (That is the traditional circumstance and arrangement in fact.)"

I would not switch until all tools (like xman, tkman, info and many others that read man pages) can cope with the new structure.

2) How should a formatted man page look like?

Let me present you an example. Below I will explain it in detail. If you read this as plain text it won't show the different typefaces (**bold** and *italics*). Please refer to the paragraph "What are the font conventions?" for further explanations. Here comes the man page for the (hypothetical) `foo` program.

```
FOO(1)                    User Manuals                    FOO(1)



NAME
     foo - frobnicate the bar library

SYNOPSIS
     foo [-bar] [-c config-file ] file ...

DESCRIPTION
     foo  frobnicates the bar library by tweaking internal symbol
     tables. By default it parses all baz segments and rearranges
```

them in reverse order by time for the **xyzzy**(1) linker to find them. The symdef entry is then compressed using the WBG (Whiz-Bang-Gizmo) algorithm. All files are processed in the order specified.

OPTIONS

-b Do not write `busy' to stdout while processing.

-c config-file
: Use the alternate system wide *config-file* instead of */etc/foo.conf*. This overrides any **FOOCONF** environment variable.

-a In addition to the baz segments, also parse the blurfl headers.

-r Recursive mode. Operates as fast as lightning at the expense of a megabyte of virtual memory.

FILES

/etc/foo.conf
: The system wide configuration file. See **foo**(5) for further details.

~/.foorc
: Per user configuration file. See **foo**(5) for further details.

ENVIRONMENT

FOOCONF
: If non-null the full pathname for an alternate system wide *foo.conf*. Overridden by the -c option.

DIAGNOSTICS

The following diagnostics may be issued on stderr:

Bad magic number.
: The input file does not look like an archive file.

Old style baz segments.
: foo can only handle new style baz segments. COBOL object libraries are not supported in this version.

BUGS

The command name should have been chosen more carefully to reflect its purpose.

AUTHOR

Jens Schweikhardt <schweikh@noc.dfn.de>

```
SEE ALSO
     bar(1), foo(5), xyzzy(1)

Linux                Last change: MARCH 1995                    2
```

Here's the explanation as I promised.

The NAME section

...is the only required section. Man pages without a name section are as useful as refrigerators at the north pole. This section also has a standardized format consisting of a comma separated list of program or function names followed by a dash followed by a short (usually one line) description what functionality the program (function, file) is supposed to provide. By means of makewhatis(8) the name sections make it into the whatis database files. Makewhatis is the reason why the name section must exist and why it must adhere to the format I described. In the groff source it must look like

.SH NAME foo \- frobnicate the bar library

The \- is of importance here. The backslash is needed to make the dash distinct from a hyphenation dash that may appear in either the command name or the one line description.

The SYNOPSIS section

...is intended to give a short overview on available program options. For functions this sections lists corresponding include files and the prototype so the programmer knows the type and number of arguments as well as the return type.

The DESCRIPTION section

...gives an eloquent explanation why your sequence of 0s and 1s is worth anything at all. Here's where you write down all your knowledge. This is the Hall Of Fame. Win other programmer's and user's admiration by making this section the source of reliable and detailed information. Explain what the arguments are for, the file format, what algorithms do the dirty jobs.

The OPTIONS section

...gives a description for any option how it affects program behaviour. You knew that, didn't you?

The FILES section

...lists files the program or function uses. For example, configuration files, startup files, files the program directly operates on. It is a good idea to give the full pathname of these files and to make the install process modify the directory part to match user preferences: the groff manuals have a default prefix of /usr/local, so they reference /usr/local/lib/groff/* by default. However, if you install using 'make prefix=/opt/gnu' the references in the man page change to /opt/gnu/lib/groff/*

The ENVIRONMENT section

...lists all environment variables that affect your program or function and tells how, of course. Most commonly the variables will hold pathnames, filenames or default options.

The DIAGNOSTICS section

...should give an overview of the most common error messages from your program and how to cope with them. There's no need to explain system error error messages (from perror(3)) or fatal signals (from psignal(3)) as they can appear during execution of any program.

The BUGS section

...should ideally be non-existent. If you're brave, you can describe here limitations, known inconveniences, features that others may regard as misfeatures. If you're not so brave, rename it the TO DO section ;-)

The AUTHOR section

...is nice to have in case there are gross errors in the documentation or program behaviour (Bzzt!) and you want to mail a bug report.

The SEE ALSO section

...is a list of related man pages in alphabetical order. Conventionally, it is the last section. You are free to invent other sections if they really don't fit in one of those described so far. So how exactly did you generate that man page? I expected that question, here's the source, Luke:

```
.\" Process this file with
.\" groff -man -Tascii foo.1
.\"
.TH FOO 1 "MARCH 1995" Linux "User Manuals"
.SH NAME
foo \- frobnicate the bar library
.SH SYNOPSIS
.B foo [-bar] [-c
.I config-file
.B ]
.I file
.B ...
.SH DESCRIPTION
.B foo
frobnicates the bar library by tweaking internal
symbol tables. By default it parses all baz segments
and rearranges them in reverse order by time for the
.BR xyzzy (1)
linker to find them. The symdef entry is then compressed
using the WBG (Whiz-Bang-Gizmo) algorithm.
All files are processed in the order specified.
.SH OPTIONS
.IP -b
Do not write `busy' to stdout while processing.
.IP "-c config-file"
Use the alternate system wide
.I config-file
instead of
.IR /etc/foo.conf .
This overrides any
```

```
.B FOOCONF
environment variable.
.IP -a
In addition to the baz segments, also parse the
blurfl headers.
.IP -r
Recursive mode. Operates as fast as lightning
at the expense of a megabyte of virtual memory.
.SH FILES
.I /etc/foo.conf
.RS
The system wide configuration file. See
.BR foo (5)
for further details.
.RE
.I ~/.foorc
.RS
Per user configuration file. See
.BR foo (5)
for further details.
.SH ENVIRONMENT
.IP FOOCONF
If non-null the full pathname for an alternate system wide
.IR foo.conf .
Overridden by the
.B -c
option.
.SH DIAGNOSTICS
The following diagnostics may be issued on stderr:

Bad magic number.
.RS
The input file does not look like an archive file.
.RE
Old style baz segments.
.RS
.B foo
can only handle new style baz segments. COBOL
object libraries are not supported in this version.
.SH BUGS
The command name should have been chosen more carefully
to reflect its purpose.
.SH AUTHOR
Jens Schweikhardt <schweikh@noc.dfn.de>
.SH "SEE ALSO"
.BR bar (1),
.BR foo (5),
.BR xyzzy (1)
```

TOC

3) How do I document several programs/functions in a single man page?

Many programs (grep, egrep) and functions (printf, fprintf, ...) are documented in a single man page. However, these man pages would be quite useless if they were only accessible under one name. We can not expect a user to remember that the egrep man page is actually the grep man page. It is therefore necessary to have the man page available under different names. You have several possibilities to achieve this:

1. have identical copies for each name.
2. connect all man pages using hard links.
3. symbolic links pointing to the actual man page.
4. use groff's `source' mechanism provided by the `.so' macro.

The first way is obviously a waste of disk space. The second is not recommended because intelligent versions of the catman program can save a lot of work by looking at the the file type or contents. Hard links will prevent catman from being clever. (catman's purpose is to format all man pages so that they can be displayed more quickly.) The third alternative has a slight drawback: if flexibility is a concern, you have to be aware that there are file systems that do not support symbolic links. The upshot of this is that the Best Thing (TM) is using groff's source mechanism. Here's how to do it: If you want to have your man page available under the names `foo' and `bar' in section 1, then put the man page in foo.1 and have bar.1 look like this:

.so man1/foo.1

It is important to specify the `man1/' directory part as well as the file name `foo.1' because when groff is run by the browser it will have the manual base directory as its current working directory (cwd) and groff interprets .so arguments relative to the cwd.

4) Which macro package should I use?

There are a number of macro packages especially designed for use in writing man pages. Usually they are in the groff macro directory /usr/lib/groff/tmac. The file names are `tmac.`<something>, where <something> is the argument to groff's -m option. Groff will use `tmac.`<something> when it is given the `-m <something>' option. Often the blank between `-m' and `<something>' is omitted so we may say `groff -man' when we are formatting man pages using the `tmac.an` macro package. That's the reason for the strange name `tmac.an'. Besides tmac.an there is another popular macro package, `tmac.doc`, which originated at the University of California at Berkeley. Many BSD man pages use it and it seems that UCB has made it its standard for documentation. The `tmac.doc` macros are much more flexible but alas, there are manual browsers that will not use them but always call `groff -man`. For example, all `xman` programs I have seen will screw up on man pages requiring `tmac.doc`. So do yourself a favor: use `tmac.an` -- use of any other macro package is considered harmful. `tmac.andoc` is a pseudo macro package that takes a look at the source and then loads either `tmac.an` or `tmac.doc`. Actually any man page browser should use it but until now not all of them do, so it is best we cling to ye olde `tmac.an`. Anything I tell you from now on and concerning macros only holds true for `tmac.an`. If you want to use the `tmac.doc` macros anyway, here is a pointer to detailed information on how to use them: http://www.bsdi.com/bsdi-man There is a searchable index form on the page. Enter `mdoc.samples` and it will find you `mdoc.samples(7)`, a tutorial sampler for writing BSD man pages.

The definitive dope for troff, with all macros explained, is the *Troff User's Manual* by Jospeh F. Ossanna and Brian W. Kernighan, revised November 1992. AT&T Bell Labs have made it publicly available. It's a 92k gzipped PostScript file. Don't forget to check out W. Richard Steven's homepage (famous for *Unix Network*

Programming as well as the *TCP/IP Illustrated* trilogy), who also has a list of Troff Resources including tbl, eqn, pic and other filters.

TOC

5) What preprocessors may I use?

Groff comes with at least three preprocessors, tbl, eqn, and pic (on some systems they are named gtbl, geqn and gpic.) Their purpose is to translate preprocessor macros and their data to regular troff input. Tbl is a table preprocessor, eqn is an equations/maths preprocessor and pic is a picture preprocessor. Please refer to the man pages for more information on what functionality they provide. To put it in a nutshell: don't write man pages requiring ANY preprocessor. Eqn will generally produce terrible output for typewriter-like devices, unfortunately the type of device 99% of all man pages are viewed on. For example, XAllocColor.3x uses a few formulas with exponentiation. Due to the nature of typewriter-like devices the exponent will be on the same line as the base. N to the power of two appears as `N2'. Tbl should be avoided because all xman programs I have seen fail on them. Xman 3.1.6 uses the following command to format man pages, e.g. signal(7):

```
gtbl /usr/man/man7/signal.7 | geqn | gtbl | groff -Tascii -man
/tmp/xmana01760 2> /dev/null
```

which screws up for sources using gtbl, because gtbl output is fed again into gtbl. The effect is a man page without your table. I don't know if it's a bug or a feature that gtbl chokes on its own output or if xman could be a little smarter not using gtbl twice... Anyway, if you want a table, format it yourself and put it between .nf .fi lines so that it will be left unformatted. You won't have bold and italics this way but this beats having your table swallowed any day. I have yet to see a man page requiring pic preprocessing. But I would not like it. As you can see above, xman will not use it and groff will certainly do the funky wadakiki on the input.

6) Should I distribute source and/or already formatted documentation?

Let me give the pros (+) and cons (-) of a few selected possibilities:

1. Source only:
   + smaller distribution package.
   - inaccessible on systems without groff.
2. Uncompressed formatted only:
   + accessible even on systems without groff.
   - the user can't generate a dvi or postscript file.
   - waste of disk space on systems that also handle compressed pages.
3. Compressed formatted only:
   + accessible even on systems without groff.
   - the user can't generate a dvi or postscript file.
   - which compression format would you use? .Z? .z? .gz? All of them?
4. Source and uncompressed formatted:
   + accessible even on systems without groff.
   - larger distribution package
   - some systems may expect compressed formatted man pages.
   - redundant information on systems equipped with groff.

IMHO it is best to distribute source only. The argument that it's inaccessible on systems without groff does not matter. The 500+ man pages of the Linux Documentation Project are source only. The man pages of XFree86 are source only. The man pages from the FSF are source only. In fact, I have rarely seen software distributed with formatted man pages. If any sysadmin is really concerned about having man pages accessible then he also

has groff installed.

7) What are the font conventions?

First of all: don't use direct font operators like \fB \fP etc. Use macros which take arguments. This way you avoid a common glitch: forgetting the font change at the end of the word and having the bold or italic extend up to the next font change. Believe me, it happens more often than you think. The `tmac.an` macros provide the following type faces:

.B Bold

.BI Bold alternating with italics

.BR Bold alternating with Roman

.I Italics

.IB Italics alternating with bold

.IR Italics alternating with Roman

.RB Roman alternating with bold

.RI Roman alternating with italics

.SM Small (scaled 9/10 of the regular size)

.SB Small bold (NOT small alternating with bold)

X alternating with Y means that the odd arguments are typeset in X while the even arguments are typeset in Y. For example

.BI "Arg 1 is Bold, " "Arg 2 is Italics, " "and Bold, " "and Italics."

The double quotes are needed to include white space into an argument. So much for what's available. Here's how you should make use of the different typefaces: (portions shamelessly stolen from man(7))

Although there are many arbitrary conventions for man pages in the UNIX world, the existence of several hundred Linux-specific man pages defines our standards: For functions, the arguments are always specified using italics, even in the SYNOPSIS section, where the rest of the function is specified in bold:

.BI "myfunction(int " argc ", char **" argv);

Filenames are always in italics, except in the SYNOPSIS section, where included files are in bold. So you should use

.I /usr/include/stdio.h

and

.B #include <stdio.h>

Special macros, which are usually in upper case, are in bold:

.B MAXINT

When enumerating a list of error codes, the codes are in bold. This list usually uses the .TP (paragraph with

hanging tag) macro as follows:

```
.TP
.B EBADF
.I fd is not a valid file descriptor.
.TP
.B EINVAL
.I fd is unsuitable for reading
```

Any reference to another man page (or to the subject of the current man page) is in bold. If the manual section number is given, it is given in roman, without any spaces:

```
.BR man (7)
```

Acronyms look best when typeset in small type face. So I recommend

```
.SM UNIX

.SM ASCII

.SM TAB

.SM NFS

.SM LALR(1)
```

8) Polishing your man page

Following are some guidelines that increase reliability, readability and 'formatability' of your documentation.

- Test examples if they work (use cut and paste to give your shell the exact wording from the man page) read output of your command into your man page, don't type what you THINK your program will print.
- Proof read, ispell, have someone else read it, especially if you are not a native English speaker. The HOWTO you are reading by now has not yet passed the latter test. Do you want to volunteer?
- Test your man page: Does groff complain when you format your man page? It's nice to have the groff command line in a comment. Does the man(1) command complain when you call `man yourprog'? Does the way how man(1) uses the formatting system produce the expected result? Will xman(1x) and tkman(1tk) cope with your manual? XFree86 3.1 has xman 3.1.6 - X11R6, it will try to uncompress using
  `gzip -c -d < %s > %s`
  `zcat < %s > %s`
- Will makewhatis(8) be able to extract the one-line description from the NAME section?

9) How do I get a plain text man page without all that ^H^_ stuff?

Have a look at `col`(1), `col` can filter out backspace sequences. Just in case you can't wait that long:

```
funnyprompt$ groff -t -e -mandoc -Tascii manpage.1 | col -bx > manpage.txt
```

The -t and -e switches tell groff to preprocess using tbl and eqn. This is overkill for man pages that don't require preprocessing but it doesn't harm apart from a few CPU cycles wasted. On the other hand, not using -t when it is actually required does harm: the table is terribly formatted. You can even find out (well, "guess" is a better word) what command is needed to format a certain groff document (not just man pages) by issuing

```
funnyprompt$ grog /usr/man/man7/signal.7 groff -t -man
/usr/man/man7/signal.7
```

"Grog" stands for "GROff Guess", and it does what it says--guess, if it were perfect we wouldn't need options any more. I've seen it guess wrong on macro packages, but never on preprocessors. Here is a little perl script I wrote that can delete the page headers and footers, therefore saving you a few pages when printing long and elaborate man pages. Save it in a file named strip-headers & chmod 755.

```
#!/usr/bin/perl -wn
#  make it slurp the whole file at once:
undef $/;
#  delete first header:
s/^\n*.*\n+//;
#  delete last footer:
s/\n+.*\n+$/\n/g;
#  delete page breaks:
s/\n\n+[^ \t].*\n\n+(\S+).*\1\n\n+/\n/g;
#  collapse two or more blank lines into a single one:
s/\n{3,}/\n\n/g;
#  see what's left...
print;
```

You have to use it as the first filter after the 'man' command as it relies on the number of newlines being output by groff. For example:

```
funnyprompt$ man bash | strip-headers | col -bx > bash.txt
```

10) How do I get a high quality PostScript man page?

```
funnyprompt$ groff -t -e -mandoc -Tps manpage.1 > manpage.ps
```

Print that using your favorite PostScript printer/interpreter. See question 9) for explanation of options.

11) How do I get `apropos' and `whatis' to work?

Suppose you wonder what compilers are installed on your system and how these can be invoked. To answer this (frequently asked) question you say

```
funnyprompt$ apropos compiler
```

f77 (1) - Fortran 77 compiler
gcc (1) - GNU C and C++ compiler
pc (1) - Pascal compiler

Apropos and whatis are used to give a quick response which man page has information on a certain topic. Both programs search a number of files named `whatis' that may be found in each of the manual base directories. Like I said before, the whatis data base files contain a one line entry for any man page in the respective directory tree. In fact, that line is exactly the NAME section (to be precise: joined on one line and with hyphenation removed, also note that the section is mentioned within parentheses). The whatis data base files are created with the makewhatis(8) program. There are several versions around, so please refer to the man page what options are available. In order for makewhatis to be able to extract the NAME sections correctly it is important that you, the manual writer, adhere to the NAME section format described under question 2). The difference between apropos and whatis is where in the line and what they are looking for. Apropos (which is

equivalent to `man -k`) searches the argument string anywhere on the line whereas whatis (equivalent to `man -f`) tries to match a complete command name only on the part before the dash. Consequently, `whatis c` will report if there is a cc manual and remain quiet for gcc.

TOC

Corrections and suggestions welcome!

A) Copying conditions

Copyright 1995-1998 by Jens Schweikhardt <schweikh@noc.dfn.de>

Voice: ++49 7151 909516

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions. All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below. In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs. If you have questions, please contact the Linux HOWTO coordinator, Tim Bynum, at linux-howto@sunsite.unc.edu via email.

Win95 + WinNT + Linux multiboot using LILO mini-HOWTO

Renzo Zanelli, rzanelli@southeast.net

v1.0, 26 March 1998

This document describes how to use the to multiboot between Windows 95, Windows NT, and Linux.

1. Introduction

2. Installing the Operating Systems

- 2.1 Installing Windows 95

3. The Linux part

4. The Windows NT part

5. The Windows 95 part

6. References

1. Introduction

There are a number of documents available that describe in detail how to multiboot just about any number of OS's using NT's OS loader. When I finally bought Windows NT, I already had Windows 95 and Linux installed on my system and I was using LILO to multiboot between them. I decided to add Windows NT to LILO instead of using NT's OS loader.

Originally I had Windows 95 on my first IDE disk drive and Linux on my second IDE disk drive. The Linux drive became my third IDE drive and I installed a new IDE disk for Windows NT as my second drive. Even though I haven't tried this in different configurations, I believe the order with which each OS is installed or which drive it is installed on is unimportant.

Just remember to create those emergemcy bootup floppies when each OS prompts you to!

Because I used Gilles Vollant's BootPart utility and BootPart only supports drives up to 4GB in size, this may not work if you have drives of larger capacity. If you need to work with drives greater than 4GB, contact Gilles directly (see BootPart documentation for Gilles' e-mail address).

2. Installing the Operating Systems

When you install the OSes, make sure that all are installed on bootable partitions. I installed both Windows 95 and Windows NT on Fat 16 partitions, so this procedure works on Fat 16 drives. If you decide to use Fat32 for Windows 95 and NTFS for Windows NT, check to see if your version of Linux supports them and, if so, to what extent.

2.1 Installing Windows 95

Install Windows 95 on your first drive. This will be your C: drive. Remember to create the Windows 95 boot floppies when prompted to do so.

I have found that it is better if the Linux drive is removed from the system when installing Windows 95, so that it is never detected and it doesn't exist as far as Windows 95 is concerned.

At this point, if you did everything correctly, you should be able to boot Windows 95.

Installing Windows NT 4.0

Install Windows NT 4.0 on your second drive. When you create the partition on this drive, make the partition bootable. Not all versions of Linux support NTFS, so I created a FAT16 partition, just to be on the safe side and because I wanted to exchange data between the Windows drives.

When Windows NT installs, it will detect the presence of Windows 95 and will create an entry for it in the NT OS loader and you should see three entries in the OS loader menu as follows:

```
Windows NT Workstation Version 4.00
Windows NT Workstation Version 4.00 [VGA mode]
Microsoft Windows
```

At this point, if you did everything correctly, you should be able to multiboot to Windows NT and Windows 95.

Installing Linux

Install Linux on your third drive. When I installed Linux, I only had Windows 95 installed so I only added the Windows 95 partition to Lilo. You can go ahead and install all OSes while installing LILO, or edit the Lilo configuration file later. However you decide to do it, I will show you how the /etc/lilo.conf file is supposed to look in the next section. Once Linux is installed, you should be able to boot to Linux.

3. The Linux part

This will involve editing the /etc/lilo.conf file and reinstalling LILO. The following should already be present in /etc/lilo.conf:

```
boot=/dev/hda
```

The Linux stanza should also be already present:

```
# Linux stanza
image=/vmlinuz
root=/dev/hdc1
label=Linux
# End Linux stanza
```

Now add the Windows 95 stanza:

```
# Windows 95 stanza
other=/dev/hda1
table=/dev/hda
label=Windows95
# End Windows 95 stanza
```

Now add the Windows NT stanza:

```
# Windows 95 stanza
other=/dev/hdb1
table=/dev/hda
loader=/boot/any_d.b
label=WindowsNT
# End Windows 95 stanza
```

Now run LILO again. At this point, if all is well, you should have Windows 95, Windows NT, and Linux as selections from LILO. You should be able to boot to Linux and Windows 95 (which should give you the Windows NT OS loader). To make the Windows NT entry operational, you still need to do some work on the Windows NT side.

4. The Windows NT part

Before you can continue, you need two do two things.

The first is to download and unzip Gilles Vollant's BootPart utility which can be found at http://www.winimage.com/bootpart.htm with some information on how to use it.

The second thing to do is copy (do not delete them yet!) the following files to the root of the Windows NT drive:

```
Boot.ini
Bootsect.dos
Ntdetect.com
Ntldr
```

These files are the Windows NT loader. They are located in the root of the Windows 95 drive and were placed there by Windows NT when you installed it.

Once this is done, boot to the Windows 95 DOS prompt (press Shift+F5 when Windows 95 loads) and issue the following command:

```
BOOTPART WINNT BOOT:D:
```

Now you must edit Boot.ini in your Windows NT root directory. You must edit three lines, one in the [boot loader] section and two in the [operating systems] section.

This is the original Boot.ini:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(1)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(1)partition(1)\WINNT="Windows NT Workstation Version 4.00"
multi(0)disk(0)rdisk(1)partition(1)\WINNT="Windows NT Workstation Version 4.00 [VGA
mode]" /basevideo /sos
C:\ = "Microsoft Windows"
```

This is the new Boot.ini:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows NT Workstation Version 4.00"
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows NT Workstation Version 4.00 [VGA
mode]" /basevideo /sos
```

Notice that all instances of rdisk(1) have been changed to rdisk(0) and the line referencing Windows 95 has been removed.

After saving Boot.ini, if you did everything correctly, you should now be able to boot to Windows NT from LILO.

5. The Windows 95 part

Create a bootable DOS floppy from the Windows 95 DOS prompt (press Shift+F5 when Windows 95 loads). It is important that this floppy be created at the DOS prompt, not from a DOS window inside Windows95! Do not boot to Windows95 until this step is complete! Copy the file C:\WINDOWS95\COMMAND\SYS.COM to the floppy (substitute your Windows95 path if different). Now boot from the floppy and enter the command:

```
SYS C:
```

This should have removed the Windows NT OS loader and you should be able to boot directly into Windows 95 from LILO.

You can now remove the NT loader files:

```
Boot.ini
Bootsect.dos
Ntdetect.com
Ntldr
```

6. References

Thanks to the following people (in no particular order):

James C. Bevier & D. Michael McFarland. In a message exchange on the reddhat-list they provided some information on how to boot NT from Linux.

Gilles Vollant. BootPart is a very useful utility and Gilles makes it available for free with some nice documentation. Thanks Gilles.

Mike Harlan. By reading his 'Linux-DOS-Win95-OS2' mini-HOWTO, it helped me make some assumptions about LILO's behavior.

NCD X terminal mini HOWTO

Ian Hodge, ihodge at nortel.ca

v1.01, 9 August 1998

This document describes how to connect an NCD X terminal to a UNIX host

1. Copyright

2. Introduction

- 2.1 Summary of steps

3. Requirements

- 3.1 The X terminal
- 3.2 Physical Connection
- 3.3 NCD X server software

4. Configuring the UNIX Host

- 4.1 TFTP and BootP

5. Configuring the NCD X terminal Boot Process

- 5.1 Boot Monitor command syntax
- 5.2 Boot Monitor Setup Menu
- 5.3 X Terminal Configuration Files
- 5.4 Logging on to the host

TOC

6. Fonts and the X terminal

- 6.1 The font server

7. Miscellaneous

- 7.1 Reference
- 7.2 Equipment used in the preparation of this document
- 7.3 Acknowledgments
- 7.4 Outstanding Issues
- 7.5 Feedback

---

1. Copyright

The NCD X terminal mini HOWTO is copyright (C) 1998 by Ian Hodge. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies.

2. Introduction

This document describes how an X terminal manufactured by NCD (Network Computing Devices) can be connected to and booted from a UNIX host using BootP (boot protocol). Many terminals are also capable of booting with RARP, NFS or locally from a PCMCIA card or over a serial link (either directly or with a modem).

Although the material in this document was prepared based on experience with a single model of X terminal, much of the information applies to other models and other X terminals generally. There is also an Linux X terminal mini HOWTO document (declared obsolete by HOWTO maintainers at the time of writing and therefore possibly not available in all Linux distributions) which overlaps material from this document. That document covers X terminal information more generally.

2.1 Summary of steps

The process of connecting an X terminal to a UNIX network can be summarized in the following steps:

- Physically connect the X terminal to the network.
- Configure the UNIX host you are going to boot from.
- Configure the X terminal boot procedure.
- Boot the X terminal.
- Log in to the network.

3. Requirements

3.1 The X terminal

An X terminal is a device which communicates and displays over a network using a distributed network window system known as X. Typically, the terminal's X software, known as the X server, is retrieved from the network at boot time. Programs other then the X server software are not run locally on the X terminal (with some exceptions); instead, the X terminal displays programs running on other hosts on the network. The X terminal, therefore, is a type of network computer which uses the X protocol to access network resources.

3.2 Physical Connection

The NCD X terminal (model Xncd19r was used in preparation of this document) has an RJ-45 (twisted pair) connector for use with 10baseT Ethernet. A hub is required to link more than two Ethernet devices using twisted pair. If the X terminal and its host are the only devices in the network, they may be connected with a 'null' cable which is described in the Linux Ethernet HOWTO document.

3.3 NCD X server software

The terminal's X server software file is available from the manufacturer and presumably is provided with the terminal upon initial purchase. This file will reside on the boot host where it can be accessed by the X terminal when it boots. This file is specific to the terminal type but independent of boot host. The terminal can boot from any host which supports the boot communication protocol (explained later). In addition to X server, the X software may also include applications, like a window manager, which can run locally on the X terminal itself.

4. Configuring the UNIX Host

At boot time, the X terminal retrieves files (including its X server software) from a remote host on the network. After the terminal boots, the X server software will control input, display, local clients and X protocol communication. The X server software is executed on the X terminal and therefore does not have to be software compatible with the host on which it resides.

4.1 TFTP and BootP

Together, tftp (trivial file transfer protocol) and BootP (boot protocol server) are used on the remote host to allow the X terminal to retrieve its X server software and configuration files over the network. Both services are typically started by inetd (Internet Daemon).

After the X terminal is powered up, if it is configured to boot from the network, it will send out a broadcast message using BootP (TCP/IP bootstrap protocol). This boot message will contain the X terminal's hardware (Ethernet) address which is used by the boot host to respond to the boot request.

When a boot request is received by the remote host, inetd (listening on a port designated in /etc/services) starts the BootP daemon specified in /etc/inetd.conf.

In file inetd.conf, create or uncomment lines that refer to TFTP and BootP. The final argument of the tftpd entry in the example below is the path of the directory containing the files required by the X terminal. Although directory names are not mandatory, for security reasons they should always be present as tftp access will then be restricted to files in specified directories.

>From a sample file /etc/inetd.conf:

```
# tftp service is provided primarily for booting.  Most sites
# run this only on machines acting as "boot servers".
tftp   dgram udp wait root /usr/sbin/tcpd /usr/sbin/in.tftpd /usr/X11/lib/X11/ncd/
bootps dgram udp wait root /usr/sbin/tcpd /usr/sbin/in.bootpd
```

Upon activation, the BootP server daemon on the host will then read its database file /etc/bootptab. An entry for the X terminal must be placed in this file. Each entry contains a set of tags separated with ':' characters. The host name must be the very first tag in an entry.

Useful bootptab tags:

ip

Address of the X terminal (eg 10.0.0.1).

sm

Subnet mask (eg 255.0.0.0). To understand the use of the subnet mask and other IP networking principles, consult the Linux NET-3 (networking) HOWTO document.

gw

IP Address of gateway (eg 10.0.0.1).

ht

Hardware type - Ethernet in this example.

ha

Hardware address of X terminal (6 byte Ethernet address)

According to the bootptab UNIX man page, the 'ha' tag must be preceded by the 'ht' tag. The X terminal's Ethernet address is displayed when the terminal is first powered on. The address appears as a series of 6 double digit hex numbers separated by colons (e.g. 00:00:A7:12:26:19).

tc

Table continuation or label of another entry in the BootP database. See the example below.

ds

IP address of domain name server (eg 10.0.0.3). Not required if DNS is not used for hostname resolution.

bf

Name of X terminal boot file (Usually the terminal model is used as the X server file name eg Xncd19r).

The following is an example of the Boot protocol server database file, /etc/bootptab. The character '\' is used to escape the end-of-line character.

```
# This is a general entry (here given the name default)
# with information common to all BootP clients
default:hd=/usr/X11/lib/X11/ncd/:\
        ds=10.0.0.3:\
        sm=255.0.0.0:\
        gw=10.0.0.1:\

# X terminal entry with hostname myxterm
# Notice the tc tag reference to the entry default
myxterm:ht=ethernet:\
        ha=0x0000a7122619:\
        ip=10.0.0.2:\
        tc=default:\
        bf=Xncd19r:
```

When a matching entry for the hardware address in the boot request is found in the bootptab file, a response is sent by bootpd with the corresponding IP address from the matching entry. File transfer can then take place over IP using TFTP.

A hostname can be assigned to the X terminal by creating an entry on the boot host in the file /etc/hosts. This file is used to map hostnames to IP addresses. In the this example, the X terminal (address 10.0.0.2) has been assigned the hostname 'myxterm'.

```
10.0.0.1        linuxhost       # The boot host
10.0.0.2        myxterm         # X terminal
```

5. Configuring the NCD X terminal Boot Process

After being powered up, the X terminal attempts to boot. This is the process where the X software is loaded into memory and executed. If the X terminal cannot boot, the Boot Monitor prompt '>' will appear. The Boot Monitor is firmware found in terminal PROMs (programmable read-only memory). With the basic Boot Monitor interface, it is possible to configure the terminal to boot and retrieve its X server software from the host. Use '?' for a list of Boot Monitor commands.

Configuration parameters set with the boot monitor are stored in NVRAM (Non-volatile Random-Access Memory) and are retained when the terminal is powered down.

>From the boot monitor, the 'bt' command or a menu system can be used to boot the terminal. Functionality of the two methods largely overlaps but the menu provides control over more boot parameters.

5.1 Boot Monitor command syntax

```
>bt file terminal_IP host_IP gateway_IP subnet_mask
```

file

The name of the file retrieved from the remote host containing the X server software used by the X terminal to boot (eg "Xncd19r"). Check that this file name is the same as the file name is found in the X terminal entry in the bootptab file on the host (explained in the previous section).

terminal_IP

The IP address assigned to the X terminal (eg 10.0.0.2). Again, this IP address should be the same as the address assigned in the X terminal entry of the bootptab file on the host.

host_IP

The IP address of the boot host (eg 10.0.0.1).

gateway

The IP address of the subnet gateway (eg 10.0.0.1)

subnet_mask

The subnet mask, specified as a decimal IP address or as a hexadecimal number (eg 255.0.0.0 or ff000000).

5.2 Boot Monitor Setup Menu

TOC

The setup menus are accessed by pressing the 'Setup' key or typing 'se' from the boot monitor '>' prompt.

Get IP Addresses From

The IP address of the X terminal should be obtained from boot monitor configuration stored in NVRAM. Only select 'Network' if you are using RARP (reverse ARP) to retrieve the X terminal's IP address from the remote host.

Terminal IP Address

The IP address assigned to the X terminal (eg 10.0.0.2). This is the same as 'terminal_IP' parameter above.

First Boot Host IP Address

The IP address of the boot host (eg 10.0.0.1). This is the same as 'host_IP' parameter above.

Gateway IP Address

The IP address of the subnet gateway. This is the same as 'gateway' parameter above.

Subnet Mask

The subnet mask, specified as a decimal IP address. This is the same as the 'subnet_mask' parameter above.

Broadcast IP Address

The IP address used to broadcast to the subnet. (eg 10.255.255.255)

Boot File

The name of the file retrieved from the remote host containing the X server software used by the X terminal to boot (eg "Xncd19r"). This is the same as 'file' parameter described above.

TFTP Boot Directory

The name of the directory on the host which contains the boot file (eg "/usr/X11/lib/X11/ncd/" or "/tftpboot/").

Config file

The name of the X terminal configuration file on the remote host (See below).

UNIX Config Directory

Name of the directory containing X terminal configuration files (eg "/usr/X11/lib/X11/ncd/").

TFTP Order, NFS Order, Local Order

Assign '1' to the preferred method for booting. Assign '1' to TFTP when booting from a host using BootP.

TOC

5.3 X Terminal Configuration Files

During the boot process, the X terminal will attempt to transfer and load files from the boot host. These files are not required for the X terminal to boot successfully. If a file is not found, the terminal will use default settings.

Configuration Files transferred to the X terminal at boot time:

- X terminal configuration file (eg ncd.conf) used to retain terminal settings.
- Color file (eg rgb.txt).
- X key symbol database (eg XKeysymDB).
- Font directory files (eg font.dir, font.alias).

After a successful boot, the X terminal console window with a menu bar should appear. The terminal setup key toggles display of this console window. From the console window 'setup' pull-down menu, terminal characteristics can be viewed, altered and saved on the boot host in the configuration file which can be used in future sessions.

If TFTP is being used to transfer files from the boot host, then file permission must be world readable. Similarly, to save a configuration file to the boot host, the file must already exist and with world write permission enabled. If secure TFTP is used (this is recommended for security reasons), then file access is possible only through specified directories.

5.4 Logging on to the host

>From the X terminal console window menu bar, select the 'terminals' pull-down menu and choose 'New Telnet...'. When the telnet window appears, insert the address of a network host in the service entry and click 'OK'. The host log in prompt should appear. After logging in, X programs, including a window manager, can be started from the telnet window.

---

6. Fonts and the X terminal

The X terminal comes with a small set of fonts. In the X terminal setup menus and configuration files, these fonts are referred to as 'built-ins'. The terminal can be operated with these fonts alone but more fonts are desirable. Fonts can be added by specifying font paths in X terminal console setup menus, configuration files or by using the xset command.

Once the X server software is running on the X terminal, the font path can modified or queried with the command xset.

To query the X server:

```
xset -q
```

To add a font entry:

```
xset +fp <path>
```

To remove a font entry:

```
xset -fp <path>
```

6.1 The font server

The font server (xfs) runs on a network host and retrieves fonts for the X terminal and other network clients. The font server improves font retrieval time and provides its clients access to more fonts then would otherwise be possible with tftp. Font server software is part of many Linux distributions and has also been incorporated into X consortium software available from ftp.x.org.

To indicate the use of a font server, a tcp service entry is used instead of a font path in the X terminal's host resident configuration file or console setup menu.

Format of tcp service entry:

```
tcp/<IP address of font server>:<port used by font server>
```

Example:

```
tcp/10.0.0.1:7100
```

An example of an entry in the X terminal configuration file found on the boot host:

```
xserver-default-font-path = {
        { "tcp/10.0.0.1:7100" }
        { "built-ins" }
}
```

The xset command with the tcp service entry can used to add the font server to the path:

```
xset +fp tcp/10.0.0.1:7100
```

WARNING: Certain versions of NCDware require the font server entry to be listed first before X terminal 'built-in' fonts. This is contrary to the NCD documentation "System Administrator's Guide for UNIX Systems". Experiment with the order of the font path and verify it with the 'xset q' command.

On the font server host, the server is started at boot time from the rc.local startup script. The font server (xfs) is started with the following command:

```
xfs -config <config file path> -port <font server port number>
```

The standard font server port is 7100.

Example:

```
xfs -config /usr/X11/lib/X11/fs/config -port 7100
```

Example font server configuration file:

```
# font server configuration file
# $XConsortium: config.cpp,v 1.7 91/08/22 11:39:59 rws Exp $
```

TOC

```
clone-self = on
use-syslog = on
catalog =
/usr/X11R6/lib/X11/fonts/misc/,/usr/X11R6/lib/X11/fonts/Speedo/,/usr/X11R6/lib/X11/fonts/Type1/,/usr/X11R6/lib/X11/fonts/75dpi/,/usr/X11R6/lib/X11/fonts/100dpi/
error-file = /usr/X11R6/lib/X11/fs/fs-errors
# in decipoints
default-point-size = 120
default-resolutions = 75,75,100,100
```

7. Miscellaneous

7.1 Reference

Unix man pages: bootpd(8), tftpd(8), bootptab(5), xdm(1x), xfs(1), fsinfo(1), xset(1), inetd(8)

Linux Ethernet HOWTO document, Linux Net-3 (Networking) HOWTO document

NCD Techtips Archive

NCD Techtips Archive Mirror at CERN

NCDware System Administrator's Guide for UNIX Systems

7.2 Equipment used in the preparation of this document

X terminal: NCD model NCD19r with 19 monochrome monitor 1280x1024 8M RAM

X server software: NCDware V3.2.CV 19r_s

Remote Host: IBM Cyrix 686 P150+ running Slackware (Linux kernel version 2.0.31)

Remote Host: IBM Cyrix M2 200 MMX running Red Hat 5.0 (Linux kernel version 2.0.32)

Network Card: 10 base T Ethernet card (ne 2000 clone) and 8 port hub

Font server version 2 release number 6300

7.3 Acknowledgments

I would like to thank Michael de Lind van Wijngaarden, Jamal Hadi-Salim and Dwight Hodge for assistance in the preparation of this document.

7.4 Outstanding Issues

- Is it possible or wise to start the font server using inetd?
- If the remote host running the font server is powered down but later comes up again, with the font server active, the X terminal may fail to retrieve its fonts. The font path entry has to be re-entered with the console menu or xset. Why?
- The date of the error file designated in the font server config file is changed when xfs is in use but the file is always empty.
- Are there host based diagnostics for the font server?
- Are the procedures for other models of X terminals roughly similar?
- What about XDM?
- What is the procedure to boot the X terminal using NFS (Network File System) or RARP (Reverse Address Resolution Protocol)?
- The following messages appear in the X terminal Console. I am suspicious that they come from xfree86 extensions to X which are not part of the X terminal server. Can someone confirm or deny this?

```
%XSERVER-I-NEWCLIENT, host "localhost" connected with blank authorization
%XSERVER-W-NOEXTENSION, client attempted to use non-existent extension "BIG-REQUESTS"
%XSERVER-W-NOEXTENSION, client attempted to use non-existent extension "XKEYBOARD"
%XSERVER-W-NOEXTENSION, client attempted to use non-existent extension "XFree86-Misc"
```

7.5 Feedback

Please write to `ihodge` at `nortel.ca` with any comments, suggestions or contributions.

NFS-Root Mini-Howto

Andreas Kostyrka, andreas@ag.or.at

V8, 8 August 1997

This Mini-HOWTO tries explains how to setup a ``disc-less'' Linux workstation, which mounts it's root filesystems via NFS. The newest version of this Mini-Howto can always be found in `ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/mini/NFS-Root` *or on any sunsite mirror NEAR YOU.*

1. Copyright

- 1.1 Contributors

2. General Overview

3. Setup on the server

- 3.1 Compiling the kernels
- 3.2 Creation of the root filesystem

4. Booting the workstation

- 4.1 Using a boot rom
- 4.2 Using a raw kernel disc
- 4.3 Using a bootloader & *RARP*
- 4.4 Using a bootloader without *RARP*

5. Known problems

- 5.1 /sbin/init doesn't start.
- 5.2 /dev troubles.

6. Other topics

1. Copyright

(c) 1996 Andreas Kostyrka (e9207884@student.tuwien.ac.at or andreas@ag.or.at)

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Andreas Kostyrka <`mailto:andreas@ag.or.at`>, the author of this mini-HOWTO, or Tim Bynum, the Linux HOWTO coordinator, at <`mailto:linux-howto@sunsite.unc.edu`> via email.

1.1 Contributors

- Avery Pennarun <`apenwarr@foxnet.net`> (how to boot without **LILO**)
- Ofer Maor `<ofer@hadar.co.il>` (providing a better mini howto about setting up discless workstations.)
- Christian Leutloff `<leutloff@sundancer.tng.oche.de>` (providing infos about netboot.)

2. General Overview

Generally speaking there are the following problems for the workstation:

- It must find out it's own IP-address, and if needed also the rest of the Ethernet configuration.
- It must know the *NFS*-server and the mount path to it's root filesystem.

The current implementation of *NFSROOT* in the Linux kernel (as of 1.3.7x) allows for the following ``solutions'':

- The IP-address may be discovered by *RARP*, or the full ethernet configuration may be passed to the kernel via kernel parameters by **LILO** or **LOADLIN**.
- The *NFS*-path to mount can be passed via kernel parameters. If this is not done, the kernel assumes the *RARP*-server also as *NFS*-server, and uses compiled in default for the path part. (current default value in the kernel: `/tftpboot/<IP-address of the machine>`.)
- The client configuration may be discovered by *BOOTP*.

Before starting to setup a discless enviroment, you should decide if you will be booting via **LILO** or **LOADLIN**. The advantage of doing so is flexibility, the disadvantage is speed. Booting a Linux kernel without **LILO** is faster. This may or may not be a consideration.

3. Setup on the server

3.1 Compiling the kernels

RARP support in the kernel of the server will probably be a good idea. You must have it if you will boot without kernel parameters. On the other hand it doesn't help you, if the client isn't on the same subnet than the server.

The kernel for the workstation needs the following as a minimum set compiled in:

- *NFS*-filesystem compiled in. (It doesn't need to have *ext2*-support compiled in, a module suffices.)
- ``Root on NFS'' must be enabled.
- The Ethernet driver for the network card of the workstation must be compiled in.
- Depending upon your needs you may have to include *RARP* or *BOOTBP* support for NFS-Root. (By this I mean the questions that are asked after the NFS question in make config.)

If the workstation will be booted without kernel parameters, you need also to set the root device to 0:255. Do this by creating a dummy device file with `mknod /dev/nfsroot b 0 255`. After having created such a device file, you can set root device of the kernel image with `rdev <kernel-image> /dev/nfsroot`.

3.2 Creation of the root filesystem

Copying the filesystem

Warning: while these instruction might work for you, they are by no means sensefull in a production enviroment. For a better way to setup a root filesystem for the clients, see the NFS-Root-Client mini howto by Ofer Maor `<ofer@hadar.co.il>`.

After having decided where to place the root tree, create it with (e.g.) `mkdir -p <directory>` and `tar cClf / - | tar xpCf <directory> -`.

If you boot your kernel without LILO, then the rootdir has to be `/tftpboot/<IP-address>`. If you don't like it, you can change it in the top Makefile in the kernel sources, look for a line like: `NFS_ROOT = -DNFS_ROOT="\"/tftpboot/%s\""` If you change this, you have to recompile the kernel.

Changes to the root filesystem

Now trim the unneeded files, and check the /etc/rc.d scripts. Some important points:

- One important thing is eth0 setup. The workstation comes up with a, at least partially, setup eth0. Setting up the IP-address of the workstation to the the IP-Address of the server is not considered a clever thing to do. (As it happened to the author on one of his early attempts.)

- Another point is the /etc/fstab of the workstation. It should be setup for nfs filesystems.
- **WARNING**: Don't confuse the server root filesystem and the workstation root filesystem. (I've already patched up a rc.inet1 on the server, and wondered why the workstation still didn't work.)

Exporting the filesystem

Export the root dir to the work station. See `exports(5)`. You most likely will have to restart the nfsd/mountd after this change. Under RedHat this can easily be done by typing `/etc/rc.d/init.d/nfs stop ; /etc/rc.d/init.d/nfs start` .

RARP setup

Setup the *RARP* somewhere on the net. If you boot without a nfsroot parameter, the *RARP* server has to be the *NFS* server. Usually this will be the *NFS* server. To do this, you will need to run a kernel with *RARP* support.

To do this, execute (and install it somewhere in `/etc/rc.d` of the server!):

`/sbin/rarp -s <ip-addr> <hardware-addr>`

where

ip-addr

is the IP address of the workstation, and

hardware-addr

is the Ethernet address of the network card of the workstation.

example: `/sbin/rarp -s 131.131.90.200 00:00:c0:47:10:12`

You can also use a symbolic name instead of the IP-address, as long the server is able to find out the IP-address. (/etc/hosts or *DNS* lookups)

BOOTP setup

For *BOOTP* setup you need to edit `/etc/bootptab`. Please consult the *bootpd(8)* and *bootptab(5)* man pages.

Finding out hardware addresses

I don't know the hardware address! How can I find it out?

- Boot the kernel disk you made, and watch for the line where the network card is recognized. It usually contains 6 hex bytes, that should be the address of the card.
- Boot the workstation with some OS with TCP/IP networking enabled. Then ping the workstation from the server. Look in the ARP-cache by executing: `/sbin/arp -a`

---

4. Booting the workstation

4.1 Using a boot rom

As I have not used such a beast myself yet, I can give you only the following tips (courtesy of Christian Leutloff `<leutloff@sundancer.tng.oche.de>`):

- You can't use ``normal'' bootroms.
- There is a `netboot` packet by Gero Kuhlmann, that provides for bootroms for Linux, and further information. `netboot` is available from the local Linux mirror, or as a Debian package (`netboot-0.4`).
- Read the documentation coming with your boot rom carefully.
- You probably will have to enable the tftpd on the server, but this depends upon your boot rom's way of loading the kernel.
- *Any informations on bootrom vendors of these Linux variety, mentioned above, as not everybody has access to prom burner :((especially in europe, as I'm located there.) welcome, I'll include them then here.*

4.2 Using a raw kernel disc

If you have exported the root filesystem with the correct name for the default naming and your *NFS* server is also the *RARP* server (which implies that the boxes are on the same subnet.), than you can just boot the kernel by `cat`ing it to a disc. (You have to set the root device in the kernel to 0:255.) This assumes, that the root directory on the server is `/tftpboot/`*IP-Address* (this value can be changed when compiling the kernel.)

4.3 Using a bootloader & *RARP*

Give the kernel all needed parameters when booting, and add `nfsroot=<`*`server-ip-addr`*`>:<`*`/path/to/mount`*`>` where *server-ip-addr* is the IP-address of your NFS-server, and */path/to/mount* is the path to the root directory.

Tips:

- When using **LILO** consider using the ``lock'' feature: Simply type in once all the correct parameters and add ``lock''. Next time when booting let LILO timeout.
- When generating a workstation specific boot disk, you can also use the `append=` feature in `lilo.conf`.

4.4 Using a bootloader without *RARP*

TOC

In addition to `nfsroot` give a `nfsaddrs=<wst-IP>:<srv-IP>:<gw-IP>:<netm-IP>:<hostname>` commandline argument for the kernel. The kernel will setup `eth0` with the given parameters:

wst-IP

machine's IP-Address

srv-IP

NFS-server IP-Address

gw-IP

gateway

netm-IP

netmask

hostname

machine name

---

5. Known problems

5.1 /sbin/init doesn't start.

A popular problem with /sbin/init is, that some (at least) current distributions come with /sbin/init dynamically linked. So you have to provide a correct /lib setup to the client. One easy thing one could try is replacing /sbin/init (for the client) with a statically linked ``Hello World'' program. This way you know if it is something more basic, or ``just'' a problem with dynamic linking.

5.2 /dev troubles.

If you get some garbled messages about ttys when booting, then you should run a MAKEDEV from the client in the /dev directory. There are rumors that this doesn't work with certain server oses which use 64-bit dev numbers, should you run into this, please mail me with which os you have the troubles. A potential solution would be to create a small /dev ram disc early in the boot process, and reinstall the device nodes each time.

6. Other topics

- There is BOOTP client:
  `ftp://sunsite.unc.edu/system/Network/admin/bootpc.v045.tgz`

  With initrd (which is included in Linux 2.0), it could be made to work for diskless stations quite nicely. initrd is actually always an advanced option for more customized setups.
- For plain bootpd based boots this is actually probably not needed as Linux 2.0 contains also the option to use BOOTP instead of RARP. (To be more precise, you can compile both in the kernel, and the faster response wins.)
- In the Documentation directory of kernel source there is a file documenting NFS-Root systems.
- There is a patch floating around, that allows for swapping over NFS. It was send to me (during a private high workload phase), but I somehow managed to loose the mail. :(

  You can get it probably from http://www.linuxhq.com/ in the unofficial patches section.
- My PGP public key can be fetched by fingering andreas@ag.or.at. The fingerprint is: F1 F7 43 D5 07 C4 6C 87 BF 6B 33 A2 2C EE 5A F9.

NFS-Root-Client Mini-Howto

Ofer Maor, *oferm@hcs.co.il*

V4.1, 02 February 1999

TOC

The purpose of this Mini-Howto is to explain how to create client root directories on a server that is using NFS Root mounted clients. You can find the latest version of this Howto (HTML or Text) at URL: http://www.hcs.co.il/oferm/NFS-Root-Client/

1. Copyright

(c) 1996 Ofer Maor (oferm@hcs.co.il)

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Ofer Maor (*oferm@hcs.co.il*), the author of this mini-HOWTO, or Greg Hankins, the Linux HOWTO coordinator, at *gregh@sunsite.unc.edu* via email, or at +1 404 853 9989.

If you have anything to add to this Mini-Howto, Please mail the author (Ofer Maor, *oferm@hcs.co.il*), with the information. Any new relevant information would be appreciated.

1.1. Thanks

I would like to express my thanks to the author of the NFS-Root Howto, Andreas Kostyrca (*andreas@medman.ag.or.at*). His Mini-Howto helped me with the first steps in creating a NFS Root Mounted client. My Mini-Howto does not, in any way, try to replace his work, but to enhance it using my experiences in this process.

I would also like to thank Mark Kushinsky (*mark026@ibm.net*) for polishing the english and spelling of this Howto, thus making it much more readable.

2. Preface

This Mini-Howto was written in order to help people who want to use NFS Root mounting to create their client's directories. Please note that there are many ways to accomplish this, depending on your needs and intent. If the clients are individual, and each client has its own users and administrator, it will be necessary to make significant parts of the client dirs not shared with other clients. On the other hand if the client is intended for multiple users, and are all administrated by the same person (for instance, a

computer class), make as many files as possible shareable in order to make administration more manageable. This Howto will focus on the second issue.

TOC

2.1. General Overview

When building a client's root directory, and trying to limit ourselves to the minimum client size, we mainly focus on which files can we share, or mount from the server. In this Howto I will recommend the configuration of a client based on my experience. But before we begin please note:

- This Mini-Howto does not explain how to do the actual NFS Root mounting. Reffer to the NFS-Root Mini-Howto if you need more information about that issue.
- I based most of my client's configuration on mounts and symbolic links. A lot of those symbolic links can be replaced by hardlinks. One should choose according to his personal preference. Putting a hardlink over a mount and a symbolic link has its advantages, but might cause confusion. A file will not be erased until all its hardlinks are removed. Thus, In order to prevent a case in which you upgrade a certain file, and the hardlinks still refer to the older version, you will have to be very careful and keep track of every link you put.
- While mounting the information from the server, two concepts can be used. The first (most common) concept, is to mount the whole server root directory under a local directory, and then just change the path or link the relevant directories there. I personally dislike mounting root partitions of a server on clients. Thus, this Howto suggests a way to mount the relevant directories of the server to the relevant places on the system.
- This Howto is based on my experience building client directories on a Slackware 3.1 based distribution. Things may be different (especially on the rc.* files), for other users, however the concepts should still remain the same.

3. Creating the client's root directory

3.1. Creating the directory tree

First of all, you need to create the directory structure itself. I created all the clients under /clients/hostname and I will use it for my examples listed below. This, however, can be changed to anything else. The first stage, then, is to create the relevant directories in the root directory. You should create the following directories:

```
bin, dev, etc, home, lib, mnt, proc, sbin, server, tmp, usr, var
```

and any other directories you might want to have on your system.

The local, proc, and dev directories will be used separately on each machine while the rest of the directories will be either party or completely shared with the rest of the clients.

3.2. Creating the minimal file system needed for boot

3.2.1. Creating the dev dir.

Although the dev dir can be shared, it is better to create a separate one for each client. You can create your client's dev directory with the appropriate MAKEDEV scripts, however in most cases it is simpler just to copy it from the server:

```
cp -a /dev /clients/hostname
```

You should keep in mind that /dev/mouse, /dev/cdrom and /dev/modem are symbolic links to actually devices, and therefore you should be sure that they are linked correctly to fit the client's hardware.

3.2.2. Copying the necessary binaries.

Although we mount everything from the server, there is a minimum that we need to copy to each client. First of all, we need "init", our system will not be able to run anything before init'ing (as the author found out in the hard way ;-). So first, you should copy /sbin/init to your client's sbin dir and then so that rc.S will run, you should copy /bin/sh to the client's bin directory. Also, in order to mount everything you need to copy /sbin/mount to the client's sbin directory. This is the minimum, assuming the first line in your rc.S is mount -av. However, I recommend copying a few more files: update, ls, rm, cp and umount, so that you will have the basic tools in case the client has problems mounting. If you choose to leave your swap on line before mount, you should also copy the swapon binary.

Since most of these binaries are by default dynamically linked, you will also need to copy a fair part of /lib:

```
cp -a /lib/ld* /lib/libc.* /lib/libcursses.* /client/hostname/lib
```

Hardlinking the binaries themselves, instead of copying them, should be considered. Please read my comments on this in part 2.1 of this Howto.

Please notice, all of the information above assumes that the kernel has been given the network parameters while booting up. If you plan to use rarp or bootp, you will probably need the relevant binaries for these as well.

Generally, you will need the minimum of files that will enable you to configure the network and run rc.S up to the point where it mounts the rest of the file system. Make sure you looked into your /etc/init and rc.S files, making sure there are no "surprises" in any of them, which will require other files to be accessed, before the first mount will take place. If you do, however, find such files, you can either copy them as well, or remove the relevant parts from your init and your rc.S files.

3.2.3. The var directory

The var directory, in most cases, should be separate for each client. However, a lot of the data can be shared. Create under the server directory a directory called var. We will mount the server's var directory there. To create the local var directory, simply type:

```
cp -a /var /clients/hostname/
```

Now, you have a choice as to what you want to separate, and what you want to share. Any directory/file that you want to share, simply remove it from the client's var dir, and symlink it to the /server/var/ directory. However please note that you should either symlink it to /server/var or to ../server/var but NOT to /clients/hostname/server/var as this will not work when the root changes.

Generally, I would recommend separating /var/run, /var/lock, /var/spool, and /var/log.

3.2.4. The rest of the directories

- etc is explained thoroughly in the next section.
- mnt and proc are for local purposes.
- usr and home are merely mount points.
- tmp is up to you. You can create a different tmp directory for each client, or create some /clients/tmp directories, and mount it for each client under /tmp. I would recommend that you provide each client with a separate tmp directory.

3.3. Building the etc directory and configuring the clients

Please Note - this section refers to building the etc directory which is mostly shared among the clients. If your diskless clients have separate system administrators, it's best to set up for each client a separate etc directory.

3.3.1. Building a clients wide etc directory

Although we separate the etc directories of the clients, we still want to share a large portion of the files there. Generally, I think sharing the etc files with the server's /etc is a bad idea, and therefore I recommend creating a /clients/etc directory, which will hold the information needed for the clients. To start with, simply copy the contents of the server's etc to the /clients/etc directory.

You should add to this directory all of the non-machine-specific configuration files, for instance motd, issue, etc. and not the client specific ones.(i.e. inittab or fstab)

The most important changes will be in your rc.d directory. First, you should change rc.inet1 to be suitable for your local setup. I pass all my network parameters to the kernel through the LILO/Loadlin, therefore I remove almost everything from rc.inet1 file. The only thing I leave there is the ifconfig and route of the localhost. If you use rarp or bootp, you will have to build it accordingly.

Secondly, you should edit your rc.S. First, remove all the parts that are responsible for the fsck check as fsck will occur when the server boots up. Then, you should find the line that mounts your fstab. This should look something like:

```
mount -avt nonfs
```

The -t nonfs is there since normal clients first run rc.S and only later on use rc.inet1 to configure the Ethernet. As this will cause no NFS partitions to be mounted this line should be deleted. Therefore, change it to mount -av. If you need to run rarp/bootp to configure your network, do it in rc.S (or call the appropriate script from rc.S), before the mount, and make sure your physical bin and sbin directories have the necessary files available.

After the mount -av is performed, you will have a working file system. Build a general fstab, so that you can later copy it to each client. Your fstab should look something like this:

```
server:/clients/hostname    /              nfs     default  1 1
server:/bin                 /bin           nfs     default  1 1
server:/usr                 /usr           nfs     default  1 1
server:/sbin                /sbin          nfs     default  1 1
server:/home                /home          nfs     default  1 1
```

```
server:/lib                 /lib            nfs     default  1
server:/clients/etc         /server/etc     nfs     default  1
server:/clients/var         /server/var     nfs     default  1 1
none                        /proc           proc    default  1 1
```

Please notice, that the keyword *deafult* might not work on all versions of mount. You might change it to *rw* or *ro* or remove all of the *default 1 1* part.

Also, make sure your server's /etc/exports looks like this:

```
/clients/hostname   hostname.domainname(rw,no_root_squash)
/clients/etc        hostname.domainname(ro,no_root_squash)
/clients/var        hostname.domainname(ro,no_root_squash)
/usr                hostname.domainname(ro,no_root_squash)
/sbin               hostname.domainname(ro,no_root_squash)
/bin                hostname.domainname(ro,no_root_squash)
/lib                hostname.domainname(ro,no_root_squash)
/home               hostname.domainname(rw,no_root_squash)
```

Other than the first line, which should be separate for each host, the rest of the lines can be replaced with a hostmask to fit all your hosts (like pc*.domain - keep in mind though, that * will substitue only strings without a dot in them). I suggest that you make most of the directories read only, but this is up to you. The no_root_squash will make sure root users on the clients have actual root permissions on the nfsd as well. Check out man exports(5). If you want users to be able to run passwd from the clients also, make sure the /etc has rw and not ro permissions. However, this is not advisable.

Please note another thing concerning the rc.S file. In Slackware, by default, it creates a new /etc/issue and /etc/motd every time it runs. This function MUST be disabled if these files are mounted ro from the server, and I would recommend that they should be disabled in any case.

Lastly, if you want to have the same userbase on the server as on the clients, you should choose between 1), using NIS (Yellow Pages - check the yp-howto), and then each client will have a separate /etc/passwd and /etc/group as it receives it from the NIS server. 2) In most cases, a simple symbolic link will suffice. Therefore, you will need to either hardlink /clients/etc/passwd to /etc/passwd, or if you prefer a symlink, link /etc/passwd to /clients/etc/passwd (and not the other way around, since the clients do not mount the server's etc directory). Do the same for /etc/group.

3.3.2. Creating a client's etc directory

Generally, most of the files in the client's etc should be symlinked to the /server/etc directory. However, some files are different for each machine, and some just have to be there when the kernel loads. The minimum you need from the etc dir is as follows:

```
resolv.conf
hosts
inittab
rc.d/rc.S
fstab
```

Since these 5 files can be identical on all clients, you can simply hardlink them or copy them again. However, with the rc.S and fstab file it is advised to keep a separate copy for each client. You will also need a separate etc/HOSTNAME for each client. I personally recommend having all of the rc.d files separate for each client, as configurationand hardware might vary from one to another.

TOC

For each client, add to the fstab the proper swap line:

```
/dev/swap_partition                    swap    swap    default  1 1
```

The rest of the /etc files of the client, you can either hardlink to the /clients/etc/* files, or symlink them to the /server/etc (which is the mount point of /clients/etc/).

Make sure your machine can resolve properly, either through a named or through etc/hosts. It is not a bad idea to keep the server's IP in the etc/hosts, instead of counting on resolving. If you will count only on named resolving, a problem in the named will prevent your clients from booting up.

3.4. Booting Up

Now, all you have to do is to boot up your machine, cross your fingers and hope everything works as it should :-).

4. Creating more clients

If you have followed my instructions so far this should be simple - cd to /clients/ and type:

```
cp -a hostname1 hostname2
```

and then make sure you check these points:

rc.d/* files matches the hardware and wanted software configuration etc/HOSTNAME is correct, fstab's swap line is correct, the symbolic links of dev/mouse, dev/modem and dev/cdrom are right.

Good Luck....

Netrom-Node mini-Howto

Karl Larsen, k5di@yahoo.com

v1.10, 19 October 1998

This document describes how to setup the ax25-utilities package for Amateur Radio such that it makes Netrom Nodes for the Node program and the BBS software from John-Paul Roubelat, F6FBB. The DOS G8BPQ Switch makes a bbs node and many features, it was expected that the Linux ax25-utils would have a similar capability. This was not the case. Help came from John Ackerman, N8UR who put a message on the Linux-Ham SIG that he had done the BBS node and the info was on his web site! When the information was tried it didn't work properly but much was learned about the technique. Help from Tomi Manninen, OH2BNS did the trick. Nodes for the BBS and the Node and the DX Cluster were made and work fine.

1. Introduction

2. How to Begin

3. Some Details of the AX.25 Utilities

4. Setting Up Netrom

5. Setting Up FBB and DXNet:

1. Introduction

It is possible, using just the ax25-util's to generate node listings for the Node application and the FBB BBS and the DXNet DX Cluster. This is done by changing the configuration files for Netrom and making a Netrom entry for each application. At present there is a kernel imposed limit of 4 Netrom entries. The new kernels are expected to drop this limit.

Now users look for CRUCES:K5DI-4 and LCBBS:K5DI-3 and LCDX:K5DI-5 on the many nodes here in New Mexico, Texas and Arizona and are connected like magic. They no longer need to remember anything.

2. How to Begin

Obtain and read the AX25-HOWTO:

`ftp:`/sunsite.unc.edu/pub/Linux/docs/HOWTO/AX25-HOWTO/

Using the AX25-HOWTO set up the normal Amateur Radio ax.25 and Netrom system and make certain it is operating properly. When the software "CALL" can be used to make either a ax25 or Netrom connection to a distant node, the system is ready to change to one using node listings like the BPQ Switch.

3. Some Details of the AX.25 Utilities

Below is a list of all the applications and files that are needed to set up a working ax.25 system. The Applications are all in the /usr/sbin/ directory and the Configuration files are in the /etc/ax25/ directory. Note: Kissattach is used only if you have TNC's in the Kiss Mode.

- kissattach Application
- call Application
- ax25d Application
- ax25d.conf Configuration file
- axspawn Application
- axspawn.conf Configuration file
- axports Configuration file

There are several names that a ax25-util user must invent. Since this paper uses the files of k5di, a listing of those names can be made.

```
Name            Call-sign       Alias           Other

ax0             k5di-9                          ax25 9600 baud
ax1             k5di-10                         ax25 1200 baud
Netrom          k5di-1          #CRUCE          Real Netrom
netnod          k5di-4          CRUCES          Node node-list
netbbs          k5di-3          LCBBS           BBS  node-list
netdx           k5di-5          LCDX            DX-Cluster
```

It's a good idea to make a list like this on paper before you start to change things. It is easy to put the wrong name in a control file.

Kissattach is an application that connects the kernel to the TNC, sets the tcp/ip address up, sets the speed of the connection, and is given the serial port to use.

Axports is a file that defines the name of the ax.25 ports and tells kissattach what call-sign, baud-rate and window size to use. Below is an example of a 2 TNC system.

```
# /etc/ax25/axports
# Be very careful with the speed setting. This is the speed in
# bits/second that data passes from the computer to the TNC, and has
# nothing to do with the radio baud rate!
#
# The format of this file is:
#
```

TOC

```
# name call-sign speed paclen window description
#
ax0     K5DI-9          9600    255     3       445.1 (9600 bps)
ax1     K5DI-10         9600    255     1       145.07 (1200 bps)
```

AX25D is the application that reads the ax25d.conf Configuration file and answers calls made to the system. Below is a sample ax25d.conf that has no Netrom defined. In fact all it will do is answer calls to k5di-9 and k5di-10. When it answers it starts the node application and logs the caller in.

```
# /etc/ax25/ax25d.conf
#
#  AX25D Configuration File.
#
# AX.25 ports begin with a '['.
#
[k5di-4 VIA ax0]
default  * *    *   *   *    0 - root   /usr/sbin/node  node
[k5di-4 VIA ax1]
default  * *    *   *   *     0 - root  /usr/sbin/node  node
#
```

The next step is to get ax25d to answer a call to the alias CRUCES as well as the call-sign. This is easy to do and is shown below:

```
# /etc/ax25/ax25d.conf
#
#  AX25D Configuration File.
#
# AX.25 ports begin with a '['.
#
[CRUCES VIA ax0]
default  * *    *   *   *    0 - root   /usr/sbin/node  node
[k5di-4 VIA ax0]
default  * *    *   *   *    0 - root   /usr/sbin/node  node
[CRUCES VIA ax1]
default  * *    *   *   *    0 - root   /usr/sbin/node  node
[k5di-4 VIA ax1]
default  * *    *   *   *     0 - root  /usr/sbin/node  node
#
```

If you have trouble, as root kill ax25d if it is running and then at the prompt type ax25d &. As ax25d loads the ax25d.conf file it will print out any errors it finds. This print out is very accurate and tells you which row in the file is wrong.

A connect from any adjacent node to k5di-4 or CRUCES will connect to the k5di node. But Netrom is not transmitting a node listing for CRUCES or k5di-4. This is done by changing some Netrom Configuration files.

4. Setting Up Netrom

Netrom has applications and files that control it's function and to achieve the G8BPQ look and function we must use these in ways never intended. Below is a list of these components of Netrom:

- nrattach Application
- netromd Application
- nrports Configuration File
- nrbroadcast Configuration File

Nrattach is the application that works with the kernel and establishes the ports and tcp-ip used by Netrom. To use nrattach you place it in your startup file and the example looks like this:

```
/usr/sbin/nrattach -i 44.30.2.5 netrom
/usr/sbin/nrattach -i 44.30.2.5 netnod
```

Nrattach gets some of it's information from a configuration file called nrports. This file is shown below:

```
# /etc/ax25/nrports
#
# The format of this file is:
#
# name call-sign alias paclen description
#
netrom          K5DI-1  #CRUCE   235    Switch
netnod          K5DI-4  CRUCES   235    Real Node
```

There is no change to the nrbroadcast file so the remaining changes will be made to the ax25d.conf file. In this file you normally put the real netrom application called k5di-1, but since a call to k5di-1 or #CRUCE gets undesirable results, leave that entry out of ax25d.conf and a user will get just a "busy" when calling.

Instead put in the netnod and that will allow ax25d to answer a call to CRUCES. This is shown in the example below:

```
# /etc/ax25/ax25d.conf
#
#  AX25D Configuration File.
#
# AX.25 ports begin with a '['.
#
[CRUCES VIA ax0]
default  * *    *   *   *    0 - root   /usr/sbin/node  node
```

TOC

```
[k5di-4 VIA ax0]
default  * *    *   *   *    0 - root   /usr/sbin/node  node
[CRUCES VIA ax1]
default  * *    *   *   *    0 - root   /usr/sbin/node  node
[k5di-4 VIA ax1]
default  * *    *   *   *     0 - root  /usr/sbin/node  node
#
# NET/ROM ports begin with a '<'.
#
<netnod>
default         * *     *   *   *   * -  root   /usr/sbin/node node
#
```

With these changes netrom node broadcasts will include the node K5DI-4:CRUCES and K5DI-1:#CRUCE. By testing it was determined that a call from any node to k5di-1 or #CRUCE got a busy, and a call to k5di-4 or CRUCES connected to the node on this system.

---

5. Setting Up FBB and DXNet:

The FBB packet BBS and DXNet Linux software are written to answer calls to a call-sign defined in the configuration files. In these examples the FBB call-sign is k5di-3 and the DXNet is k5di-5.

Since calls to k5di-3 and k5di-5 are answered by other software, ax25d is not used and these calls should NEVER be found in a ax25d.conf file. But the nrports file needs to have the information added and 2 more nrattach lines are added to the start file. The nrattach lines (4 each) and the file "nrports" are shown below:

```
/usr/sbin/nrattach -i 44.30.2.5 netrom
/usr/sbin/nrattach -i 44.30.2.5 netbbs
/usr/sbin/nrattach -i 44.30.2.5 netnod
/usr/sbin/nrattach -i 44.30.2.5 netdx

# /etc/ax25/nrports
#
# The format of this file is:
#
# name call-sign alias paclen description
#
netrom          K5DI-1  #CRUCE   235    Switch
netnod          K5DI-4  CRUCES   235    Real Node
netbbs          K5DI-3  LCBBS    235    FBB BBS
netdx           K5DI-5  LCDX     235    DXNet DX Cluster
```

These changes will make the node listings wanted but a call to LCBBS will not work yet. Recall that FBB answers a call to k5di-3 but not the alias. To achieve this a change to the /usr/local/fbb/system/port.sys file is required. Before these changes port.sys had a listing for the name "netrom". With these changes replace "netrom" with "netbbs". That section of port.sys is shown below:

```
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode  Freq
 0   0    0   0      0     0     0     0      00/01   ----  File-fwd.
 1   8    1   ax0    250   4     1     10     30/60   XUWY  UHF port
 2   2    1   ax1    250   4     1     10     00/60   XUWY  VHF port
 3   6    1 netbbs   250   4     4     10     30/60   XUWY  BPQ look
 4   8    2   0      250   5     4     1000    5/15   TUWY  Telnet
#
```

A similar change is made to the "dxnet.cfg" file where netrom is replaced with netdx. When these changes are made and a few hours have passed to let Netrom send node lists, any nearby node will have nodes listed to your Netrom for CRUCES and LCBBS and LCDX, and they will all work just as they do when using the G8BPQ Switch under DOS.

HOW-TO for inHouse IntraNet

Author: Sarma Seetamraju
EMail: (sarma@usa.net)

Date: August 1997

Place: on the Amtrack and Path Trains on the way to Downtown Manhattan.
Notepad Used: a 16-Mhz 386 SX circa 1991 Magnavox notebook running linux.
(Just to show that if you ever complain about linux not running AS IS on your computer, I am going to shove that computer up your ....)

Reformatted as HTML for: All abnormal people who cannot stand illegible plain text.

Important: SOCKS is a FREE package for UNIX systems. I doubt its available for OTHER platforms. If you wish to influence NETSCAPE to keep supporting SOCKS, email me with your supporting statement (saying how you are using SOCKS).

This document describes the procedure to set up a NETWORK (INTRANET) at your home. Then we shall setup up the network such that NETSCAPE Clients can be used on ANY machine to access the internet...

The network we are talking about,

- has TWO or more computers...
- wherein, there is ONE (ONLY ONE) linux machine
- and the rest are Win95 or WinNT machines. (I doubt things will be any different for a MACintosh).
- Only the LINUX machine has PPP access to the internet. The other machines MAY have modems. I shall ignore those modems.
- if ANY of your clients are UNIX machines, you are perhaps better off reading the "sockd" package's documentation, since you may be needing the use of "rlogin", "ftp" etc... from within the UNIX CLIENTS. This document will not help you in that aspect.
- The computer network is assumed to be TCP/IP over ethernet. No netbeui, etc...

The "single linux" machine will be referred to as the "LINUX SYSTEM". while all others are referred to as "OTHER MACHINES" or also as "CLIENT MACHINES". The linux machine is also referrred to as the "SERVER" sometimes.

If you do not understand the next para, then jump to the FOR NETWORK NOVICES ONLY section. Then come back here...

All of the following assumes that there is an IP address assigned (using "ifconfig") to the eth0 port of your LINUX server.

Also, matter, this document does not restrict you to PPP only (it could be SLIP, PPTP, etc...) The IP address of the "ppp0" port is absolutely irrelevant. This document assumes you have one such port, and that its UP.

WHY WE NEED SUCH AN ARRANGEMENT:

- The linux machine is to be used to connect to the world. Only the linux machine has a REAL-WORLD IP address. (see below). (see "ON-LINE services" section below). The linux machine has a non-persistent PPP link to the world.
- The other machines in the network have IP addresses that are either invalid or are unknown to the world.

TOC

- You need to use the "Other Machines", and NOT the linux machine to access the internet, VIA NETSCAPE ONLY
- I have no need to "telnet" or FTP directly from the "other m/c" to the world. If I ever need to, I telnet manually into the linux machine, and then into the world.

I did NOT want to spend much on a linux m/c that didn't run an X server (much less any X applications). I bought a 486Dx/4 100 Mhz PCI board (since I didn't want to be stuck with plain ISA slots), with a $20 SVGA card and a $20 NE2000 compatible card, and an extra 20$ for terminators+co-ax ('cos I didn't know how to convert a regular Ethernet Hub connecter into a pt-2-pt connector).

And $90 worth of memory (it went all the way to $60 for 16megs) and I had a fully functional linux system for $270. Don't intend to burden that system with NT or any other memory-disk-cpu hogging OSes. Ofcourse, my client machine is a 32-meg P100 machine with two hard disks (one of which was transplanted as a linux machine's HD) and runs 95.

The linux system is sitting on top of a clean PizzaHut pizza box. I couldn't affort another $50 for a tower, since I was getting a power supply module from one of my friends.

The reason I chose NETSCAPE is that I no longer use FTP manually. Its simply, out of fashion. Every ftp site, worth its name and every company, has a web site that lets me use the Netscape browser to access their ftp site. I do need to telnet frequently, but go thru the trouble of going from my win95/winNT4.0 machine to the linux machine, and from there... Secondly, I am hooked to QuickTime and all those net audio sites. And LINUX versions of those tools, do not exist. So, I have to run stuff from Netscape ON windows platforms. And my LINUX machine recvs EMail using sendmail... (remember, I have a fixed IP address. Such fixed IP addresses are better ONLY for things like recving email. Its no benefit for Surfing, Telnetting, etc...)

Lastly, we will never have a "Microsoft Explorer Browser" for linux, and hence I never even considered using Explorer. Also something tells me that its NOT going to be this easy to configure the Explorer as it was to configure the NETSCAPE on the CLIENT machines (i.e., the other machines).

PROXY SERVER

I am not giving directions to installing a PROXY server. This is about installing a "socks" facility on the LINUX machine, which NETSCAPE on the client machines can use to access the internet. NETSCAPE (as far as I know) is the ONLY application that runs on NON-UNIX machines and is aware of the SOCKS facility.

INTERNET Addresses

If you have a TCP/IP network, then you MUST have ATLEAST two IP addresses for the machines (one for the LINUX machine and another for the one of the Client Machines, and more IP addresses if you have more than one client machine).

Read the other HOW-TOs on how to assign IP addresses to ALL your machines on the TCP/IP network. (ESPECIALLY IF YOU DO NOT have a REGISTERED internet domain).

I created a network 10.0.1.x out of the single LINUX machine and the single Win95 machine. They were assigned 10.0.1.1 and 10.0.1.2 respectively. The 10.0.1.1 is the IP address of the ETHERNET port (eth0) of the LINUX machine. The ppp0 port has another IP (which [lucky me] has a fixed IP address). That IP address is irrelevant to us, and also, being withheld for security reasons.

I also have a fixed domain name server on the other end of the PPP link. (University machine).

The linux machine has a modem and CRONTAB entries, that automatically dial up to the internet at fixed times daily. I also manually connect to the internet, when I want to go surfing.

If you connect to the internet via ON-LINE services, see below...

ON-LINE services

If you connect to the world using ON-LINE services like AOL, Compuserve, Sprynet, Netcom etc... then you may NOT have a fixed IP address. That is of little relevance in getting your intrAnet hooked up to the world. If you do not believe that, I request you to read on... and become a believer...

Some Background Information

(For those who are like me and want to know what the hell is happening...). Others may skip this section....

1. ... since you have ONLY one ethernet network, you do NOT need routing within that network. And you perhaps have manually hardcoded the IP addresses (10.0.1.1, 10.0.1.2) of ALL your machines in /etc/hosts. If you did that, you are smart person. Using "named" for a two or three computer network at home, is like using the bulldozer instead of a spoon to eat.
2. What we would ideally like, is for ALL IP packets from the client machines to go to the LINUX machine, which will then route accordingly. Problem with this, you are exposing your computer to hackers because if the LINUX machine routes, you DO NOT have firewalling or proxy or whatever. Here in this document, we will do firewalling unintentionally! while trying to get NETSCAPE to access internet from the client machines.

   One problem with this "re-routing" desired from the linux machine, is that the clients MUST actually SEND ALL packets to the linux machine, no matter what the destination address. To that end, Win95 and WinNT will ONLY allow "proxy servers" (Which I intend to figure out, and write another document on).
3. IF you are well versed with various free utils, you may have heard of "term" package. It was designed simply because its easier to configure networks being a "simple" user and NOT AS A ROOT/ADMIN (on both client and server sides). The same logic goes with NETSCAPE on the clients. It is easier to JUST GET the netscape to access the internet and leave the rest of the features (FTP, TELNET) unsupported.
4. If you think, having ONLY netscape access and NOT telnet / ftp access to internet from the client machines, is a bummer, then you are a dinosaur. Wake Up, Mr./Ms. Rip Van Winkle.
5. (TECHNICAL) The "named" which remained unused (as mentioned above) will be put to use to support NETSCAPE (so that http:/www.sex.org will be resolved right from the client machine).
6. (TECHNICAL) You will have to REBUILD your LINUX kernel to disable IP forwarding. I intend to rebuild my kernel with forwarding ENABLED and see if the socks package still works (I am betting it will). If it does, then you will find a newer version of this document. (What this means, is that, you can use the kernel installed by your favorite LINUX installation package).
7. You will need ROOT access on the linux machine :-) You will need to download the socks package and COMPILE it. It will NOT compile 'cos the MAKE file is bad.
8. (TECHNICAL) be prepared to edit the socks.c file, to comment out ONLY two lines which place an entry in your syslog file (/var/adm) for every data transfer via socks. For eg: a single page on WWW.CNN.COM will have 10 pictures atleast and 5 separate text objects. For each of them you will find an entry in syslog (that it was transferred!). My syslog keeps filling up. I do NOT like that. Maybe you might not mind.
9. This sockd package supports CLIENT machines ONLY. All applications on the LINUX machine DO NOT need the sockd or any other package to access the internet, since this LINUX machine connects to internet directly using PPP.

PREPARING YOUR LINUX MACHINE

Read the NET-HOW-TO in `/usr/doc/faq/howto` on your linux machine (if its slakware), or go to the `www.linux.org` and read the same NET-HOW-TO there...

In that you will find how to down load the socks package and compile it. You NEED TO READ the instructions there to setup the in-house network. But you are welcome to read this :-) .

That document spends a lot of time, explaining how to configure UNIX clients. Especially for "rlogin" "telnet" "ftp" etc... If you do not have UNIX clients, then after compiling the SOCKS package, start reading this document again, for using the socks

package rather than the readme file in that package.

I placed the tar file in `/usr/local/ProxyServer` and untarred it, creating a "sockd4.2b" subdirectory within which there is a "Makefile". As mentioned in the howto document, I had to struggle to successfully do a make on the MAKEFILE.

For your convenience, the MAKEFILE is included at end...

Hopefully, you will have change line # 9 of my copy of the Makefile, only.

Then I moved the sockd directory contents into its parent and changed the line # 9 and did a make again -- successfully. So I guess I "fixed" the MAKEFILE.

Setting UP the sockd daemon

You will find an executable called "sockd" in the sockd subdirectory.

```
Once you are done compiling, COPY the following files to /usr/local/etc
(They SHOULD be in the same dir as the sockd directory)
        sockd                   (The executable a.k.a daemon)
        sockd.conf              (configuration file)
        sockd.route             (configuration file)
        socks.conf              (configuration file)

# ### make a link called "socks" which points to "sockd" within the same dir.
# cd /usr/local/etc
# ln -s sockd socks
```

Then edit those three configuration files so that they are similar to the ones given below (these are my settings for a two computer network, made up of a LINUX "server" and a Win95/WinNT client machine).

My sockd.conf file

```
permit  10.0.1.2    0.0.0.0
deny    0.0.0.0 0.0.0.0 : /usr/ucb/finger @%A | /usr/ucb/mail -s 'SOCKD: rejected --
from %u@%A to host %Z (service %S)' root
#BAD_ID: /usr/ucb/finger @%A | /usr/ucb/mail -s '%U pretends to be %u on host %A'
root@%A root
#NO_IDENTD: /usr/ucb/mail -s 'Please run identd on %A' %u@%A root@%A
#[EOF]
```

NOTE: 10.0.1.2 is my Win95/WinNT client machine's IP address. This sockd.conf file MUST be on your LINUX server (in my case that the ethernet port of the LINUX server has an IP address = 10.0.1.1)

NOTE: This sockd program is for CLIENT machines ONLY. All applications on the LINUX machine DO NOT need the sockd or any other package to access the internet, since this LINUX machine connects to internet directly using PPP.

My sockd.route file

```
#! NoShell
10.0.1.1        10.0.1.0        255.255.255.0
```

#[EOF]

NOTE: The first IP address is the address of the LINUX machine's eth0 PORT. The second IP number is NOT an IP add
Its the NETWORK address (basically, convert the last of the FOUR numbers of the IP address into a ZERO).

TOC

My socks.conf file

```
direct  127.0.0.1       255.255.255.255
direct  10.0.1.1        255.255.255.255
direct  10.0.1.2        255.255.255.255
sockd   @=199.99.99.99  10.0.1.1 0.0.0.0
```

Now to configure the LINUX machine

- **Step # 1:** Check to see if "named" is already running in your system. If it is -- then, you are on your own. Unless you know the concepts of DNS very well, you may not be able to adapt the contents of this document to suit your needs.
- **Step # 2:** Copy the "named.boot" file given below into your machine.
- **Step # 3**: copy the "root.cache" file given below into your machine (follow instructions that come with it).

/etc/host.conf file

All programs that run on the LINUX machine WILL (you cannot prevent that) use the resolver libraries -- which depend on the file /etc/host.conf

You must make sure that NONE of these programs ever access the "named" daemon on THAT VERY linux machine. To do that we shall specify to the resolver routines (i.e., routines which convert www.cnn.com into the numerical ip address) that those resolver routines MUST either check the /etc/hosts file and then check the DNS servers mentioned in /etc/resolv.conf

How do we do that? Simply, make sure the /etc/host.conf file is :-

```
order hosts, bind
multi on
```

If there is anything else, remove it, unless you know a lot about DNS and "named".

The reason I insist on preventing the LINUX machine's applications from accessing its own "named" server, is because it makes no sense. And from my experience, such a "unnecessaries" may look technically safe and harmless but will cause enough grief sooner or later...

The linux machine is obviously doing just great accessing the internet via the PPP (or whatever link) link. We are installing "sockd" package and the "named" daemon for the client machines. Let's not disturb the LINUX system.

You DO NOT NEED to change the `"/etc/gateways" or "/etc/hosts" file or the "hosts.allow" or the "hosts.deny"` file in order to get your socks working.

Do not change any file unless someone suggests a change to that file...

I will also assume that you have setup "resolv.conf" properly, to enable your LINUX server to access the internet and the DNS (on the "other end" of the PPP connection). My sample resolv.conf file is available as a sample at the very end.

```
**********************************************
                 WARNING
**********************************************
For your own good, I suggest that you setup your
```

```
machine through the linux installation programs
(i.e., while installing linux on your computer.)
**************************************************
```


My named.boot file

```
;    boot file for name server
forwarders 128.112.129.111
directory /etc
cache      .       root.cache
primary    1.0.10.in-addr.arpa   named_DNS_for_inTi_xwk
                                 ^^^^^^^^^^^^^^^^^^^^^^
```

NOTE: line # 2, contains the IP address of the DNS server in the network to which your LINUX machine connects to using PPP(or whatever).

*** How to determine this IP Address ****

SIMPLE ! on a command prompt type in the command "nslookup". The response you see will CONTAIN such an IP address. (After noting the DNS' IP address, exist "nslookup" using <CTL-D>.

NOTE: The LAST line contains the **name of a file** called "named_DNS_for_inTi_xwk' which MUST be in the "/etc" directory. The contents of this file, is given below (you are free to give it a better name :-))

My "named_DNS_for_inTi_xwk" file

```
@                IN      SOA     10.0.1.0 hostmaster.10.0.1.0 (
                                 1       ; Serial
                                 28800   ; Refresh
                                 7200    ; Retry
                                 604800  ; Expire
                                 86400)  ; Minimum TTL
                         NS      10.0.1.1
1                        PTR     MyLinuxMachine
```

NOTE: The last line (starts with a 1) contains the name "MyLinuxMachine". replace it with the name in /etc/HOSTNAME. **NOTE**: Again , as you have been doing so far, replace "10.0.1.1" with that of your LINUX machine eth0 port's address, and replace "10.0.1.0" with that of the network address of that port. **NOTE**: I really do not understand every character of the above file. You will be better off statisfying your curiousity by studying the documentation for the NAMED daemon.

My root.cache file

To get this file read the NET-HOW-TO documentation and the documentation that comes with the socks package.

Those instructions are VERY simple...

All I did was to run a command mentioned there, and redirected it into a file and called it `"/etc/root.cache"`

```
; <<>> DiG 2.1 <<>> ns
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr rd ra; Ques: 1, Ans: 9, Auth: 0, Addit: 9
```

TOC

```
;; QUESTIONS:
;;      ., type = NS, class = IN

;; ANSWERS:
.       137030  NS      C.ROOT-SERVERS.NET.
.       137030  NS      D.ROOT-SERVERS.NET.
.       137030  NS      E.ROOT-SERVERS.NET.
.       137030  NS      I.ROOT-SERVERS.NET.
.       137030  NS      F.ROOT-SERVERS.NET.
.       137030  NS      G.ROOT-SERVERS.NET.
.       137030  NS      A.ROOT-SERVERS.NET.
.       137030  NS      H.ROOT-SERVERS.NET.
.       137030  NS      B.ROOT-SERVERS.NET.

;; ADDITIONAL RECORDS:
C.ROOT-SERVERS.NET.     410161  A       192.33.4.12
D.ROOT-SERVERS.NET.     410161  A       128.8.10.90
E.ROOT-SERVERS.NET.     410161  A       192.203.230.10
I.ROOT-SERVERS.NET.     167767  A       192.36.148.17
F.ROOT-SERVERS.NET.     410161  A       192.5.5.241
G.ROOT-SERVERS.NET.     410161  A       192.112.36.4
A.ROOT-SERVERS.NET.     410161  A       198.41.0.4
B.ROOT-SERVERS.NET.     410161  A       128.9.0.107
H.ROOT-SERVERS.NET.     410161  A       128.63.2.53

;; Total query time: 334 msec
;; FROM: svathyam to SERVER: default -- 128.112.129.111
;; WHEN: Sat Sep 28 21:38:04 1996
;; MSG SIZE  sent: 17  rcvd: 312
```

Add this to /etc/services

(add the single line starting with "socks"...)

```
# services      This file describes the various services that are
#               available from the TCP/IP subsystem.  It should be
#               consulted instead of using the numbers in the ARPA
#               include files, or, worse, just guessing them.
# Version:      @(#)/etc/services       3.02    02/21/93
# Author:       Fred N. van Kempen,

... &LT;lines delete&GT;

socks           1080/tcp                        # sarma: Sep.15.96: Got this from the
~sockd/include/socks.h file.

... &LT;lines delete&GT;

# End of services.
```

NOTE: This line is read ONLY by inetd daemon I think. This tells the inetd to invoke the "socks" program for all tcp connections to the port # 1080.

Add this to your /etc/inetd.conf file

TOC

```
# I am just following instructions from ~sockd/doc/sockd.1 man pages...
socks   stream  tcp     nowait  root    /usr/local/etc/socks
```

- NOTE: Make sure /usr/local/etc is in the SYSTEMS's default PATH.
- NOTE: For more instructions, read the SOCKD package's instruction file. In that this very same line is mentioned, and also you will get to know what it means...
- NOTE: As the filename indicates, this file tells the "inetd" daemon where it can find the "socks" program, and what arguments to pass it (always) etc...

LET'S GET STARTED !

Now reboot your system (if you know how, you may instead send HUP to the appropriate daemons). Your LINUX server is now set.

do a `"tail -f /var/adm/messages" and a "tail -f /var/adm/syslog"` simultaneously and attempt to connect using NETSCAPE from your CLIENT machines.

Now let's configure the client machine's Netscape...

Configuring NETSCAPE 3.0 Client

This is to be DONE on the client machines ONLY
DO NOT bother doing this on the LINUX server.

- Pull down the menu called "options" in Netscape.
- Choose "Network Preferences".
- You MUST see a dialog box (new window) with "TABS" (layers)...
- One of the "tabs" will be labelled "PROXIES"
- Click on that layer/tab.
- You will see a radio button labelled "manual proxy configuration" with a button beside.
- CLick on that button to open up another dialog box.
- The second last set of entry slots in that new window will show you...

```
--------- --------------------------                --------
| socks | |                         |               | 1080 |
--------- --------------------------                --------
```

This shows that netscape is already aware of socks. All you have to do is to tell NETSCAPE where the socks daemon is running.

Type in the **`eth0 port IP address`** of the linux server in the middle box shown above...

Save this setting and get going... !!

For Networking NOVICES

If you have already used your Win95 or WinNT machines to connect to the internet via PPP, this documents is of absolutely NO help to you. Anyways, why bother using linux to connect to the internet when you can do so via the client machine's built-in PPP?

If you haven't been able to connect to the internet via the LINUX server, then stop reading this document and read the other HOW-TO documents to setup your LINUX machine to access the internet via the PPP link.

I hope you know the concept of IP addresses. In short IP addresses have "mnemonics" formats (like www.cnn.com) as well as numeric versions like "198.20.186.4". If you type the former "www.cnn.com" someone must HELP your computer convert that name into the numerical format.

Why the numerical format? 'cos, that numerical format encodes a very efficient system of telling each computer HOW to send out communication capsules to OTHER computers THAT IT WANTS TO communicate to.

So, if you type in "www.cnn.com" on your NETSCAPE browser, then a UNIX computer called a "DNS server" will convert that name into a number for your computer. Then your computer will use that numerical format of the IP address to actually CONNECT to www.cnn.com and show you their latest news.

So, the gist being that : to use the internet you need a DNS server. This document includes instructions on setting up your computer to HOOK up to your NEIGHBORHOOD DNS server.

Your LINUX machine MUST have ALL of the following :-

1. A modem, through which you can connect to INTERNET *** DIRECTLY ***
2. An ethernet card, to which you have the ethernet cable hooked up (the other ends of which you have your client machines hooked up to...)
3. A Working PPP connection.
4. A Valid DNS server information (use nslookup --- if that program retuns invalid values, stop reading this document. You WILL NOT be able to proceed...)
5. Netscape 2.0 or later on your client machines.)

The "modem" is technically referred to as the "ppp0 port" as far as this document is concerned. By "port" I mean something similar to a "Sea-Port". This modem or PPP port enables you to "explore the world" (go on a "vacation" from daily chores) :-)

The "ethernet card" is your "eth0" port. That ethernet "port" lets you explore the ethernet network to which its connected.

Since your client machines are connected via the ethernet cable to the LINUX machine, anything that your client machine communicates to the LINUX machine will ONLY REACH the linux machine VIA the "eth0" port. ANything that the outside world sends to your LINUX machine will ONLY REACH via the "ppp0 port". So, its very important that these two ports be given "DIFFERENT ADDRESSES".

To make things easier for you, if you ALREADY successfully connected to the world using PPP, then, you have UNKNOWINGLY (or knowingly) assigned an IP numerical address to your linux machine's PPP port.

The MAKEFILE for sockd compilation

```
        SHELL=/bin/bash
        #SOCKS=-DSOCKS
        # or
        SOCKS=-Dconnect=Rconnect -Dgetsockname=Rgetsockname -Dlisten=Rlisten
-Daccept=Raccept -Drcmd=Rrcmd -Dbind=Rbind -Dselect=Rselect
        CFLAGS="$(SOCKS)"

        # If your system doesn't have PWD defined, define it here:
```

TOC

```
PWD="/usr/local/ProxyServer/socks42b"
# It should be this current directory.

# If your system has getcwd() but no getwd(), uncomment the next line:
#GETCWD=-DGETCWD

# Define FASCIST if you want ftp (rftp) to log names of all files transferred
#FASCIST=-DFASCIST

# Define RCMD and SUPPORT_RCMD if you want to support Rrcmd, which is
required
# for SOCKSified rlogin, rsh, and rcp.
RCMD=Rrcmd.o
SUPPORT_RCMD=-DSUPPORT_RCMD

# Define FOR_PS if your system is not SYSV and you want to have the
# command 'ps' show some details of sockd's activity.
FOR_PS=-DFOR_PS

# Define SHORTENED_RBIND to make Rbind() take exactly the same
# argument list as the regular bind(), i.e., without the additional
# 'remhost' argument.
SHORTENED_RBIND=-DSHORTENED_RBIND

# optimization flag for cc
#OPTIMIZE=-g
OPTIMIZE=-O6 -fomit-frame-pointer -pipe -m486
# Be careful with the OPTIMIZE flag. SunPro's SC2.0.1, for example, is
# knwon to produce incorrect code when -O is used.

# Directory into which to install the man pages
MAN_DEST_DIR = /usr/local/man

# Directory into which the SOCKS server should be installed
SERVER_BIN_DIR = /usr/local/ProxyServer
## This was defalted to /usr/local/etc

# Directory into the client programs should be installed
CLIENTS_BIN_DIR = /usr/local/ProxyServer
## This was defaulted to /usr/local/bin

# LINUX should use
CC=gcc
RANLIB=ranlib
RESOLV_LIB=
#OTHER_CFLAGS=-traditional -DLINUX  $(GETCWD) $(FASCIST) $(SHORTENED_RBIND)
-DCOMPAT
OTHER_CFLAGS=-DLINUX  $(GETCWD) $(FASCIST) $(SHORTENED_RBIND) -DCOMPAT
OS=linux
INSTALL=install
GETPASS=getpass.o

# Remember to include -Dindex=strchr -Drindex=strrchr in OTHER_CFLAGS if
# you don't have index() and rindex() (Sys-V camp)
```

TOC

```
        # <<<----------------

        # The Internet Whois server; used to be nic.ddn.mil.
        WHOIS_SERVER= WHOIS_SERVER=-DWHOIS_SERVER\'=\"rs.internic.net\"\'

        SOCKS_LIB=$(PWD)/lib/libsocks.a
        IDENT_LIB=$(PWD)/libident/libident.a

        all: LIB LIBIDENT server clients

        server: LIB LIBIDENT
                (cd sockd; $(MAKE) CC="$(CC)" RESOLV_LIB="$(RESOLV_LIB)" \
                        OPTIMIZE="$(OPTIMIZE)" \
                        SOCKS_LIB="$(SOCKS_LIB)" SUPPORT_RCMD="$(SUPPORT_RCMD)" \
                        IDENT_LIB="$(IDENT_LIB)" \
                        OTHER_CFLAGS="$(OTHER_CFLAGS) $(FOR_PS)")

        clients: RFINGER RFTP RTELNET

        LIB:
                (cd lib; $(MAKE) CC="$(CC)" GETPASS="$(GETPASS)" \
                        OPTIMIZE="$(OPTIMIZE)" \
                        RCMD="$(RCMD)" SUPPORT_RCMD="$(SUPPORT_RCMD)" \
                        OTHER_CFLAGS="$(OTHER_CFLAGS) " RANLIB="$(RANLIB)")

        LIBIDENT:
                (cd libident; $(MAKE) CC="$(CC)" OTHER_CFLAGS="$(OTHER_CFLAGS)" \
                        OPTIMIZE="$(OPTIMIZE)" RANLIB="$(RANLIB)")

        RFINGER: LIB
        # This also build rwhois
                (cd rfinger; $(MAKE) CC="$(CC)" $(WHOIS_SERVER) \
                        OPTIMIZE="$(OPTIMIZE)" SOCKS="$(SOCKS)" \
                        RESOLV_LIB="$(RESOLV_LIB)" SOCKS_LIB="$(SOCKS_LIB)" \
                        OTHER_CFLAGS="$(OTHER_CFLAGS) ")

        RTELNET: LIB
                (cd rtelnet; $(MAKE) CC="$(CC)" OS="$(OS)" SOCKS_LIB="$(SOCKS_LIB)" \
                        OPTIMIZE="$(OPTIMIZE)" SOCKS="$(SOCKS)" \
                        RESOLV_LIB="$(RESOLV_LIB)" OTHER_CFLAGS="$(OTHER_CFLAGS)")

        RFTP: LIB
                (cd rftp; $(MAKE) CC="$(CC)" SOCKS_LIB="$(SOCKS_LIB)" \
                        OPTIMIZE="$(OPTIMIZE)" SOCKS="$(SOCKS)" \
                        RESOLV_LIB="$(RESOLV_LIB)" OTHER_CFLAGS="$(OTHER_CFLAGS)")

        install.server:
                (cd sockd; $(MAKE) INSTALL="$(INSTALL)"
MAN_DEST_DIR="$(MAN_DEST_DIR)" \
                        SERVER_BIN_DIR="$(SERVER_BIN_DIR)" install install.man)

        install.clients: install.man
                for i in rfinger rftp rtelnet; do \
                        (cd $$i ; $(MAKE) INSTALL="$(INSTALL)" \
                                CLIENTS_BIN_DIR="$(CLIENTS_BIN_DIR)" \
```

```
                                install) done
        install.man:
                (cd doc; $(MAKE) INSTALL="$(INSTALL)" MAN_DEST_DIR="$(MAN_DEST_DIR)" \
                        install)

        clean:
                for i in lib libident sockd rfinger rftp rtelnet ; do \
                        ( cd $$i ; $(MAKE) clean) done
```

The ppp-on script

NOTE: This script is being provided as a sample. Having this sample is not a guarantee that you will have an internet connection.

```
        #!/bin/csh
        #
        #       ppp-on
        #
        #       Set up a PPP link

        set LOCKDIR=/var/spool/uucp
        set DEVICE=cua3

        set OUR_IP_ADDR=128.000.111.222

        if ( -f $LOCKDIR/LCK..$DEVICE ) then
            echo 'PPP device is locked'
            exit 1
        endif

        route del default
        # Just in case the Ethernetwork (In-House ethernet network) is up....
        #       if its NOT, then the above command is harmless...
        route ## To show that the above was successful...

        /usr/lib/ppp/fix-cua $DEVICE

            unalias pushd
            unalias popd
            pushd /usr/lib/ppp
            # stty 19200 -tostop

            # The original code has been commented out below...
            # if chat -l LCK..$DEVICE ABORT "NO CARRIER" ABORT BUSY "" ATZ OK
ATs50=255s111=0DT$PHONE CONNECT "" ogin: $USER ssword: \\q$PASSWORD
            echo $cwd
            ls -l ./comserv.dip
            dip ./comserv.dip       ## I removed the -v (DEBUG&VERBOSE) option to
'dip'.
            set dip_status=$status
            # echo the return value of dip is $dip_status

            if $dip_status == '0' then
```

TOC

```
                # Now please wait for 10 seconds, while the link is being
auto-verified by dip.
                echo 'About to fork-off pppd (after a delay of 10 secs)...'
                date
                echo 'If you see any error msgs below, then we are having SERIOUS
problems...'
                sleep 10
                pppd -detach crtscts defaultroute domain remote.princeton.edu mru
1005 mtu 1005 $OUR_IP_ADDR{}: /dev/$DEVICE 38400 &
                ###### we dont need this for the previous line...    < /dev/$DEVICE >
/dev/$DEVICE ) &
                #    The pppd deamon is FORKED OFF.  See the "&" at the END of above
line...
                # By using "locl" option, I am requesting that /var/spool/uucp be the
dir
                #    in which the LOCKS are created...
                echo 'Now wait another 10 seconds, before I auto-verify internet
connection.'
                sleep 10
                cat ~root/@utils/.line
                ping -v -c 5 genius.eng.wayne.edu
                cat ~root/@utils/.line
                traceroute physics.iisc.ernet.in >&! /tmp/$$
                                                cat  /tmp/$$
                                                \rm -f /tmp/$$
                cat ~root/@utils/.line
                exit 0
            else
                echo 'PPP setup failed'
                exit 1
            endif
            popd
        # [EoF]
```

The comserv.dip script to dial up PPP.

NOTE: This script is being provided as a sample. Having this sample is not a guarantee that you will have an internet connection. This script is called from the PPP-ON script given above...

```
        #******************************** comserv.dip
********************************
        #
        # Connection script for SLIP to ........
        # STATUS code for "dip" when it executes this script are:
        #  0 - all ok.
        #  1 - basic failures, in initializing the modem.
        #  2 - Failed in the crucial "dial" command.
        #  3 - Though DIAL command was successful, this script couldnt recognize
        #      the VERY FIRST responses from the other modem. (i.e., Training occured
        #      but, after that nothing happened that was intelligible to this script.
        #  4 - Modems could nicely link up.  But remote server HAS CHANGED syntax.
        #      i.e., the strings output by the server, are assumed to arrive in a
        #      PARTICULAR sequence.  If server s/w has changed, then we have this
        #      problem.  SOLUTION!!!! Manually connect and note all the strings &
```

```
#      all the sequence of interactions...  Then reprogram the script bel
# 10 - though dialing and connecting (modem-wise) is successful, ppp fail
##########################################################################
main:
redial:
  # Set the desired serial port and speed.
  port cua3
  speed 38400
  # term
  get $mtu 1005
  # Reset the modem and terminal line.
  reset
  # Without doing the above reset, nothing below will work!
  # Initialize the modem and dial comserv.
  # send ATQ0V1E1X4L1S0 0 \r
  # wait OK 5
  send ATZ\r
  wait OK 5
  if $errlvl != 0 goto error
  # send ATTQ0V1E1X4S0=0&C1\r
  # wait OK 5
  # if $errlvl != 0 goto error
  ## For Dial Tone use :- send AT&D2\r
  send AT&DP\r
  wait OK 5
  if $errlvl != 0 goto error
  # send ATS10=1\r
  # wait OK 5
  # if $errlvl != 0 goto error
  print if the line is busy, the dial command will realize that after 30 secs
ONLY.
  dial 258-0000   30
  print Return value of DIAL is $errlvl
  if $errlvl == 1 goto Continue1
  if $errlvl == 3 goto busy
  print unknown error with DIAL command in "dip" script.
  quit 2 # unknown error with crucial DIAL command...
busy:
  print telephone number is busy... Continue (1) or terminate (2)?
  get $input ask
  if $input == 1 goto redial
  print You have requested to cancel PPP.  Quitting...
  quit 10 # terminated...
Continue1:
  # wait V32 30
  # wait CONNECT 10
  # if $errlvl != 0 print Couldn't detect a CONNECT
  # if $errlvl != 0 goto connect_fail
  # print CONNECT was detected...
  # We are connected.  Login to the system.
login:
  sleep 3
  # send \r\r
  wait Username: 20
  if $errlvl != 0 goto error2
```

TOC

```
          send USERID\r
          wait Password: 5
          if $errlvl != 0 goto error2
          send __Password+_::\r\r
          wait comserv> 15
          # print Reached Comserv prompt...
          if $errlvl != 0 goto error2
        slipon:
          send ppp\r
          wait PPP_STARTED 25
          if $errlvl != 0 goto error2
          print CONNECTION completed...
          mode ppp
          exit 0
        error:
          print Total failure to interact with MODEM!!!
          quit 1 # basic failure in working with modem, etc...
        connect_fail:
          print Couldnt detect a "CONNECT 14400" kind of string after dial in...
          quit 3
        error2:
          print Modems could nicely link up.  But remote server HAS CHANGED syntax/
interaction sequence...
          quit 4 # basic failure in working with modem, etc...
        #==================================== EOF ====================================
```

Linux Netstation mini-HOWTO

Kris Buytaert, Kris.Buytaert@advalvas.be

v0.98p14, 22 February 1998

This document tries to describe how to hook up a IBM Netstation to your local network using a Linux box as server.

1. Introduction

2. Requirements

- 2.1 Hardware
- 2.2 Software

3. Other Usefull Documentation

4. Setting up the Serverside

- 4.1 AIX License
- 4.2 The Real Operating System

5. Configuring the Thin Client

- 5.1 Setting up TCP/IP
- 5.2 Setting Up the Boot Parameters
- 5.3 Setting Up the Monitor
- 5.4 Further Configuration

TOC

6. Misc Stuf

- 6.1 TroubleShooting
- 6.2 Experiences

7. Credits

---

1. Introduction

Some unused piece of delicious hardware floating around on my desk. Screaming to be used. 40 Mb's of ram and a 403 PPC inside.

So I decided to give it a try and connect it to my local Linux network. I want to thank the company I work for because they gave me the chance to experiment with it.

In this (Mini-Howto) we'll be dealing with an IBM Network Station model 8361-100, other models may be featured in the future.

It`s trying to describe how I setup the NC, there are probably lots of other ways to set up this machine, this one works fine. Any other remarks you might have from your own experience is welcome. (Free Hardware also ;-))

This MINI Howto is not trying to be the Bible on NC's and Linux, its trying to be something for you to get started from nowhere. Contributions to this NC are welcome

Things we still have to implement in this Howto

- How to export your homedir
- How to run applications
- How to run Java Applets

2. Requirements

2.1 Hardware

An NC, connected to your local Network, most likely by UTP, thus by a cross-cable or HUB connecting to an X Server. Herafter called the Linux Box. Basically the NC is Made to work with either AS/400 , Windows NT or AIX Servers. These are all expensive solutions, however working with thin clients doesnt have to cost that much.

The model I have at my disposal is an IBM Network Station model 8361-100, On the inside you can find an 403PPC chip, some S3 vga components, an PCMCIA slot, and normal 72pins (parity ??) sims. I found 8+32 Mb in my edition.

Actually those 40Mb were major overkill. In full operation modus the NC only uses about with lot's of configuration panels a couple of telnet sessions and a X -query open only took up about 4Mb of ram.) So taking out the 32Mb showed absolutely no significant loss of speed.

We proved it was no problem setting it up on a Linux only network. You`ll need a server that can provide you both with about 25Mb of free diskspace for the software, and the capacities to run a X -query. In the setup overhere we used a 486DX50 with 8Mb as a fileserver and I switch between another 486DX266 (PS/2 85) with 32Mb, and my Multia with 48Mb as the X server Off course, the fatster machines the better.

2.2 Software

IBM Software.

NC Kernel, fonts etc.

In order to boot the NC you will need its Kernel and fonts. About 25Mb of files are needed on the server. They can be found on an AIX 4.X with the Netstation modules installed. Or from the IBM Netstation Download Page

NFS Server.

A working NFS server, like in every default Linux distribution. Approx 85Mb of diskspace has to be exported to the NC.

TOC

X Server.

Any machine running XDM with enough memory, processor power will do. You don't need to have X configured on the machine itselve, it can perfectly be a monitorless server. Basic X Windows install will provide you with the necessary deamons.

Optionally

DHCP Mini Howto on sunsite

---

3. Other Usefull Documentation

Mainly the IBM website. It contains lot's of docuentation on the NC, most of it can be found in the .pdf format.

Latest updates can be found at http://www.as4000.ibm.com/networkstation/rs6000/ including

- IBM Network Station Runtime Environment for RS/6000 Users's Guide
- IBM Network Station Runtime Environment for RS/6000 System Administrator's Guide
- IBM Network Station Runtime Environment for RS/6000 System Navio NC Navigator Browser Guide

  Or from http://www.ibm.com/nc/pubs/
- IBM Network Station Setup and Use

4. Setting up the Serverside

You found all the software you needed. Now let's install them.

4.1 AIX License

Before making the tarball on the aix machine, make sure you run /usr/netstation/bin/agree in order to make the kernel in a usable format. Probably you will have to do the same thing with the tarball you get from the IBM website. So finding the kernel separatly might be another solution. Find some drive with about 25Mb of free diskspace, I use /usr/netstation/, and unpack the tarbal either from the IBM website or from an AIX machine. There is no need in leaving the approx 60Mb from /usr/netstation/doc on the disk if you don't have enough diskspace.

4.2 The Real Operating System

Setting up the nfs server

Edit your /etc/exports, add the line /usr/netstation 10.0.0.50(rw) Where 10.0.0.50 is the ipnr you want to give to the NC. Restart your nfs deamon.

[root@velvet sdog]# ps auxf |grep rpc sdog 4145 0.0 5.8 828 384 p1 S 03:55 0:00 _ grep rpc root 3120 0.0 5.7 944 380 ? S Feb 27 0:00 rpc.mountd root 3129 0.0 1.5 880 100 ? S Feb 27 0:10 rpc.nfsd [root@velvet sdog]# kill -9 3120 3129 ; /usr/sbin/rpc.mountd ; /usr/sbin/rpc.nfsd

or on a RedHat alike system easier

[root@velvet init.d]# pwd /etc/rc.d/init.d [root@velvet init.d]# ./nfs restart Restarting NFS services: rpc.nfsd rpc.mountd done. [root@velvet init.d]#

Your NC should now be able to mount the /usr/netsation by NFS.

Bascially if you don`t need X-Windows this is as far as it gets. You can easily telnet from your NC with nothing more installed. Hoever the beauty of this thing is it`s X capacities

Setting up the X server

Next we have to set up the X server. Basically I didn`t need to set up anything, all of my machines that ran X-Windows were configured to accept connections. I just started up the NC for the first time and it showed me all the machines that ran an XDM (cfr running an X -indirect). So any machine that can run xdm can be used as X Server. Just make sure XDM is So any machine that can run xdm can be used as X Server. Just make sure XDM is started.

Everything on the server side should be setup now. Lets try the NC side.

5. Configuring the Thin Client

Unplug the network cable from the NC. Then boot it up.

It will boot after checking its Memory, VGA etc, in a screen IBM Network Station Setup Utility. Basically you can manage everything from this menu system. The main parts will be described here.

5.1 Setting up TCP/IP

Section 5 (F5) : Your network setup should look something like this :

IP Addressed From NVRAM Network Station IP Adress 10.0.0.50 First Boot Host IP Adress 10.0.0.1 Second Boot Host IP Adress 10.0.0.11 Third Boot Host IP Adress 0.0.0.0 Gateway IP Adress 10.0.0.1 Subnet Mask 255.255.255.0 Broadcast IP Adress 10.0.0.255 Ethernet Standard Version 2

Resembles my setup. The NC itselve has 10.0.0.50 While 10.0.0.1 is the NFS Server, 10.0.0.11 is a secondary NFS server, just in case.

5.2 Setting Up the Boot Parameters

Section 6 (F6): Looks like this in my setup.

Boot file kernel TFTP Boot Directory /usr/netstation/ NFS Boot Directory /usr/netstation/ Configuration File /usr/netstation/configs/standard.nsm Configuration Directory /usr/netstation/configs TFTP Order 2 NFS Order 1 MOP Order Disabled Local Order Disabled

Where /usr/netstation is the directory export on 10.0.0.1

5.3 Setting Up the Monitor

Section 7 (F7): You can choos the right resolution / Refresh rate from a nice menu. I have mine running on an old 14" VGA Screen running in 1024x768 @60Hz

Basically thats all you need to install.

Just plug in your Network cable again. And Reboot the NC. After testing the Video / DRAM, the NC will search the Host system and request the startup information, download the Kernel from the NFS and boot up.

It might occur that the NC first gets a new FirmWare and upgrades it

When starting the graphical Console you should get a menu bar and a screen where you can select the X Hosts. You are now in the CLE (Common Login Environment): it's a desktop from where you can start

all sessions, either telnet or X . It uses a local window manager which is a small window manager ba on Motiv.

Your NC is setup now. Congratulations

5.4 Further Configuration

Further configuration of the NC can e.g. be setting the colors, window sizes, default keymaps etc. I'm not going to describe these in detail. If you have troubles finding your way through the menu system. The IBM Network Station Runtime Environment for RS/6000 User's Guide which can be found in either .pdf or .ps at http://www.as400.ibm.com/networkstation/rs6000/ and will provide you with detailed information.

---

6. Misc Stuf

6.1 TroubleShooting

Q: I get an Invalid Kernel Type while tryng to boot the NC. A: You didn't run the agree script. Q: I log in at the remote host and my keyboard settings fuck up. A: I use no .Xmodap when working on the NC. This keeps my keyboardsettings perfect.

6.2 Experiences

Using the NC as a plain terminal with multiple consoles. The Built-in Motiv extensions are a lot easier to use than the default setup ;-) Pressing the Pause/Break key will pop up the NC Menu at any time.

Port 5978

Remote Logging It`s the same as the messages console in your CLE.

Port 5999

Remote managment for the NC This is acutally a call for help. Strobing the NC , I found some usefull ports. Among them 5999, I still haven't found the exact way how to deal with it. However here are some commands that showed me to be usefull. You can apparently configure the NC remotely.

- help
- get boot
- get tcpip
- get file
- get nfs
- get tftp

7. Credits

Lots of thanks must go out to Wouter Cloetens, wcloeten@raleigh.ibm.com for getting me started and to Bart Geens ,Bart.Geens@advalvas.be for rereading this howto and findin uot lost of splelling errosr

If somebody has more info on these, help is welcome. They seem to be some SNMP thing but I have no expericience at all on that matter.

News Leafsite mini-HOWTO

Florian Kuehnert, sutok@gmx.de

v0.3, 4 January 1998

This HOWTO will help you to configure a small leafsite for Usenet News using the free software package Leafnode. For any questions, suggestions and comments, please write to Florian Kuehnert (sutok@gmx.de). Please send any bugs you found in this document to me as well. (C) 1998 by Florian Kuehnert.

1. Why to use Leafnode?
2. Where to get Leafnode?
3. How do I install it?
4. How do I maintain leafnode?
5. How does it work?
6. What newsreader should I use?
7. Where do I get more information?
8. Thanks

1. Why to use Leafnode?

In normal cases if you want to read news offline on your local computer you have to install a news server software like INN or CNews. Also you are in the need for an nntp or uucp connection to your newsfeed. In fact that such packages contain much more features than you need, you run better by installing Leafnode.

Leafnode is much simpler to use and very small but there are some disadvantages: Leafnode is slow and loses news in about any error situation. That's why you shouldn't use it for a big news server, nevertheless, it´s appropriate for private users who don´t want spend much time configuring INN.

2. Where to get Leafnode?

Leafnode is available at ftp://ftp.troll.no/pub/freebies/ and has been developed by Arnt Gulbrandsen, an employee of Troll Tech AS. The recent version is 1.4 and in some distributions (for example Debian), Leafnode is included. However, be sure to use at least version 1.4 as several critical bugs were fixed.

3. How do I install it?

A small installation help comes with the package, but let's do it together :-) -- If Leafnode >=1.4 is shipped with your distribution, it would be the wisest to use the pre-compiled version and leave out the steps 1 to 4.

1.) Be sure that there is no other news server running on your computer. When you type
$ telnet localhost nntp
you should get an error message. If you get a connection get back to the prompt of your shell, you should uninstall INN, CNews or whatever you're running now and comment out the nntp-line in your /etc/inetd.conf.

2.) Make sure that there is a user called "news", check in /etc/password for the name. If there isn't, create one (either typing
$ adduser news
or using a tool shipped with your distribution).

3.) Unpack the sources:
$ tar xfz leafnode-1.4.tar.gz
and change into the source directory
$ cd leafnode-1.4

4.) Compile the program and install it
$ make; make install

5.) Edit /usr/lib/leafnode/config (it may be on any other place like /etc/leafnode when you use a pre-compiled version of your Linux distribution). The line "server =" should point to the news server of your ISP.

6.) Edit /etc/nntpserver. It should include your local hostname (localhost or whatever your computer name is, the command hostname should help you). If in some startup file like /etc/profile or /.bash_profile the environment variable is defined, you should adjust to your computer´s name as well.

7.) Edit the /etc/inetd.conf: Make sure that there is no line beginning with "nntp". If there is such a line, comment it out putting a "#" before it. Then add the following line:
nntp stream tcp nowait news /usr/sbin/tcpd /usr/local/sbin/leafnode
When someone (for example you :-) connects to your computer on the NNTP port, leafnode is started as server process.

8.) Go online and run the program "fetch" as root or news. The first time fetch is started, it will download a list of your ISP's newsgroups. This may take some time depending on the speed on your connection and the number of groups your ISP has in its active-File.

9.) Start your favorite newsreader (slrn, (r)tin and knews are not a bad start) and subscribe to all your groups you read. Be careful not just to subscribe these groups, but also to enter them, even when they're

empty.

10.) Start fetch again to download all the news of the groups want to get.

4. How do I maintain leafnode?

Now you have got a working news system up and running, but there are still some things to do. You may edit the file /usr/lib/leafnode/config to set the expire dates of your groups. This number means, when old messages should be deleted. The standard time of 20 days is much often too long if you read some groups with much traffic, 4 days or a week are in most cases a good time for your system. You may change the value for all groups ("expire = n" to hold all groups n days), but you tell leafnode to change this time for some separate groups writing
groupexpire foo.bar n
to set the expire time for the group foo.bar to n days.

This setting alone won't make leafnode deleting old messages, a separate program is responsible for this: texpire. It may be started as a cron job or by command line. If your computer is up all the time, you may want to add the following line to news' crontab file (to edit it, log in as news and type "crontab -e" or as type root "crontab -u news -e"):
0 19 * * * /usr/local/sbin/texpire
This line causes the cron daemon to star texpire every day at 19:00. Check the crontab manual page for further adjustment. If your computer is not regularly switched on, you may start texpire just from time to time, when you notice that fetch gets slower. It works fine as with the "cron-method".

5. How does it work?

Leafnode is a "real" NNTP server, which means that you can also login from a different computer (via Internet, the local network etc.). Every time you enter a group in your newsreader, your reader sends the information to leafnode and requests it. If the group does not exist, leafnode will create an empty file /var/spool/news/interesting.groups, named like the group. When you run fetch the next time, it will fetch the messages of the group. If a newsgroup has not been visited for a certain time, leafnode will stop to fetch its articles and delete its name in /var/spool/news/interesting.groups. So if you just subscribed to a high traffic newsgroup by accident, you may delete its file there by hand for that you won't have to download all the postings there for the next week.

A week is not enough for you? You want to go on holiday for three weeks and still get news? Unfortunately, there is no option in leafnode to change it. But you can edit the file leafnode.h and recompile it. The #defined constants are TIMEOUT_LONG and TIMEOUT_SHORT, just set the time in seconds up. Another, simpler solution is to define a cron-job that does "touch /var/spool/news/interesting.groups/*" every night.

If you want to get a list of all avaible groups, look into the file /usr/lib/leafnode/groupinfo, where you will find a short description on the group.

If you want to re-read the list of newsgroups from your newsfeed (for example when you want to read a new group), just delete the file /var/spool/news/active.read. Fetch will create it the next time and get the new list. Fetch will also re-read the grouplist from time to time, so you don't have to do it by hand.

6. What newsreader should I use?

There is not *the* newsreader for Linux, like there is not *the* editor. My favourite newsreader is emacs in gnus mode which is the most configurable reader for Linux. Many people are using slrn and tin on a terminal, many people use knews under X. There are also trn, nn and a lot more reader, so just try what you like. The only reader you shouldn't use is Netscape, it is big, feature-less, unstable, and it creates sometimes broken postings. However, it is your personal decision.

Anyway, knews is no bad idea for your first experiences as it is very user-friendly and easy to understand.

7. Where do I get more information?

Some documentation is provided with the Leafnode package (read the files INSTALL and README, the sources are also quite interesting). If you want to know more about "professional" and "big" news servers, check the INN FAQ (they are provided with the INN package). To get information about your newsreader, type man "name of your newsreader" or check for other files in /usr/doc.

If you have any questions concerning the news system, just ask in an appropriate newsgroup (look in the news.software.ALL hierarchy).

If you have any question, comments or corrections concerning this HOWTO, just write to me (sutok@gmx.de).

8. Thanks

I would like to thank Michael Schulz (michaels@home.on-luebeck.de) for his help concerning some language problems and Cornelius Krasel (krasel@wpxx02.toxi.uni-wuerzburg.de) for his "touch *"-trick.

Linux off-line mailing method (offline mailaddr with 1 account)

Gunther Voet, freaker@tuc.ml.org - Belgium (CompuMed/TuCSRV)

v1.3.3, 4 June 1998

Use your linux mailing system offline, receive mail for multiple users with only one email address, and without being 24-24 online on the net. If you are unable to pay a direct line to stay online for 24-24 and still want your users to receive mail on your linux box; as well not pay for a multi-drop box at your isp, you can use this system using only one email address to divide to your users email addresses. It is as well 24-24 reachable since the server where your account resists will receive the mail. *This howto is Copyright (c)1997-98 by Gunther Voet.*

1. Notes by the author (preface).

- 1.1 Legal stuff
- 1.2 Copyrights
- 1.3 Feedback
- 1.4 Distribution
- 1.5 Changes
- 1.6 What am i using ?
- 1.7 Thanks go out to ...

2. Preliminaries

- 2.1 Preface & Description
- 2.2 Points of interest ...
- 2.3 Things you should know
- 2.4 there are 2 different ways ...

TOC

3. Requirements

- 3.1 A linux machine
- 3.2 A extra account at your linux machine
- 3.3 Fetchmail
- 3.4 Procmail
- 3.5 A mail account

4. I got it all, what now ?

- 4.1 Creating a .fetchmailrc file
- 4.2 Creating a .procmailrc file
- 4.3 "nosuchuserfile"
- 4.4 "crontab files".
- 4.5 "At the admins site".

5. Automation

- 5.1 The "skeleton".
- 5.2 addmail script

6. Help! (sigh)

- 6.1 The automation script just doesn't work:
- 6.2 What do you mean by "anonymous mailserver"?
- 6.3 My cat died
- 6.4 My dog died
- 6.5 Linux?
- 6.6 Can you help me with finding a mail account?
- 6.7 Why is the "maybe later i'll make some addition ..." removed ?
- 6.8 How do i get a "domain" ?
- 6.9 Why are you so f*cking lame using this?
- 6.10 Nosuchuserfile?
- 6.11 Can my users write/send mail too?
- 6.12 Does every user need a shell account at my server?

- 6.13 skeletion?
- 6.14 Addmail?
- 6.15 Why are you so cruel?
- 6.16 Didn't you get a complaint of excessive language ?
- 6.17 Why is this howto different than most others?
- 6.18 locally my domain works, but remote it seems not to receive
- 6.19 My dog died

7. The End

---

1. Notes by the author (preface).

In this chapter i'm just going to put myself safe for any damages and flames - since even *I* could be wrong ... If you got any questions or suggestions to add to this faq, even if you find any faults - there is a feedback section in it ...

Some "need to know" point is, all filenames & files/types are indicated with the line:

```
(*** < file > *** text ***) .procmailrc
```

This means, this is a file, called ".procmailrc" containing text. The text variable can also be code. It shows you what name of file to use. Without a path means it can be variable/choosen by you (in the most cases this files resists in the homedirectory of the mail"user" like /home/mailer).

1.1 Legal stuff

Neither the author nor the distributors of this HOWTO are in any way responsible for physical, financial, moral damage incurred by following the suggestions and examples of this text. The information in this document contains the best of my knowledge and experience, but i could still make any mistakes as well in the information as in the examples. Any trademarks are property of their respective holders (i ain't using any commercial thingy's here - but well - IF i'm going to add some text, i don't need to change this disclaimer). If your cat dies 'coz this document was too heavy - i am not responsible as well ...

1.2 Copyrights

This document and contents are Copyright (c)1997-98 by Gunther Voet. Unauthorized (re)production in any form is explicitly allowed and even strongly encouraged as long you don't change the contents of it without contacting the author (Gunther Voet). If you quote the document as whole or a part of it, there needs to be a Copyright "hint" or link to the derived work. "The HOWTO documents are copyrighted by their respective authors". The "HOWTO copyright" will discuss what can be done and what cannot be done with this document. If it is used in a commercial way, the author should been noticed for such distributions. Exceptions on this copyright may be granted under certain conditions with a written letter or e-mail to the author. For more info about the standard HOWTO disclaimer, please contact linux-howto@sunsite.unc.edu.

PLEASE *IF* you are going to make a system based on this HOWTO, then PLEASE give me the copyright :) heh ... - i am not that cruel :)). I mean by a system "a package" - NOT the mailsystem itself - altough it is some appreciative thingy i would be included :)) i can always use one email address more for more flaming :)) (just kiddin').

1.3 Feedback

Well, i expect from you and the users who read this HOWTO, they will make this HOWTO useful. If you got any suggestions, corrections, comments (except flame-mail :)) - please send them to me at freaker@freestamp.com or freaker@tuc.ml.org and i will make the corrections, comments or suggestions happen in the next revision. If you publish this document on a commercial way, a complimentary copy would be appreciated - you can mail me for my postal address. For flames - you could send them to me, but they will end at /dev/null ... so - don't even mind to

waste bandwidth for it !. Since i can exclude some users to receive mail from (explained in this document) (ev grin). My alias (name) is freaker btw :)). You can always try "freaker@freestamp.com". (note the changed ad !!)

TOC

1.4 Distribution

The latest revision of this document can been get from:

http://sunsite.unc.edu/pub/Linux/docs/HOWTO/mini/Offline-Mailing

-and-

http://tuc.ml.org/om/

(also checkout http://tuc.ml.org)

tuc.ml.org could be down at the time of current writing!

1.5 Changes

```
v1.0.0        - Preliminary release, internal testing, we tried it before
                we gave it out to you :) .. since - it GOT to be usefull
                and needs to work? rite ? :)
v1.1.0        - Fixed a lot of errors in the texts, put chapters in it,
              - fixed some small errors and typo's ...
v1.2.0        - fixed grammatical errors.
              - added a "what-are-we-using" chapter.
              - added Sunsite address.
v1.3.0        - Added "virtual mail support" for admins
              - New site and email addresses
              - Thanks go to ...
v1.3.1        - Converted to SGML and fixed some minor details.
v1.3.2        - Major errors corrected (filenames .fetchmailrc to
                .procmailrc) etc... (thx to Tetsu Isaji)
              - Added support for the "new" sendmail, v8.8.0+
              - Fixed some minor errors
              - Upgraded my system YAY!
              - Japanese version will come out soon, location
                will be mentioned when known ... (thx to Tetsu Isaji)
v1.3.3        - Added CC support, changed email addr to freaker@freestamp.com
```

1.6 What am i using ?

I'm using Linux 2.0.30, Pentium-166, ATI pci-mach64 card, Accelerated-X (Metro) and a connection to my isp (currently ibmnet) via ppp using a ZyXEL 28k8 modem, and occasionally a Bausch 28k8.

Now i upgraded to Linux 2.0.33, installed a USRobotics modem, it still works as it should work :) (everybody should upgrade their linux versions as soon there comes out a patch :) believe me :) you could spare a lot of time & seeking for errors when doing this one :)

TOC

1.7 Thanks go out to ...

- Fred, durban.hebel.net (for allowing the pages & bot)
- Lifesaver, parkside.net (for allowing the new pages, helping with this faq for mailing, since, it is all working overthere and here ... make some visit to his cyber-cafe ... heard they got great coffee :))
- The server admins at lodus.net to continue this project.
- Now i'll need a new server since lodus.net has been sold :(

---

2. Preliminaries

Well, how does this system works ? what can you do with it ? what is it anyway ? and ... am i used to read it ? ... to all those questions - there is an answer ...

2.1 Preface & Description

This HOWTO is dedicated to all the off-line linux users/servers ... You got a group, or organization, and you want all your members to have a contact (email) address, without paying for the 24/24 direct line or for the email accounts ... Or you got a internal network - with a masq'd server and you want to give all the users using it a email address, you could use this way to do it. I personally am using it to give my users & members of a demogroup all a e-mail account without being 24/24 online; since it is a non-commercial demogroup - and we don't got THAT much money to pay a direct line with instant email access - and different ip's. Everybody got a email address, even when they don't got internet access. The ones with internet access can receive their mail on their personal email address - viavia the system described below. Your users can still send mail as they used to do, since - your system will hold the mail queue from the users. The only thing what should be changed for the users - is that they add their "email address" wether (described below) it is the method with the subject line, or with the header "to" ...

2.2 Points of interest ...

- A organization with members, for all a email address, so they can be contacted when neccesary.
- A anonymous mailservice - they can't read your "mail setup file" - so they don't know where the mail is forwarded/rerouted to.
- A group who doesn't got enuf money (don't laugh), or is free, and don't WANT to spend money for it (eheh) - and still wants to provide the members a email address.
- A masq'd network - with a server connected to the internet, where the users using the masq'd network need to have a email account
- You want to put some documents "autoreplied" - like i do with my HOWTO, statistics, or documents ... like "info@yourmail.dom" or "document1@yourmail.dom".
- You only need one account at the "main" server to fetch from, for the +200 email accounts you can create at your server ...
- I used the "B" method with +- 300 email accounts - with a public server; 80486DX4-100 - and was processed in 1 minute.

TOC

2.3 Things you should know

- Not every (free) mailservice is happy when you are going to distribute mail to other users, or putting up an anonymous mail server, so, please try to contact your mailserver administrator before "just doing it".
- It is slower than a direct 24/24 connection, since your server will need to get online before it will process the mail.
- It needs a administrative force to administrate the "mailrouting" and to add the user ... you will need work with it ! - it is not that automatic. (well - i made it "semi-automatic" with my server :) you could try it as well).
- Bandwidth and space is needed when you got many users - on the mailserver you are working on ... and sure - when they send files - you need to have enough space on it !.
- All mail will be transferred into "one step" - so - the last mail received on your server will be last sent to the recipient. "first in - first out".
- In the first method (to:) there is no prob with the usernames - but in the second "cheaper" method - users can't put in a subject line, since that one will be used to send the mail to the recipient. Many users forget to use the "subject" ... - or they will need to know it clearly.
- If you poll enough for your mail, (for example 4 times a day) - the mail will be sure be delivered to as from a recipient the least every 6hrs. So the processing will go a little smoother.
- you CAN use it as a "online" mail server as well :) since it will poll for it's mail for every XX minutes. but - when you are online - why using another server ? and not asking your own domain for your own server ? (dough).

2.4 there are 2 different ways ...

1. method 'A' - now called "(m(A))". The "mailsystem" will look to the header, and will see to the "to:" line. This is the best way, since it are "real email addresses". You will need another account on some server *AND* you need a DNS (MX) entry (your own domain for instance). If the system administrator wants to help you with that, then there will be no problem !. The system administrator will need to reroute ALL mail sent to your (MX) maildomain -> to your account !. This will take some time for him to do it :) so be really nice to him :)) Thanks to my system administrator (Fred) i got my maildomain and the user- services of my "free" demogroup. It can be a DNS entry costs something, what you will need to consider to take it or not.
2. method 'B' - now called "(m(B))". The "mailsystem" looks to the subject line, and will forward to the user indicated into the subject line. If the user is not found, it will bounce a mail back. This will need a pop server with only one useraccount, and don't need your own DNS or you don't need to be nice to your system- administrator who arranges you the mail address :). You can use a public mailserver, as well your own mailserver or a private one. You can even sacrifice your own email address for it :). I used this approach as first what worked for me for over a year now ... I know this is sloppy, but, i ain't telling you need to use this way, so don't shoot ME - since i'm only offering you some solutions to your mail problems :) - you could use method A what isn't that sloppy !

---

3. Requirements

Well, you need (of'course) a Linux machine, what can be connected to the internet - So! (i am smart - i am smart :)) - you probably got a modem, right? a phoneline - or any way to connect yourself with the server you are running. You will also need some software like Fetchmail, Procmail, an extra account, and your email account or DNS with email account.

3.1 A linux machine

This can be ANYthing ... even a 8086 ... - but preferable - since there will be some load on it - minimally a 80386 :). as faster the machine - as faster the software will run (rite eh)? :).

3.2 A extra account at your linux machine

You need to create a extra account on your Linux machine. I am using as example (in this HOWTO) "mailservice". This can be as well "mailserver" or "mailtousers" - as long it is some name what can be recognized by you!.

! It doesn't need to have ANY root privileges !

This mailaccount will forward all mail to the users on your system, or to external users (not on your system :)). Just create it as an ordinary user. You will need to test things under this account, and, the mail "administrator" can use this account to administrate the mail account without being root even .. so - pretty safe !.

3.3 Fetchmail

I am using v1.9 patch level 9 ... i found it at sunsite.unc.edu, and, i am using only this mail-fetcher in my example - if you want to use another one, you are on your own ! since i am happy with it :). For extended features you should read the man of Fetchmail :). This can be installed as root - as well for the user itself ... - best is to install it as root :) since i know it will work for sure then :)))

3.4 Procmail

I am using v3.10 - found it again at sunsite.unc.edu, and - i like this as the best mda (mail delivery agent) around ... It delivers nicely - and it is better than using it by the "original" system mda - since it will deliver user-per-user ! ... The same as above - it can be installed as root, or as user - and :) best is to install it as root - since i know it will work for sure then (again!) :)... how repetetive ...

3.5 A mail account

TOC

You always need to have a mail account - where you can "poll" your mail from. The server needs to be a POP server - where you can poll from when you want, and where users can send their mail 24/24 to. It can be a free-mail service, as well commercial - even your mailaccount is sufficient enough. In the "subject" method (m(B)) - you don't need something more - in method A - with the "to" fields - you NEED more - as written below !

A DNS record (MX)

Needed to run your own "domain" - email domain ... it is needed since the administrator of the server can't use it's own domain for your users - since there are users who need to be contacted on his server as well - what could interfere with your users. And your mailheader must be filtered on that domain. So - an apart domain - pointing to HIS server. This can be an additional cost - or you can ask your system administrator to add a "non- authorative address" - pointing to his server ... - however - it needs to be another domain than he is using !

A "forward" to your account-account :)

Well, the system administrator needs to forward *ALL* mail to your domain to YOUR account ... - it is easy for him when he knows how to do it :) .. else he will need to read the sendmail manual :)) (or qmail - whatever). You will poll for your mail at your account and voila ! filtering on YOUR own email domain.

---

4. I got it all, what now ?

Now you created the extra account, you got the mail address - and/or the DNS entry & forwarding to your account ... As well installed Procmail & Fetchmail so we can rock the place ! :) .. here we go !

1. You need to create a .procmailrc file, what will contain the "delivery" info to your users.
2. You need to create a "nosuchuserfile" - so the writer knows his mail isn't delivered well.
3. For best work :) you could use crontab to check for mail. This is a easy way to check your mail every XX minutes when on the internet.

4.1 Creating a .fetchmailrc file

You will need to create a .fetchmailrc file, what will contain the information (username & password, as well the Mail Delivery Agent (mda) to proces the mail to). Here is some example file ...

```
(*** < file > *** text ***) .fetchmailrc
```

```
server my.mail.server.com
proto pop3
user myaccountthere
pass deepestsecrets
flush
mda /usr/bin/procmail
```

This file will be used to fetch your mail. Please test it by using the fetchmail program "fetchmail -vv" - and see your mail is being transferred right ... There will be some errors - since the procmail control file hasn't been created yet. You can wait by testing AFTER making the procmailrc file, but - i'll warn ya - IF there is something fault :) it CAN be this file :) It needs to be owned by the user account itself - in my case "mailservice" and needs to be "user readable" but NOT group/world readable - since it contains the "main password" :)). (chmod 600 .fetchmailrc will do).

4.2 Creating a .procmailrc file

This control file will forward all mail to the users in it. There are 2 ways as described before - the "to:" (header) way - and the "subject" (sloppy) way. The file will contain the usernames to transport to. All the "#" are comments and are absolutely not needed when not wanted - it's only so you know what i am doing ... - you can as well best chmod it 600 - so the rest of the world or group doesn't need those private addresses eh :) ... It needs also to be owned by the user (like "mailservice") :)). The "nosuchuserfile" is a "bounce" to the writer - if the user isn't found (so mail not delivered) in the procmailrc file ... - so the writer knows the mail isn't delivered well.

For "header (to:)" transportation

```
(*** < file > *** text ***) .procmailrc
```

```
# this line is for debugging purposes only ! it should be removed for
# ethical purposes - since you can read all mail passed trough your mail-
# server ... - all mail will be copied to the file "passtrough" before
# going to the users ... herein you can look what went wrong ...
:0 c
        passtrough

# the mail with header "to: freaker@mydom.com" will be forwarded directly
```

TOC

```
# to me, the other mail will pass this option ...
:0
* ^To:.*freaker@mydom.com
! freaker

# the mail to root@mydom.com will be forwarded to root ... as well postmaster!
:0
* ^To:.*root@mydom.com
! root

:0
* ^To:.*postmaster@mydom.com
! postmaster


# the mail to barbara@mydom.com will be forwarded to barbara AND will be
# forwarded to her private email address !

:0 c
* ^To:.*barbara@mydom.com
! barbara@her.private.one

:0
* ^To:.*barbara@mydom.com
! barbara

# the mail to johnny@mydom.com and johnny@hisdom.com will be forwarded to johnny

:0
* (^To:.*johnny@mydom.com)|(^To:.*johnny@hisdom.com)
! freaker

# the mail to hans@mydom.com and all carbon copys will be forwarded to hans

:0
* (^To:.*hans@mydom.com)|(^CC:.*hans@mydom.com)
! hans

# this lines will BOUNCE the mail to the sender - when it is not delivered to
# one of above users ... it will send the file "nosuchuser" into the mail
# body as reply ... be aware ! you need to make such file ! - mine contains
# "well, the user you wanted to reach does not exist on this server, please
# try again, it could be the user is not present anymore".
#
:0
  |(/usr/bin/formail -r -k \
     -A"X-loop: mailservice@mydomain.dom "| \
       /usr/bin/gawk '{print }\
       /^/ && !HEADER \
         { system("/bin/cat nosuchuser"); \
         print"--" ;\
         HEADER=1 }' ) |\
         /usr/bin/sendmail -t


exit
```

For "subject: touser" transportation

TOC

(*** < file > *** text ***) .procmailrc

```
# this line is for debugging purposes only ! it should be removed for
# ethical purposes - since you can read all mail passed trough your mail-
# server ... - all mail will be copied to the file "passtrough" before
# going to the users ... herein you can look what went wrong ...
:0 c
        passtrough

# the mail with header "to: freaker@ibm.net" will be forwarded directly
# to me, the other mail will pass this option ... When you got a "dedicated"
# email address to receive your "mailservice thingy's" on - you don't need
# to use this line :)
:0
* ^To:.*freaker@ibm.net
! freaker

# all mail with as subject "root" will be forwarded to root !
:0
* ^Subject:.root
! root

# all mail to "subject: barbara" will be forwarded to barbara ...
:0
* ^Subject:.barbara
! barbara

# all mail to "subject: paul" will be forwarded to his external email addr.
:0
* ^Subject:.paul
! paul@his.personal.emailaddress

# all mail to "subject: john" will be forwarded to his account at your server
# and a copy will go to his private email address ...
:0 c
* ^Subject:.john
! john@his.personal.emailaddress

:0
* ^Subject:.john
! john

# All the mail from ibm, with their updates and information, will go to
# freaker, as he is the one who will administrate the mailservice, and
# as ibm doesn't want to get the bounce putten below !! ... this is
# neccesary if your mail provider sends "newsletters" etc...
:0
* ^From:.*newsletter@ibm.net
! freaker

# All messages from the daemon should been thrown away, or in my case, will
# be saved to a file ... (use /dev/null to throw to endless pits of The Abyss)
:0
* ^FROM_DAEMON
```

TOC

```
throwaway

# this lines will BOUNCE the mail to the sender - when it is not delivered to
# one of above users ... it will send the file "nosuchuser" into the mail
# body as reply ... be aware ! you need to make such file ! - mine contains
# some text like "user not found in subject line, please use "Subject: user"
# to write a mail to the user, like example "subject: freaker" would send a
# mail to freaker." The file can be long, but also small :) ... the
# "mailservice@mydomain.dom" will prevent to loop between your server and
# the other server - it needs to have the EXACT email address used !.
# Else you could create an endless loop with a server what sends mail
# to "your email address" with as subject something like "don't spend 500$
# at your ..." etc...
:0
  |(/usr/bin/formail -r -k \
     -A"X-loop: mailservice@mydomain.dom "| \
       /usr/bin/gawk '{print }\
       /^/ && !HEADER \
         { system("/bin/cat nosuchuser"); \
         print"--" ;\
         HEADER=1 }' ) |\
         /usr/bin/sendmail -t

exit
```

4.3 "nosuchuserfile"

```
(*** < file > *** text ***) nosuchuser
```

```
The user you wanted to contact is not present at this system.

Please use the subject line as recipient - example "subject: freaker" would
send mail to freaker on this system.
```

4.4 "crontab files".

If you don't know how crontab works :) better read the manual :) ... You need to create a "checkmail" file - what will see if the link is up, as well the cronfile itself ... - i am using a ppp link :) so - this is an example how to look when the ppp link is up - and to poll every 10 minutes using cron. Looks sloppy - but isn't !.

checkformail

the .checkformail file will be called (needs to be executable as well) - and will look if the ppp link is up. If it is up - then it will fetch for mail. Crontab will use this file when you are using the below cronentry ...

```
(*** < file > *** code ***) .checkformail
```

```
#!/bin/sh
#

        cd /home/mailservice
```

```
        if [ -f /var/run/ppp0.pid ]; then
        /usr/local/bin/fetchmail -s > /dev/null 2>&1
fi
```

TOC

crontab

This cronentry file needs to been loaded into crontab, and will call the .checkformail - every 10 minutes. It won't write any mail or give any info to the console - since i'm redirecting everything to null.

```
(*** < file > *** text ***) cronentry
```

```
0,10,20,30,40,50 * * * *  /home/mailservice/.checkformail 1> /dev/null 2> /dev/null
```

4.5 "At the admins site".

Well, this should be done when using the "A method" ... at the admin's site, so the email goes all from a complete domain, to one username. It is pretty simple, and once you've done it - it works like hell. this is NOT neccesary if your system administrator (the uplink) got another method, and is NOT neccesary at YOUR side !!!!

When using a newer version of sendmail, the "old sendmail" trick probably won't work, so please refer to the "new sendmail" topics to let your mailrouting work.

(old sendmail) add some lines to sendmail.cf

add the following lines to your /etc/sendmail.cf file, so the domains file will be read. please be noted that the "ruleset 98" is added as underhere, since - once you got errors :) it's a hell to find 'm out ! (and i can know it :) DuH).

```
(*** < file > *** add ***) /etc/sendmail.cf
```

```
# Database of handled domains

Kmaildomains btree /etc/maildomains.db

# Add these lines *IN* Ruleset 98 ! (under Ruleset 98).

R$+ < @ $+  . >              $: $1 < @ $2 > .
R$+ < @ $+ > $*              $: $(maildomains $1@$2 $: $1 < @ $2 > $3 $)
R$+ < @ $+ > $*              $: $(maildomains $2 $: $1 < # $2 > $3 $)
R$+ < @ $* > .               $: $1 < @ $2 . >
```

(new sendmail) Adding some lines to sendmail.cf

With the newer sendmail releases (tested with sendmail v8.8.7, 8.8.8). Ignore method A, and add the next lines ...

```
(*** < file > *** add ***) /etc/sendmail.cf
```

```
# Database of handled domains

Fw/etc/sendmail.cw
Kvirtuser btree /etc/maildomains.db

*OR*

Fw/yourhomedir/sendmail.cw
```

```
Kvirtuser btree /yourhomedir/maildomains.db
```

TOC

(new sendmail)editing the /etc/sendmail.cw (or /yourdir/sendmail.cw) file

If you are using another "location" for the sendmail.cw file, then please replace the "/etc/sendmail.cw" to "/yourhomedirectory/sendmail.cw". The pro points of putting this sendmail.cw file into your homedirectory is that you don't need root to change the domains to receive on. tough - this can give security risks if not used properly !

This file can already exist, or needs to be created, if it already exists be sure you don't overwrite the older data - or i need to refer you to my fine disclaimer :)

First create a /etc/sendmail.cw file, what will be used to "send" a domain to a specific user ... here is an example ... (as you already knew, the name "mailservice" can be anything you want - it can even be your loginname (like mine is freaker).

```
(*** < file > *** text ***) /etc/sendmail.cw
```

```
mydomain.dom            mailservice
```

creating a /etc/maildomains file

First create a /etc/maildomains file, what will be used to "send" a domain to a specific user ... here is an example ... (as you already knew, the name "mailservice" can be anything you want - it can even be your loginname (like mine is freaker). (you could have this /etc/maildomains in /yourhomedir/maildomains as mentioned before, just change the paths :)

With the OLDER sendmail versions:

```
(*** < file > *** text ***) /etc/maildomains
```

```
mydomain.dom            mailservice
```

With the NEWER sendmail versions:

```
(*** < file > *** text ***) /etc/maildomains
```

```
@mydomain.dom           mailservice
```

let it work !

With the old & new sendmail versionsyou need to generate the btree (database) files, you'll need to do the following:

```
cd /etc    (or /yourhomedir)
makemap btree maildomains < maildomains
```

after that, kill the sendmail daemon, and restart it. it should now WORK! good luck :)

5. Automation

Well, now, everything above works ... (if it doesn't work - don't even think about automation before it WILL work ... - now - we need some script so it isn't a pain in the ass for the "mailadministrator" or the root user to add users to the procmailrc file ... The below example will be for "more experienced users" - since some things NEED to be changed as well ... - it is an example with the "B method - Subject lines". It can be easily adapted to the "A method". Since the "B method" will be more used (cheaper) than the "A" method - i decided to use the "B method" ... Am i sounding repetitive or not ? .. well - i meant to write it in this way :)) to bug you :)))..

5.1 The "skeleton".

Since the footer needs to be as footer (everything below won't be processed since we are bouncing there); there needs to be a header and a footer file.

the ".procmailrc-header" file

This file will now be the "header & user" file .. since here will users been added and removed - it will be a important file ... - best take a backup from it each time you add a user ... - there COULD be something wrong sometime .. A system can fail ...

```
(*** < file > *** text ***) .procmailrc-header
```

```
:0 c
        passtrough

:0
* ^To:.*freaker@ibm.net
! freaker

:0
* ^Subject:.root
! root

:0
* ^Subject:.barbara
! barbara

:0
* ^Subject:.paul
! paul@his.personal.emailaddress

:0 c
* ^Subject:.john
! john@his.personal.emailaddress

:0
* ^Subject:.john
! john
```

TOC

```
:0
* ^From:.*newsletter@ibm.net
! freaker

:0
* ^FROM_DAEMON
throwaway
```

---

the .procmailrc-footer file ..

As mentioned above, this file needs to be as footer - since all data BELOW it won't be used to deliver - this footer contains the "bounce" code to bounce the users not found *ABOVE* this footer !. it is the ABSOLUTE end of the file !

```
(*** < file > *** text ***) .procmailrc-footer
```

```
:0
  |(/usr/bin/formail -r -k \
     -A"X-loop: mailservice@mydomain.dom "| \
       /usr/bin/gawk '{print }\
       /^/ && !HEADER \
         { system("/bin/cat nosuchuser"); \
         print"--" ;\
         HEADER=1 }' ) |\
         /usr/bin/sendmail -t


exit
```

5.2 addmail script

This script will add a user to the header file, attach the header & footer to eachother - so it will be a complete .procmailrc file. the "#" (comments) are not really needed - and are for your info :).

```
(*** < file > *** code ***) addmail (* chmod 500 *)
```

```
#/bin/sh
#
# Copyright (c)1997 by Gunther Voet. rev 1.0.1
# please leave the Copyright in it when it is distributed with any
# system using this thingy ...

echo ""
echo "Addmail v1.0.1 by Gunther Voet, Freaker / TuC'97-98 (21/04/97)"
echo ""

if [ $1 ]; then

        if [ $2 ]; then
```

```
# make a backup file !

        cp /home/mailserv/.procmailrc-header /home/mailserv/.procmailrc-backup

# APPEND (>>) the information to the header file ...

        echo ":0" >> /home/mailserv/.procmailrc-header
        echo "* ^Subject:.$1" >> /home/mailserv/.procmailrc-header
        echo "! $2" >> /home/mailserv/.procmailrc-header
        echo "" >> /home/mailserv/.procmailrc-header

# copy the header file to .procmailrc - and append the footer file to it !

        cat /home/mailserv/.procmailrc-header > /home/mailserv/.procmailrc
        cat /home/mailserv/.procmailrc-footer >> /home/mailserv/.procmailrc

# make sure it is owned by "mailserv" and the read/write priveleges are ONLY
# for the user "mailserv" itself ...

        chown mailserv /home/mailserv/.procmailrc
        chgrp users /home/mailserv/.procmailrc
        chmod 600 /home/mailserv/.procmailrc
else
        echo "No DESTINATION mail address has been given ..."
fi

else
        echo "usage:"
        echo ""
        echo "syntax:  addmail from_user to_user(domain)"
        echo ""
        echo "example: addmail freaker freaker@myemail.dom"
        echo ""
fi
```

Now, this script will append the information of the user, as well the email address to the header file, it will copy it to the .procmailrc file, and will add the footer to it, so you got a complete .procmailrc to process the mail. If you want to delete a user- just edit the .procmailrc-header file, and at the next user added it will be deleted at the .procmailrc. To do a instant delete, just delete the user from both the files .procmailrc & .procmailrc- header.

I guess you are smart enough to write a script that'll automatically add your users when using both methods - when doing a "adduser" at your box.

---

6. Help! (sigh)

Well, you need help - isn't :) ... hmm .. i could be sarcastic and just say "you could better do it again, 'coz it looks a messy enuf" - or .. i could help ... - I wrote down some common problems - IF you got any problem NOT listed in this HOWTO (section) - then mail me - and i'll put it in this howto even with your name/email addr in it :). **BEFORE** mailing to me - please look if you didn't forgot anything - and IF you want some help from me - send me the MOST DETAILED information - included the scripts & things you needed. i DON'T need any binaries - since i won't run them.

6.1 The automation script just doesn't work:

- is it executable ?
- is your (default shell) located at /bin/sh ?

6.2 What do you mean by "anonymous mailserver"?

- Well, you could make accounts like "anon0001@yourdom.dom" - and forward it to another email address ... nobody needs to read the .procmailrc file, so YOU ONLY know the address !.

6.3 My cat died

- Well, next time don't print this HOWTO out on 200 gram papers, since it are 10 pages it would be 2KG for the cat - it is JUST TOO HEAVY !

6.4 My dog died

- hmm - can't do anything about that one - why askin' me ? ... Just bury it ....

6.5 Linux?

- A free-unix - posix compatible - made by Linus Torvalds ... Why are you reading this if you even don't know what Linux is ?

TOC

6.6 Can you help me with finding a mail account?

- No! - this is a howto for YOUR side, i don't care about the side of your isp, nor how to get your email address.

6.7 Why is the "maybe later i'll make some addition ..." removed ?

- Read point 5.5 very carefully :) and you'll see why .. it's 'coz i needed to add it on general request :)

6.8 How do i get a "domain" ?

- Ask your local ISP/provider - he will help you with it. It could take days/ even weeks when asking to the Internic - your provider can help.

6.9 Why are you so f*cking lame using this?

- coz i don't want to spend money - and - 'coz i feel like being f*cking lame USING it - i *AM* using it - so why bother?

6.10 Nosuchuserfile?

- You can put in it what you want - as long you put some "needed" info so the original writer knows what happened with his "never delivered" mail.

6.11 Can my users write/send mail too?

- This has nothing to do with the system i explained to you, read the sendmail manual ... - this is to RECEIVE mail - to be "always available at an email address".

6.12 Does every user need a shell account at my server?

- Nope .. - but - it will be really hard for users not having internet access :) ... You could have a masq'd network - and use a computer connected to it - so the users can get their mail that way, or you can generate a link between a bulletin board and his mail, you could even forward it to a fidonet gate :) ... reasons enough not to give a shell!.

TOC

6.13 skeletion?

- i know it is a typo - but - i like this word better.

6.14 Addmail?

- No questions - for automation - you better be sure about what you are going to do ... since - it COULD be a trojan ya know :))) (it isn't but what means you need to know what scripting is before asking questions) It IS easy enough to interprete - that's even why i put the comments with it. *IF* you are going to distribute this script - leave my Copyright in it please ! thanks :).

6.15 Why are you so cruel?

- I am not cruel :) i am nice :) i am the nicest guy of the world, of the universe ! NOW SCRAM! (i just want to be complete in my HOWTO, and not TOO much "drifting away" from my original point in the doc - so :) that's all).

6.16 Didn't you get a complaint of excessive language ?

- Not yet, but, could be i'm filtering everything containing 'excessive' and language :) i don't know :))

6.17 Why is this howto different than most others?

- coz sometimes reading plain howto's CAN be boring ... i wanted to add something next to it ...

6.18 locally my domain works, but remote it seems not to receive

- Check out your "sendmail.cw" file, since the domains NEED to be added in it !

6.19 My dog died

- the rulesets don't work, i can't receive any mail, or sendmail dies Refer to the new section "new sendmail" - and disregard the old sendmail tricks. The newer sendmail should be easier to use with virtual domains.

---

7. The End

This sounds like the end ... If you find any unwanted bugs (or features :)), then leave some feedback ... any comments & suggestions -> mail them :). if you are still bored after reading this, please go to: http://tuc.ml.org/ hehehe.

My thanks go out to:

Hannes van de Vel: for supporting me (hum) Tetsu Isaji: the japanese offline-mailing & notifying me about errors :) Greg Hankins: for notifying me about some errors in the sgml version. Linus torvalds: of'course ... without his help this howto wouldn't be here!

LINUX PLIP MINI-HOWTO

Andrea Controzzi, controzz@cli.di.unipi.it

v2.1, 12 March 1998

This HOWTO will hopefully let you build and use a Parallel Line Interface Protocol.

NOTE

This is a new release. There are many changes and lots of enhancements, but there will be still grammar and spelling errors (english is not my native language) and, unlikely but possible, some wrong or outdated info. PLEASE let me know about any errors to help me provide the correct information for everybody.

The biggest changes in the release are:

- SGML format
- a general answer to the "PLIP with win95" most asked question (the answer is negative, sorry)
- bigger and better FAQ section (thanks to the reports of countless users: many of them will find their comments there, as I promised!)
- removal of the Quick PLIP Installation files, that was useless and outdated.
- updated Dos-Linux PLIP link addendum

For any question, error correction, comment and/or suggestion, my E-Mail address is: controzz@cli.di.unipi.it. Feel free to mail me any time you need help. Sometimes I won't answer immediately, but I'll answer. If it happens that you don't receive anything from me after 2 weeks, this means that our mail system has trouble: do not hesitate to mail me again, it's my duty to help you. I can guarantee I'll solve your problems, but I'll try. If after many mails you don't receive answer, check your return address. Several times my answers did not arrive due to delivery problems, usually because the return address was root@myhost.

Before sending mail read the FAQ, my answer to a question already present in the FAQ will not be better (but likely worse or less complete) than the answer you can already read.

For questions about PLIP with DOS and Win95 please send mail to the authors of these chapters, I can't help you.

First of all, a lot of technical information come from the net-2-HOWTO, by Terry Dawson. This mini-HOWTO is not supposed to cover other aspects and/or replace the net-2-HOWTO: my goal is to give you a way to install a PLIP permanent connection quickly, *ONLY* this. All the other

TOC

info come from my personal experience and the help of many users that sent me comments an information.

Read the net-2-HOWTO and the other docs for the general information about the network and the config files I suggest to change.

1. Introduction: what is PLIP and why should I use it?

2. Hardware required to use PLIP.

3. Reconfigure the kernel.

4. Kernel messages about the PLIP interface.

5. Setting up the configuration files.

6. Activate the PLIP link.

7. FAQ.

8. Patches to make PLIP and LP live together.

- 8.1 PLIP and LP together on the same port.
- 8.2 PLIP and LP on different ports.

9. A PLIP link between DOS and Linux

- 9.1 DOS-Linux link.
- 9.2 A DOS-Linux PLIP link experience.

10. PLIP between Linux and Windows 95.

11. Questions? Comments? Send me feedback.

12. Where to find new releases of this mini-howto.

13. Credits.

14. Copyright message.

---

1. Introduction: what is PLIP and why should I use it?

There are many ways to create a connection between multiple hosts. PLIP, like SLIP, allow a local connection between two machines, but uses the parallel ports.

Parallel ports transfer more than one bit at a time, this means it is possible to achieve higher speeds than with a serial interface.

The speed achieved depends completely on your hardware (CPU and parallel port) and system load, in general it may be from 5 Kb/sec up to even 40 Kb/sec.

The PLIP interface is fast enough to allow some decent tcp/ip functions, like NFS. So, you may have a computer with all your Linux stuff and another with only the minimal system, where you can mount all the rest from the main machine.

The disadvantage is that most users have only one parallel port, this means that you won't be able to print and use PLIP together. Even with two parallel ports it is impossible to print and use PLIP without using kernel modules.

This disvantage can be also eliminated, if you have two or more parallel ports, applying a patch that you can find in this Mini-HOWTO.

Finally I am now able to give a hopefully good way to set up a PLIP link between DOS and Linux.

I won't stress it enough: so far nobody reported a successful link between Linux and Windows95.

2. Hardware required to use PLIP.

The hardware required to set up a PLIP interface is (obviously) a free parallel port in both the machines and the cable. If you can configure it with your BIOS, set it at least as "bi-directional", but if possible in ECP or EPP mode.

About the cable, this is what is written in the plip.c file, in the kernel 2.0.33 source:

```
The cable used is a de facto standard parallel null cable -- sold as
a "LapLink" cable by various places.  You'll need a 12-conductor cable to
make one yourself.  The wiring is:
  SLCTIN      17 - 17
  GROUND      25 - 25
  D0->ERROR   2 - 15          15 - 2
  D1->SLCT    3 - 13          13 - 3
  D2->PAPOUT  4 - 12          12 - 4
  D3->ACK     5 - 10          10 - 5
  D4->BUSY    6 - 11          11 - 6
Do not connect the other pins.  They are
  D5,D6,D7 are 7,8,9
  STROBE is 1, FEED is 14, INIT is 16
  extra grounds are 18,19,20,21,22,23,24
```

But I strongly advice you to read the /usr/src/linux/drivers/net/README1.PLIP and README2.PLIP files for more info about the cable.

In my opinion you should avoid building your own parallel null cable. A self-made cable may save very little money, but can add lots of headaches. If you wish to build your parallel cable, remember that you're doing it at your own risk, I reported exactly what is written in plip.c but I don't give warranties.

A final word about cable length: long cables (i.e. more than 10 feet or 3 meters) may bring problems due to radio interference. If you need long cables you should use good and well shielded cables, but very long cables are not recommended: I think the maximal cable lenght should be 15 meters (50 feet).

Anyway, someone mailed me that his/her 100 feet (30 meters) cable works fine; if someone really wants to try a PLIP connection between the office and his/her home (200 meters away), and has the money to spend, can try it, but is at his/her risk.

3. Reconfigure the kernel.

You're supposed to already know how to configure and compile the kernel, otherwise you must get some doc (kernel-howto or other guides). Thanks to the cool work made by the kernel guys, recompiling the last kernels is a really easy jobs also for "common" people, so just do it. Anyway, for the sake of completeness, here is a quick summary of what you must do:

NOTE: I suppose you are using the 2.0.xx kernel series. Now there is no need to keep the 1.2.xx kernels. There are no istructions about the 2.1.xx kernel series, since they are for development.

I will suppose that you use menuconfig to set up the kernel options, but the other tools are equivalent. I'll show how to do it with menuconfig:

```
#make menuconfig
```

I strongly advice to select

```
Loadable module support  --->
```

and enable the

```
[*] Enable loadable module support
```

and, if possible (i.e. you have modules.2.0.0) the

```
[*] Kernel daemon support (e.g. autoload of modules)
```

Then go back and choose

```
Networking options  --->
```

where you should choose at least

```
[*] Network firewalls
[*] TCP/IP networking
[*] IP: forwarding/gatewaying
```

The go back and choose at least

```
[*] Network device support
<M> PLIP (parallel port) support
```

If you use modules I definitely advice you to set up PLIP as a module. If you do so you can also, if you

need to use a printer, go to

```
Character devices  --->
```

and set up as a module the

```
<M> Parallel printer support
```

Now you have enabled the kernel support for PLIP. If it's the first time that you compile the kernel look at all the other options then save and exit.

Finally compile with

```
#make dep ; make clean
#make zlilo
```

And, if you use modules

```
# make modules
# make modules_install
```

Now reboot your system.

---

4. Kernel messages about the PLIP interface.

After you've reconfigured and compiled the kernel with PLIP support enabled, when you boot the system, if the kernel supports PLIP directly, or when you load (later, see below) the PLIP module if you compiled PLIP as modules, you should get something like this (numbers may differ):

```
NET3 PLIP version 2.2 gniibe@mri.co.jp
plip1: Parallel port at 0x378, using assigned IRQ 7.
```

Depending upon your klogd and syslogd configuration the plip message could have been stored in your system log files: don't panic if you don't see the above message. If you compiled PLIP as a module and lsmod shows that the plip module is loaded, then it's enough.

Please take notice of the interface name. Usually is plip1, but may be plip0 or even plip2, plip3, and so on. It depends on the IO Address.

5. Setting up the configuration files.

NOTE: Some distributions, like Debian, use different config files. If you have a standard installation and you don't find the rc.inet* files, look for (different) config files in the /etc/init.d directory.

First of all remember to backup all the files you will change,

```
#cp rc.inet1 rc.inet1.BACKUP
```

may be a good idea.

Now, if you don't have it done already, you must choose the IP addresses of the two machines. In my examples I'll use a couple of example IPs for the IPs that you'll write, in the standard xxx.xxx.xxx.xxx format.

In the /etc/rc.d/inet1.rc file of both the machines add this (better if in the last part of the file):

```
/sbin/route add -net ${NETWORK} netmask ${NETMASK}
```

Where NETWORK and NETMASK should be set up previously. If you don't know how to do it, please read the NET-2-HOWTO.

If after this route command you get a message like this:

```
SIOCADDRT: network unreachable
```

then use this instead:

```
/sbin/route add -net ${NETWORK} netmask ${NETMASK} dev plip1
```

---

where, as usually, you'll have to use the interface name reported by the kernel messages (see above).

You may safely ignore these variables only in the following case:

If you only want to connect two machines on a standalone network, you may pick-up any IP address, say 200.0.0.1 and 200.0.0.2 respectively. In this case you can safely put NETWORK="200.0.0.0" and NETMASK="255.255.255 .0". These are the example IPs that I use in my Quick PLIP Installation (see below).

NOTE: 200.0.0.1 and 200.0.0.2 are only example IPs, I advice not to use these numbers definitively because they could be the addresses of real hosts on Internet!

I strongly advice to choose your address between the "private address" intervals:

```
10.0.0.0        -   10.255.255.255
172.16.0.0      -   172.31.255.255
192.168.0.0     -   192.168.255.255
```

In the file /etc/hosts of both the machines you should add the entries with the IP of the machines that you connect via PLIP. In my example, the entries are:

```
200.0.0.1       one             # this is the "one" IP address
200.0.0.2       two             # this is the "two" IP address
```

Where one and two are the names you have chosen for the two hosts.

If you want to activate the NFS, beside answering yes during the kernel configuration, you must add in /etc/exports the entries that describe the directories that you wish to export. In my example, to be able to mount the directory /usr, you should add this entry:

```
/usr                            two (ro)
```

For more informations about NFS, please read the specific documentation; don't report me problems with the NFS, I won't be able to help.

Now reboot your system.

---

6. Activate the PLIP link.

Finally, these are the commands, that must be executed with root rights, that activate the PLIP interface (of course the cable must be already plugged correctly).

NOTE: If something unexpected happens, please doublecheck the cable and the spelling of the commands. If you followed the istructions correctly but the are still errors, read the FAQ paragraph, a lot of answers are already available.

First of all confirm that there is no lp device present:

```
# cat /proc/devices
```

You mustn't see any reference to lp like this:

```
 6 lp
```

If you see it, please remove (temporanely) the lp device before going on, if PLIP works then you can try it with lp later. To remove the lp device you'll have to use the rmmod if it's a module; if instead it's built in the kernel, you'll need to recompile the kernel with lp as a module (a much wiser idea).

Again I use the name one and two, as example. On one you'll have to do the following steps.

If you don't have the module automounter daemon and you compiled PLIP as a module, you must mount it:

```
# insmod plip
```

NOTE: if your parallel port is on an IRQ different from 7 and/or is on a IO Address different from 0x378, then you'll have to tell it to insmod. Find your real IRQ and IO Address (the DOS command MSD is likely to be ok, but don't trust it too much) and write something like this:

```
# insmod plip io=0x278 irq=5
```

Usually IRQ is 7 or 5, while IO Address is 0x378, 0x278 or 0x3bc. It is important that you check that the address and IRQ match the hardware settings (jumpers on old boards, BIOS on modern motherboards).

If you are paranoic check that the module has been loaded with:

```
  # lsmod

Module:         #pages:  Used by:
plip                3           0
```

Take notice of the interface name (plip0, plip1, and so on; for more details read the kernel messages

chapter above), then set up the PLIP interface:

```
# ifconfig plip1 one pointopoint two up
```

NOTE: if your parallel port is on an IRQ different from 7 and/or is on a IO Address different from 0x378, then you'll have to tell it to ifconfig. Use the same IRQ and IO Address reported by the kernel messages and write something like this:

```
# ifconfig plip1 irq 7
# ifconfig plip1 io_addr 0x3bc
```

Usually IRQ is 7 or 5, while IO Address is 0x378, 0x278 or 0x3bc.

Now check that it worked...

```
# ifconfig




 .....
 .....
 plip1     Link encap:10Mbps Ethernet  HWaddr FC:FC:C8:00:00:01
           inet addr:200.0.0.1  P-t-P:200.0.0.2  Mask:255.255.255.0
           UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0
           TX packets:0 errors:0 dropped:0 overruns:0
           Interrupt:7 Base address:0x378
```

Add the route to two...

```
# route add two plip1
```

And, if you want also the NFS for two:

```
# rpc.portmap
# rpc.mountd
# rpc.nfsd
```

On "two" the commands are the same, but you must write one instead of two and vice versa.

One of your machines is likely to have only the PLIP connection, if this is true and that machine is two, you may also type:

```
# route add default gw one
```

on that machine. In my example above, two is a laptop with only a PLIP connection with one, so I type the above line on two.

Finally check with a

```
# ping two
```

from one and a

```
# ping one
```

from two to see that all is working.

Of course you may want to have all these commands automatically done by a script or at boot time. You must only create a script that execute these commands: now you may invoke it as root when you need, or you may add a command (in /etc/rc.d/rc.inet2) that calls it at boot time.

To tune your PLIP, you can use the plipconfig command, see the man page for more informations.

To shutdown PLIP, you need only to do:

```
# ifconfig plip1 down
```

which removes also the route entries. If you don't have the automounter daemon, then remove also the module:

```
# rmmod plip
```

---

7. FAQ.

This section will (hopefully) solve your problems. If you have any other question, feel free to mail me anytime.

- I get these messages at boot time (or when I load the plip module):

  ```
  SIOCSIFADDR: No such device
  SIOCADDRT: Network is unreachable
  ```

  and when I try to set up the link as written above, I get again error messages like:

  ```
  SIOCSIFADDR: No such device
  SIOCSIFDSTADDR: No such device
  SIOCADDRT: Network is unreachable
  mount clntudp_create: RPC: Port Mapper failure - RPC: Unable to send
  ```

  - The kernel, for some reason, hasn't PLIP support enabled. This could be due to:
    - You didn't answer yes to "PLIP support? " during kernel configuration.
    - You answered yes to "Printer support? " during kernel configuration.
    - You compiled PLIP as a module, so you must load it.
    - You are addressing the wrong port, i.e. you wrote for instance plip1 instead of plip0.
- Is there a way to support both PLIP and LP, beside modules, perhaps with two parallel ports?
  - Yes, so far there are two ways, described in the "Patches to make PLIP and LP live together":
    - You can apply a patch to make the kernel support both.
    - You can apply another patch to make the kernel use a parallel port for PLIP and another for LP.
- I have created the script that connects my 2 computers. I set up the link automatically in my rc.inet2, where I call a script that creates the link and enables NFS. My "two" hosts mounts some "one"'s directories; I have added the correct entries in "two"'s /etc/fstab. If I boot "two" when "one" is down, "two" halts for some minutes on the "mounting remote file systems...".
  - This happens because "two" waits to mount the "one" filesystems, but if "one" is down you must wait until "two" is bored of waiting. To avoid this, you may:
    - Comment out in rc.inet2 the command that mounts the remote filesystems
    - Remove the entry in "two"'s /etc/fstab and mount the remote filesystems manually when and if you need.
    - A better solution would be for "two" to detect upon booting whether "one" is up, and mount the filesystem if it is. This can be accomplished by replacing the mount command in rc.d or whereever with something like the following:

      ```
      if ping -c 5 one ; then
        mount one:/.....
      fi
      ```

- My link is up, but ping fails. I receive the following message from the kernel:

  ```
  plip1: timed out (1, 89)
  ```

or similar messages.

- This means that the "your side of the link" is working, your machine sends the signal, but the "other side" isn't answering or your side is not waiting at the proper IRQ/IO Address. This is the most common problem and, alas, has a lot of possible reasons, usually bad cable or wrong IRQ and/or IO Address. The wrong IRQ is the source of over 60% of the problems, so it's very likely that changing it will remove the problem. Here is a detailed list of possible reasons:
  - The cable isn't plugged properly or is broken or is wrong. Check it, if possible, between two Linux hosts which already work with PLIP. If it is not possible, then at least test the cable with a tester. The fact that the cable worked/not worked with DOS/win95 is a good/bad omen but is not a proof.
  - The "other side" machine has not PLIP up.
  - You are linked with a notebook with a not proper parallel port, see below.
  - You have a really cheap parallel port that is a simple "printer" port, so can send and not receive.
  - Your parallel port is not set as (at least) bi-directional. Do it in the BIOS configuration. Advanced parallel port settings like EPP or ECP are ok.
  - The parallel ports have different irq, so you have to load the plip module (or the lp module) with a different irq. Go back to the chapter "Activate the PLIP link" and choose a different irq.
  - Some other device may have shared your irq (which usually is irq 7), it may be a sound card. Do not trust DOS programs like MSD, instead try to load the plip module with a different irq.

- I put the right IRQ and IO Address, but it still doesn't work. I got the addresses from the MSD command.
  - I got a report from MSD giving wrong port addresses. Try to use this program: http://www.cs.caltech.edu/ huny/para13.zip.
- My link is up, and ping works. I sometimes receive the following message from the kernel:

```
plip1: timed out (1, 89)
```

  or similar messages.
  - This means that the other side has not answered before the timeout. If all is working, you can ignore these messages: usually means that the other side is much slower than yours, either due to older hardware or more load. You can try to tune PLIP with the plipconfig command.
- I have installed the PLIP connection but if I ping I get 100% data loss. I connected my desktop with a notebook.
  - Some notebook's parallel ports aren't good for PLIP, because they are only "printer ports", i.e. they can only transmit but not receive the data. So far I don't know if there is a way to make them work. The only hope is:
    - Look at your notebook setup, perhaps there is a way to configure the parallel port as a parallel port instead of a printer port. Usually is called "parallel enhanced mode".
    - Try plip mode 0. Alas I don't know how to do it and/or if it works or is still available in the last kernels.
- What speeds can I achieve with PLIP?
  - This is an hard question to answer to, because there are MANY factors that can change deeply your performance:
    - The CPU speed on both the sides of the link.
    - The parallel port type and settings.
    - The system load.
    - What do you use PLIP for.

    Just to give a rough idea, you should achieve about 40Kbytes/sec, much faster than any serial rate and near to a low-level ethernet card.
- What happens if I need to ifconfig up and ifconfig down many times plip1?

TOC

  - Seems that you need to add a -arp to the ifconfig command, except for the first time after each boot. I don't need, but perhaps someone does.
- I have read the IP numbers reserved for private networks and your 200.0.0.1 and 200.0.0.2 are not in these ranges. Shouldn't they be changed?
  - Yes, they should. But as I underline since the beginning I choose these IP addresses only because of their simplicity, you are free to change them as you wish. Here is a cut from the net-2-howto:

```
RFC1597 has specifically reserved some IP addresses for private
networks.  You should use these as they prevent anything nasty
happening if you accidentally get connected to the Internet. The
addresses reserved are:

  10.0.0.0        -   10.255.255.255
  172.16.0.0      -   172.31.255.255
  192.168.0.0     -   192.168.255.255
```

- Is there a way to fine tune PLIP parameters without editing the source code?
  - Yes, there is. Try the /sbin/plipconfig command. See the man page for more info.
- I'm running Debian GNU/Linux, and under Debian, the files /etc/rc.d/rc.inet1 and 2 do not exist. Where must I write the plip configuration commands?
  - In Debian GNU/Linux you must edit /etc/init.d/network, where you have to put all the commands that should stay in rc.inet1 and 2.
- I have some problems linking two hosts with PLIP. The first has the latest kernel, the second still uses the 1.0.x PLIP version: is this a problem?
  - Yes, it's much better, where is possible, to have the same PLIP version on both ends. In the plip.c is written that the actual PLIP cannot work with the 1.0.xx PLIP.
- Right now PLIP works with 4 bits, what about the 8 bit PLIP I've read in the kernel docs? I think is called Mode 1.
  - This Mini-HowTo is for configuration, for technical informations please read the /usr/src/linux/drivers/net/README*.PLIP files or contact the author. What I know is only this: the standard PLIP uses "null printer" cables and is the Mode 0 (don't confuse it with plip0, which is the interface name), which uses 4 bits; Mode 1 uses 8 bits and should be available already, but will need an handmade cable and will work only between 2 Linux hosts. I don't know, once you got the cable, how to set up the Mode 1 PLIP link; if somebody does, please let me know.

---

8. Patches to make PLIP and LP live together.

The best way to make PLIP and LP live together is to use kernel modules: you can load plip.o and unload it when you need to print or vice versa. If you do really need to use both PLIP and LP, try the following patches.

8.1 PLIP and LP together on the same port.

If for some reason you wish PLIP and LP supported directly by the kernel, you can try these patches.

You must modify the following pieces of code, but *backup* the files before:

```
******** modifications to linux/drivers/char/lp.c **********************
struct lp_struct lp_table[] = {
        { 0x3bc, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL,
NULL, },
/*      { 0x378, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL,
NULL, },
        { 0x278, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL,
NULL, },
*/
};
#define LP_NO 1

******** modifications to linux/drivers/net/Space.c *******************
#if defined(PLIP) || defined(CONFIG_PLIP)
    extern int plip_init(struct device *);
    static struct device plip2_dev = {
        "plip2", 0, 0, 0, 0, 0x278, 2, 0, 0, 0, NEXT_DEV, plip_init, };
    static struct device plip1_dev = {
        "plip1", 0, 0, 0, 0, 0x378, 7, 0, 0, 0, &plip2_dev, plip_init, };
/*    static struct device plip0_dev = {
        "plip0", 0, 0, 0, 0, 0x3BC, 5, 0, 0, 0, &plip1_dev, plip_init, };
*/
#   undef NEXT_DEV
#   define NEXT_DEV     (&plip1_dev)
#endif  /* PLIP */
```

Of course there is the standard disclaimer: *I received these patches and I put them "as I got them". This means that you try them at your own risk.* Anyway, your biggest trouble should be only restore the original files and recompile.

8.2 PLIP and LP on different ports.

If you have at least 2 parallel ports you can try these patches, that should allow you to use PLIP on a port and LP on the other.

1. Comment out one line in kernel source file, drivers/char/lp.c.

```
struct lp_struct lp_table[] = {
{ 0x3bc, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
{ 0x378, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
/* { 0x278, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, }, */
};
    3 -> 2
```

---

2. Kernel configuration

```
PLIP (parallel port) support (CONFIG_PLIP) [n] y

Parallel Printer support [y] y
```

3. Kernel message at startup

```
lp1 at 0x0378, using polling driver
.....
NET3 PLIP version 2.0 gniibe@mri.co.jp
plip2: Parallel port at 0x278, using assigned IRQ 5.
```

Again the standard disclaimer, like section 8.1.

---

9. A PLIP link between DOS and Linux

After the first release of this Mini-HowTo many people wrote for info about a link between Linux and DOS (or Windows) computer. The general interest lead me to add this chapter, I hope will be of help to everybody.

This section comes from an article I've found on Linux Gazzette by James McDuffie <mcduffie@scsn.net> . It covers the basic installation of a PLIP link between Linux and a DOS computer using Windows and Trumpet WinSock and gives the address of a cool program that let's you run X-Windows programs on Windows.

The last section is an addendum sent by James Vahn jvahn@short.circuit.com where he describes deeply how to set up this link and how to solve many problems.

For any questions about this chapter please contact him, not me.

9.1 DOS-Linux link.

I suppose you have already set up properly the PLIP support on the Linux side and you have got the right cable, else go back to the previous chapters.

Now, for the DOS side, you need first of all a packet driver. It can be found here:

ftp://ftp.crynwr.com/drivers/plip.zip

The program runs under DOS and acts like a Ethernet Packet driver. If you want to use PLIP with Windows you need also Trumpet Winsock. This serves as the TCP/IP interface. Otherwise, you can probably find TCP/IP software for DOS.

Now go back to the Linux computer and add the DOS computer address to /etc/hosts. If your DOS computer does not have a registered IP address you may choose any address (remember the warning of chapter 3 about IP addresses).

Now let's suppose you chose the name linux for the Linux computer and dos for the DOS one. You have to type:

```
ifconfig plip1 linux pointopoint dos arp up
route add dos
```

Of course if you want to have this done every time you boot the linux computer you may add these lines to the file /etc/rc.d/rc.inet1:

```
/sbin/ifconfig plip1 linux pointopoint dos arp up
/sbin/route add dos
```

This sets up the interface and then adds a route to it. Of course if you are using the second parallel port you have to write plip2 instead.

Go back to the DOS/Windows computer and edit autoexec.bat, you have to add the following lines.

```
c:\plip\plip.com 0x60
c:\tcpip\winsock\winpkt.com 0x60
```

Of course I suppose you put plip.com (the packet driver) in the directory c:/plip and the winpkt.com in c:/tcpip, else need to put the right path.

This sets the plip.com program on packet vector 0x60 and then loads the winpkt.com program that comes with trumpet winsock on the same vector. If the cable is something other than lpt1 you will have to tell plip.com the irq number and io address. Also, winpkt.com needs to run to make the packet vector avaliable to Windows. From here we go to the actual setup under Trumpet Winsock. All you have to do is unselect SLIP or PPP and enter 60 into the box labeled Packet vector. Then tell it the IP address you gave it, the IP address of the Linux computer as the default gateway and the Name Server as either you computer's ip or your ISP's address for its nameservers if your going to connect it to the Internet (more on this later). Close the setup and re-run Winsock and you should have it! Put winsock in your startup group and you have everything setup automatically!

If you want to access the Internet through the Linux computer on the Windows computer you will need to set up IP Masquerading, for info on this see the NET-2-HOWTO. This simply masquerades the Windows computer with your Linux computer's IP address.

Also I have found a program that lets you run X-Windows programs under Windows! It is located at:

http://www.tucows.com/

Set it up according to directions and then all you have to do is telnet in from the Windows computer then set the display to the Windows computer (`DISPLAY=duncan:0.0` for instance) and run the program desired. There is nothing cooler than running xv under Windows! Hope all this helped.

9.2 A DOS-Linux PLIP link experience.

NOTE: I received this document from James Vahn jvahn@short.circuit.com. I put it here unchanged. This means that **for any question about this section he's much better qualified than me so please mail to him than to me**. His experience with a PLIP connection of a floppy-only DOS computer to a Linux one is the perfect example of how to work-around common problems.

Last Update 11 July 1996

My floppy-only DOS box is networked via PLIP to the second printer port on the Linux machine. The first Linux printer port has a printer on it, both are permanently connected and the DOS box is telnet'd into Linux. These are my notes on what I did to accomplish this.

When the kernel probes for printer ports, it will grab all of them unless you remove one from the probe. Otherwise PLIP will get nothing. One method is to load the drivers as modules when needed...

<gniibe@mri.co.jp> writes:

I keep recommending using PLIP/LP as kernel module, since

- modules are flexible for change of configuration
- (re)compiling the kernel is not easy for novice users
- co-existing PLIP and LP is only feasible by the modules

With PLIP/LP as kernel module, you can specify which port is PLIP and which port is LP. Here is example:

```
# insmod lp.o io=0x378
# insmod plip.o io=0x278 irq=2
```

Even you can use two parallel ports:

```
# insmod plip.o io=0x278,0x3bc irq=2,5
```

TOC

In the example above,

plip0 is assigned on 0x278 and it's irq is 2,

plip1 is assigned on 0x3bc and it's irq is 5, respectively.

Using modules certainly sounds like the way to go. The following method shows how to patch the kernel to allow both a printer and PLIP on different ports, without modules. If you are unfamiliar with the module concept, you might find this quicker to set up.

You will need to modify two files in the kernel source tree. I'm using kernel 1.2.13 and found some changes were needed in ../linux/drivers/net/Space.c to accommodate my system. Look at around line 205 for the PLIP definitions to make sure your port and IRQ match, and make a note of which driver it will be (plip0, plip1, plip2). In my case port 0x278 uses IRQ 5 (the card is jumpered that way) but Space.c defined it with IRQ 2. I made the changes here, rather than opening up the box and changing jumpers. The alternative is to specify the IRQ through ifconfig later on, but the kernel will boot up with the wrong IRQ for PLIP and it may annoy you. It is a simple (single character) change.

The next, and more difficult step:

In .../drivers/char/lp.c you will find the following at around line 38:

```
struct lp_struct lp_table[] = {
    { 0x3bc, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
    { 0x378, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
/*  { 0x278, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
 * 0x278 reserved for plip1
 *
 * };
 * #define LP_NO 3
 */
};
#define LP_NO 2
```

Notice the changes to make- one port is commented out, so now only 2 ports are defined. Port 0x3BC will probably not work for PLIP- the IRQ line is usually broken on these ports, as found on old monochrome adapters (MDA).

You made backups of these file before you changed them, right? Now make a new kernel with printer, net, dummy, and plip support.

Configure the system. This is my /etc/rc.d/rc.inet1 file:

```
#!/bin/bash
#
/sbin/ifconfig lo 127.0.0.1
/sbin/route add -net 127.0.0.0

/sbin/ifconfig dummy 200.0.0.1
/sbin/route add -net 200.0.0.0 netmask 255.255.255.0
/sbin/ifconfig plip1 arp 200.0.0.1 pointopoint 200.0.0.2 up
/sbin/route add 200.0.0.2
/sbin/ifconfig dummy down
```

TOC

Notice that arp is used for the DOS-to-Linux connection, apparently not used on Linux-to-Linux connections.

And in /etc/hosts you can add these, just to give the two machines names:

```
200.0.0.1       console1
200.0.0.2       console2
```

The DOS box is console2. Note Andrea's warning about these, better to use official numbering schemes.

Reboot so all of these changes and the new kernel will take effect. During the boot sequence (or by running dmesg) if you made the patches, otherwise when the modules are loaded:

```
lp0 at 0x03bc, using polling driver
lp1 at 0x0378, using polling driver
[....]
NET3 PLIP version 2.0 gniibe@mri.co.jp
plip1: Parallel port at 0x278, using assigned IRQ 5.
```

The "route" command shows this:

```
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
console2        *               255.255.255.255 UH    1436   0      136 plip1
loopback        *               255.0.0.0       U     1936   0      109 lo
```

And "ifconfig plip1" shows:

```
plip1     Link encap:10Mbps Ethernet  HWaddr FC:FC:C8:00:00:01
          inet addr:200.0.0.1  P-t-P:200.0.0.2  Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:1500  Metric:1
          RX packets:132 errors:0 dropped:0 overruns:0
          TX packets:136 errors:0 dropped:0 overruns:0
          Interrupt:5 Base address:0x278
```

Look at /etc/inetd.conf and see if telnet is enabled. You might want to read the man page for tcpd, and the use of /etc/hosts.allow (ALL: LOCAL) and /etc/hosts.deny (ALL: ALL). You should be able to "telnet localhost".

Linux is done, now the DOS side. Again, be suspicious of port 0x3BC if one is present.

I'm using NCSA's telnet and Crynwr's PLIP driver found at these sites:

ftp://ftp.ncsa.uiuc.edu/Telnet/DOS/ncsa/tel2308b.zip

ftp://ftp.crynwr.com/drivers/plip.zip

Be sure to use NCSA's version 2.3.08 telnet and version 11.1 of Crynwr's PLIP driver. Please find and read Crynwr's SUPPORT.DOC located elswhere.

The CONFIG.TEL file. Most of it is the default and to save some space I've tried to cut it back here to just the info you need (hopefully). The second port on this machine is setup as 0x278 on IRQ 5.

```
myip=200.0.0.2
netmask=255.255.255.0       # subnetting mask
hardware=packet             # network adapter board (packet driver interface)
interrupt=5                 # IRQ which adapter is set to
ioaddr=60                   # software interrupt vector driver is using
#
#[...lots unchanged...]
#
# at the end of the file, put this line:
name=console1 ; hostip=200.0.0.1 ; nameserver=1 ; gateway=1
```

(console1 is the name of the Linux machine, you can use whatever you like)

I made a 12 foot null cable between both machines, and (after initially finding it miswired) there have been no problems. A standard 11-wire null printer cable should work too. The Linux plip.c source shows the wiring. Although my cable has the 17-17 connection, I don't think it is used for anything and was not present on a ready-made cable.

```
@echo off
plip.com 0x60 5 0x278
telbin -s console1
```

That should connect you to the Linux box on /dev/ttyp. NCSA's telnet provides for 8 virtual screens and also acts as an ftp server. The PLIP interface provides a fair throughput, I'm getting 6.5K/s file transfers with my antiques. Let's hope you can do better. :-)

---

10. PLIP between Linux and Windows 95.

This section is empty. I use windows 95 for nothing serious but games, so I don't try and don't care about a PLIP link with Linux. The questions about such a link have won the most asked question contest, so I give here a (so far) definitive answer.

No, so far nobody reported me a successful link between Linux and Windows 95. if somebody succeds in setting up this link, please let me know immediately: thousand of PLIP users await these news!

11. Questions? Comments? Send me feedback.

For any questions and comments you can find me via e-mail at the address controzz@cli.di.unipi.it

Feedback is welcome, any error report is precious. The next release will have an even larger FAQ section, if you send questions and, of course, the answers if you find them by yourself.

Please do not send questions already present in the FAQ.

If you have to ask me for help, please be sure to let me know any information that can help me, at least: kernel version, commands used, error messages, the cable you used and any other system message related to PLIP.

Please remember not to send me any question about PLIP with DOS/Windows 3.1/Windows 95, I can't help you. These questions should be sent to James Vahn jvahn@short.circuit.com, who sent me the DOS addendum. Again: it's useless to ask him or me about PLIP with Windows 95.

12. Where to find new releases of this mini-howto.

This mini-HOWTO is maintained by the HOWTO coordinators and is posted monthly on comp.os.linux.answers and can be found in the HOWTO directory at sunsite and at sunsite's mirrors.

Another way to find the mini-HOWTO (and to contact me) is on my Home Page,

http://www.cli.di.unipi.it/~controzz/intro.html (italian language)

http://www.cli.di.unipi.it/~controzz/intro_e.html (english language)

13. Credits.

Many thanks to:

- Rick Lim <ricklim@freenet.vancouver.bc.ca> for the patches to make PLIP and LP live together.
- Takeshi Okazaki <GBA03552@niftyserve.or.jp> for the patches to use PLIP and LP on two different parallel ports.
- Jim Van Zandt <jrv@vanzandt.mv.com> for some advice on the "tutorial" part of this HOWTO.
- Fernando Molina <fmolina@nexo.es> for useful comments about IRQs and IO Addresses.
- James Vahn <jvahn@short.circuit.com> for the cool addendum on the PLIP between DOS and Linux chapter.
- To all the users that posted PLIP-related articles on the linux newsgroups and/or mailed me. The list of all the people that helped me with info and comments could be longer than the Mini-HOWTO itself: thank you all!

14. Copyright message.

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Greg Hankins, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu via email.

Linux Partition HOWTO

Kristan Koehntopp, kris@koehntopp.de

v2.4, 3 November 1997

This Linux Mini-HOWTO teaches you how to plan and layout disk space for your Linux system. It talks about disk hardware, partitions, swap space sizing and positioning considerations. file systems, file system types and related topics. The intent is to teach some background knowledge, not procedures.

1. Introduction

- 1.1 What is this?
- 1.2 What is in it? and related HOWTO documents

2. What is a partition anyway?

- 2.1 Backups are important
- 2.2 Device numbers and device names

3. What Partitions do I need?

- 3.1 How many partitions do I need?
- 3.2 How large should my swap space be?
- 3.3 Where should I put my swap space?
- 3.4 Some facts about file systems and fragmentation
- 3.5 File lifetimes and backup cycles as partitioning criteria

4. An example

- 4.1 A recommended model for ambitious beginners

5. How I did it on my machine

1. Introduction

1.1 What is this?

This is a Linux Mini-HOWTO text. A Mini-HOWTO is a small text explaining some business related to Linux installation and maintenance tutorial style. It's mini, because either the text or the topic it discusses are too small for a real HOWTO or even a book. A HOWTO is not a reference: that's what manual pages are for.

1.2 What is in it? and related HOWTO documents

This particular Mini-HOWTO teaches you how to plan and layout disk space for your Linux system. It talks about disk hardware, partitions, swap space sizing and positioning considerations, file systems, file system types and related topics. The intent is to teach some background knowlegde, so we are talking mainly principles and not tools in this text.

Ideally, this document should be read before your first installation, but this is somehow difficult for most people. First timers have other problems than disk layout optimization, too. So you are probably someone who just finished a Linux installation and is now thinking about ways to optimize this installation or how to avoid some nasty miscalculations in the next one. Well, expect some desire to tear down and rebuild your installation when you are finished with this text. :-)

This Mini-HOWTO limits itself to planning and layouting disk space most of the time. It does not discuss the usage of fdisk, LILO, mke2fs or backup programs. There are other HOWTOs that address these problems. Please see the Linux HOWTO Index for current information on Linux HOWTOs. There are instructions for obtaining HOWTO documents in the index, too.

To learn how to estimate the various size and speed requirements for different parts of the filesystem, see "Linux Multiple Disks Layout mini-HOWTO", by Gjoen Stein <gjoen@nyx.net>.

For instructions and considerations regarding disks with more than 1024 cylinders, see "Linux Large Disk mini-HOWTO", Andries Brouwer <aeb@cwi.nl>.

For instructions on limiting disk space usage per user (quotas), see "Linux Quota mini-HOWTO", by Albert M.C. Tam <bertie@scn.org>

Currently, there is no general document on disk backup, but there are several documents with pointers to specific backup solutions. See "Linux ADSM Backup mini-HOWTO", by Thomas Koenig <Thomas.Koenig@ciw.uni-karlsruhe.de> for instructions on integrating Linux into an IBM ADSM backup environment. See "Linux Backup with MSDOS mini-HOWTO", by Christopher Neufeld <neufeld@physics.utoronto.ca> for information about MS-DOS driven Linux backups.

For instructions on writing and submitting a HOWTO document, see the Linux HOWTO Index, by Tim

Bynum <linux-howto@sunsite.unc.edu>.

Browsing through /usr/src/linux/Documentation can be very instructive, too. See ide.txt and scsi.txt for some background information on the properties of your disk drivers and have a look at the filesystems/ subdirectory.

---

2. What is a partition anyway?

When PC hard disks were invented people soon wanted to install multiple operating systems, even if their system had only one disk. So a mechanism was needed to divide a single physical disk into multiple logical disks. So that's what a partition is: A contiguous section of blocks on your hard disk that is treated like a completely seperate disk by most operating systems.

It is fairly clear that partitions must not overlap: An operating system will certainly not be pleased, if another operating system installed on the same machine were overwriting important information because of overlapping partitions. There should be no gap between adjacent partitions, too. While this constellation is not harmful, you are wasting precious disk space by leaving space between partitions.

A disk need not be partitioned completely. You may decide to leave some space at the end of your disk that is not assigned to any of your installed operating systems, yet. Later, when it is clear which installation is used by you most of the time, you can partition this left over space and put a file system on it.

Partitions can not be moved nor can they be resized without destroying the file system contained in it. So repartitioning usually involves backup and restore of all file systems touched during the repartitioning. In fact it is fairly common to mess up things completely during repartitioning, so you should back up anything on any disk on that particular machine before even touching things like `fdisk`.

Well, some partitions with certain file system types on them actually *can* be split into two without losing any data (if you are lucky). For example there is a program called "fips" for splitting MS-DOS partitions into two to make room for a Linux installation without having to reinstall MS-DOS. You are still not going to touch these things without carefully backing up everything on that machine, aren't you?

2.1 Backups are important

Tapes are your friend for backups. They are fast, reliable and easy to use, so you can make backups often, preferably automatically and without hassle.

Step on soapbox: And I am talking about real tapes, not that disk controller driven ftape crap. Consider buying SCSI: Linux does support SCSI natively. You don't need to load ASPI drivers, you are not losing precious HMA under Linux and once the SCSI host adapter is installed, you just attach additional disks, tapes and CD-ROMs to it. No more I/O addresses, IRQ juggling or Master/Slave and PIO-level matching.

Plus: Proper SCSI host adapters give you high I/O performance without much CPU load. Even under heavy disk activity you will experience good response times. If you are planning to use a Linux system as a major USENET news feed or if you are about to enter the ISP business, don't even think about deploying a system without SCSI. Climb of soapbox.

2.2 Device numbers and device names

TOC

The number of partitions on an Intel based system was limited from the very beginning: The original partition table was installed as part of the boot sector and held space for only four partition entries. These partitions are now called primary partitions. When it became clear that people needed more partitions on their systems, logical partitions were invented. The number of logical partitions is not limited: Each logical partition contains a pointer to the next logical partition, so you can have a potentially unlimited chain of partition entries.

For compatibility reasons, the space occupied by all logical partitions had to be accounted for. If you are using logical partitions, one primary partition entry is marked as "extended partition" and its starting and ending block mark the area occupied by your logical partitions. This implies that the space assigned to all logical partitions has to be contiguous. There can be only one extended partition: no `fdisk` program will create more than one extended partition.

Linux cannot handle more than a limited number of partitions per drive. So in Linux you have 4 primary partitions (3 of them useable, if you are using logical partitions) and at most 15 partitions altogether on an SCSI disk (63 altogether on an IDE disk).

In Linux, partitions are represented by device files. A device file is a file with type c (for "character" devices, devices that do not use the buffer cache) or b (for "block" devices, which go through the buffer cache). In Linux, all disks are represented as block devices only. Unlike other Unices, Linux does not offer "raw" character versions of disks and their partitions.

The only important thing with a device file are its major and minor device number, shown instead of the files size:

```
$ ls -l /dev/hda
brw-rw----   1 root     disk       3,   0 Jul 18  1994 /dev/hda
                                   ^    ^
                                   |    minor device number
                                   major device number
```

When accessing a device file, the major number selects which device driver is being called to perform the input/output operation. This call is being done with the minor number as a parameter and it is entirely up to the driver how the minor number is being interpreted. The driver documentation usually describes how the driver uses minor numbers. For IDE disks, this documentation is in `/usr/src/linux/Documentation/ide.txt`. For SCSI disks, one would expect such documentation in `/usr/src/linux/Documentation/scsi.txt`, but it isn't there. One has to look at the driver source to be sure (`/usr/src/linux/driver/scsi/sd.c:184-196`). Fortunately, there is Peter Anvin's list of device numbers and names in `/usr/src/linux/Documentation/devices.txt`; see the entries for block devices, major 3, 22, 33, 34 for IDE and major 8 for SCSI disks. The major and minor numbers are a byte each and that is why the number of partitions per disk is limited.

TOC

By convention device files have certain names and many system programs have knowledge about the names compiled in. They expect your IDE disks to be named `/dev/hd*` and your SCSI disks to be named `/dev/sd*`. Disks are numbered a, b, c and so on, so `/dev/hda` is your first IDE disk and `/dev/sda` is your first SCSI disk. Both devices represent entire disks, starting at block one. Writing to these devices with the wrong tools will destroy the master boot loader and partition table on these disks, rendering all data on this disk unusable or making your system unbootable. Know what you are doing and, again, back up before you do it.

Primary partitions on a disk are 1, 2, 3 and 4. So `/dev/hda1` is the first primary partition on the first IDE disk and so on. Logical partitions have numbers 5 and up, so `/dev/sdb5` is the first logical partition on the second SCSI disk.

Each partition entry has a starting and an ending block address assigned to it and a type. The type is a numerical code (a byte) which designates a particular partition to a certain type of operating system. For the benefit of computing consultants partition type codes are not really unique, so there is always the probability of two operating systems using the same type code.

Linux reserves the type code 0x82 for swap partitions and 0x83 for "native" file systems (that's ext2 for almost all of you). The once popular, now outdated Linux/Minix file system used the type code 0x81 for partitions. OS/2 marks it's partitions with a 0x07 type and so does Windows NT's NTFS. MS-DOS allocates several type codes for its various flavors of FAT file systems: 0x01, 0x04 and 0x06 are known. DR-DOS used 0x81 to indicate protected FAT partitions, creating a type clash with Linux/Minix at that time, but neither Linux/Minix nor DR-DOS are widely used any more. The extended partition which is used as a container for logical partitions has a type of 0x05, by the way.

Partitions are created and deleted with the `fdisk` program. Every self respecting operating system program comes with an `fdisk` and traditionally it is even called `fdisk` (or `FDISK.EXE`) in almost all OSes. Some `fdisk`s, noteable the DOS one, are somehow limited when they have to deal with other operating systems partitions. Such limitations include the complete inability to deal with anything with a foreign type code, the inability to deal with cylinder numbers above 1024 and the inability to create or even understand partitions that do not end on a cylinder boundary. For example, the MS-DOS fdisk can't delete NTFS partitions, the OS/2 fdisk has been known to silently "correct" partitions created by the Linux fdisk that do not end on a cylinder boundary and both, the DOS and the OS/2 fdisk, have had problems with disks with more than 1024 cylinders (see the "large-disk" Mini-Howto for details on such disks).

---

3. What Partitions do I need?

3.1 How many partitions do I need?

Okay, so what partitions do you need? Well, some operating systems do not believe in booting from logical partitions for reasons that are beyond the scope of any sane mind. So you probably want to reserve your primary partitions as boot partitions for your MS-DOS, OS/2 and Linux or whatever you are using. Remember that one primary partition is needed as an extended partition, which acts as a container for the rest of your disk with logical partitions.

Booting operating systems is a real-mode thing involving BIOSes and 1024 cylinder limitations. So you probably want to put all your boot partitions into the first 1024 cylinders of your hard disk, just to avoid problems. Again, read the "large-disk" Mini-Howto for the gory details.

To install Linux, you will need at least one partition. If the kernel is loaded from this partition (for example by LILO), this partition must be readable by your BIOS. If you are using other means to load your kernel (for example a boot disk or the LOADLIN.EXE MS-DOS based Linux loader) the partition can be anywhere. In any case this partition will be of type 0x83 "Linux native".

Your system will need some swap space. Unless you swap to files you will need a dedicated swap partition. Since this partition is only accessed by the Linux kernel and the Linux kernel does not suffer from PC BIOS deficiencies, the swap partition may be positioned anywhere. I recommed using a logical partition for it (/dev/?d?5 and higher). Dedicated Linux swap partitions are of type 0x82 "Linux swap".

These are minimal partition requirements. It may be useful to create more partitions for Linux. Read on.

3.2 How large should my swap space be?

If you have decided to use a dedicated swap partition, which is generally a Good Idea [tm], follow these guidelines for estimating its size:

- In Linux RAM and swap space add up (This is not true for all Unices). For example, if you have 8 MB of RAM and 12 MB swap space, you have a total of about 20 MB virtual memory.
- When sizing your swap space, you should have at least 16 MB of total virtual memory. So for 4 MB of RAM consider at least 12 MB of swap, for 8 MB of RAM consider at least 8 MB of swap.
- In Linux, a single swap partition can not be larger than 128 MB. That is, the partition may be larger than 128 MB, but excess space is never used. If you want more than 128 MB of swap, you have to create multiple swap partitions.
- When sizing swap space, keep in mind that too much swap space may not be useful at all. Every process has a "working set". This is a set of in-memory pages which will be referenced by the processor in the very near future. Linux tries to predict these memory accesses (assuming that recently used pages will be used again in the near future) and keeps these pages in RAM if

possible. If the program has a good "locality of reference" this assumption will be true and prediction algorithm will work. Holding a working set in main memory does only work if there enough main memory. If you have too many processes running on a machine, the kernel is forced to put pages on disk that it will reference again in the very near future (forcing a page-out of a page from another working set and then a page-in of the page referenced). Usually this results in a very heavy increase in paging activity and in a sustantial drop of performance. A machine in this state is said to be "thrashing" (For you german readers: That's "thrashing" ("dreschen", "schlagen", "haemmern") and not trashing ("muellen")). On a thrashing machine the processes are essentially running from disk and not from RAM. Expect performance to drop by approximately the ratio between memory access speed and disk access speed. A very old rule of thumb in the days of the PDP and the Vax was that the size of the working set of a program is about 25% of its virtual size. Thus it is probably useless to provide more swap than three times your RAM. But keep in mind that this is just a rule of thumb. It is easily possible to create scenarios where programs have extremely large or extremely small working sets. For example, a simulation program with a large data set that is accessed in a very random fashion would have almost no noticeable locality of reference in its data segment, so its working set would be quite large. On the other hand, an xv with many simultaneously opened JPEGs, all but one iconified, would have a very large data segment. But image transformations are all done on one single image, most of the memory occupied by xv is never touched. The same is true for an editor with many editor windows where only one window is being modified at a time. These programs have - if they are designed properly - a very high locality of reference and large parts of them can be kept swapped out without too severe performance impact. One could suspect that the 25% number from the age of the command line is no longer true for modern GUI programs editing multiple documents, but I know of no newer papers that try to verify these numbers.

So for a configuration with 16 MB RAM, no swap is needed for a minimal configuration and more than 48 MB of swap are probably useless. The exact amount of memory needed depends on the application mix on the machine (what did you expect?).

3.3 Where should I put my swap space?

- Mechanics are slow, electronics are fast. Modern hard disks have many heads. Switching between heads of the same track is fast, since it is purely electronic. Switching between tracks is slow, since it involves moving real world matter. So if you have a disk with many heads and one with less heads and both are identical in other parameters, the disk with many heads will be faster. Splitting swap and putting it on both disks will be even faster, though.
- Older disks have the same number of sectors on all tracks. With this disks it will be fastest to put your swap in the middle of the disks, assuming that your disk head will move from a random track towards the swap area.
- Newer disks use ZBR (zone bit recording). They have more sectors on the outer tracks. With a constant number of rpms, this yields a far greater performance on the outer tracks than on the inner ones. Put your swap on the fast tracks.
- Of course your disk head will not move randomly. If you have swap space in the middle of a disk between a constantly busy home partition and an almost unused archive partition, you would be better of if your swap were in the middle of the home partition for even shorter head movements.

You would be even better off, if you had your swap on another otherwise unused disk, though.

Summary: Put your swap on a fast disk with many heads that is not busy doing other things. If you have multiple disks: Split swap and scatter it over all your disks or even different controllers.

Even better: Buy more RAM.

3.4 Some facts about file systems and fragmentation

Disk space is administered by the operating system in units of blocks and fragments of blocks. In ext2, fragments and blocks have to be of the same size, so we can limit our discussion to blocks.

Files come in any size. They don't end on block boundaries. So with every file a part of the last block of every file is wasted. Assuming that file sizes are random, there is approximately a half block of waste for each file on your disk. Tanenbaum calls this "internal fragmentation" in his book "Operating Systems".

You can guess the number of files on your disk by the number of allocated inodes on a disk. On my disk

```
# df -i
Filesystem            Inodes   IUsed   IFree  %IUsed Mounted on
/dev/hda3               64256   12234   52022     19%  /
/dev/hda5               96000   43058   52942     45%  /var
```

there are about 12000 files on `/` and about 44000 files on `/var`. At a block size of 1 KB, about 6+22 = 28 MB of disk space are lost in the tail blocks of files. Had I chosen a block size of 4 KB, I had lost 4 times this space.

Data transfer is faster for large contiguous chunks of data, though. That's why ext2 tries to preallocate space in units of 8 contigous blocks for growing files. Unused preallocation is released when the file is closed, so no space is wasted.

Noncontiguous placement of blocks in a file is bad for performance, since files are often accessed in a sequential manner. It forces the operating system to split a disk access and the disk to move the head. This is called "external fragmentation" or simply "fragmentation" and is a common problem with DOS file systems.

ext2 has several strategies to avoid external fragmentation. Normally fragmentation is not a large problem in ext2, not even on heavily used partitions such as a USENET news spool. While there is a tool for defragmentation of ext2 file systems, nobody ever uses it and it is not up to date with the current release of ext2. Use it, but do so on your own risk.

The MS-DOS file system is well known for its pathological managment of disk space. In conjunction with the abysmal buffer cache used by MS-DOS the effects of file fragmentation on performance are very noticeable. DOS users are accustomed to defragging their disks every few weeks and some have even developed some ritualistic beliefs regarding defragmentation. None of these habits should be carried

TOC

over to Linux and ext2. Linux native file systems do not need defragmentation under normal use and includes any condition with at least 5% of free space on a disk.

The MS-DOS file system is also known to lose large amounts of disk space due to internal fragmentation. For partitions larger than 256 MB, DOS block sizes grow so large that they are no longer useful (This has been corrected to some extent with FAT32).

ext2 does not force you to choose large blocks for large file systems, except for very large file systems in the 0.5 TB range (that's terabytes with 1 TB equaling 1024 GB) and above, where small block sizes become inefficient. So unlike DOS there is no need to split up large disks into multiple partitions to keep block size down. Use the 1 KB default block size if possible. You may want to experiment with a block size of 2 KB for some partitions, but expect to meet some seldom exercised bugs: Most people use the default.

3.5 File lifetimes and backup cycles as partitioning criteria

With ext2, Partitioning decisions should be governed by backup considerations and to avoid external fragmentation from different file lifetimes.

Files have lifetimes. After a file has been created, it will remain some time on the system and then be removed. File lifetime varies greatly throughout the system and is partly dependent on the pathname of the file. For example, files in `/bin`, `/sbin`, `/usr/sbin`, `/usr/bin` and similar directories are likely to have a very long lifetime: many months and above. Files in `/home` are likely to have a medium lifetime: several weeks or so. File in `/var` are usually short lived: Almost no file in `/var/spool/news` will remain longer than a few days, files in `/var/spool/lpd` measure their lifetime in minutes or less.

For backup it is useful if the amount of daily backup is smaller than the capacity of a single backup medium. A daily backup can be a complete backup or an incremental backup.

You can decide to keep your partition sizes small enough that they fit completely onto one backup medium (choose daily full backups). In any case a partition should be small enough that its daily delta (all modified files) fits onto one backup medium (choose incremental backup and expect to change backup media for the weekly/monthly full dump - no unattended operation possible).

Your backup strategy depends on that decision.

When planning and buying disk space, remember to set aside a sufficient amount of money for backup! Unbackuped data is worthless! Data reproduction costs are much higher than backup costs for virtually everyone!

For performance it is useful to keep files of different lifetimes on different partitions. This way the short lived files on the news partition may be fragmented very heavily. This has no impact on the performance of the `/` or `/home` partition.

---

4. An example

4.1 A recommended model for ambitious beginners

A common model creates `/`, `/home` and `/var` partitions as discussed above. This is simple to install and maintain and differentiates well enough to avoid adverse effects from different lifetimes. It fits well into a backup model, too: Almost noone bothers to backup USENET news spools and only some files in `/var` are worth backing up (`/var/spool/mail` comes to mind). On the other hand, `/` changes infrequently and can be backuped upon demand (after configuration changes) and is small enough to fit on most modern backup media as a full backup (plan 250 to 500 MB depending on the amount of installed software). `/home` contains valuable user data and should be backuped daily. Some installations have very large `/home`s and must use incremental backups.

Some systems put `/tmp` onto a seperate partition as well, others symlink it to `/var/tmp` to achieve the same effect (note that this can affect single user mode, where `/var` will be unavailable and the system will have no `/tmp` until you create one or mount `/var` manually) or put it onto a RAM disk (Solaris does this for example). This keeps `/tmp` out of `/`, a good idea.

This model is convenient for upgrades or reinstallations as well: Save your configuration files (or the entire `/etc`) to some `/home` directory, scrap your `/`, reinstall and fetch the old configurations from the save directory on `/home`.

5. How I did it on my machine

There was this old ISA bus 386/40 sitting on my shelf that I abandoned two years ago because it no longer cut it. I was planning to turn it into a small X-less server for my household LAN.

Here is how I did it: I took that 386 and put 16 MB RAM into it. Added a cheap EIDE disk, the smallest I could get (800 MB) and an ethernet card. Added an old Hercules because I still had a monitor for it. Installed Linux on it and there I have my local NFS, SMB, HTTP, LPD/LPR and NNTP server as well as my mail router and POP3 server. With an additional ISDN card the machine became my TCP/IP router and firewall, too.

Most of the disk space on this machine went into the `/var` directories, `/var/spool/mail`, `/var/spool/news` and `/var/httpd/html`. I put `/var` on a separate partition and made this one large. There will be almost no users on this machine, so I created no home partition and mounted `/home` from some other workstation via NFS.

Linux without X plus several locally installed utilities will be fine with a 250 MB partition as `/`. The machine has 16 MB of RAM, but it will be running many servers. 16 MB swap should be in order, 32 MB should be plenty. We are not short on disk space, so the machine will get 32 MB. Out of sentimentality a MS-DOS partition of some 20 MB is kept on it. I decided to import `/home` from another machine, so the remaining 500+ MB will end up as `/var`. This is more than sufficient for a household USENET news feed.

We get

```
Device      Mounted on                        Size
/dev/hda1   /dos_c                             25 MB
/dev/hda2   - (Swapspace)                      32 MB
/dev/hda3   /                                 250 MB
/dev/hda4   - (Extended Container)            500 MB
/dev/hda5   /var                              500 MB

homeserver:/home /home                        1.6 GB
```

I am backing up this machine via the network using the tape in `homeserver`. Since everything on this machine has been installed from CD-ROM all I have to save are some configuration files from `/etc`, my customized locally installed *.tgz files from `/root/Source/Installed` and `/var/spool/mail` as well as `/var/httpd/html`. I copy these files into a dedicated directory `/home/backmeup` on `homeserver` every night, where the regular homeserver backup picks them up.

The Linux Partition-Rescue mini-HOWTO

Rolf Klausen, <rolfk@romsdal.vgs.no>

Version 1.1, 22 Oct 1997

This mini-HOWTO document describes how to rescue your Linux partition if M$-DOG has deleted it for you.

1.Introduction

- 1.1 Disclaimer
- 1.2 Legal stuff

2.Some preparations and needs

- 2.1 What you need
- 2.2 I assume that...

3.Let's do it! I want my partition back!

- 3.1 Using the Partition Magic program to recover your partition
- 3.2 Changing the partition type to Linux Native
- 3.3 The final touches

4.Contributors

- 4.1 Authors
- 4.2 Thanx go out to:
- 4.3 Thanx does NOT go out to:

---

1. Introduction

On my computer I have installed Linux, and I have used it for over an year now, and I think it is the best OS ever created. I also have M$-DOG and Windows NT 4.0 installed on my computer. The reason that I use M$-DOG is simply because, as Lars Wirzenius once wrote: "MS-DOS - you can't live with it, you can't live without it". I use it to play games and run demos and stuff with.

But enough about that. I have two DOS partitions, one primary partition and onelogical drive (I assume you know what a partition is - if not read the Partition mini-HOWTO. It is located at sunsite.unc.edu somewhere and probably many other places too.). They are about 500 MB each (I have one 3.2 GB disk and one 1.2 GB disk :). My Linux partition is 1200 MB. I have a free space of about 800 MB which I don't know what do do with yet, and I was wondering about what I should do if I needed more space for DOS (some large game or something), because I wasn't sure how many logical drives DOS could use / access. I don't know why, but for some reason I thought that DOS could use only one (in addition to the primary DOS partition). So I thought that I could check out if DOS could access more logical drives. I fired up fdisk (in DOS), and used the menu system to create a "logical DOS drive", and that worked fine. I created one with the size of 100 MB. Then I formatted it and it worked fine! Great! I thought. Then I knew that! But I didn't need that drive now, so I decided to delete it. I used DOS' fdisk again, chose "delete DOS primary partition or logical drive" (or something like that) on the menu, selected drive E to be deleted, typed the volume label, which was nothing (I just tapped ENTER) and fdisk told me that the disk was successfully deleted. Then I rebooted. On my computer I use a program called System Commander. It is a great program which displays a nice menu (a la OS/2's Boot Manager) with icons and many nice features. It detects and removes new/deleted OSes automatically etc. But when I rebooted, Linux wasn't on the menu anymore! I started DOS, typed e: and enter, and there it was still! Then I started fdisk. It chose to print the partition table, and it said that drive E was there, and where my Linux partition was before, now only a free space was left! SHOCK! HORROR! I started to sweat and shake! What should I do now! I couldn't just reinstall everything! It had to be a way to get it all back. And, yes, it was! Thank God (even though I don't believe in him), it was! I had lots of useful information and programs on that partition! Before I figured out how to rescue my partition, I wished Bill Gates was never even born! Have you experienced the same thing recently? Don't be afraid! The rescue is here! Just read on a little bit more...!
Oh, BTW: Due to the fact that I live in Norway, my english is probably not so good, but I hope you understand it anyway :-)

1.1 Disclaimer

This mini-HOWTO is Copyright (C) 1997 Rolf Klausen. All rights reserved. You may do (almost) whatever you want with it. I don't care. Just be sure to keep my name intact. I just hope it becomes useful for any person in the Linux community. But please read section 1.2 below.

IMPORTANT!!!!

I am not responsible of any damage on any computer as a result of anyone reading this HOWTO. If you do any damage, **it is YOUR fault, NOT MINE!** Be careful when partitioning disks, and don't make any mistakes, because it can be fatal! Backup all your important data and check that everything you do is correct! What is described here worked on my computer, but it may or may not work on your computer. Although it should work for everyone, I can't garantee anything. This is the last warning you get: **BACKUP IMPORTANT DATA!** Or, to put it short: Use at your own risk!

1.2 Legal stuff

Previous Next Table of Contents

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have any questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu via email.

---

2 Some preparations and needs

Previous Next Table of Contents

2.1 What you need

Previous Next Table of Contents

OK! So, MS-DOG has deleted your Linux partition? Here is what you need to bring it

back from the dead:

- A working DOS installation, preferrably on a harddrive, maybe a floppy will work, I don't know
- A dos program called Partition Magic. It is shareware, afaik, and you should be able to get it on the net, either one way or the other ;^)
- A Linux Boot floppy and a Linux Root Floppy. I used the ones from my Slackware distribution. You must have Linux fdisk on the root floppy.

2.2 I assume that...

Previous Next Table of Contents

- ... M$-DOG has deleted your Linux partition (accidently?)
- ... you haven't changed anything in your partition table since that happened.
- ... you satisfy the needs described in section 2.1

If this suites you, then the chances for succesfully recovering your partition are very large!

---

3. Let's do it! I want my partiton back!

Previous Next Table of Contents

OK! Get ready to rescue... grab a cup of coffee, and relax. Everything will be just fine :)
BTW - Can you imagine how I felt when MeSsy-DOS deleted my partiton? I have never been so angry in my entire life (almost). But I found a way to rescue the partition. Here it is...:

3.1 Using the Partition Magic program to recover your partition.

Previous Next Table of Contents

Get the partition magic program from the internet somewhere. Fire it up, and you will get a nice win95 look-alike user interface. It is very easy to use, and if you have used windoze 95, you should be able to use this program too. Press the OK button, and then a graphical presentation (is that the right word to use? I'm not so very good in english.) of your disk should appear on your disk. The different partitions and filesystems have different colors. I have a DOS primary partiton, and I assume you have the same. One of my logical drives is my Linux partition. Where that previously were, now only a grey bar of free space is left. To the right or left of it you should have a small logical drive which is you Linux Swap partition. Now right click on the free space and select "create", or choose from the menus. You should be able to figure out how it works. It is, as said earlier, a very easy to use interface. When the create

partition dialog box appears, select "Unformatted" where it says "FAT". You can choose from "FAT", "HPFS" and "Unformatted". If you choose FAT or HPFS, a FAT or HPFS filesystem will be created on that partition and your linux filesystem destroyed, so make sure that you have selected the right value! Now select OK, and the partition will be created. But the partition is now marked as a DOS FAT16 partition (in the partition table, every partition has a mark (a hexadecimal number) which shows what OS the partition belongs to.). What do you do then? Goes on to section 3.2 :)

3.2 Changing the partition type to Linux Native

Previous Next Table of Contents

Now you need the boot and root floppies, so I hope you have them. It is also possible to use loadlin. Just make sure you can start Linux and run fdisk under it. First thing to do is login ;). fire up fdisk by typing fdisk at the shell prompt. If you have more than one hd, then check that fdisk is using the right one. Type 'p' (only the letter p) and tap ENTER. You should now get a listing of your partition table. It should have one "DOS 16-BIT >= 32M" more than normal. Find out which one is your Linux partition (look at the sizes etc) and remember its number (i.e. mine is /dev/hda6, so I would have to remember the number 6 :). Now type 't' and ENTER to change a partitions type, or ID as it is called in Linux fdisk (i'm not sure which one is right; type or id, but I use the word type). Now type the number you should remember and press ENTER. Now you shall type the hex code of the OS which the partition belongs to. Type the number 83 and press ENTER. 83 means that the partition is a Linux Native partition. Linux Swap partitions have the type set to 82. You can type 'L' to see a list of codes when you are asked to type the hex code of the partition.

When you have done all this, and when you have done everything right, your partition shall be OK!! YES! It is true! It did work for me, and should work for you too! All you now have to do is the final step:

3.3 The final touches

Previous Next Table of Contents

There are some small details still remaining. First of all, reboot and fire up Linux the way you usually do it (LILO etc) and see if everything works. If it doesn't, then you have odne something wrong or my recovery mothod doesn't work for you :(But if it boots correctly, then congratulations! You have done everything correctly! Now fsck will check you filesystem once (it checked mine twice, I dunno why ;), and then everything shall work fine again! If not, then don't blame me. As said, it worked for me, but maybe not for you. If I saved your partition, then a e-mail would be very nice :)

---

4. Contributors

Previous Next Table of Contents

4.1 Authors

Previous Next Table of Contents

There are only one author, and that one is me:


```
Rolf Klausen
E-mail: rolfk@romsdal.vgs.no
WWW: http://www2.romsdal.vgs.no/~rolfk/
Snail mail:

Rolf Klausen
Neptunvn. 10
6400 Molde
NORWAY
```


4.2 Thanx go out to:

Previous Next Table of Contents

- Eivind Stormyr (eivind@ardata.no) for introducing me to Linux, the best OS ever created :) and learning me lots of things about computers.
- Bjorn Olav Stormyr (bjorn@ardata.no) for giving me large discounts on every part of my computer :) Whitout him I couldn't have afforded to buy the computer I own :)
- Every other member of the Linux community and everybody who supports Linux and writes documentation and programs for Linux and all the authors of the LDP and virtually any person involved in anything which has to do with Linux. Particularly Linus B. Thorvalds - he is **The King** !!!

- And of course:
- Tone B. Skjoelsvik - the most beautiful girl in the world :-)

4.3. Thanx does NOT go out to:

Bill Gates (bill.gates@microsoft.com (?)) for creating the worst "operating systems" in the world and for deleting my Linux partition (Yes, I blame him. If he wasn't born, then my partition wouldn't have been deleted and I wouldn't have had to recover it or even write this little HOWTO.

PATH HOWTO

Esa Turtiainen `etu@dna.fi`

v0.4, 15 November 1997

1. Introduction

2. Copyright

3. General

4. Init

5. Login

6. Shells

- 6.1 bash
- 6.2 tcsh

7. Changing user ID

- 7.1 su
- 7.2 sudo

8. Network servers

- 8.1 inetd
- 8.2 rsh
- 8.3 rlogin
- 8.4 telnet

- 8.5 ssh

9. XFree86

- 9.1 XDM
- 9.2 xterm -ls
- 9.3 Window manager menus and buttons

10. Delayed commands cron and at

- 10.1 cron
- 10.2 at

11. Some examples

- 11.1 magicfilter
- 11.2 Printing from X applications

12. Security concerns

13. How to debug problems?

14. Some strategies to get the same path for all the users

15. Acknowledgements

1. Introduction

This document describes common tricks and problems with Unix / Linux environment variables, especially with PATH variable. PATH is a list of directories where commands are looked for. The details apply for Debian Linux 1.3 distribution.

Note! This document is in beta release status. Please send comments and corrections.

2. Copyright

This documentation is free documentation; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This documentation is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this documentation; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

3. General

All the Unix processes contain an "environment". This is a list of variables that contain name and value, both just strings that can contain most characters. All Unix processes have a parent process - the process that created this process as child. Child processes inherit environment from parent process. They can make some modifications to the environment before passing it in turn to their child processes.

One important environment variable is PATH, a list of directories separated by colons (':'). These directories are searched through to find commands. If you try to invoke command 'foo', all the directories in PATH (in that order) are searched for an executable file 'foo' (one with x-bit on). If a file is found, it is executed.

In this howto, I use term 'command' to refer executable program that is meant to be called with short names, using the path mechanism.

In Linux, even the low level operating system calls to start processes (the exec family of calls) searches through directories in the PATH variable: you can use the path mechanism anywhere where you try to execute a command. If exec operating system call gets a file name that does not contain '/', it evaluates the PATH environment variable. Even if there is no variable PATH in the environment, at least directories /bin and /usr/bin are looked for suitable commands.

In sh you use export command to set environment, in csh you use setenv command. For example:

sh:

```
PATH=/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games:.
```

csh:

```
setenv PATH /usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games:.
```

C-programs can use setenv() library call to change environment. Perl has environment in an associative array %ENV, you can set PATH as $ENV{PATH}="/bin".

env command is the basic way of asking the current environment variables. It can be used to modify it as well.

More information of the basic environment mechanism can be found from manual pages 'environ', 'execl', 'setenv', info file 'env' and documentation of shells.

When Linux boots up, the first normal process that starts is the init process. It is a special process because it does not have parent. However, it is the ancestor of all the other processes. Init environment will remain as environment of all the processes if they do not touch it explicitly. Most processes do touch.

Init starts a group of processes. File /etc/inittab tells what processes the system starts. These processes

work in the environment that is directly inherited from init - typically they are processes like 'getty', program that writes 'login:' to console. If you start PPP connections here, you must remember that you are working in the init environment. The system initialization is often a script that is started here. In Debian 1.3 initialization script /etc/init.d/rc and it calls other initialization scripts in turn.

The system contains many running servers (daemons) that may or may not use the default environment. Most servers are started from the initialization scripts and thus they have the init environment.

When user logs in to the system, the environment is affected by the settings that are compiled into the programs, system wide initialization scripts and user initialization scripts. This is pretty complicated and the current situation is not completely satisfactory. It is totally different if user logs in from text console, XDM or from network.

---

4. Init

Init is a parent process for all the other processes of the system. Other processes inherit environment of the init process and the path is the init path in the rare case that no other path is set.

The 'init path' is fixed in the source of the init program and it is:

```
/usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin
```

Note that init path does not contain /usr/local/bin.

All the programs that are started from /etc/inittab work in init environment, especially system initialization scripts in /etc/init.d (Debian 1.3).

Everything that is started from system initialization scripts has init environment as default environment. For example, syslogd, kerneld, pppd (when started from startup), gpm and most importantly lpd and inetd have init environment and they do not change it.

A group of programs are started from startup scripts but the PATH environment variable is explicitly set in the startup script. Examples are: atd, sendmail, apache and squid.

There are other programs that are started from boot scripts but they change the path completely. One such example is cron.

5. Login

In text console there is a getty program waiting for user login. It writes 'login:' and other messages. It is working in init environment. When getty gets user to log in to the system, it invokes the 'login' program. This program sets the user environment and invokes the shell.

Login program sets path as defined in /usr/include/paths.h. This 'login path' is different for root users and other users.

for common users (_PATH_DEFPATH):

```
/usr/local/bin:/usr/bin:/bin:.
```

for root (_PATH_DEFPATH_ROOT):

```
/sbin:/bin:/usr/sbin:/usr/bin
```

Common user's path does not contain any sbin directories. However, it contains the current directory, '.', which is considered dangerous for the root user. Not even /usr/local/bin is available for the root user.

Login path is often overwritten by shell initialization. However, it is possible to use other programs in /etc/passwd as user shells. For example, I have used the following line to start PPP when I log in using special user name. In this case, the pppd has exactly login path.

```
etu-ppp:viYabVlxPwzDl:1000:1000:Esa Turtiainen, PPP:/:/usr/sbin/pppd
```

6. Shells

Often user processes are children processes of the shell mentioned in /etc/passwd for this user. Initialization files of shells often modify path.

In login, the name of the shell is preceded with '-', for example bash is called as '-bash'. This signals to the shell that it is a 'login' shell. In this case, the shell executes the 'login' initialization files. Otherwise some lighter initialization is performed. Additionally, the shell checks if it is interactive - are the commands coming from file or interactive tty. This modifies the shell initialization so that a non-interactive non-login shell is initialized very lightly - bash do not execute any initialization file in this case!

6.1 bash

As a normal login shell, bash 'sources' system-wide file /etc/profile, where the system environment and path can be set for bash users. However, it is not run when the system interprets the shell as non-interactive. The most important case is in rsh, where remote command is executed in the neighboring machine. The /etc/profile is not run and the path is inherited from rsh daemon.

bash receives command line arguments -login and -i that can be used to set the shell as a login shell or interactive shell respectively.

The user can overwrite values set in /etc/profile by creating a file ~/.bash_profile, ~/.bash_login or ~/.profile. Note that just the first one of these is executed thus differing of the logic of csh initialization. ~/.bash_login is not executed specially for login shells and if .bash_profile exists, it is not executed at all!

If bash is used with name sh instead of the name bash, it emulates original Bourne shell initialization: it sources just files /etc/profile and ~/.profile and just for login shells.

6.2 tcsh

As a login shell tcsh executes the following files in this order:

- /etc/csh.cshrc
- /etc/csh.login
- ~/.tcshrc
- ~/.cshrc (if .tcshrc is not found)
- ~/.history
- ~/.login
- ~/.cshdirs

tcsh can be compiled to execute login scripts before cshrc scripts. Beware!

TOC

Non-interactive shells execute just the *cshrc scripts. *login scripts can be used to set the path just o in the login.

7. Changing user ID

7.1 su

Command su sets a new user id to use. If no user id is given, root is used.

Normally su invokes a subshell with a different user id. With argument '-' (more recent synonyms -l or --login) su invokes shell like login shell. However, it does not use login program to do this but uses a yet another built-in path for login 'simulation' (term used in the source code). It is:

for normal users

```
/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:.
```

for root user

```
/sbin:/bin:/usr/sbin:/usr/bin:/usr/bin/X11:/usr/local/sbin:/usr/local/bin
```

su makes many quite subtle environment changes as well.

7.2 sudo

There is a group of commands that make use of super user commands safer. They allow better logging, user-based restrictions and usage of individual passwords. Most widely used is sudo.

```
$ sudo env
```

executes command env as super user (if it is configured to allow it).

sudo command has again a different approach to path handling. It modifies the search path so that the current directory is always the last one. However, it does not modify PATH environment variable. 'sudo env' and 'env' give the same value for PATH variable. Sudo adds just couple of environment variables like SUDO_USER.

8. Network servers

Most network servers should not invoke subprocesses of any kind. For security reasons, their path should be minimal.

An important exception is all the services that allow logging in to the system from network. This section describes what is the environment in these cases. If the command is executed in the remote machine with rsh it gets different path than if it is executed with ssh. Similarly, logging in with rlogin, Telnet or ssh is different.

8.1 inetd

Most network servers do not have process of their own waiting for requests all the time. This work is delegated to an Internet super server called inetd. Inetd listens for all the defined network ports and starts the appropriate server when there is an incoming request. This behaviour is defined in /etc/inetd.conf.

inetd is started from system startup scripts. It inherits just path of init process. It does not modify it and all the servers started from inetd has init path. An example of such a server is imapd, the server of IMAP post office protocol.

Other examples of inetd processes are telnetd, rlogind, talkd, ftp, popd, many http servers and so on.

Often usage of inetd is still complicated by using a separate tcpd program to start the real server. It is a program that makes additional security checks before starting the real application. It does not affect the path (not verified).

8.2 rsh

rsh daemon sets the path from _PATH_DEFPATH (/usr/include/paths.h) that is the same path that login program uses for normal users. Root will get the same path than the normal user.

Actually, rshd executes the command it gets with the command line:

```
shell -c command-line
```

and shell is not a login shell. It is desirable that all the shells mentioned in /etc/passwd support -c option to give on the command line.

TOC

8.3 rlogin

Rlogin is invokes login to make the real login procedure. If you login with rlogin, you get the same path than in login. Most other ways to log in to a Linux computer do not use login. Note the difference with rsh.

The login command actually used is

```
login -p -h host-name user-name
```

-p preserves the environment except the variables HOME, PATH, SHELL, TERM, MAIL and LOGNAME. -h tells the remote host name for logging.

8.4 telnet

Telnet is similar than rlogin. It uses the login program and the command line to invoke it in a similar way.

8.5 ssh

ssh has a path setting of it's own. It has a fixed path where it adds the directory where ssh is. Often this means that /usr/bin is in the path twice:

```
/usr/local/bin:/usr/bin:/bin:.:/usr/bin
```

The path does not contain /usr/X11/bin and shell invoked by ssh command is not a login shell. Thus

```
ssh remotehost xterm
```

never works and anything in /etc/profile or /etc/csh.cshrc can change this. You must always use explicit path /usr/bin/X11/xterm.

ssh searches environment variables of form VAR=VALUE from file /etc/environment. Unfortunately this causes some problems with XFree86.

---

9. XFree86

9.1 XDM

XDM is the most common way to log in to a graphical terminal. It a bit looks like login but it is internally totally different.

In directory /etc/X11/xdm there are configuration files that are executed on different login phases. Xstartup (and Xstartup_0 specially for screen 0) contains commands to be run after the user has logged in (commands are run as user root).

The path that is set for users is in /etc/X11/xdm/xdm-config. There are lines:

```
DisplayManager*userPath: /usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games
DisplayManager*systemPath:
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11
```

That will be a default path for normal and root users respectively. It is very important that /usr/bin/X11 is available for X users. If X user logs in to another machine to start and X client application, he should get /usr/bin/X11 to his path even he don't seem to come directly from X terminal.

After running Xstartup the XDM runs /etc/X11/Xsession that is run as the final user. Local configuration is meant to be done in /etc/environment that is sourced (included) from Xsession if available (Xsession is run with /bin/sh and thus /etc/environment must be a sh file). This clashes with ssh that supposes that /etc/environment is a file that contains just lines of form VAR=VALUE.

9.2 xterm -ls

By default the path for all the commands invoked from X window manager menus is the path inherited from XDM. To use something different it must be set explicitly. To start a terminal emulator with a path that is "normal" some special option must be used. In xterm the option -ls (login shell) must be used to get a login shell with path specified in shell login initialization files.

9.3 Window manager menus and buttons

Window manager inherits environment of XDM. All the programs started by the window manager inherit the environment of the window manager.

User shell environment does not affect the programs that are started from window manager buttons and menus. For example, if program is started from 'xterm -ls', it has the default environment of login shell but if it is started from menu, it has just environment of the window manager.

10. Delayed commands cron and at

10.1 cron

Cron is a command that executes commands periodically as specified in /etc/crontab and user-defined crontabs. In Debian 1.3 there is a standard mechanism to execute commands in /etc/cron.daily, /etc/cron.weekly and /etc/cron.monthly.

Cron is started from boot scripts but it seems to change it's PATH to a pretty strange one:

```
/usr/bin:/binn:/sbin:/bin:/usr/sbin:/usr/bin
```

THIS IS LIKELY A BUG IN CRON. This is the init path where there is /usr/bin:/bin written over the beginning without terminating 0! This bug does not exist in all the systems.

In crontab there can be PATH definition. In Debian 1.3 there is the following default line in the beginning of /etc/crontab:

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

Because of this, the PATH of crond program is never used in user programs. All the scripts in /etc/cron.* directories get this path by default. This path is used even if a program is executed as non-root.

10.2 at

at is a command that can be used to run a one-time program at specific time.

atd is run using init path. However, the user programs are always run in the user environment using sh command. Therefore the usual shell overwrites apply. Look the chapter on bash.

11. Some examples

11.1 magicfilter

magicfilter is a common tool to manipulate files for printer. It analyzes the type of the file to be printed and invokes a filter script to make appropriate pretty-printing. These scripts are invoked from lpd that is started from /etc/init.d/lpd that is started from init. Thus, the path is that of init. That does not contain /usr/bin/X11!

You might want to insert printing of PDF files to magicfilter. It is possible to do this by using /usr/bin/X11/xpdf. Now you must remember to insert full directory path to the file name because magicfilter would not find it otherwise. Most programs used in magicfilter do not need full path, because they are on /bin or /usr/bin.

11.2 Printing from X applications

You may use PRINTER environment variable to show what is the printer that you are using. However, you may notice that in some cases in X applications it is sometimes lost.

You must remember that if the X session is started from XDM, the window manager has never evaluated your shell login scripts. All the X applications that you have started from xterm have your PRINTER variable. However, if the same application is started from menu or window manager button, it does not contain your PRINTER variable.

In some cases this can be inherited to an even lower layer: for example a Netscape helper application can have or have not your PRINTER definition.

12. Security concerns

The path is sometimes a big security problem. It is a very common way to hack into a system using some mistakes in path settings. It is easy to make Trojan horse attacks if hacker gets root or other users to execute his versions of commands.

A common mistake in the past (?) was to keep '.' in the root's path. Malicious hacker makes program 'ls' in his home directory. If root makes

```
# cd ~hacker
# ls
```

he executes ls command of hacker's.

Indirectly, this same applies to all the programs that are executed as root. Any of the important daemon processes should never execute anything that some other user can write into. In some systems, /usr/local/bin is allowed to contain programs with less strict security screening - it is just removed from the path of the root user. However, if it is known that some daemon executes 'foo' using path '/usr/local/bin/:...', it may be possible to cheat daemon to execute '/usr/local/bin/foo' instead of '/bin/foo'. Likely anybody who can write to '/usr/local/bin' is able to break into the system.

It is very important to consider in what order the directories are in the path. If /usr/local/bin is before /bin, it is a security risk - if it is after, it is not possible to overwrite command /bin/foo with some localized modification in /usr/local/bin/foo.

In Linux it should be remembered that the path evaluation is done in the operating system call level. Everywhere where an executable file path is given you can give a short name that is searched at least from /bin and /usr/bin - likely from many other places as well.

13. How to debug problems?

The basic command to read environment is /usr/bin/env.

It is possible to use /proc directory to find out path of any program. First you must know the process number - use ps command to get that. For example, if xterm is process number 1088, you can find it's environment with command

```
# more /proc/1088/environ
```

This does not work with daemon processes like xdm. To access environment of system processes or other user processes, root access is required.

To debug Netscape, you can create a script /tmp/test:

```
$ cat > /tmp/test
#!/bin/sh
/usr/bin/env > /tmp/env
^d
$ chmod +x /tmp/test
```

Then set some helper application, for example RealAudio, audio/x-pn-realaudio to call program "/tmp/test". When you try to browse some RealAudio link (something from http://www.realaudio.com/showcase), Netscape calls the dummy program that stores environment to /tmp/env.

14. Some strategies to get the same path for all the users

The most important settings is possible to set in the global shell initialization files for login shells: /etc/csh.login for tcsh and /etc/profile for bash.

Exceptions that do not get the right path from these files are rsh commands, ssh commands, menu items from X window manager that do not explicitly start login shell, commands invoked from inittab, cron jobs, daemons jobs like magic filters started from lprd, WWW CGI scripts, and so on.

If the path is set in /etc/csh.cshrc, the path is right even when rsh or ssh execute command in remote machine with account using tcsh/csh. However, it is not possible to set path if account uses bash/sh.

It is possible to combine path setting to one file, for example to a file /etc/environment-common. There we write:

```
${EXPORT}PATH${EQ}/bin:/usr/bin:/sbin:/usr/sbin:/usr/bin/X11:/usr/local/bin:/usr/games:.
```

This can be used from /etc/csh.login (for tcsh and csh)

```
set EQ=" " set EXPORT="setenv " source /etc/environment-common
```

And from /etc/profile (for bash, doesn't work for ordinary sh)

```
EQ='=' EXPORT="export " . /etc/environment-common
```

And from /etc/environment (for XDM)

```
EQ="=" EXPORT="export " . /etc/environment-common
```

This strategy works mostly but ssh will complain of the lines in /etc/environment (and defined environment variables EQ and EXPORT). And still, rsh commands executed with bash won't get this path.

15. Acknowledgements

One reason to start writing this document was the big frustration of Ari Mujunen. Juha Takala gave some valuable comments.

Linux pre-installation checklist v2.2, 29 August 1998

TOC

from
Algologic Research & Solutions

How many times have you found yourself fumbling in the middle of a Linux installation process, because some vital information about the target hardware is not known ?

Have you ever crossed your fingers when you were asked by the installation procedure, a simple question, like: "to which COM port is the mouse connected ?

Let's face it, all of us have, at one time or another, felt we should have been better prepared before starting to install Linux. Here is a simple solution, brought to you by

Algologic Research & Solutions.

What does this checklist achieve ?
Who must use this list ?
Where do I find more about installing Linux ?
Japanese version of the checklist
The check list
Give us your feedback

What does this checklist (aim to) achieve ?

This is an aide-memoire which may make installation of Linux a pleasant experience for you. It fills an important role in the promotion of Linux.
It will help you to be prepared with all the answers which may be needed during the course of installation of Linux on your machine or on anybody else's machine. Sometimes, it is very inconvenient and even impossible to find out some details in the middle of an installation procedure. This becomes all the more difficult when you are installing Linux on somebody else's machine, and in somebody else's office.
This checklist, when filled up, will help you go through your Linux documentation and verify in advance whether certain hardware is supported by the distribution you are planning to install. It will also help you to be prepared with the packages you would need to install, and the configuration choices which need to be done.
The idea is to reduce the chances of frustrating surprises.
This checklist is only for installing Linux on PCs. You may use this checklist as a model for other classes of machines (e.g.SUN, Alpha, ...) also.

Where do I find more about Linux ?

This checklist does not give any details about hardware compatible with Linux, or other details about actually procuring or installing Linux. If you want details, just click here.
If you are the impatient type, you can go to the Linux Installation HOWTO directly, by clicking here..

Goto top

Who must use this checklist ?

- If you are planning to install Linux on somebody else's machine, or in a location which is not your own work place, you must absolutely get this checklist completed in advance. It is important to get this questionnaire answered by your host, so that you go adequately prepared for the installation.

TOC

- If you are planning to organise an "Install fest" (we Indians call it "Linux mela"), ask your participants to come wi checklist answered.
- Conversely, if you are going to an Install fest (aka Linux mela), to get Linux installed on your machine, do your home-work. Go with answers to this checklist.
- If you are going to ask someone to come over and install Linux for you, send him/her this questionnaire with as many responses as you know.
- If you are the D-I-Y type, you must start by answering all these questions, so that you can save time for enjoying your Linux !

Goto top

Japanese Version

This checklist is also available in Japanese. The Japanese translation is available as: a plain text file, or as a html file, at the URL given below:

- **Japanese Text Version:** Click here for a text-only version.
- **Japanese html Version:** Click here for a html version.

Click here to send a mail to Mrs. Chie Nakatani who did the translation into Japanese. You can also visit her home page, by clicking here.

Goto top

Linux Pre-installation checklist

```
-------------------------------------------------------------
                            LINUX
        PRE-INSTALLATION CHECK LIST CUM QUESTIONNAIRE
                   (Ver. 2.2 of 1998-08-29)
-------------------------------------------------------------
                         prepared by
     Algologic  Research  &  Solutions,   Secunderabad,
                            India

                       S. Parthasarathy
                   algolog@hd1.vsnl.net.in

Updated versions of this checklist will be posted at:
         http://members.tripod.com/~algolog/lnxchk.htm

Please  make  sure that you have the latest version  of  this
checklist.

-------------------------------------------------------------

                           Welcome

Welcome to the world of Linux. Your first step in  installing
Linux on your PC will begin by answering a few (?), simple(?)
questions about your hardware.

This  is a FREE service from Algologic. Algologic offers   no
warranties, and accepts no liabilities in  this exercise. You
are using this checklist at your own risk and responsibility.

This checklist will make it easy for you and us to go through
```

TOC

the installation, painlessly and without surprises. Please take your own time and fill up the following checklist. Wherever possible, and necessary, we have given some suggestions and hints, for your convenience.

If you don't know all the details, you should be able to find them in your computer's manuals. Or, ask your hardware vendor, or your local hardware "guru".

If you still have problems finding answers, don't worry -- you can just ignore any areas you don't know. Bring in your machine and ALL manuals you have. The installation may succeed with some trials and a lot of luck.

Remember, Linux, like many other software products, is a growing product. Some of your hardware may still not be supported in the version of Linux which is being installed.

Each question below can be identified by a letter followed by a number (e.g. G3 is a question on processor speed). If you need any assistance/clarifications about the questions asked in this form, please specify the chapter identifier (an alpha) and the question number (a numeral), and send an e mail to:

algolog@hd1.vsnl.net.in

Assumptions:

1. You are interested in installing Linux on a PC (386 upwards).

2. You have a CDROM drive on your PC. You have made sure that your proposed version of Linux supports this CDROM drive. Check latest version of Hardware HOWTO of the Linux Documentation Project (LDP).

3. You have at least one 3.5" floppy diskette drive on your system.

4. The 3.5" diskette drive is a BOOTABLE drive.

5. You will keep readily available, three, bad-sector-free, formatted (3.5"), blank diskettes, for use in the installation procedure.

6. If you wish to install Linux over a LAN, you have confirmed that your network interface card is supported by your version of Linux. Check latest version of Hardware HOWTO of LDP.

CHECKLIST

A. ABOUT YOU

TOC

1* Your Name _____

2* Your contact phone numbers:

3* Your email adress _____

4* Have you used Linux before ? What distribution?

5* Have you used any version of Unix before ? Which one ?
On what hardware platform ?

6* Is this an upgrade to an existing Linux distribution ?

7* What will be the end-use of this Linux installation ?

8* Do you wish to retain DOS/Win/any other OS along with
Linux, on your system ?

9* Any other detail you want to add:

_____

_____

G. GENERAL

1* CPU type [i386, i486 DX2...]:

2* Bus type [ISA, EISA, VESA, PCI, MCA ...]:

3* Processor speed (MHz) ?

4* What size is your RAM (minimum 4 MB needed) ?

5* Do you want X Window System (i.e. Graphic User
Interface for Linux) to be installed on your system ?

6* How many serial ports (COM ports) do you have on your PC ?

7* Are you planning to add dumb terminals to your PC ?
How many ?

8* Please mention below, any additional details you wish to
add:

_____

_____

_____

M. MOUSE

TOC

1* To which port is your mouse connected ? What IRQ ?

2* What type of mouse are you using
[Microsoft, Busmouse, PS/2 ..] ?

3* How many buttons does the mouse have ?

C. CDROM

Useful tips: If you are not sure if your CDROM is bootable or not, power on the machine and go to BIOS settings (do not let the PC bootup). Look for bootable drive options in the BIOS settings.

1* What is the make and type of CDROM drive
[e.g. IDE ATAPI, Sony, S-B Pro/Panasonic, SCSI, true IDE ..] ?

2* On which drive interface [1st IDE, 2nd IDE, 3rd IDE, 4th IDE, SCSI, proprietary interface] and which drive [drive 1 , drive 2, SCSI-ID 5] is the CDROM connected (e.g. drive 2 of 1st IDE, SCSI-ID 5 of built-in SCSI interface) ?

3* Is your CDROM a bootable drive ?

F. FLOPPY DRIVES

1* Do you have a 3 1/2" floppy drive?

2* Is the 3 1/2 " floppy drive a bootable drive ?

3* Do you have a 5 1/4" floppy drive?

4* Is the 5 1/4 " drive a bootable drive ?

T. TAPE DRIVES

1* What type of tape drive do you have (can be NONE)
[floppy interface type, SCSI, parallel port ..] ?

2* Do you have a secondary floppy controller for your tape drive?

D. DISK

TOC

Useful tips: At this point, you must be clear about how your Linux will be installed and on which disk etc. You must also be clear how your Linux will be booted (there are different possibilities).

Read the installation HOWTO before you answer the following questions.

1* What type of drive do you have [IDE, SCSI, RLL, MFM ....] ?

2* How many partitions are there on your first disk drive ? Name, type, size ? Collect these details also for other disk drives you may have on this machine.

3* On which drive are you planning to install Linux (e.g. drive 2 of first IDE, or SCSI-ID 5 of built-in SCSI interface)

4* Can we re-partition your disk ? (All existing data will be lost)

5* Which partitions can be reformatted to become Linux partitions ? Note: All information recorded in this partition will be erased.

6* How much of disk space can you give for Linux [50 MB minimum] ?

7* What is your second hard disk (if any)?

8* What is your third hard disk (if any)?

9* What is your fourth hard disk (if any)?

10* Do you have any SCSI device ? Which SCSI controller do you have (if any)?

11* Do you plan to boot from your hard disk or floppy?

12* On which hard disk partition do you want LILO boot loader to reside?

13* Do you wish to retain the original boot sector/master boot record that is currently on your hard disk?

14* On which disk partitions will your Linux root reside?

15* Do you access the IDE drive in LBA mode (check your BIOS settings) ?

V. Video Display Unit (Monitor and controller)

Useful tips: If you do not have good documentation on your VDU, try to look at the sticker (nameplate) on the back of

your VDU, you may find the answers to some of the questions given below.

1* What kind of monitor do you have [Monochrome, Colour] [EGA, VGA, CGA ..] ? What make ?

2* Size of screen [14", 15", 19"] ? Is it an interlaced monitor ?

3* What are the resolutions, and number of colours possible on this monitor [600x480, 800x600, 1024x768]?

4* What is your monitor's vertical refresh rate / vertical sync. frequency (could be a range)?

5* What is your monitor's horizontal refresh rate / horizontal sync. frquency (could be a range) ?

6* Monitor's bandwidth ?

7* Video RAM size ?

8* VDU Control card : make and model ?

9* Chip set used in VDU control card ?

N. NETWORKING

1* Are you using an internal modem | external modem ?

2* What make ?

3* On which COM port is the modem connected ? What IRQ ?

4* What is the speed of the modem [bps] ?

5* What serial interface chip does your computer use (e.g. 8450, 16550C) ?

5* Is it a "Windows only" modem [WIN-MODEM...] ?

6* Which ethernet card do you have [10base2 or 10baseT; NONE for none] ? What type ?

7* Will this machine be a part of Internet or any Intranet ?

8* Will this machine use a dialup connection / PPP connection to an Internet/Intranet server ?

9* Will this machine be used as an Internet or Intranet server ?

TOC

10* What will be the hostname of your computer (e.g. penguin) ?

11* What will be the domain name, if any, of your computer (e.g. family_name.city.country)

12* What will be the IP address, if any, of your computer (e.g. 123.1.2.3) ?

13* What is the IP address of the DNS server (can be none) ?

14* What is the IP address of the default router (can be none) ?

P. PRINTER

1* Have you connected any printer on the parallel port of your PC ?

2* What is the type, make of the printer ?

L. LAPTOP

1* Number and type of PCMCIA slots on your Laptop ?

2* PC cards you plan to use ?

3* Built-in devices on your laptop (e.g. trackball, pointer, touchpad) ?

4* COM ports used for the above built-in devices ?

X. MISCELLANEOUS

Useful tip: Try to make a consolidated hardcopy list of I/O addresses, IRQs, DMA-addresses used by the different devices in your machine. Take care to find out the configuration details of your VDU-driver card, Ethernet card, SCSI card etc.

1* Does your keyboard use a round connector (DIN-5) or PS/2 connector?

2* Do you plan to use keyboard mapping to any non-US keyboards [French, Greek, Cyrillic, ..] ?

3* Are you using any Plug & play devices ?

4* Are you using any sound card ? What card ? What I/O, IRQ, DMA does it use ?

TOC

5* How many ISA spare slots do you have ?

6* How many EISA spare slots do you have ?

7* How many VESA spare slots do you have ?

8* How many PCI spare slots do you have ?

9* How many other (describe) spare slots?

10* What kind of other removable media do you have (can be NONE)?

11* Do you have any other interesting hardware?

12* Please add any details you wish to highlight:

_____

_____

_____

Did you find this form clear ? Any suggestions for improvement ?

_____

_____

_____

That is all. Thanks for filling out the form.

Please crosscheck the above responses, and keep the checklist handy at the time of installation.

ACKNOWLEDGEMENTS

Algologic acknowledges with gratitude the useful suggestions and inputs from many members of the Linux community, for preparation of this checklist. This checklist is proof of the cooperation and goodwill which exists between Linux users. Algologic submits this checklist to the Linux community, as a humble tribute to the "spirit of Linux".

********Endchecklist

Goto top

TOC

Your feedback is IMPORTANT

Send us an email
Tell us if you found this webpage clear.
Tell us how we can improve this questionnaire and make it more useful.
Tell us how you located this page.
Visit our home page at: Algologic Research & Solutions.

Goto top

------ http://members.tripod.com/~algolog/lnxchk.htm -----
------- This page last modified on 1998-08-29 ----------

Thank you for your visit.
Algologic Research & Solutions.

How To Enable Process Accounting on Linux

TOC

Last updated: Fri Aug 8 09:25:58 HKT 1997

Preamble: This document is copylefted by Albert M.C. Tam (bertie@scn.org). Permission to use, copy, distribute this document for non-commerical purposes is hereby granted, provided that the author's / editor's name and this notice appear in all copies and/or supporting documents; that this document is not modified. This document is distributed in hope that it will be useful, but WITHOUT ANY WARRANTY, either expressed or implied. While every effort has been taken to ensure the accuracy of the information documented herein, the author / editor / maintainer assumes NO RESPONSIBILITY for errors, or for damages results for the use of the information documented herein.

This document describes how to enable system process accounting on a Linux host, and the usage of various process accounting commands. It is intended for users running kernel version greater than or equal to 1.3.73 (recently tested on RedHat 4.1 kernel 2.0.27). Kernels older than 1.3.73 may need a patch in order to use the process accounting feature.

Feel free to send any feedback or comments to bertie@scn.org if you find an error, or if any information is missing. I appreciate it.

What is Process Accounting?

Process accounting is the method of recording and summarizing commands executed on Linux. The modern Linux kernel is capable of keeping process accounting records for the commands being run, the user who executed the command, the CPU time, and much more.

Process accounting enables you to keep detailed accounting information for the system resources used, their allocation among users, and system monitoring.

Current Status of Process Accounting under Linux

Process accounting support has been integrated into the newer kernels (version >= 1.3.73). If you are running an older kernel, you may need some patch files. The patches are available from

ftp://iguana.hut.fi/pub/linux/Kernel/process_accounting

Requirements for Process Accounting on Linux

Kernel

Linux Kernel version greater than or equal to version 1.3.73, I recommended 2.x. The kernel source is

available from

http://sunsite.unc.edu/pub/Linux/kernel/v2.0

Process accounting software

Depending on the Linux distribution you have, you may, or may not have the process accounting software package installed on your system. If you don't have it, try downloading the package from

http://sunsite.unc.edu/pub/Linux/system/admin/quota-acct-modified.tgz

---

Process Accounting Setup on Linux

1. Compile and install process accounting softwares

The process accounting software package is available from

http://sunsite.unc.edu/pub/Linux/system/admin/quota-acct-modified.tgz

2. Modify your system init script and turn on process accounting at boot time

Here's an example:

```
# Turn process accounting on.
if [ -x /sbin/accton ]
then
        /sbin/accton /var/log/pacct
        echo "Process accounting turned on."
fi
```

3. Create accounting record file "pacct"

Your process accounting softwares will print out all commands executed to the file /var/log/pacct by default.

To create the accounting record file:

```
touch /var/log/pacct
```

This record file should be owned by root, has read-write permission for root, and read permission for anybody else:

```
chown root /var/log/pacct
chmod 0644 /var/log/pacct
```

4. Reboot

Now reboot your system for changes you made to take effect.

Miscellaneous Process Accounting Commands

TOC

ac

ac prints out statistics about users' connection time in hours, based on the logins and logouts in the current /var/log/wtmp file. ac is also capable of printing out time totals for each day (-d option), and for each user (-p option).

accton

accton is used to turn on or turn off process accounting. The file is normally executed at system bootup or shutdown via system init scripts.

last

last goes through the /var/log/wtmp file and prints out information about connection times of users.

sa

sa summarizes accounting information from previously executed commands, software I/O operation times, CPU times, as recorded in the accounting record file /var/account/pacct.

lastcomm

lastcomm prints out the information about all previously executed commands, recorded in /var/account/pacct.

ProxyARP Subnetting HOWTO

Copyright 1997 by Bob Edwards, email: <Robert.Edwards@anu.edu.au>

http://spigot.anu.edu.au/people/bob/home.html

See further information on copying conditions below.

Last update: August 1997. Click here to browse the author's latest version of this document. Corrections and suggestions welcome!

This HOWTO discusses using Proxy Address Resolution Protocol (ARP) with subnetting in order to make a small network of machines visible on another Internet Protocol (IP) subnet (I call it sub-subnetting). This makes all the machines on the local network (network 0 from now on) appear as if they are connected to the main network (network 1).

This is only relevent if all machines are connected by Ethernet or **ether** devices (ie. it won't work for SLIP/PPP/CSLIP etc.)

Table of contents

- Why use Proxy ARP with subnetting?
- How Proxy ARP with subnetting works
- Setting up Proxy ARP with subnetting
- Other alternatives to Proxy ARP with subnetting
- Other applications of Proxy ARP with subnetting
- Copying conditions

Acknowledgements

This document, and my Proxy ARP implementation could not have been made possible without the help of:

- Andrew Tridgell, who implemented the subnetting options for arp in Linux, and who personally assisted me in getting it working
- the Proxy-ARP mini-HOWTO, by Al Longyear
- the Multiple-Ethernet mini-HOWTO, by Don Becker
- the arp(8) source code and man page by Fred N. van Kempen and Bernd Eckenfels

Why use Proxy ARP with subnetting?

The applications for using Proxy ARP with subnetting are fairly specific.

In my case, I had a wireless Ethernet card that plugs into an 8-bit ISA slot. I wanted to use this card to provide connectivity for a number of machines at once. Being an ISA card, I could use it on a Linux machine, after I had written an appropriate device driver for it - this is the subject of another document. From here, it was only necessary to add a second Ethernet interface to the Linux machine and then use some mechanism to join the two networks together.

For the purposes of discussion, let network 0 be the local Ethernet connected to the Linux box via an NE-2000 clone Ethernet interface on eth0. Network 1 is the main network connected via the wireless Ethernet card on eth1. Machine A is the Linux box with both interfaces. Machine B is any TCP/IP machine on network 0 and machine C is likewise on network 1.

Normally, to provide the connectivity, I would have done one of the following:

- Used the IP-Bridge software (see the Bridge mini-HOWTO) to bridge the traffic between the two network interfaces. Unfortunately, the wireless Ethernet interface cannot be put into "Promiscuous" mode (ie. it can't see all packets on network 1). This is mainly due to the lower bandwidth of the wireless Ethernet (2MBit/sec) meaning that we don't want to carry any traffic not specifically destined to another wireless Ethernet machine - in our case machine A - or broadcasts. Also, bridging is rather CPU intensive!
- Alternatively, use subnets and an IP-router to pass packets between the two networks (see the IP-Subnetworking mini-HOWTO). This is a protocol specific solution, where the Linux kernel can handle the Internet Protocol (IP) packets, but other protocols (such as AppleTalk) need extra software to route. This also requires the allocation of a new IP subnet (network) number, which is not always an option.

In my case, getting a new subnet (network) number was not an option, so I wanted a solution that allowed all the machines on network 0 to appear as if they were on network 1. This is where Proxy ARP comes in. Other solutions are used to connect other (non-IP) protocols, such as netatalk to provide AppleTalk routing.

How Proxy ARP with subnetting works

The Proxy ARP is actually only used to get packets from network 1 to network 0. To get packets back the other way, the normal IP routing functionality is employed.

In my case, network 1 has an 8-bit subnet mask (255.255.255.0). I have chosen the subnet mask for network 0 to be 4-bit (255.255.255.240), allowing 14 IP nodes on network 0 (2 ^ 4 = 16, less two for the all zeros and all ones cases). Note that any size of subnet mask up to, but not including, the size of the mask of the other network is allowable here (eg. 2, 3, 4, 5, 6 or 7 bits in this case - for one bit, just use normal Proxy ARP!)

All the IP numbers for network 0 (16 in total) appear in network 1 as a subset. Note that it is very important, in this case, not to allow any machine connected directly to network 1 to have an IP number in this range! In my case, I have "reserved" the IP numbers of network 1 ending in 64 .. 79 for network 0. In this case, the IP numbers ending in 64 and 79 can't actually be used by nodes - 79 is the broadcast address for network 0.

Machine A is allocated two IP numbers, one within the network 0 range for it's real Ethernet interface (eth0) and the other within the network 1 range, but outside of the network 0 range, for the wireless Ethernet interface (eth1).

Say machine C (on network 1) wants to send a packet to machine B (on network 0). Because the IP number of machine B makes it look to machine C as though it is on the same physical network, machine C will use the Address Resolution Protocol (ARP) to send a broadcast message on network 1 requesting the machine with the IP number of machine B to respond with it's hardware (Ethernet or MAC layer) address. Machine B won't see this request, as it isn't actually on network 1, but machine A, on both networks, will see it.

The first bit of magic now happens as the Linux kernel arp code on machine A, with a properly configured Proxy ARP with subnetting entry, determines that the ARP request has come in on the network 1 interface (eth1) and that the IP number being ARP'd for is in the subnet range for network 0. Machine A then sends it's own hardware (Ethernet) address back to machine C as an ARP response packet.

Machine C then updates it's ARP cache with an entry for machine B, but with the hardware (Ethernet) address of machine A (in this case, the wireless Ethernet interface). Machine C can now send the packet for machine B to this hardware (Ethernet) address, and machine A receives it.

Machine A notices that the destination IP number in the packet is that of machine B, not itself. Machine A's Linux kernel IP routing code attempts to forward the packet to machine B by looking at it's routing tables to determine which interface contains the network number for machine B. However, the IP number for machine B is valid for both the network 0 interface (eth0), and for the network 1 interface (eth1).

At this point, something else clever happens. Because the subnet mask for the network 0 interface has more 1 bits (it

TOC

is more specific) than the subnet mask for the network 1 interface, the Linux kernel routing code will match the number for machine B to the network 0 interface, and not keep looking for the potential match with the network interface (the one the packet came in on).

Now machine A needs to find out the "real" hardware (Ethernet) address for machine B (assuming that it doesn't already have it in the ARP cache). Machine A uses an ARP request, but this time the Linux kernel arp code notes that the request isn't coming from the network 1 interface (eth1), and so doesn't respond with the Proxy address of eth1. Instead, it sends the ARP request on the network 0 interface (eth0), where machine B will see it and respond with it's own (real) hardware (Ethernet) address. Now machine A can send the packet (from machine C) onto machine B.

Machine B gets the packet from machine C (via machine A) and then wants to send back a response. This time, machine B notices that machine C in on a different subnet (machine B's subnet mask of 255.255.255.240 excludes all machines not in the network 0 IP address range). Machine B is setup with a "default" route to machine A's network 0 IP number and sends the packet to machine A. This time, machine A's Linux kernel routing code determines the destination IP number (of machine C) as being on network 1 and sends the packet onto machine C via Ethernet interface eth1.

Similar (less complicated) things occur for packets originating from and destined to machine A from other machines on either of the two networks.

Similarly, it should be obvious that if another machine (D) on network 0 ARP's for machine B, machine A will receive the ARP request on it's network 0 interface (eth0) and won't respond to the request as it is set up to only Proxy on it's network 1 interface (eth1).

Note also that all of machines B and C (and D) are not required to do anything unusual, IP-wise. In my case, there is a mixture of Suns, Macs and PC/Windoze 95 machines on network 0 all connecting through Linux machine A to the rest of the world.

Finally, note that once the hardware (Ethernet) addresses are discovered by each of machines A, B, C (and D), they are placed in the ARP cache and subsequent packet transfers occur without the ARP overhead. The ARP caches normally expire entries after 5 minutes of non-activity.

Setting up Proxy ARP with subnetting

I set up Proxy ARP with subnetting on a Linux kernel version 2.0.30 machine, but I am told that the code works right back to some kernel version in the 1.2.x era.

The first thing to note is that the ARP code is in two parts: the part inside the kernel that sends and receives ARP requests and responses and updates the ARP cache etc.; and other part is the arp(8) command that allows the super user to modify the ARP cache manually and anyone to examine it.

The first problem I had was that the arp(8) command that came with my Slackware 3.1 distribution was ancient (1994 era!!!) and didn't communicate with the kernel arp code correctly at all (mainly evidenced by the strange output that it gave for "arp -a").

The arp(8) command in "net-tools-1.33a" available from a variety of places, including (from the README file that came with it) ftp.linux.org.uk:/pub/linux/Networking/PROGRAMS/NetTools/ works properly and includes new man pages that explain stuff a lot better than the older arp(8) man page.

Armed with a decent arp(8) command, all the changes I made were in the /etc/rc.d/rc.inet1 script (for Slackware - probably different for other flavours). First of all, we need to change the broadcast address, network number and netmask of eth0:

```
NETMASK=255.255.255.240 # for a 4-bit host part
NETWORK=x.y.z.64        # our new network number (replace x.y.z with your net)
```

```
BROADCAST=x.y.z.79      # in my case
```

Then a line needs to be added to configure the second Ethernet port (after any module loading that might be required to load the driver code):

```
/sbin/ifconfig eth1 <name on net 1> broadcast <x.y.z.255> netmask 255.255.255.0
```

Then we add a route for the new interface:

/sbin/route add -net <x.y.z.0> netmask 255.255.255.0

And you will probably need to change the default gateway to the one for network 1.

At this point, it is appropriate to add the Proxy ARP entry:

```
/sbin/arp -i eth1 -Ds ${NETWORK} eth1 netmask ${NETMASK} pub
```

This tells ARP to add a static entry (the **s**) to the cache for network **${NETWORK}**. The **-D** tells ARP to use the same hardware address as interface **eth1** (the second **eth1**), thus saving us from having to look up the hardware address for eth1 and hardcoding it in. The **netmask** option tells ARP that we want to use subnetting (ie. Proxy for all (IP number) & ${NETMASK} == ${NETWORK} & ${NETMASK}). The **pub** option tells ARP to publish this ARP entry, ie. it is a Proxy entry, so respond on behalf of these IP numbers. The **-i eth1** option tells ARP to only respond to requests that come in on interface eth1.

Hopefully, at this point, when the machine is rebooted, all the machines on network 0 will appear to be on network 1. You can check that the Proxy ARP with subnetting entry has been correctly installed on machine A. On my machine (names changed to protect the innocent) it is:

```
#/sbin/arp -an
Address                  HWtype  HWaddress           Flags Mask            Iface
x.y.z.1                  ether   00:00:0C:13:6F:17   C     *               eth1
x.y.z.65                 ether   00:40:05:49:77:01   C     *               eth0
x.y.z.67                 ether   08:00:20:0B:79:47   C     *               eth0
x.y.z.5                  ether   00:00:3B:80:18:E5   C     *               eth1
x.y.z.64                 ether   00:40:96:20:CD:D2   CMP   255.255.255.240 eth1
```

Alternatively, you can examine the /proc/net/arp file with eg. cat(1).

The last line is the proxy entry for the subnet. The CMP flags indicate that it is a static (Manually entered) entry and that it is to be Published. The entry is only going to reply to ARP requests on eth1 where the requested IP number, once masked, matches the network number, also masked. Note that arp(8) has automatically determined the hardware address of eth1 and inserted this for the address to use (the -Ds option).

Likewise, it is probably prudent to check that the routing table has been set up correctly. Here is mine (again, the names are changed to protect the innocent):

```
#/bin/netstat -rn
Kernel routing table
Destination     Gateway         Genmask         Flags Metric Ref Use    Iface
x.y.z.64        0.0.0.0         255.255.255.240 U     0      0       71 eth0
x.y.z.0         0.0.0.0         255.255.255.0   U     0      0      389 eth1
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        7 lo
0.0.0.0         x.y.z.1         0.0.0.0         UG    1      0      573 eth1
```

Alternatively, you can examine the /proc/net/route file with eg. cat(1).

TOC

Note that the first entry is a proper subset of the second, but the routing table has ranked them in netmask order, eth0 entry will be checked before the eth1 entry.

Other alternatives to Proxy ARP with subnetting

There are several other alternatives to using Proxy ARP with subnetting in this situation, apart from the ones mentioned about (bridging and straight routing):

- IP-Masquerading (see the IP-Masquerade mini-HOWTO), in which network 0 is "hidden" behind machine A from the rest of the Internet. As machines on network 0 attempt to connect outside through machine A, it re-addresses the source address and port number of the packets and makes them look like they are coming from itself, rather than from the machine on the hidden network 0. This is an elegant solution, although it prevents any machine on network 1 from initiating a connection to any machine on network 0, as the machines on network 0 effectively don't exist outside of network 0. This effectively increases security of the machines on network 0, but is also means that servers on network 1 cannot check the identity of clients on network 0 using IP numbers (eg. NFS servers use IP hostnames for access to mountable file systems).
- Another option is IP in IP tunneling, which isn't supported on all platforms (such as Macs and Windoze machines) so I opted not to go this way.
- Use Proxy ARP without subnetting. This is certainly possible, it just means that a separate entry needs to be created for each machine on network 0, instead of a single entry for all machines (current and future) on network 0.
- Possibly IP Aliasing might also be useful here, but I haven't looked into this at all.

Other Applications of Proxy ARP with subnetting

There is only one other application that I know about that uses Proxy ARP with subnetting, also here at the Australian National University. It is the one that Andrew Tridgell originally wrote the subnetting extensions to Proxy ARP for. However, Andrew reliably informs me that there are, in fact, several other sites around the world using it as well (I don't have any details).

The other A.N.U. application involves a teaching lab set up to teach students how to configure machines to use TCP/IP, including setting up the gateway. The network used is a Class C network, and Andrew needed to "subnet" it for security, traffic control and the educational reason mentioned above. He did this using Proxy ARP, and then decided that a single entry in the ARP cache for the whole subnet would be faster and cleaner than one for each host on the subnet. Voila...Proxy ARP with subnetting!

Corrections and suggestions welcome!

Copying conditions

Copyright 1997 by Bob Edwards <Robert.Edwards@anu.edu.au>

Voice: (+61) 2 6249 4090

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions. All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below. In short, we wish to promote dissemination of this information through as many channels as possible.

However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans
redistribute the HOWTOs. If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, a
linux-howto@sunsite.unc.edu via email.

The Linux Public Web Browser mini-HOWTO

Donald B. Marti Jr., dmarti@best.com

v0.3, 5 January 1998

The basic idea here is to give web access to people who wander by, while limiting their ability to mess anything up.

1. Copyright and Disclaimer

2. Introduction

3. Before you begin

- 3.1 You need a graphical browser
- 3.2 You need to be able to add an account
- 3.3 You need `httpd` for a stand-alone web browsing station

4. Add the guest account

5. Create or edit the following files in `/home/guest`

- 5.1 File name: `.bash_login`
- 5.2 File name: `.Xclients`
- 5.3 File name: `.xsession`
- 5.4 File name: `.Xdefaults`

6. Make a `.netscape` directory for `guest`

7. Try it

8. Changing preferences

1. Copyright and Disclaimer

Copyright 1997 Donald B. Marti Jr. This document may be redistributed under the terms of the Linux Documentation Project license.

This document currently contains information for Netscape Navigator only, but I plan to add notes for other browsers too as I get the necessary information. If you try this with a different browser, please let me know.

2. Introduction

The basic idea here is to give web access to people who wander by, while limiting their ability to mess anything up.

This setup was originally intended for trade shows, but it might be applicable other places you want to have a web browser going without having to babysit a computer.

Following these instructions does **not** make your system bulletproof or idiot-proof.

3. Before you begin

3.1 You need a graphical browser

This document assumes that you already have a running graphical web browser, such as Netscape Navigator, on your system. You should have permission to use your graphical web browser. If you want to use Netscape Navigator in a commercial setting, you can buy a copy with appropriate license through Caldera.

3.2 You need to be able to add an account

If you don't have the right to be **root**, get the system administrator to add the ```guest`'' account and give you ownership of `guest`'s home directory. Skip to the ``Create or edit the following files'' step (Create or edit the following files in /home/guest) when he or she is done.

3.3 You need `httpd` for a stand-alone web browsing station

If you are setting up a web browsing station to run stand-alone, without a network connection, you should have `httpd` working and the web documents installed. To tell if this is the case, enter:

```
lynx -dump http://localhost/
```

You should get the text of the home page on your system.

4. Add the guest account

As **root**, run `adduser` to add a user named `guest`. Then enter

```
passwd guest
```

to set the password for the `guest` account. This should be something easy to remember, like ``guest''. You will be telling people this password. Don't make it the same as your own password.

Then make `guest`'s home directory owned by you. Enter

```
chown me.mygroup /home/guest
```

Replace ``me'' with your regular username and ``mygroup'' with your group name. (On Red Hat Linux, these will be the same, since every user has his or her own group.)

You should now exit and do the rest of the steps as yourself, not **root**.

5. Create or edit the following files in `/home/guest`

5.1 File name: `.bash_login`

```
exec startx
```

This means that when `guest` logs in, the login shell will start up the X Window System right away.

5.2 File name: `.Xclients`

```
netscape
```

This means that when X starts, `guest` just gets the web browser, no window manager. If you prefer another web browser, do something else.

The file `.Xclients` should be executable by `guest`. Enter

```
chmod 755 /home/guest/.Xclients
```

to make it so.

5.3 File name: `.xsession`

```
#!/bin/sh
netscape
```

If you use `xdm`(1) to log people in, this file should make guest get the web browser as if he or she had logged in normally. The file `.xsession` should be executable by `guest`. Enter

```
chmod 755 /home/guest/.xsession
```

to make it so.

5.4 File name: `.Xdefaults`

```
! Disable drag-to-select.
*hysteresis:                              3000
```

```
! Make visited and unvisited links the same color by default
*linkForeground:                        #0000EE
*vlinkForeground:                       #0000EE

Netscape.Navigator.geometry: =NETSCAPE_GEOMETRY

! Disable some of the keyboard commands.
*globalTranslations:

! Mouse bindings: make all mouse buttons do the same thing.
*drawingArea.translations:              #replace                        \
<Btn1Down>:                     ArmLink()                       \n\
<Btn2Down>:                     ArmLink()                       \n\
<Btn3Down>:                     ArmLink()                       \n\
~Shift<Btn1Up>:                 ActivateLink()                  \
                                DisarmLink()                    \n\
~Shift<Btn2Up>:                 ActivateLink()                  \
                                DisarmLink()                    \n\
~Shift<Btn3Up>:                 ActivateLink()                  \
                                DisarmLink()                    \n\
Shift<Btn1Up>:                  ActivateLink()                  \
                                DisarmLink()                    \n\
Shift<Btn2Up>:                  ActivateLink()                  \
                                DisarmLink()                    \n\
Shift<Btn3Up>:                  ActivateLink()                  \
                                DisarmLink()                    \n\
<Btn1Motion>:                   DisarmLinkIfMoved()             \n\
<Btn2Motion>:                   DisarmLinkIfMoved()             \n\
<Btn3Motion>:                   DisarmLinkIfMoved()             \n\
<Motion>:                       DescribeLink()                  \n\
```

This file disables blink tags, drag-to-select, and some of the keyboard commands. It also makes all mouse buttons do the same thing, hides the menu bar, and makes visited and unvisited links the same colour, so each visitor gets nice clean blue links, not ones that other people have been thumbing through and staining purple.

You should replace the `NETSCAPE_GEOMETRY` in this file with an X geometry that looks like this: `XxY+0-0`, where `X` is the width of your screen and `Y` is the height of your screen `+ 32`. This will position the Netscape menu bar off the top of the screen, so the user won't be distracted. For example, if your screen is 800x600, the geometry should be `800x632+0-0`.

6. Make a `.netscape` directory for `guest`

Enter

```
mkdir /home/guest/.netscape
chmod 777 /home/guest/.netscape
```

to create `guest`'s `.netscape` directory and make it world-writable.

7. Try it

Log out, then log in as `guest`.

8. Changing preferences

Since you won't be able to use the menu bar as `guest`, you should edit guest's preferences manually if you need to change them, or change your own preferences to what you want `guest`'s to be and copy the preferences file.

mini-HOWTO install qmail with MH

Christopher Richardson (rdn@tara.n.eunet.de)

v1.4, 5 March 1998

I am just documenting my installation experiences to offer some help to other users who wish to use the above combination for their email. v1.4 - I have finally got a new Linux box running so I decided to update this mini-howto

1. Introduction

2. My System Details

3. Qmail Installation

- 3.1 Maildir2smtp

4. MH Installation

- 4.1 mtstailor
- 4.2 mh_profile

5. Fetchmail

6. Exmh

7. Procmail

8. ISDN

9. Sources

10. Disclaimers

11. Postscript

1. Introduction

My thanks to all netizens who have helped me, especially Tony Nugent (tony@trishul.sci.gu.edu.au), David Summers (david@summersoft.fay.ar.us) and S.u.S.E (Linux distribution) who has made installing Linux so much easier, and the authors of the above excellent programs.

What is qmail and why should I use it? Here is the author´s (Dan Bernstein) blurb:

qmail is a secure, reliable, efficient, simple message transfer agent. It is meant as a replacement for the entire sendmail-binmail system on typical Internet-connected UNIX hosts.

Secure: Security isn't just a goal, but an absolute requirement. Mail delivery is critical for users; it cannot be turned off, so it must be completely secure. (This is why I started writing qmail: I was sick of the security holes in sendmail and other MTAs.)

Reliable: qmail's straight-paper-path philosophy guarantees that a message, once accepted into the system, will never be lost. qmail also supports maildir, a new, super-reliable user mailbox format. Maildirs, unlike mbox files and mh folders, won't be corrupted if the system crashes during delivery. Even better, not only can a user safely read his mail over NFS, but any number of NFS clients can deliver mail to him at the same time.

Efficient: On a Pentium under BSD/OS, qmail can easily sustain 200000 local messages per day---that's separate messages injected and delivered to mailboxes in a real test! Although remote deliveries are inherently limited by the slowness of DNS and SMTP, qmail overlaps 20 simultaneous deliveries by default, so it zooms quickly through mailing lists. (This is why I finished qmail: I had to get a big mailing list set up.)

Simple: qmail is vastly smaller than any other Internet MTA. Some reasons why: (1) Other MTAs have separate forwarding, aliasing, and mailing list mechanisms. qmail has one simple forwarding mechanism that lets users handle their own mailing lists. (2) Other MTAs offer a spectrum of delivery modes, from fast+unsafe to slow+queued. qmail-send is instantly triggered by new items in the queue, so the qmail system has just one delivery mode: fast+queued. (3) Other MTAs include, in effect, a specialized version of inetd that watches the load average. qmail's design inherently limits the machine load, so qmail-smtpd can safely run from your system's inetd.

Replacement for sendmail: qmail supports host and user masquerading, full host hiding, virtual domains, null clients, list-owner rewriting, relay control, double-bounce recording, arbitrary RFC 822 address lists, cross-host mailing list loop detection, per-recipient checkpointing, downed host backoffs, independent message retry schedules, etc. In short, it's up to speed on modern MTA features. qmail also includes a drop-in ``sendmail'' wrapper so that it will be used transparently by your current UAs.

2. My System Details

SuSE Linux Distribution 5.1 with 2.0.33 kernal.

PPP link to ISP

3. Qmail Installation

Follow the INSTALL instructions exactly.

Notes:

Please take the time to read the Fine documentation completely. The numerals refer to the installation steps in the above INSTALL doc.

- 2 - I had to set up the groups and users manually as per INSTALL.ids
- 7 - ./qmail-makectl did not work on my system. I added my domain name (mickey.n.eunet.de) manually in /var/qmail/control/me
- 23 - Make sure qmail-smtpd is spelt correctly in the inetd-conf file. (I spelt it incorrectly i.e. qmail-smptd, which took me two days to find:( ) smtp stream tcp nowait qmaild /var/qmail/bin/tcp-env tcp-env /var/qmail/bin/qmail-smtpd

3.1 Maildir2smtp

Dan Bernstein has provided a package for sending queued email to an ISP via dial-in. This package is available as serialmailxxx from his site.

Install this package as described in the man page (Thanks Rupert Mazzucco (maz@pap.univie.ac.at), it works out of the box!

```
maildir2smtp - blast a maildir across SMTP

maildir2smtp is designed to pass messages along a SLIP  or
PPP  link.  To set this up on the disconnected end, create
a new maildir in alias:

   # maildirmake ~alias/pppdir
   # chown -R alias ~alias/pppdir

Put

   :alias-ppp

into control/virtualdomains and

   ./pppdir/
```

```
into ~alias/.qmail-ppp-default.  Don't  forget  the  ext
slash in pppdir/.  Then, in the PPP startup script, do

   maildir2smtp ~alias/pppdir alias-ppp- $IP `hostname`

replacing  $IP with the remote IP address.
```

Notes:

- Please read the Fine manual page completely.
- Maildir2smtp requires the dotted IP address of your mail server. If you do not have this then ping YourMail.host.country which will return the IP.
- This command can be included in your login script to flush all queued mail after logging in to your ISP.

---

4. MH Installation

In addition to this, I also replaced /mh-6.8.4/mts/sendmail/smail.c with Dan Bernstein´s mh-qmail-smail.c

This is what my mh-6.8.4/conf/MH looks like:

```
bin     /usr/bin/mh
etc     /usr/lib/mh
#mail
#mandir /usr/man
#manuals        standard
chown   /bin/chown
#cp     cp
#ln     ln
#remove mv -f
cc      gcc
ccoptions -traditional -O2 -m486 -D_NFILE='getdtablesize()'
          -DSIGEMT=SIGUSR1
curses  -lncurses
#ldoptions      -s
#ldoptlibs
lex     flex
#oldload        off
#ranlib on
mts     sendmail
#mf     off
#bboards        off
#bbdelivery     off
#bbhome /usr/spool/bboards
pop     on
popdir  /usr/lib/mh
sharedlib       sys5
slflags -fPIC
slibdir /usr/lib
mailgroup       mail
signal  void
sprintf int
#editor prompter
```

TOC

```
#debug  off
#regtest        off
options ATHENA
options BIND
options DPOP
options DUMB
options FCNTL
options MHE
options MHRC
options MIME
options MORE='"/usr/bin/less"'
options OVERHEAD
options POP2
options POPSERVICE='"pop3"'
options RENAME
options RPATHS
options RPOP
options SOCKETS
options SVR4
options SYS5
options SYS5DIR
options TERMINFO
options UNISTD
options VSPRINTF
```

Notes:

- I have only compiled ``mts sendmail'' - read in comp.mail.mh somewhere that /smtp can cause problems. Dominic Mitchell (hdm@demon.net) wrote in comp.mail.mh (13 June 1997): ``Not quite. With this option MH still talks SMTP, just over a pipe and not over a network. You *really* need a line in your /.mh_profile which says: postproc: /usr/local/nmh/lib/spost Or whever it's kept on your system. This will pass the message directly to sendmail in the traditional manner. You're using qmail of course, so sendmail will be qmail's wrapper script, but that's just fine.'' Thanks Dominic.
- I have remmed out ``mail'' because I want to control it via mtstailor

4.1 mtstailor

As qmail delivers mail to the home directory (~/Mailbox). I added the following to my mtstailor

```
localname:      mickey
localdomain:    n.eunet.de
mmdfldir:
```

```
mmdflfil:       Mailbox
uucpldir:
uucplfil:
mmdelim1:       \001\001\001\001\n
mmdelim2:       \001\001\001\001\n
mmailid:        0
umincproc:
lockldir:
sendmail:       /usr/lib/sendmail
```

Notes:

- sendmail: /usr/lib/sendmail is a link to the qmail sendmail wrapper in /var/qmail/bin
- MH does not like the tilde notation (~/) use /home instead or leave blank which according to the docs defaults to $HOME.
- I recently installed MH and qmail on my office machine which is connect via ethernet. I added the following line to mtstailor: servers: mailserver.company.country

4.2 mh_profile

Here is my .mh_profile

```
Path: Mail
draft-folder: drafts
unseen-sequence: unseen
AliasFile: /home/rdn/.mh_aliases
send: -msgid
comp: -form /home/rdn/.mymh-components
MailDrop: /home/rdn/Mailbox
```

Notes:

- I put in the MailDrop line to be ``sure to be sure''.

---

5. Fetchmail

I decided to use fetchmail because I have a multiuser (my family :). Linux and fetchmail delivers mail to the smtp port where qmail takes over.

Installation was no problem, multidrop works with the following .fetchmailrc :

```
poll PersonalMail.Germany.EU.net
protocol pop3
username myname
password mypassword
# the next two lines do the trick for multidrop
localdomains mydomain.de
is * here
# T2 of the fetchmail FAQ states that qmail needs this
forcecr
```

6. Exmh

This is my mailer by choice. I love it.

There is one problem - most pre-compiled TCL/TK packages have the security option compiled in. The following script .xserverrc.secure which came with SuSE solves this.

```
#!/bin/sh

#
# move this file to ~/.xserverrc, if you don't want to allow everybody to
# get access to your X-Server
#
if [ -x /usr/bin/keygen ]; then
    if [ ! -x /usr/bin/hostname -a ! -x /bin/hostname \
         -a ! -x /usr/bsd/hostname ];
    then
        echo "startx: can't get my hostname - exiting"
        exit 1
    else
        host=`hostname`
    fi

    xauth add $host:0 . `/usr/bin/keygen`
    sleep 2
    xauth add $host/"unix":0 . `/usr/bin/keygen`
    exec X :0 -auth .Xauthority $*
else
    exec X :0 $*
fi
```

7. Procmail

The qmail FAQ gives this command:

In /.qmail add the line

```
| preline procmail
```

Version 3.11pre7 has changed the default mail box variable. It used to be in config.h. It is now in src/authenticate.c:

```
#define MAILSPOOLHOME "/Mailbox"        /* watch the leading / */
                                        /* delivers to $HOME/Mailbox */
```

I have culled from comp.mail.mh ans comp.mail.misc some hints on a .procmailrc file. Thanks to everyone!

```
# A SAMPLE .PROCMAILRC FILE FOR NOVICES
# Written by Catherine Hampton <ariel@best.com>
# Version 1.1
# Updated 1/25/98
#
# Released to the Public Domain.
#
#
# SET VARIABLES

# Internal Variables
# the following have be modified by rdn 19980303

# Everyone says that the SHELL environment is essential
SHELL=/bin/sh               #Shell used to run procmail.  Be sure this points to
                            #your system's copy of sh.  DO NOT substitute a
                            #different shell unless you really know UNIX

LINEBUF=4096                #Needed to keep Procmail from choking on long
                            #"recipes", or instructions on what to do with
                            #particular kinds of email.

PATH=$HOME/bin:/bin:/usr/bin:/usr/local/bin:/usr/bin/mh:/usr/lib/mh:
                            #Path for your programs -- this is probably best
                            #left alone.

VERBOSE=off                 #Change this to "on" when you try a new recipe
                            #so that Procmail will log literally every step
                            #it takes.  DO NOT LEAVE IT ON, though, because
                            #it creates huge logfiles.

# Default Program & file locations

MAILDIR=$HOME/Mail          #you'd better make sure this directory exists
```

```
DEFAULT=$HOME/Mailbox         #default incoming mailbox for shell2-5 users
                            #on Best Internet.  Substitute the correct
                            #setting for your system.


LOGFILE=$MAILDIR/procmail.log #Logs message disposition.  Recommended -- otherwise
                            #errors are emailed to you. :/

FORMAIL=/usr/bin/formail        #useful for autoreply recipes.  If you
                                #are not on Best Internet, modify this to
                                #your system's copy of formail.

SENDMAIL=/usr/sbin/sendmail    #useful for autoreply recipes.  If you are
                               #not on Best Internet, modify this to point
                               #to your system's copy of sendmail.
```

Procmail is an excellently documented program. Read the man pages for examples on how to set up your .procmailrc file.

---

8. ISDN

I am including this although this has nothing to do with qmail or mh. But without a PPP line to your ISP there is no email at all. I had quite a bit of bother to get my ISDN working. The SusE distribution includes a configuration for ISDN, but I wanted something simpler. The stuff here was adapted from Bernhard Hailer's scripts. (Vielen, vielen dank!)

The following rc.config loads the necessary modules during initialisation:

```
#!/bin/bash
# This is adapted Bernhard Hailer's old script

LOCAL_NUMBER="91311234"        # tel no. 091311234
REMOTE_NUMBER="0911123456"     # ISP tel no.
LOCAL_IP="192.168.0.99"        # I have dynamic IP so this will do
REMOTE_IP="195.112.123.11"     # your ISP's gateway
DEVICE="ippp0"

SYSPATH="/sbin"
ISDNCTRL="$SYSPATH/isdnctrl"

case "$1" in
start)
        # turn on isdn
        insmod /lib/modules/2.0.33/net/slhc.o
        insmod /lib/modules/2.0.33/misc/isdn.o
        sleep 1
        # load the hisax module
        insmod /lib/modules/2.0.33/misc/hisax.o
                                id=Tel0 type=5 protocol=2 irq=10 io=0x300
        echo "starting isdn4linux"
        # global
        $ISDNCTRL verbose 0

        $ISDNCTRL addif $DEVICE         # create new interface
        $ISDNCTRL addphone $DEVICE in $REMOTE_NUMBER
        $ISDNCTRL addphone $DEVICE out $REMOTE_NUMBER
        $ISDNCTRL eaz $DEVICE $LOCAL_NUMBER
        $ISDNCTRL l2_prot $DEVICE hdlc
        $ISDNCTRL l3_prot $DEVICE trans
        $ISDNCTRL encap $DEVICE syncppp
        $ISDNCTRL huptimeout $DEVICE 300
        $ISDNCTRL chargehup  $DEVICE off
        $ISDNCTRL secure $DEVICE on

        $SYSPATH/ifconfig $DEVICE $LOCAL_IP pointopoint $REMOTE_IP metric 1
        $SYSPATH/route add default $DEVICE
        $SYSPATH/ipppd /dev/ippp0 file /etc/ppp/options.ipppd &
        $SYSPATH/route del default

        ;;
stop)
```

TOC

```
        #turn off isdn
        rmmod hisax.o
        sleep 1
        rmmod isdn.o
        rmmod slhc.o
        echo "Shutting down isdn4linux"
        $ISDNCTRL delif ippp0
        ;;
*)
        echo "Usage: $0 (start|stop)"
        exit 1
        ;;
esac
```

I use the following script to dial out, it is called simply isdn on|off

```
#!/bin/bash
# This is based on an old script from Bernhard Hailer

IP_ADDRESS="195.112.123.11"

case "$1" in
on)


        echo "Calling ippp0"
        /sbin/isdnctrl dial ippp0
# the sleep is important as it gives the PPP time to settle down
        echo "Sleep for 8s for PPP handshake"
        sleep 8s
        /sbin/route add default ippp0
        echo "line open - checking...."

# check whether PPP negotiation was successful:
        set `ping -qc3 -i1 $IP_ADDRESS 2>/dev/null | grep transmitted`
        if [ $4 -gt 0 ];
        then
                echo "succeeded."
                echo "Starting fetchmail daemon"
                /usr/bin/fetchmail -d 600 -k -v -a -L /var/log/fetchmail
                echo "Flushing mail queue...."
                /usr/local/bin/serialmail/maildir2smtp
                            ~alias/pppdir alias-ppp- mail.server.ip.no `hostname`
        else
                echo "failed!"
                /sbin/isdnctrl hangup ippp0
        fi

;;

off)
                echo -n "Shutting down fetchmail daemon"
                /usr/bin/fetchmail --quit
```

TOC

```
                  /sbin/isdnctrl hangup ippp0
                  /sbin/route del default         # and delete route
                  echo "You're off line"
    ;;

    *)
            echo -e "\aUsage:"
            echo "isdn on"
            echo "isdn off"
    ;;

    esac
```

The next lot is the ipppd options file /etc/ppp/options.ipppd

```
    # Based on:
    # Klaus Franken, kfr@suse.de
    # Version: 27.08.97 (5.1)
    #
    # This file is copy by YaST from /etc/ppp/ioptions.YaST
    #   to options.<device>

    user "myuserid"

    # my system name (only for CHAP!)
    # name my_system_name

    # accept IP addresses from peer
    # use with dynamic IP
    ipcp-accept-local
    ipcp-accept-remote
    noipdefault

    # try to get IP address from interface
    # option specific to ipppd (as opposed to pppd)
    # use only with static IP
    #useifip

    # disable all header-compression
    -vj
    -vjccomp
    -ac
    -pc
    -bsdcomp

    # sometimes you need this:
    #noccp

    # max receive unit
    mru 1524
    # max transmit unit
    mtu 1500

    # If this machine is a server, force authentication by uncommenting one
    # of the following. However, if this machine is a client, doing this will
    # prevent a succesful connection! (message "peer refused to authenticate").
```

```
# So, only uncomment on a server.
# "+pap" / "+chap" NUR AKTIVIEREN, WENN DIES EIN SERVER IST!!!
#+pap
#+chap

# if you have problems with handshaking (no response for first
# lcp-package) try to decrease the retry-cycle. Default is 3 sec,
# try for example 2 sec:
# lcp-restart 2
```

---

9. Sources

Required Packages:

The net is so dynamic that it is pretty pointless to give sources. but for what it's worth:

- Find Qmail, setserial on http://www.qmail.org/
- Find MH on http://www.ics.uci.edu/~mh/
- Find glimpse on http://glimpse.cs.arizona.edu/
- Find Fetchmail on http://sagan.earthspace.net/~esr/fetchmail
- Find Exmh http://www.beedub.com/exmh
- Find Procmail ftp.informatik.rwth-aachen.de/pub/packages/promail

10. Disclaimers

The usual no guarantees, no money back, use at your own risk.

11. Postscript

Has anyone got mh working with Maildir? I have not tried - the principle of not fixing a running system. If you have got it working please mail me your instruction for inclusion in the next revision

Quota mini-HOWTO

Albert M.C. Tam bertie@scn.org

v0.0, 8 August 1997

Preamble: This document is copylefted by Albert M.C. Tam (bertie@scn.org). Permission to use, copy, distribute this document for non-commerical purposes is hereby granted, provided that the author's / editor's name and this notice appear in all copies and/or supporting documents; that this document is not modified. This document is distributed in hope that it will be useful, but WITHOUT ANY WARRANTY, either expressed or implied. While every effort has been taken to ensure the accuracy of the information documented herein, the author / editor / maintainer assumes NO RESPONSIBILITY for errors, or for damages results for the use of the information documented herein. *This document describes how to enable file system quota on a Linux host, assigning quota for users and groups, as well as the usage of miscellaneous quota commands. It is intended for users running kernel 2.x (recently tested on RedHat 4.1 running kernel 2.0.27). Users running older kernels may need to upgrade to a newer kernel version in order to take advantage of quota. Feel free to send feedbacks or comments to bertie@scn.org if you find an error, or if any information is missing. I appreciate it.*

1. What is Quota?

2. Current Status of Quota on Linux

3. Requirements for Using Quota on Linux

- 3.1 Kernel
- 3.2 Quota software

4. Quota Setup on Linux - Part I: The Configuration

- 4.1 Reconfigure your kernel
- 4.2 Compile and install the quota softwares
- 4.3 Modify your system init script to check quota and turn quota on at boot
- 4.4 Modify /etc/fstab
- 4.5 Create quota record "quota.user" and "quota.group"

- 4.6 Reboot

5. Quota Setup on Linux - Part II: Assigning Quota for Users and Groups

- 5.1 Assigning quota for a particular user
- 5.2 Assigning quota for a particular group
- 5.3 Assigning quota for a bunch of users with the same value
- 5.4 Soft Limit
- 5.5 Hard Limit
- 5.6 Grace Period

6. Miscellaneous Quota Commands

- 6.1 Quotacheck
- 6.2 Repquota
- 6.3 Quotaon and Quotaoff

---

1. What is Quota?

Quota allows you to specify limits on two aspects of disk storage: the number of inodes a user or a group of users may possess; and the number of disk blocks that may be allocated to a user or a group of users.

The idea behind quota is that users are forced to stay under their disk comsumption limit, taking away their ability to comsume unlimited disk space on a system. Quota is handled on a per user, per file system basis. If there is more than one file system which a user is expected to create files, then quota must be set for each file system seperately.

2. Current Status of Quota on Linux

Quota support has been integrated into kernel since version 1.3.8x I heard. Now it is part of the 2.0 release of the Linux kernel. If your system doesn't support quota, I really recommend an upgrade.

Currently, quota works for ext2 type file system only.

3. Requirements for Using Quota on Linux

3.1 Kernel

The 2.x kernel source is available from

```
http://sunsite.unc.edu/pub/Linux/kernel/v2.0
```

3.2 Quota software

Depending on the Linux distribution you have, you may, or may not have the quota softwares installed on your system. If you don't, then download the quota software source from

---

```
ftp://ftp.funet.fi/pub/Linux/PEOPLE/Linus/subsystems/quota/all.tar.gz.
```

---

4. Quota Setup on Linux - Part I: The Configuration

4.1 Reconfigure your kernel

Reconfigure your kernel and add quota support by typing y to:

```
Quota support (CONFIG_QUOTA) [n] y
```

4.2 Compile and install the quota softwares

The quota software source is available from

```
ftp://ftp.funet.fi/pub/Linux/PEOPLE/Linus/subsystems/quota/all.tar.gz
```

4.3 Modify your system init script to check quota and turn quota on at boot time

Here's an example:

```
# Check quota and then turn quota on.
if [ -x /usr/sbin/quotacheck ]
        then
               echo "Checking quotas. This may take some time."
               /usr/sbin/quotacheck -avug
               echo " Done."
        fi
         if [ -x /usr/sbin/quotaon ]
        then
                echo "Turning on quota."
                /usr/sbin/quotaon -avug
        fi
```

The golden rule is that always turn quota on after your file systems in /etc/fstab have been mounted, otherwise quota will fail to work. I recommend turning quota on at the end of your system init script, or,

if you like, right after the part where file systems are mounted in your system init script.

4.4 Modify /etc/fstab

Partitions that you have not yet enabled quota normally look something like:

```
/dev/hda1       /       ext2    defaults        1       1
/dev/hda2       /usr    ext2    defaults        1       1
```

To enable user quota support on a file system, add "usrquota" to the fourth field containing the word "defaults" (man fstab for details).

```
/dev/hda1       /       ext2    defaults        1       1
/dev/hda2       /usr    ext2    defaults,usrquota       1       1
```

Replace "usrquota" with "grpquota", should you need group quota support on a file system.

```
/dev/hda1       /       ext2    defaults        1       1
/dev/hda2       /usr    ext2    defaults,grpquota       1       1
```

Need both user quota and group quota support on a file system?

```
/dev/hda1       /       ext2    defaults        1       1
/dev/hda2       /usr    ext2    defaults,usrquota,grpquota       1   1
```

4.5 Create quota record "quota.user" and "quota.group"

Both quota record files, quota.user and quota.group, should be owned by root, and read-write permission for root and none for anybody else.

Login as root. Go to the root of the partition you wish to enable quota, then create quota.user and quota.group by doing:

```
touch /partition/quota.user
touch /partition/quota.group
chmod 600 /partition/quota.user
```

```
chmod 600 /partition/quota.group
```


4.6 Reboot

Now reboot system for the changes you have made to take effect.

Also note that subsequent partitions you wish to enable quota in the future only require step 4, 5, and 6.

5. Quota Setup on Linux - Part II: Assigning Quota for Users and Groups

This operation is performed with the edquota command (man edquota for details).

I would normally run quotacheck with the flags -avug to obtain the most updated filesystems usage prior to editing quota. This is just a personal habit, and not a required step however.

5.1 Assigning quota for a particular user

Here's an example. I have a user with the login id bob on my system. The command "edquota -u bob" takes me into vi (or editor specified in my $EDITOR environment variable) to edit quota for user bob on each partition that has quota enabled:

```
Quotas for user bob:
/dev/hda2: blocks in use: 2594, limits (soft = 5000, hard = 6500)
         inodes in use: 356, limits (soft = 1000, hard = 1500)
```

"blocks in use" is the total number of blocks (in kilobytes) a user has comsumed on a partition.

"inodes in use" is the total number of files a user has on a partition.

5.2 Assigning quota for a particular group

Now I have a group games on my system. "edquota -g games" takes me into the vi editor again to edit quota for the group games:

```
        Quotas for group games:
        /dev/hda4: blocks in use: 5799, limits (soft = 8000, hard = 10000)
                inodes in use: 1454, limits (soft = 3000, hard = 4000)
```

5.3 Assigning quota for a bunch of users with the same value

To rapidly set quotas for, say 100 users, on my system to the same value as my user bob, I would first edit bob's quota information by hand, then execute:

```
edquota -p bob `awk -F: '$3 > 499 {print $1}' /etc/passwd`
```

assuming that you are using csh, and that you assign your user UID's starting with 500.

In addition to edquota, there are 3 terms which you should familiarize yourself with: Soft Limit, Hard Limit, and Grace Period.

5.4 Soft Limit

Soft limit indicates the maximum amount of disk usage a quota user has on a partition. When combined with grace period, it acts as the border line, which a quota user is issued warnings about his impending quota violation when passed.

5.5 Hard Limit

Hard limit works only when grace period is set. It specifies the absolute limit on the disk usage, which a quota user can't go beyond his hard limit.

5.6 Grace Period

Executed with the command "edquota -t", grace period is a time limit before the soft limit is enforced for a file system with quota enabled. Time units of sec(onds), min(utes), hour(s), day(s), week(s), and month(s) can be used. This is what you'll see with the command "edquota -t":

```
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for users:
/dev/hda2: block grace period: 0 days, file grace period: 0 days
```

Change the 0 days part to any length of time you feel reasonable. I personally would choose 7 days (or 1 week).

6. Miscellaneous Quota Commands

6.1 Quotacheck

Quotacheck is used to scan a file system for disk usages, and updates the quota record file "quota.user" to the most recent state. I recommend running quotacheck at system bootup, or via cronjob periodically (say, every week?).

6.2 Repquota

Repquota produces a summarized quota information for a file system. Here is a sample output repquota gives:

```
# repquota -a
                              Block limits                File limits
        User            used    soft    hard  grace    used  soft  hard  grace
        root      --  175419       0       0          14679     0     0
        bin       --   18000       0       0            735     0     0
        uucp      --     729       0       0             23     0     0
        man       --      57       0       0             10     0     0
        user1     --   13046   15360   19200            806  1500  2250
        user2     --    2838    5120    6400            377  1000  1500
```

6.3 Quotaon and Quotaoff

Quotaon is used to turn on quota accouting; quotaoff to turn it off. Actually both files are similar. They are executed at system startup and shutdown.

The RCS MINI-HOWTO

Robert Kiesling

v1.4, 14 August 1997

This document covers basic installation and usage of RCS, the GNU Revision Control System, under Linux. It also covers the installation of the `diff(1)` and `diff3(1)` utilities, which are necessary for RCS to operate. This document may be reproduced freely, in whole or in part, provided that any usage of this document conforms to the general copyright notice of the HOWTO series of the Linux Documentation Project. See the file COPYRIGHT for details. Send all complaints, suggestions, errata, and any miscellany to kiesling@terracom.net, so I can keep this document as complete and up to date as possible.

1. Overview of RCS.

2. System requirements.

3. Compiling RCS from Source.

4. Creating and maintaining archives.

5. `ci(1)` and `co(1)`.

6. Revision histories.

7. Including RCS data in working files.

8. RCS and `emacs(1)` Version Control.

1. Overview of RCS.

RCS, the revision control system, is a suite of programs that tracks changes in text files and controls shared access to files in work group situations. It is generally used to maintain source code modules. It lends itself to tracking revisions of document files as well.

RCS was written by Walter F. Tichy and Paul Eggert. The latest version which has been ported to Linux is RCS Version 5.7. There is also a semi-official, threaded version available. Much of the information in this HOWTO is taken from the RCS man pages.

RCS includes the `rcs(1)` program, which controls RCS archive file attributes, `ci(1)` and `co(1)`, which check files in and out of RCS archives, `ident(1)`, which searches RCS archives by keyword identifiers, `rcsclean(1)`, a program to clean up files that are not being worked on or haven't changed, `rcsdiff(1)`, which runs `diff(1)` to compare the revisions, `rcsmerge(1),` which merges two RCS branches into a single working file, and `rlog(1),` which prints RCS log messages.

Files archived by RCS may be text of any format, or binary if the `diff` program used to generate change files handles 8-bit data. Files may optionally include identification strings to aid in tracking by `ident(1)`. RCS uses the utilities `diff(1)` and `diff3(3)` to generate the change files between revisions. A RCS archive consists of the initial revision of a file, which is version 1.1, and a series of change files, one for each revision. Each time a file is checked out of an archive with `co(1)`, edited, and checked back into the archive with `ci(1)`, the version number is increased, for example, to 1.2, 1.3, 1.4, and so on for successive revisions.

The archives themselves commonly reside in a `./RCS` subdirectory, although RCS has other options for archive storage.

For an overview of RCS, see the `rcsintro(1)` manual page.

2. System requirements.

RCS needs `diff(1)` and `diff3(3)` to generate the context diff files between revisions. The diff utilities suite needs to be installed on your system, and when you install RCS, the software will check for its presence.

Precompiled diffutils binaries are available at:

`ftp://sunsite.unc.edu/pub/Linux/utils/text/diffutils-2.6.bin.ELF.tar.gz`
and its mirror sites. If you need to compile `diff(1)`, et al., from source, it is located at:

`ftp://prep.ai.mit.edu/pub/gnu/diffutils-2.7.tar.gz`
and its mirror sites.

You will also need to have the ELF libraries installed on your system if you want to install pre-built binaries. See the ELF-HOWTO for further details.

3. Compiling RCS from Source.

Get the source distribution of RCS Version 5.7. It is available at

`ftp://sunsite.unc.edu/pub/Linux/devel/vc/rcs-5.7.src.tar.gz`

and its mirrors. After you have unpacked the archive into your source tree, you need to configure RCS for your system. This is done via the `configure` script in the source directory, which you need to execute first. This will generate a `Makefile` and the appropriate `conf.sh` for your system. You can then type

```
make install
```

which will build the binaries. At some point you may need to `su` to root so the binaries can be installed in the correct directories.

4. Creating and maintaining archives.

The program `rcs(1)` does the work or creating archives and modifying their attributes. A summary of `rcs(1)` options may be found in the `rcs(1)` manual page.

The easiest way to create an archive is first to `mkdir RCS` in the current directory, then initialize the archive with the

```
rcs -i name_of_work_file
```

command. This creates and archive with the name `./RCS/name_of_work_file,v` and requests a text message describing the archive, but it does not deposit any revisions in the archive. You can turn on or off strict archive locking with the commands

```
rcs -L name_of_work_file
```

and

```
rcs -U name_of_work_file
```

respectively. There are other options for controlling access to the archive, setting its format, and setting revision numbers, which are covered in the `rcs(1)` manual page.

5. ci(1) and co(1).

`ci(1)` and `co(1)` are the commands used to check files in and out of their RCS archives. The `ci(1)` command may also be used to a check a file both in and out of an archive. In their simplest forms, `ci(1)` and `co(1)` take only the name of the working file.

```
ci name_of_work_file
```

and

```
co name_of_work_file
```

The command form

```
ci -l name_of_work_file
```

checks in the file with locking enabled, and

```
co -l name_of_work_file
```

is performed automatically. That is, `ci -l` checks the file out again with locking enabled.

```
ci -u name_of_work_file
```

checks the file into the archive, and checks it out again with locking disabled. In all cases, the user is prompted for a log message.

`ci(1)` will also create a RCS archive if one does not exist already.

If you don't specify a revision, `ci(1)` increments the version number of the last revision locked in the archive, and appends the revised working file to it. If you specify a revision on an existing branch, it must be higher than the existing revision numbers. `ci(1)` will also create a new branch if you specify the revision of a branch which does not exist. See the `ci(1)` and `co(1)` man pages for details.

`ci(1)` and `co(1)` have various options for interactive and non-interactive use. Again, see the `ci(1)` and `co(1)` man pages for details.

6. Revision histories.

The `rlog(1)` program provides information about the archive file and the logs of each revision stored in it. A command like

```
rlog work_file_name
```

will print the version history of the file, each revision's creation date and `userids` of author and the person who locked the file. You can specify archive attributes and revision parameters to view.

7. Including RCS data in working files.

`co(1)` maintains a list of keywords of the RCS database which are expanded when the working file is checked out. The keyword `$Id$` in a document will expand to a string which contains the file name, revision number, the date checked out, the author, the revision status, and the locker, if any. Including the keyword `$Log$` will expand to the document's revision history log.

These and other keywords may be used as search criteria of the RCS archive. See the `ident(1)` man page for further details.

8. RCS and `emacs(1)` Version Control.

The Version Control facility of `emacs(1)` works as a front end to RCS. This information applies specifically to Version 19.34 of GNU Emacs, which is provided with the major Linux distributions. When editing a file with `emacs(1)` which is registered with RCS, the command `vc-toggle-read-only` (bound to `C-x C-q` by default) will check a file in to the emacs's Version Control, and then into RCS. Emacs will open a buffer where you can type a log message to be included in the RCS log. When you are finished typing a log entry, type `C-c C-c` to terminate your input and proceed with the check-in process.

If you have selected strict locking for the file with RCS, you must re-lock the file for editing by `emacs(1)`. You can check the file out for emacs's Version Control with the command `%` in buffer-menu mode.

For more information, see the GNU Emacs Manual and the Emacs info pages.

RPM+Slackware Mini-Howto

Dave Whitinger, dave@whitinger.net

v1.3, 13 April 1998

This document describes how to get RPM installed and working properly under Slackware. The information contained herein, however, is probably applicable to any Linux distribution.

1. Introduction
2. Obtaining the software
3. Installing the software
4. Bugs (Important!)
5. Acknowledgements
6. Copyright

1. Introduction

RPM is the "Red Hat Package Manager" and is the heart of the Red Hat Linux distribution. It's most basic functionality is to install and de-install packages.

This document is geared toward installing RPM on a slackware system using an Intel processor, but the information contained herein should be applicable to any distribution.

The latest version of this HOWTO is always available at http://www.threepoint.com/HOWTO/RPM+Slackware.html

For further reading, consult the RPM-HOWTO (available at your neighborhood LDP mirror). Also, consider buying a copy of the excellent book, Maximum RPM, by Ed Bailey of Red Hat Software, Inc.

2. Obtaining the software

The newest release of RPM is always available from:

```
ftp.rpm.org/pub/rpm/dist/latest
```

As of this writing, the latest version was

```
rpm-2.4.12-1.i386.tar.gz
```

Notice the .i386 section. This means that it is a binary package for the Intel architecture, ready to untar and run. Make sure that the file you download has the i386 in the filename, otherwise the following instructions will not work.

3. Installing the software

The easiest way to install RPM is to use Slackware's native package manager.

You must be root to install RPM.

```
installpkg /home/dave/rpm-2.4.12-1.i386.tar.gz
```

Of course, replace the /home/dave with the correct path for the filename.

(NOTE!) If that fails, simply untar the file with these commands:

```
cd / ; tar zxvpf /home/dave/rpm-2.4.12-1.i386.tar.gz
```

Next, you have to create a directory called "rpm" under the /var/lib tree.

mkdir /var/lib/rpm

Now type 'rpm --initdb' to initialize the rpm database.

If everything has gone correctly up to this point, you will have a rpm-capable system! Test it out by grabbing any rpm file and installing it with 'rpm -Uvh filename.rpm'

4. Bugs (Important!)

Be aware that on several versions of RPM, the tar file has been created using incorrect permissions. As soon as you install RPM, check your permissions of various directories (/bin, /usr, etc). If the permissions are of 700 (drwx------), then you have been infected by the bug.

To fix these permissions problems, run this shell script:

```
#!/bin/sh

chmod 755 /bin
chmod 755 /usr
chmod 755 /usr/bin
chmod 755 /usr/doc
chmod 755 /usr/lib
chmod 755 /usr/man
chmod 755 /usr/man/man8
chmod 755 /usr/share
chmod 755 /usr/share/locale
chmod 755 /usr/share/locale/de
chmod 755 /usr/share/locale/de/LC_MESSAGES
chmod 755 /usr/share/locale/pt-br
chmod 755 /usr/share/locale/pt-br/LC_MESSAGES
chmod 755 /usr/share/locale/sv
chmod 755 /usr/share/locale/sv/LC_MESSAGES
chmod 755 /usr/src
```

Feel free to E-Mail me if you have any questions about this.

5. Acknowledgements

I recognize Red Hat Software, Inc. and Patrick Volkerding for their fine Linux distributions.

Thanks to Milan Kopacka (mkop5230@ss1000.ms.mff.cuni.cz) for bringing to my attention a better way to install RPM (installpkg).

6. Copyright

This HOWTO is copyright 1998 by Dave Whitinger, and is a free document. You can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version.

Burning a RedHat CD mini-HOWTO

Morten Kjeldgaard, mok@imsb.au.dk and Peter von der Ahé, pahe+rhcd@daimi.au.dk

v1.18, 27 December 1998

This document describes how to make your own CDs from the Red Hat Linux distribution equivalent to the ones commercially available from Red Hat. The structure of the distribution is described, as well as the procedure needed to include updated RPMS into the distribution. Prerequisites are a good network connection, and a CD-writer.

1. Introduction

2. Anatomy of the Red Hat FTP site

- 2.1 The top level directory
- 2.2 The "RedHat" directory -- the core of the distribution

3. RPM packages

4. Obtaining your local copy of the distribution

5. Incorporating the updates

- 5.1 Correcting the file protection modes
- 5.2 Replacing the updated RPMS
- 5.3 Generating a new hdlist file

TOC

6. At last: burning the CD

- 6.1 Creating a bootable CD

7. Installing from the CD

8. This document...

- 8.1 Related documentation
- 8.2 Acknowledgements

9. DISCLAIMER

---

1. Introduction

There may be several reasons for making your own CD. Perhaps you're a cheapskate and want to save the $50 cost of the Red Hat distribution. Or, perhaps you want a distribution CD containing the latest distribution with all current updates. This is highly relevant, because after each major release of the Red Hat distribution, there have been loads of updates, several of which are security related. Just take a look at the updates/00README.errata file. A specific errata sheet also exists for each supported platform. See for example the Intel errata sheet

2. Anatomy of the Red Hat FTP site

In the spirit of the Linux community, Red Hat Software has made available their Linux distributions for several platforms on their FTP site. These are all available from the top distribution directory.

2.1 The top level directory

The toplevel directory for RedHat Linux release 5.1 (pub/redhat/redhat-5.1) contains distributions for the different platforms, and a directory containing updates and corrections to program packages that have been published since the release.

```
SRPMS/     alpha/     i386/      sparc/     updates/
```

In this document, we use the i386 distribution as an example. The procedure given in this document is likely to work on all platforms supported by Red Hat (Alpha, SPARC, ppc, etc.), but we have only tested it on the i386 platform (the authors would be most interested in additional information). The root of the i386 directory looks like this:

```
-rw-r--r--   8 ftpuser  ftpusers     19686 May 27  1997 COPYING
-rw-r--r--   1 ftpuser  ftpusers      3023 May  7 09:58 README
-rw-r--r--  10 ftpuser  ftpusers      2751 Sep 18  1997 RPM-PGP-KEY
drwxr-xr-x   5 ftpuser  ftpusers        96 Jul 15 08:34 RedHat/
drwxr-xr-x   5 ftpuser  ftpusers      8192 Jul 15 08:35 doc/
drwxr-xr-x   5 ftpuser  ftpusers      8192 Jul 15 08:35 dosutils/
drwxr-xr-x   5 ftpuser  ftpusers      8192 Jul 15 08:33 gnome/
drwxr-xr-x   2 ftpuser  ftpusers        96 Jun  7 02:47 images/
drwxr-xr-x   4 ftpuser  ftpusers        96 Jun  5 12:24 misc/
```

The `doc` directory contains an abundance of information. Most importantly, the RedHat installation manual can be found in HTML format in the directory doc/rhmanual/manual/. Next, there is a number of FAQs, and finally, the entire collection of HOWTOs and mini-HOWTOs.

The `images` directory contains boot floppy images that must be copied to a diskette. In the most recent distribution (5.1), there are two disk images available. The boot image is called `boot.img`, which is required when installation is performed directly from a CD-ROM. If installation from a local hard disk, NFS mounted disk or FTP is required, the supplementary disk image `supp.img` might be needed. See section Installing from the CD and references therein for details.

The `misc` directory contains source and executables of a number of programs needed for the installation.

2.2 The "RedHat" directory -- the core of the distribution

TOC

The most important part of the directory tree is rooted in the `RedHat` directory:

```
drwxr-xr-x   2 ftpuser  ftpusers     24576 Jul 15 08:35 RPMS/
drwxr-xr-x   2 ftpuser  ftpusers      8192 Jul 15 08:32 base/
-rw-rw-rw-  59 ftpuser  ftpusers         0 Aug 15 14:21 i386
drwxr-xr-x   4 ftpuser  ftpusers        96 Jun  5 12:24 instimage/
```

The `RPMS` directory contains the major part of the Red Hat distribution consisting of a set of RPM (Redhat Package Manager) files. An RPM package typically contains binary executables, along with relevant configuration files and documentation. See the section RPM packages for more information.

The `base` directory holds different book-keeping files needed during the installation process, e.g. the `comps` file, which defines the *components* (groups of packages) used during the "Choose packages to install" phase. Another important file in the `base` directory is the `hdlist` file containing most of the header fields from all the RPMs in the `RPMS` directory.

This means that all the interdependencies among RPM packages can be determined just by reading `hdlist` without having to read all the RPM packages which is quite convenient especially during FTP installs.

Another use of `hdlist` is mapping package names to file name, eg. `perl` to `perl-5.004-6.i386.rpm`. This means that if you want to incorporate updates from RedHat (see section Incorporating the updates) or add your own packages to the `RPMS` directory, you need to update `hdlist`. This is descriped later in Generating a new hdlist file.

The `instimage` directory contains a bare-bones live file system with a number of programs and shared libraries needed during the installation procedure.

---

3. RPM packages

The major part of the Red Hat distribution consists of a set of RPM (Redhat Package Manager) files. An RPM package typically contains binary executables, along with relevant configuration files and documentation. The `rpm` program is a powerful package manager, which can be used to install, query, verify, update, erase and build software packages in the RPM format. `Rpm` convieniently maintains a database of all the software packages it has installed, so information on the installed software is available at any time.

The binary RPM files in the distribution have been built on a system running the distribution itself. This is important, because most of the programs in the packages rely on shared libraries. From RedHat version 5.0, the new version 2 of the GNU standard C library (which is 64-bit clean) has been used. This version of the library is commonly referred to as `glibc` or in Linux: `libc 6`. All executables in the distribution have been linked against this library. If you attempt to install binary files from a different distribution, chances are that they will not work, unless you install the libc5 package for backwards compability.

The names of the RPM packages contain the suffix *.arch*.`rpm`, where *arch* is the architechture, having the value `i386` for Intel platform binaries. The packages you install must match the versions of the shared libraries available on the machine. The `rpm` program is usually quite good at ensuring that this is indeed the case, however, there are ways around this check, and you should be sure that you know what you are doing if you force installation of packages this way. However, using the RedHat installation boot disk, it is ensured that the correct set of RPM packages are installed on the machine.

If you discover an RPM package that was not installed on your system during the installation process, don't despair. At any time, you may (as root) install RPM packages, for example:

```
rpm --install  WindowMaker-0.18-1b.i386.rpm
```

You can even install directly from the Internet, if you know the URL of an RPM package:

```
rpm --install ftp://rufus.w3.org/redhat-contrib/noarch/mirror-2.9-2.noarch.rpm
```

Another version of the RPM packages contain the original sources used to build the binaries. These packages have the suffix `.src.rpm` and are situated in the `SRPMS` directory. These packages are not needed on the installation CD, and in fact, there is not even enough disk space on an 74 minute burnable CD to accomodate them. Of course, you can make a separate CD with the SRPMS.

4. Obtaining your local copy of the distribution

You need a copy of the distribution on a writable disk which is accessible from the computer having the CD writer (duh!). If you want to incorporate the latest updates, this directory should (also) be accessible from from a Linux machine, either from a local disk, an NFS mounted disk on a different computer, or a JAZ disk.

You could copy the distribution from a RedHat CD, or you could get it via FTP. If you choose to use FTP, the best way to get a correct copy of the distribution is to use the `mirror` package.

Mirror is a sophisticated perl script that compares the content of a directory on a remote site with a local directory. It will use FTP to fetch the files that are on the remote site but not the local site, and delete files on the local site that are not on the remote site. The mirror program is configured with a configuration file. The mirror package is available as an RPM from rufus.w3.org.

Make your local copy `mirror.redhat` of the mirror configuration file, and edit the relevant fields at the top of the file. After the default section, define these packages:

```
package=updates
       site=ftp.sunsite.auc.dk
       exclude_patt=(alpha/|sparc/)
       remote_dir=/disk1/ftp.redhat.com/pub/redhat/redhat-5.1/updates
       local_dir=/jaz/redhat-5.1/updates

package=dist
       site=ftp.sunsite.auc.dk
       exclude_patt=(alpha/|sparc/)
       remote_dir=/disk1/ftp.redhat.com/pub/redhat/redhat-5.1/i386
       local_dir=/jaz/redhat-5.1/i386
```

The following command will download a copy of the entire RedHat tree on your local disk. **Think** before you do this, you are about to transfer approximately 350Mb of data.

```
mirror -pdist mirror.redhat
```

This will mirror the Red Hat FTP site on your local disk. The content of a Red Hat distribution does not change between releases, so you only need to download this package *ONCE*. All changes to the distribution are in the `updates` directory. Thus, if you want to keep an up-to-date mirror of the Red Hat distribution, you only need to keep the `updates` directory current. This is done using the command

```
mirror -pupdates mirror.redhat
```

You can run this regularly, say, once a week, through a cron script. The RedHat distribution is available on a

great number of FTP servers around the world, which are updated daily from the master site at ftp://ftp.redhat.com/pub. You should choose an FTP site close to you, see the RedHat FAQ

5. Incorporating the updates

To incorporate the updates, you need write access to the distribution directory from a Linux machine, with a working version of rpm installed. There are three steps involved:

1. Correct the file protection modes.
2. Replace updated RPMs.
3. Generate the hdlist file

If you maintain a mirror of the `updates` directory, you can at any time produce a CD including the current updates by repeating these steps.

5.1 Correcting the file protection modes

During the installation process, some programs are run directly off the CD. Unfortunately, the FTP program does not always preserve the protection modes of the files and directories that are copied. Therefore, it is necessary to make sure that execute permission is given to programs, shell scripts and shared libraries, before the directory is burned on the CD. This is done by running the `updatePerm` script on your local copy of the distribution:

```
#!/bin/bash

RHVERSION=5.1

LIST=/tmp/er3hd3w25
CDDIR=/jaz/redhat-${RHVERSION}

# Find all directories, and make sure they have +x permission
find $CDDIR -type d -exec chmod -c 755 {} \;

# Find all files that are executables, shell or perl scripts
find $CDDIR -type f | file -f - | grep -v RPM \
   | egrep -i 'executable|perl|bourne|shell' | cut -f1 -d: > $LIST

# Find shared libraries
find $CDDIR -name \*.so >> $LIST

# Make them executable
while read file
do
   if [ ! -x $file ] ; then
      chmod -c 755 $file
   fi
done < $LIST

/bin/rm $LIST
```

```
exit 0
```

TOC

5.2 Replacing the updated RPMS

The following script called `updateCD` copies all files from the update directory to the RPMS directory. The script uses some nifty rpm tricks to determine what packages in the updates directory are more recent. Older packages are moved to the `${OLD}` directory.

```
#! /bin/bash
# This script updates rpms in a RedHat distribution found in $RPMDIR.
# The old rpms will be placed in $OLDDIR.
# The new rpms should be located in $UPDDIR.
# The architechture is $ARCH.

RHVERSION=5.1
ARCH=i386

CDDIR=/jaz/redhat-${RHVERSION}
RPMDIR=${CDDIR}/${ARCH}/RedHat/RPMS
UPDDIR=${CDDIR}/updates/${ARCH}
OLDDIR=${CDDIR}/old

if [ ! -d $OLDDIR ] ; then
   echo making directory $OLDDIR
   mkdir $OLDDIR
fi

allow_null_glob_expansion=1

for rpm in ${UPDDIR}/*.rpm ; do
  NAME=`rpm --queryformat "%{NAME}" -qp $rpm`
  unset OLDNAME
  for oldrpm in ${RPMDIR}/${NAME}*.rpm ; do
    if [ `rpm --queryformat "%{NAME}" -qp $oldrpm` = "$NAME" ]; then
      OLDNAME=$oldrpm;
      break
    fi
  done
  if [ -z "$OLDNAME" ]; then
    echo $NAME is new
    cp -pv $rpm $RPMDIR
  else
    if [ `basename $rpm` != `basename $OLDNAME` ]; then
      mv $OLDNAME $OLDDIR
      cp -pv $rpm $RPMDIR
    fi
  fi
done
```

TOC

```
# Copy new boot image files to the right place...
for newfile in ${UPDDIR}/images/* ; do
  file=${CDDIR}$/${ARCH}/images/$(basename ${newfile})
  if [ $newfile -nt $file ] ; then
     cp -pv $newfile $file
  fi
done

exit 0
```

5.3 Generating a new hdlist file

When installing from the CD, the installation program on the CD relies on the file `RedHat/base/hdlist` describing what RPM packages are available on the CD. The `hdlist` file can be generated by the program `misc/src/install/genhdlist`. This program must be run with the root name of the distribution as the only argument. Here is the `updateHdlist` script which calls that program:

```
#!/bin/bash

RHVERSION=5.1
ARCH=i386

echo generating hdlist...
CDDIR=/jaz/redhat-${RHVERSION}
GENHDDIR=${CDDIR}/${ARCH}/misc/src/install

chmod u+x ${GENHDDIR}/genhdlist
chmod 644 ${CDDIR}/${ARCH}/RedHat/base/hdlist
${GENHDDIR}/genhdlist ${CDDIR}/${ARCH} || echo "*** GENHDLIST FAILED ***"

exit 0
```

NOTE: After having incorporated the updates in the main `RedHat/RPMS` directory, your copy of the distribution is no longer a mirror of the Red Hat distribution site. Actually, it is more up-to-date! Therefore, if you attempt to mirror the distribution, older versions of the RPM's that have been updated will be downloaded once more, and the updates deleted.

Important note for RedHat 5.2

As distributed with RedHat version 5.2 and earlier, `genhdlist` CRASHES if there are files in the `RedHat/RPMS` directory which are *not* RPM files! This causes problems, because in the 5.2 distribution, there are a couple of non-RPM files named `ls-lR` and `ls-lR.gz` in `RedHat/RPMS`. Therefore, you must remove all non-RPM files from the directory. Alternatively, you can apply the following patch to `misc/src/install/genhdlist.c` and do a make. The patch will cause `genhdlist` to ignore any non-RPM files.

TOC

```
*** genhdlist.c.orig    Fri Nov 27 12:08:13 1998
--- genhdlist.c Fri Nov 27 12:08:20 1998
***************
*** 12,23 ****
--- 12,26 ----

  #define FILENAME_TAG 1000000

+ /* Not used apparently...
+
  int tags[] =  { RPMTAG_NAME, RPMTAG_VERSION, RPMTAG_RELEASE, RPMTAG_SERIAL,
                RPMTAG_FILENAMES, RPMTAG_FILESIZES, RPMTAG_GROUP,
                RPMTAG_REQUIREFLAGS, RPMTAG_REQUIRENAME, RPMTAG_REQUIREVERSION,
                RPMTAG_DESCRIPTION, RPMTAG_SUMMARY, RPMTAG_PROVIDES,
                RPMTAG_SIZE, RPMTAG_OBSOLETES };
  int numTags = sizeof(tags) / sizeof(int);
+ */

  int main(int argc, char ** argv) {
      char buf[300];
***************
*** 26,34 ****
--- 29,39 ----
      struct dirent * ent;
      int fd, rc, isSource;
      Header h;
+     /* not used
      int count, type;
      int i;
      void * ptr;
+     */

      if (argc != 2) {
        fprintf(stderr, "usage: genhdlist <dir>\n");
***************
*** 74,79 ****
--- 79,85 ----

            rc = rpmReadPackageHeader(fd, , , NULL, NULL);

+           if (!rc) {
            headerRemoveEntry(h, RPMTAG_POSTIN);
            headerRemoveEntry(h, RPMTAG_POSTUN);
            headerRemoveEntry(h, RPMTAG_PREIN);
***************
*** 110,115 ****
--- 116,122 ----
            headerWrite(outfd, h, HEADER_MAGIC_YES);
            headerFree(h);
            close(fd);
+           }
```

```
        }

        errno = 0;
```

6. At last: burning the CD

As we assume that you have a working CD-writer on your system, and you know how to use it, we wont go into much detail about burning the CD. If you are burning your CD on a Linux system, you can install the excellent XCDroast package. In XCDRoast, switch on the Rock Ridge extensions, and the creation of a TRANS.TBL file.

Make sure the *top* directory of the CD contains at least the following files and directories:

```
COPYING      RPM-PGP-KEY README       RedHat/
```

The following directories might come in handy:

```
doc/         gnome/      misc/        dosutils/    images/
```

6.1 Creating a bootable CD

(This section, thanks to Dawn Endico dawn@math.wayne.edu). Since XCDroast doesn't support creation of bootable disks you'll need to use other tools, for example `mkisofs` and `cdrecord`. Get the latest RPMs for these packages from ftp://contrib.redhat.com. You'll need to create an image file which will be written to the CD. This file will be 500Mb or more so find a partition with enough free space and change the path for `redhat.img` in the following commands if necessary. You may need to be root to use `mount` and `cdrecord`.

Create disk image

Change directory to the place in your mirror that will be the root directory of the cd. For instance, redhat-5.2/i386.

```
mkisofs -v -r -T -J -V "Red Hat 5.2" -b images/boot.img -c misc/boot/boot.cat -o
/tmp/redhat.img .
```

Test the image

If you're paranoid you can test your new disk image by mounting it. If you forgot to fix the file permissions or set the rock ridge extensions then the error will be obvious here since the file names and directory structure will be wrong.

```
mount -t iso9660 -o ro,loop=/dev/loop0 /tmp/redhat.img /mnt/cdrom
```

When you're done, don't forget to unmount it.

```
umount /mnt/cdrom
```

Burn the disk

Be sure to set the correct speed for your device. This command is for a 4X CDR. You may have a 1X or 2X drive.

```
cdrecord -v speed=4 dev=0,0 /tmp/redhat.img
```

7. Installing from the CD

When installing from the new CD, you first need to create a bootable installation diskette. IMPORTANT: use a NEW, freshly MS-DOS formatted diskette! Using an old, worn-out, faulty diskette can result in strange problems during the installation!

On a Linux system, you can create the diskette using the `dd` command:

```
dd if=/mnt/cdrom/images/boot.img of=/dev/fd0 bs=1440k
```

On a system running DOS or Windows-9x, you need to use the `RAWRITE.EXE` program, which is found on the CD in the `dosutils` directory.

Shut down the machine you want to install (or do a system upgrade) on, insert the boot diskette and your freshly burned CD, and let the machine boot from the diskette. For more information on the installation process, se the documents and the Installation-HOWTO or the Bootdisk-HOWTO which are on the CD in the `doc/HOWTO` directory.

8. This document...

The SGML source of the most recent version of this document can be retrieved from http://imsb.au.dk/~mok/linux/doc/RedHat-CD.sgml.

8.1 Related documentation

Ed Schlunder <zilym@asu.edu> has written a utility called `fix-rhcd` to let you check your Red Hat Linux distribution mirror for matching file sizes, names, permissions, and symlinks against an "ls -lNR" listing from the offical Red Hat ftp site. Any permissions that are wrong are changed to match the ls listing. See the fix-rhcd homepage.

8.2 Acknowledgements

Thanks to the following people for valuable input:

- Lars Christensen <larsch@cs.auc.dk>
- Thomas Duffy <tbd@cs.brown.edu>
- Dawn Endico <dawn@math.wayne.edu>
- Seva <seva@null.cc.uic.edu>

9. DISCLAIMER

While the given information in this document is believed to be correct, the authors assume no responsibility whatsoever for any damage to hardware and/or software, or any loss of data resulting from the procedures outlined in this document.

Linux Remote-Boot mini-HOWTO: Configuring Remote-Boot Workstations with Linux, DOS, Windows 95/98 and Windows NT

Marc Vuilleumier Stückelberg, David Clerc

v3.19, February 1999

This document describes how to set up a very robust and secure server-based configuration for a cluster of PCs, allowing each client to choose at boot-time which operating system to run. The key of this configuration is a bootprom based program, which let the user choose at boot time one of several boot images. This configuration is applicable using InCom TCP/IP Bootprom (add-on for most network cards) or any PXE-compliant Boot ROM (ready-to-use in most recent PC with built-in network cards). The most up-to-date version of this document, with hypertext links to downloadable software and other related materials, can be found at the address `http://cuiwww.unige.ch/info/pc/remote-boot/howto.html`. *Linuxdoc-SGML*, *DVI* *and* *PostScript* *versions are available in the same directory. If you are interested in getting info on further developpments, send an E-mail to* `David.Clerc@cui.unige.ch`.

1. Disclaimer and Copyrights

2. What has changed...

- 2.1 ...since version 2.x ?
- 2.2 ...since version 3.0 ?

3. Introduction

- 3.1 Boot ROM and Hard-disk
- 3.2 The Network
- 3.3 How it Works

- 3.4 Related non-commercial documentations

4. The Configuration How-To

- 4.1 Server-side configuration
- 4.2 Client-side configuration
- 4.3 Setting Up the Boot Process
- 4.4 Setting Up Linux
- 4.5 Setting up DOS 6 and Windows 3.1
- 4.6 Setting up Windows 95
- 4.7 Setting up Windows NT
- 4.8 Troubleshooting (FAQ)

5. Remote-Boot Tools Reference Manual

- 5.1 BpBatch, MrBatch and MrZip
- 5.2 NoBreak.sys

6. Special TFTP Servers

- 6.1 Incom Enhanced TFTP Server
- 6.2 Linux Enhanced TFTP Server
- 6.3 The Security Gateway
- 6.4 The Broadcast TFTP Server

---

1. Disclaimer and Copyrights

This document and the related software are provided as is to the Linux and Internet community, with no form of warranty. Please note that **some operations related in this document may destroy the content of your hard-disk**. We assume no liability for any use, correct or not, of this document and of the related software.

You are free to do anything you want with the remote-boot tools as long as you do not make money by selling them or by distributing them with a commercial product. If you want to commercialize a product derived from these tools, please contact the authors first to make a commercial agreement. These remote-boot tools will remain available for free forever, but we may authorize derived commercial tools.

These provisions shall be interpreted under and in accordance with the laws of Switzerland, canton of Geneva. All disputes, defenses, controversies or claims arrising in conncetion with this document and the related software, shall be subject to the exclusive juridiction of the courts of the canton of Geneva, Switzerland.

If you like this program, you can send us a Postcard and/or make a gift to the International Committee of the Red Cross (ICRC) or to the UNICEF.

2. What has changed...

2.1 ...since version 2.x ?

To say it frankly, almost everything. The underlying concepts are the same, but the software part has been completly redesigned to overcome the limitations of previous versions and to make it easier to use. An highlight of the new features :

- All functions (bpmenu, bpclean, bpunzip) are encompassed in a single program.
- The program can run not only from the boot rom, but also under DOS, Windows 95 and Linux.
- The program can now restore images of FAT16, FAT32 and EXT2FS partitions. If someone want to write NTFS support, let me know... For now, NT users still have to stick to FAT16.
- The program can not only restore disk images but also add and patch individual files in order to customize the client behaviour.
- Disk images are not any more bound to 87 MB. They are now file-system independant archives.
- We provide a mean for automatically downloading a disk image to an arbitrary big number of clients at the same time (broadcast).
- You can now write your own secure boot script, that will determine the behaviour of the machine before the real boot.
- You can now boot any Linux kernel, without applying any patch. Its is also possible to provide a command line and a ramdisk image.
- You can authenticate users at boot time using a Unix, NT or Radius server and deny them any access to the machine.
- Full national language support is included.
- And many, many other new features...

Is there a program for converting old archives to the new format ?

No, because the internal format is radically different. But you can easily do the conversion by yourself:

1. Boot an old image (unzip it to your disk)
2. Remove calls to the old `unzipreg` utility and replace them by the adequate `patch` commands (it is very easy, see the detailed instructions below)
3. Run the new `mrzip` program to create a new-style disk image

2.2 ...since version 3.0 ?

Version 3.0 was the beta-release. A dozen of sites around the world have tested it during a month and given much of their time to help us finding bugs and to suggest enhancements. Thanks to all of them for their patience, and in particular to Maciek Uhlig, Dick Velders and Jeff Teeters.

A few minor features have been added since 3.01, such as support for diskless Linux boot (by disabling the cache).

Version 3.10 introduced compatibility with Intel's *Wired for Management 1.1a* NetPC standard. The tools now work with any PXE-compliant boot ROM (as are most on-board boot ROMs) available today. Thanks to InCom GmbH for giving us the PXE bootprom that permitted this developpment. We also succesfully tested the tools with the PXE Boot ROM that I found incidentally in my Dell computer with onboard network card (called LanDesk Service Agent).

Version 3.11 to 3.12 added UNIX server-side tools (a PXE Proxy DHCP server for Solaris and Linux, and an enhanced TFTP server for Linux), as well as detailled informations on server-side setup and the PXE booting process.

Version 3.13 added Advanced Power Management support (PowerOff command).

Version 3.14 added minor enhancements and some corrections. We fixed a problem with the terminal under RedHat 5.1, and another problem in the syntax of the "if" command. We added some features suggested by the Laboratori de Càlcul de la Facultat d'Informàtica de Barcelona (LCFIB) :

- A new APM variable let you know if your system support the Advanced Power Management (i.e it supports the poweroff command).
- A "beep" command.
- A new parameter to DrawWindow, to include a title at the window creation. You can now do DrawWindow 200 200 400 200 "Title".

Version 3.15 added full VESA support. BpBatch now support several video modes, to accomodate old computers not being able to display 800x600 graphics. A new parameter has been added to InitGraph to specify the video mode, and a list of detected video mode can be retrieved from the new VESA-Modes variable.

Version 3.16 fixes the following bugs:

- "Malloc failed" during the Fullunzip process of a multiple fragments image. Many thanks to Christian Meyer for his collaboration.
- A bug which prevented the linux version of MrBatch to properly fullunzip images. This bug was located in the low-level functions of MrBatch, so it may fix other problems encountered in the *linux* version of MrBatch. Many thanks to Jeff Teeters for his collaboration.
- An error in the codepage translation tables. This bug was found by the Laboratori de Càlcul de la Facultat d'Informàtica de Barcelona (LCFIB). You can find the bug report in the BpBatch forum.

Version 3.17 adds some minor features and fixes bugs:

- Fullunzip was turning Extended Memory off

- Booting on the RedHat boot disk now works
- When extracting images with a large number of directories, the resulting FAT file system was corrupted.
- We added retries to text TFTP transfers. BpBatch will now retry three times before saying "Could not transfer the file".
- Timestamps are now correctly updated in FAT. (thank to Francis Chan)

Version 3.18 fixes a bug with the IncrUnzip function. Thanks to Gary Pike for its collaboration.

Version 3.19 fixed a bug in the error handling of the `delete` command on ext2fs, as well as the inappropriate handling of names starting with A: under Linux. The following new features were also added:

- A new `if valid disk:partition` syntax can be used to check if a partition has been formatted
- FAT32 disk images are now fully functional (they now boot properly)
- Linux EXT2 partitions bigger than 2 GB are now supported
- Linux Swap partitions bigger than 128 MB are now supported (this feature needs a recent kernel, at least 2.1.x)
- FullUnzip is now also possible without a cache partition, by setting `CacheNever` to `"ON"`. This might be usefull for a unique installation, but is not recommended in general is it results in a high network load.

Thanks to Ruben Schattevoy for its help and contributions to this release.

---

3. Introduction

The configuration described here was developped since Summer 1996 at the CUI, University of Geneva. The Computer Science Department uses several servers and a number of PCs, which fall into two classes:

- computers devoted to students
- computers devoted to research and teaching assistants

We developped the current configuration with the following aims:

- Every computer should be able to run under Linux, DOS, Windows 3.1, Windows 95 or Windows NT. One should be able to choose the desired operating system for each session.
- All softwares, including operating systems, should be take from the server, in order to facilitate the installations and upgrades.
- Clients computers should be able to run without any write-access on the server (for security reasons), except for their home directory.
- Client-side configuration should be reduced to its very minimum. Clients should automatically get their IP configuration parameters from the server, and this information should be located in a single file, used for all operating systems.
- Since almost every computer now has a hard-disk, clients should be able to take profit of it for reducing network load and as temporary storage space for the user.
- Users *must* have a login to be able to use any of the computers.
- The login should be the same for all operating system and should let the user access its unique home directory, common to all operating systems.
- Student (and secretary :-) computers should be fully cleaned up at each start. That is, the PC should always look like if it were just installed.
- Every computer has to be protected from virus attacks.

These constraints lead us to base our configuration on bootprom tools. We first developped new tools for the excellent TCP/IP Bootprom from InCom GmbH. Now that a standard for preboot execution environments as finally emerged, we ported the tools so that it now also works for any PXE-compliant bootprom. PXE boot roms, also called LanDesk Service Agent, are now distributed with almost all on-board network adapter. For more info on PXE and Intel *Wired for Management* standard in general, read from `http://www.intel.com/managedpc`.

3.1 Boot ROM and Hard-disk

Bootproms exist for quite a long time, but until recently, they were solely used with diskless computers. Since 1996, this How-to has been claiming that bootproms are even more interesting for computers which have a local harddisk, since they allow to take profit of both sides:

- A boot rom make the configurations more robust, since it ensure that the computer will always boot the same way, no matter any virus or partition table crash. It can be used, as we did, to cleanup the harddisk even before the operating system is loaded.
- A local harddisk make the configuration more efficient, since it can reduce the network trafic through caching, and allows for efficient swap.

Today, we have the pleasure to see that all computer manufacturers have come to the same point and provide boot roms as part of new computer standards.

Note that you can still use the tools described below in an *old fashioned* way, that is as a simple kernel/ramdisk loader, even for diskless computers. However, we do not encourage this use.

3.2 The Network

The University of Geneva owns a class B domain, subdivided into several subnets. The CUI uses four subnets, among them one is dedicated to students.

Originally, our PCs were concerned about two network protocols: IPX and IP. On the IPX side, we used a single Novell Netware 3 server for sharing software and users files for DOS and Windows. On the IP side, we used a SUN server for sharing software and users partitions for Linux, with NFS.

In our latest configuration, we do not any more use IPX. There is a single Unix server (which could be Linux as well as a SUN), sharing software and user files using NFS for Linux clients and using SMB (NetBIOS) over TCP/IP for Dos and Windows clients. In this way, we have a single home directory used by all operating systems.

3.3 How it Works

1. When a client PC is turned on, it first performs the traditional system checks before the TCP/IP Bootprom or PXE Boot ROM takes the control.
2. The bootprom issues a BOOTP/DHCP request in order to get its IP configuration parameters.
3. If the server knows the PC issuing the request, it will send back a BOOTP/DHCP reply with informations such as the client's IP address, the default gateway, and which bootdisk image to use.
4. In case of a PXE boot ROM, there might be some more exchanges between the client and the server to determine installation parameters.
5. The bootprom then downloads the boot image from the server using the TFTP protocol. The boot image happens to be a small program called `bpbatch`, our boot-time batch file interpreter.
6. The batch interpreter is started. At this time, it is almost alone in the computer memory. There is no operating system loaded, except the preboot execution environment (offered by the Boot ROM).
7. The batch interpreter look in the BOOTP/DHCP reply for command-line options, and in particular for the name of the batch to execute.
8. According to the instructions in the batch file, it will for instance:
    1. Load a national keyboard mapping

2. Authenticate the user according to a remote server (Unix, Radius or Windows NT)
3. Let the user choose between the available operating systems
4. According to the operating system choosen, repartition the hard-disk and quick-format some partitions
5. Check if an up-to-date compressed image of the selected OS is present at the end of the disk. If not, it download it using TFTP
6. Uncompress the selected OS to the main partition
7. If the selected OS is Linux, load a kernel and start it
8. If the selected OS is DOS or Windows, simply let the computer boot on its fresh new hard-disk

For **DOS and Windows 3.1**, we use the freely available Microsoft LanManager for DOS (search the network for the mirror nearest to you; the distribution consists of three files named `disk1` to `disk4`) as SMB client. Microsoft LanManager supports dynamic configuration using DHCP. After logging in, the user is faced to DOS, and can start Windows 3.1 by typing the traditional `win` command. Note that at this point, DOS and Windows 3.1 appear to be installed locally. For **Windows 95 and Windows NT**, we also use Microsoft SMB client (called *Client for the Microsoft Network*), that supports dynamic configuration using DHCP. We reduce network load using Shared LAN Cache, a nice and powerful network-to-disk cache program.

Students computers can be turned off *the hard way* at any time without risks, since the hard disk is reinitialized at each start.

For "safe" computers (ie. for assistants computers), once the computer has been booted once using the above described system, the boot script simply redirect the boot to the local hard-disk, without cleaning it again. This allow users to leave data on their local hard disk. But whenever the configuration gets corrupted, the user can simply choose from the boot menu in order to have a fresh installation.

3.4 Related non-commercial documentations

This configuration has been successfully reproduced at several places around the world. A few people have written some hints and tricks that complement this How-To. If you did so and that your page is not already referenced in this documentation, please send an e-mail to `Marc.VuilleumierStuckelberg@cui.unige.ch`. And if you experience problems while reproducing this configuration, have a look at these pages !

- `http://www.br.fgov.be/RESEARCH/INFORMATICS/info/bootp.html`, by Alain Empain of the Belgium National Botanic Garden. Many useful sample scripts, and a nice PERL program to automatically generate graphic menus and corresponding HTML documentation from a higher level description.
- `http://www.katedral.se/system/elevsyst`, by Johan Carlstedt of The Cathedral School of Uppsala, Sweden. *At this day, the configuration described at this place is still based on the previous version of the remote-boot tools. However, almost everything remains applicable, given a few changes*.
- `http://vitoria.upf.tche.br/~fred/`, in portuguese, by Frederico Goldschmidt of the

Passo Fundo University, Brasil.

- http://www.etse.urv.es/~larinyo, in spanish, by Lluis Arino, of the Escola Tecnica Superio d'Enginyeria, Spain.

You can also send me your BpBatch script if you want me to include it in the sample scripts collection.

---

4. The Configuration How-To

First, arrange to have the following two machines within arm's reach:

- the **server**, usually a Unix or Windows NT machine
- the **client**, a PC with a bootprom enabled, and nothing valuable on the hard disk.

If you want to test the configuration but you do not yet have a bootprom, you can download the TCP/IP BootProm demo diskette from InCom GmbH at `http://www.incom.de`. This diskette will make your computer behave like if it had a TCP/IP Bootprom plugged in.

If you already have a Boot ROM, you need to enable it. If you are using Incom TCP/IP Bootprom, you can do that using a special program from your network card manufacturer. If you have a PXE Bootprom, you can do it simply from BIOS setup, by changing the default boot device.

For student computers, we configured the boot on network first, and disabled hard-disk and floppy-disk boot. For assistant computers, we also configured network-boot first, but we allow hard-disk and floppy-disk boot.

4.1 Server-side configuration

On the server, you will need the following services:

1. A BOOTP/DHCP server
2. May be a Proxy DHCP server
3. A TFTP server

Note for PXE Boot ROM users: We found after severals hours of tedious search that PXE Boot ROMs with version before 0.99 do not follow the IP protocol and discard all packets that have the *Don't Fragment* (DF) flag set. That means, you will have to disable *Path MTU Discovery* on the server, or the Boot ROM will not see any of its packets. On Solaris, use `ndd /dev/ip ip_path_mtu_discovery` to see if you have it enabled and `ndd -set /dev/ip ip_path_mtu_discovery 0` to disable it. However, this fix only works for non-broadcast packets (ask SUN why...). That means, it will work for TFTP but not for DHCP :-(. Intel has recently fixed this bug, and if you bought your computer after June 1998, you surely have a corrected PXE implementation.

Setting up DHCP

The role of the DHCP server is to give to the client an IP address and to make it load the file named `bpbatch.P` from the TFTP server. DHCP is a superprotocol over BOOTP. If you are using InCom TCP/IP Bootprom, you may live without DHCP (using an old BOOTP server).

On Windows NT, you will probably use the native DHCP server. If you are using InCom TCP/IP Bootprom, you will have to use a special trick to specify the boot file name (get more info from InCom WWW site). If you are using a PXE Bootrom, you will need a Proxy DHCP server, but no other trick is needed as the boot file name will be provided by the Proxy DHCP server.

On Linux, the best choice is the standard DHCP server from the Internet Software Consortium. If you are using a PXE Bootrom, in addition to the usual options, you will need to add the following ones:

- `option dhcp-class-identifier "PXEClient"`
- `option vendor-encapsulated-options ff;`

On Solaris, you can either use the Internet Software Consortium DHCP server (available on the Web), or use Solaris DHCP server (available since Solaris 2.5). However, as Solaris DHCP server does not seems to be able to insert a client class identifier in its DHCP offer, you must install a Proxy DHCP server. Morever, this Proxy DHCP server must reside on

another computer since Solaris DHCP server locks the DHCP port.

We suggest giving infinite lease time for remote-boot clients. Don't forget that BOOTP/DHCP requests are bounded by subnets. If the client and the server do not reside on the same subnet, you should install a BOOTP/DHCP Relay agent on any computer between the two. For now, just assume that both machines are on the same subnet.

Setting up a Proxy DHCP

The role of the Proxy DHCP server is to overcome limitions of some DHCP servers and to provide PXE specific extensions. A proxy DHCP server only makes sense for a PXE Boot rom.

As BpBatch itself is quite powerfull, you wont need to use any PXE specific DHCP extension (menus, etc.). However, if your DHCP server is not able to show minimal PXE compliance, you will need a Proxy DHCP server or your PXE Boot ROM will not accept to go further.

On Windows NT, you can try to use Intel WfM PDK (available from their web site), but it is not very easy to use. We rather suggest having a Linux machine on the subnet and using our small Proxy DHCP. The major advantage of our Proxy DHCP Server for BpBatch is that our server will let you specify an option 155 vendor tag that will be interpreted by BpBatch as a command line.

On Linux and Solaris, you can run our Proxy DHCP program, that simply takes as argument the TFTP server IP address, boot file name and optional arguments, and does everything for you. If the DHCP port on the server is already requested by another daemon, the proxy DHCP server will run on port 4011. In this case, it is necessary that the other daemon on DHCP port answer a DHCP offer with client class `PXEClient` so that the PXE client knows that it must try on port 4011.

If you want to understand better PXE extensions to DHCP, there is an extensive description available on Intel WWW site. However, be warned that the documents are quite confusing, as the protocol has been extended to a number of optional stages, in order to allow for a maximal flexibility. The key to understand it is that all what a PXE client needs is a complete *enhanced DHCP answer*. If it receives only a standard DHCP offer, it will look further until it gets

1. a client class (T60) set to `PXEClient`
2. vendor encapsulated options (T43) (possibly empty, ie. hex `ff`)
3. a non-empty boot filename

The PXE specific negociation ends as soon as all these infos are received, but can lead to a very complex process (install server discovery, etc.) if some are missing.

Setting up TFTP

The TFTP server is a very simple file server. In its basic version, TFTP use 512 bytes data blocks, which are quite inefficients. InCom TCP/IP Bootprom and PXE Boot ROMs allow to use larger blocks (1408 bytes), which speeds up transfers a lot. However, this can only work with an enhanced TFTP server.

On Windows NT, we suggest using InCom enhanced TFTP server, available on their web site.

On Linux, you can use our enhanced TFTP server, available at `http://cuiwww.unige.ch/info/pc/remote-boot/soft/etftpd.tar.gz`.

On Solaris, you should use InCom enhanced TFTP serer, available on the utility disk provided with the TCP/IP Bootprom.

If you prefer using a standard TFTP daemon, remove the `P` in all boot image name extensions, in order to tell the Bootprom to use only the standard TFTP port (This trick was introduced by InCom GmbH for the TCP/IP Bootprom. We still use it as an easy way to select the default TFTP port with PXE bootproms).

4.2 Client-side configuration

First, we will do set up the part common to all operating systems, ie. the batch-file interpreter. Then, for each operating system, we will go through the following steps:

1. Setup a stand-alone client
2. Save its configuration on the server
3. Test it as a remote-boot client
4. Adapt it so that it works for any similar client machine

Once this is done, you will be able to setup any supplemental client just by plugging a Boot ROM in it (or buying a Wired for Management ready computer...) and adding one line in the DHCP configuration file.

Our examples assume that you have a hard disk of 1.4 Gb or more. If you have less, reduce the sizes of the partitions, but remember the you need to leave a few hundreds megabytes unallocated (that is, the last partition must not take up to the last cylinder) to leave free room for the special cache partition. Moreover, as the cache always starts at the cylinder following the last allocated cylinder, if you do not use the same total size for all your tests, you will have to download several times the same files (the cache will be automatically cleared).

Never despair. If you can't get it to work, first look in the *Troubleshooting* section if your problem is not already solved (get the latest version from the Web). Then, take a look in the BpBatch forum. Perhaps someone else had the same troubles as you have, and the answer can be found in the forum. Forum's URL : `http://cuiwww.unige.ch/info/pc/remote-boot/forum/`. If it still does not work, think about monitoring network traffic for network related problems (use `tcpdump` on Linux or `snoop` on Solaris). If you really cannot get it to work, you can send an E-mail to `David.Clerc@cui.unige.ch` or `Marc.VuilleumierStuckelberg@cui.unige.ch`. If your problem is strictly related with the remote-boot configuration and if we are not overflowed, we will try to solve your problem.

4.3 Setting Up the Boot Process

Get the `BpBatch` software, either as `.zip` or as `.tar.gz`. The executables are available at

- `http://cuiwww.unige.ch/info/pc/remote-boot/soft/bpb-exe.zip`
- `http://cuiwww.unige.ch/info/pc/remote-boot/soft/bpb-exe.tar.gz`

The source code (Assembler and C) is also available on request.

In the server `/tftpboot` directory, put the following three special boot images, which together make our pre-boot batch file interpreter:

- `bpbatch.P`, the dynamic loader (respect the uppercase !)
- `bpbatch.ovl`, the relocated interpreter
- `bpbatch.hlp`, the on-line help file

Then add an entry in the DHCP configuration file for your client, with the boot file set to `"bpbatch.P"`. Define a vendor option tag 155 (decimal) with the value `"-i"` (on the standard DHCP server, this is done by the following command: `option option-155 "-i";`). It is interpreted by `bpbatch` as the command line, and `-i` stands for "interactive".

Boot the client computer. You might shortly see

- The Boot ROM copyright
- The string `DHCP` while the client waits for a DHCP reply
- The string `TFTP` while the client waits for the first TFTP packet
- The string `Loading BpBatch` while the loader download the interpreter
- And finaly our banner, followed by a nice *greather-than* prompt

Congratulations ! You have started the batch interpreter... If you are curious about what you can do with it, continue reading

the next section. If you are on a hurry, skip it and go directly install the operating system of your choice. If you have a doubt about a command within the interpreter, type `help`.

Note that you can run the same interpreter within DOS and Linux by running the `MrBatch` program. There are a only very few differences (the Linux version do not have graphics support and the DOS version can only send BOOTP and TFTP requests if the BootProm is not hidden by the operating system).

It may be a good idea to read now the section about the *Syntax Rules* of `BpBatch`, and in particular the paragraphs on *File References* and on *The Cache Filesystem*. This will help you understand the examples.

Once all operating systems will be set up, you will have to make a menu to let the user choose the one he wants. You should be able to discover by yourself how to make such a menu. All necessary commands are documented at the end of this document.

Discovering BpBatch

Try to type `LogVars`. You should get about thirty variables listed. Roughly, the first are BpBatch settings, then come all parameters extracted from the BOOTP/DHCP reply, and the last variable is a list of disks sizes, in Megabytes.

Type `GetPartitions part`, then `LogVars` again. There should be one more variable containing the list of defined partitions on your first hard-drive. Assuming that the first partition is either BIGDOS, FAT32 or LINUX-EXT2, try `LogDir "{:1}"` to get the content of the root directory, then `LogDir "{:1}/usr"` if there is an `usr` directory. You can even try `LogTree "{:1}/etc"` to get a directory tree.

Put a GIF file (format GIF-87a, interlaced or not, but NOT GIF-89a) on your TFTP server. We will suppose that the file is named `image.gif`. You can copy it wherever you want with the following command: `Copy "image.gif" "{:1}/temp/image.gif"`. Or you can use it directly from the server. Now type `Logvars "V*"` and look at the value of the `VESA` variable. If it is `On`, which is most probable, that means you have a VESA-compliant video adapter. You can list the available video modes using `Echo "$VESA-Modes"`. To display your image try the following command: `DrawGif "image.gif"`. The image should be on the upper left corner of the screen. You can draw it on another place by specifying X and Y coordinates after the image name. You can also draw text with `DrawText 200 200 "Hello world" yellow`. Or draw an empty window with `DrawWindow 200 200 300 150`. To insert a title when you create a new window, try `DrawWindow 200 200 300 150 "My Window"`. When you are tired of graphic mode, simply type `CloseGraph`.

Note on graphics : by default, all graphical routines work in the 800x600 VESA mode (with 256 colors), which is the first field of the `VESA-Modes` variable. If you want to use a different video mode, change the variable in order to have the requested video mode as the first field of the list.

Now take a text editor, and create a file named `test.bpb` in the `tftpboot` directory with the following content:

```
:again
DrawWindow 150 200 400 160 "Identity check"
TextAttr Black LightGray
At 15,20 Print "Username : "
Input username 8
At 17,20 Print "Password : "
Getpasswd userpass 8
if "$username" != "smith" goto again
if not "$userpass" match-passwd "BpR8oiIlRR9bo" goto again
#
clear
DrawWindow 200 200 150 100 green blue "Congratulations"
DrawText 220 250 "You got it !" yellow
WaitForKey 3
```

```
CloseGraph
interact
```

In your BOOTP/DHCP configuration, change the option-155 from `"-i"` to `"test"`, and reboot the client computer. The small script should run automatically, and ask you for a username and password. If you do not type `smith` and `justdoit`, you wont be able to boot the computer. Later you will learn how to use a Unix, NT or Radius server to check valid user names.

4.4 Setting Up Linux

In order to set up Linux, you will need to boot the floppy disk provided with the RedHat Linux distribution. BpBatch includes a command that can redirect the boot to the floppy: `FloppyBoot`.

Set up RedHat Linux on your client, with network support, and any packages you may want. You may want to recompile the kernel to better fit your hardware, but it is not necessary.

Configuring the Client

It is probably a good idea to include BOOTP support to the kernel, so that you do not have to customize the client IP address manually.

In order to reduce network load, you might also want to setup the `filecache` for caching on the hard disk files that are loaded by NFS. Roughly, the principle of the filecache is that whenever a symbolic link from the `cache` subdirectory is followed, it is replaced by its target. If the target is itself a subdirectory, each entry of the subdirectory becomes a symbolic link to the original entry of the foreign filesystem. The filecache has been written by Unifix GmbH, and is part of Unifix Linux 2.0. It is freely distributable, and you can get the necessary files from `http://cuiwww.unige.ch/info/pc/remote-boot/soft/filecache.tar.gz`. In order to use the filecache, you have to

- apply a patch to the kernel (file `patch-filecache`), enable filecache support through `make config` or whatever you prefer, and recompile the kernel
- copy the filecache binary file to `/sbin`
- create a mount point called `/mnt/nfs` (using `mkdir`)
- copy `filecache.conf` to `/etc`. This file contains the following lines:

  ```
  Max 100 MB 50 % #
  Cache /mnt/nfs/usr /usr
  Cache /mnt/nfs/opt /opt
  ```
- copy the content of `/usr` and `/opt` to the server, export them read-only with `anon=0` (for allowing root access) and mount them under `/mnt/nfs` (add a line for that in `/etc/fstab`)
- rename `/usr` as `/usr.orig`
- link `/usr` to `/mnt/nfs/usr`
- rename `/opt` as `/opt.orig`
- link `/opt` to `/mnt/nfs/opt`
- ensure that `/usr` and `/opt` are not empty and contains the correct directories
- recursively remove `/usr.orig` and `/opt.orig`
- copy `filecache.init` to `/etc/rc.d/init.d`
- And finally link `/etc/rc.d/rc3.d/S35filecache` to `/etc/rc.d/init.d/filecache.init`

If you successfully followed each of these steps, you should have the filecache working next time you boot, as long as you do not forget to use your patched kernel.

Testing the Configuration

Copy your compressed kernel image (`zImage`, `bzImage`, `vmlinuz` or whatever you call it) to the server `/tftpboot` directory as `linux.krn`. If you had to unplug the bootprom from the PC, you can now plug it again. When `BpBatch` starts, type `LinuxBoot "linux.krn" "root=/dev/hda1 BOOT_IMAGE=linux"` (assuming that the root ext2 filesystem is on the first partition). Alternatively, if you did setup your configuration on a computer without bootprom, just boot let it boot using the loader you installed (lilo, ...). But in the later case, if you want the filecache to work, you should have explicitely installed your kernel with filecache support at the right place.

Wait until the system comes up. If you installed the filecache, you can check that `/usr` has exploded into a directory with some symlinks and some already-exploded directories. Now start the programs that the end-users will use most of the time, in order to load them once for all to the hard disk.

You can still make adjustements to your configuration, like on any stand-alone linux station.

Building the Disk Image

When you are happy with your configuration, login as `root`, go to the `/tmp` directory and run our `mrzip` program. `MrZip` is a command interpreter like `BpBatch`, but it can understand more commands than `BpBatch` does. In particular, it can understand the following commands:

```
showlog
filter -"tmp/*"
filter -"var/log/*"
fullzip "/" "/tmp/linux.imz"
```

This will create a disk image in `/tmp/linux.imz`. Move it to the server `/tftpboot` directory. Then copy the following batch file to `/tftpboot/linux.bpb`:

```
hidelog
setpartitions "linux-ext2:992 linux-swap:32"
fullunzip "linux.imz" 1
clean 2
linuxboot "linux.krn" "root=/dev/hda1 BOOT_IMAGE=linux"
```

The `BOOT_IMAGE` argument is to stay compatible with `lilo` for RedHat 5.1 and later `rc.sysinit`.

Your remote-boot linux configuration is ready ! You can now either set the BOOTP-option-155 to `"linux"`, or type `include "linux.bpb"` from within BpBatch to test it.

System Maintenance and Upgrades

If you want later to upgrade software, install bug fixes and security fixes, proceed as follow:

- Remote-boot a client computer to get a fresh linux install
- Make your changes
- Redo the disk image
- Copy the new image in place of the old one on the server

That means, you can upgrade software on your server-based configuration as if it were a purely local install.

4.5 Setting up DOS 6 and Windows 3.1

On the client computer, boot on your favorite dos floppy disk (either remove the bootprom or type `FloppyBoot` within BpBatch). Format the dos partition of your hard-drive with the `/S` option, in order to put the operating system on it. The size of the partition is not important, as disk archives created with `MrZip` Create a `DOS` subdirectory, copy DOS in it. Install your favorite network client (for instance Microsoft LanManager), Windows 3.1, and so on. If you use Microsoft LanManager, do not use DHCP for the IP configuration as it is a very poor implementation that will almost surely fail with reasonable network load. To do that, add the following lines in your `protocol.ref` file, in the section that loads `tcptsr` (of course, replaces the `xxx` by your true IP parameters):

```
IPADDRESS0 = xxx xxx xxx xxx
SUBNETMASK0 = 255 255 xxx xxx
DEFAULTGATEWAY0 = xxx xxx xxx xxx
DISABLEDHCP = 1
```

Do not be afraid to use EMM386 to optimize the memory usage, and even to include the area where you put your network adapter ROM, since it is not used anymore at this time. But carefully exclude the network adapter RAM, or you will not be able to connect to your server. Use the `NOEMS` parameter.

If you want to ensure that the client machine cannot be used without a valid login name, download our `nobreak` pseudo-device driver (available at http://cuiwww.unige.ch/info/pc/remote-boot/soft/nobreak.zip) and run it at the beginning of your `config.sys`. Then add something like this to your `autoexec.bat`:

```
rem -- we use the dummy file c:\logged as a flag
del c:\logged >nul
:loginneeded
cls
echo Please type in your login name and password
echo.
net logon *
rem -- the login script should have created c:\logged
if not exist c:\logged goto loginneeded
del c:\logged
rem -- now enable break again
echo Yes >NOBRK
```

Ensure that your client boot well by rebooting the client and evaluating the following commands within `BpBatch` interactive mode:

```
HideBootprom
HdBoot
```

Building the Disk Image

On the server, make a share called `admin` for instance, on which you will put some stuff for the system administrator. If the server is a Unix machine, it is a good opportunity to put in `admin` a softlink to the `/tftpboot` subdirectory, so that you can put images in it directly from the client. Within `admin`, create a `/utils` subdirectory and put the following files in it:

- `mrbatch.exe`, the DOS version of `BpBatch`
- `mrzip.exe`, the DOS version of the program for building disk images
- `bpbatch.hlp`, the on-line help file

You might also like to put in the same directory a simple MrZip script named `zipdos.mrz` file that contains the

TOC

commands needed for building a DOS image, like this one:

```
showlog
filter -"lanman.dos/lmuser.ini"
filter -"temp/*"
filter -"*.swp"
fullzip "c:/" "L:/tftpboot/dos.imz"
```

Now go back to your client, mount the `admin` volume on drive `L:`, go to your `utils` directory and type the following command:

```
mrzip -b zipdos
```

One minute later, you will have a new file in the server `/tftpboot` subdirectory called `dos.imz`, which is a compressed image of your hard disk. Copy the following batch file to `/tftpboot/dos.bpb`:

```
hidelog
setpartitions "bigdos:1024"
setbootpart 1
fullunzip "dos.imz" 1
hidebootprom
hdboot :1
```

Your remote-boot DOS configuration is ready ! You can now either set the BOOTP-option-155 to `"dos"`, or type `include "dos.bpb"` from within BpBatch to test it.

Adapting the configuration for other machines

If you want to customize some settings according to the machine, typically the IP settings since Micro$oft DHCP is buggy, you can setup `BpBatch` to change some files before booting. Firsti go to the `lanman.dos` directory and do

```
copy *.ini *.ref
```

Then edit the `.ref` files and replace all fixed parameters with BOOTP variable names as in the following examples:

```
computername = ${BOOTP-Host-Name}
ipaddress0 = ${MS-IPAddress}
subnetmask0 = ${MS-IPSubnet}
defaultgateway = ${MS-IPRouter}
```

Then rebuild the disk image as previously. Note that for IP parameters, we do not use the BOOTP variables directly because LanManager needs then as space-separated numbers instead of dot-separated numbers. Change `dos.bpb` to the following:

```
hidelog
setpartitions "bigdos:1024"
setbootpart 1
fullunzip "dos.imz" 1
set MS-IPAddress="$BOOTP-Your-IP"/.= /
set MS-IPSubnet="$BOOTP-Subnet-Mask"/.= /
set MS-IPRouter="$BOOTP-Routers"/.= /
patch "{:1}lanman.dos/protocol.ref" "{:1}lanman.dos/protocol.ini"
```

```
patch "{:1}lanman.dos/tcpputils.ref" "{:1}lanman.dos/tcputils.ini"
patch "{:1}lanman.dos/lanman.ref" "{:1}lanman.dos/lanman.ini"
hidebootprom
hdboot :1
```

If you prefer, you can also put the `.ref` files in the server `/tftpboot` directory instead of in the disk image.

We like to be able to easily change the computers configuration without rebuilding the image. To do that, copy your `autoexec.bat` and `config.sys` as `autoexec.ref` and `config.ref` to the server `/tftpboot` and add the following two lines to the batch file above:

```
patch "autoexec.ref" "{:1}autoexec.bat"
patch "config.ref" "{:1}config.sys"
```

You can then freely change the files and even customize them with machine-dependant values obtained from BOOTP.

After making any change to the client machine configuration, do not forget to rebuild the disk image using `mrzip` if you want to preserve your changes.

System Maintenance and Upgrades

If you want later to add new software or change anything else, proceed as follow:

- Remote-boot a client computer to get a fresh install
- Make your changes
- Redo the disk image
- Copy the new image in place of the old one on the server

That means, you can upgrade software on your server-based configuration as if it were a purely local install.

4.6 Setting up Windows 95

In previous versions of this document, we used the Microsoft server-based installation of Windows 95, but it was really too much pain and not much worth:

- It is very, very bogus
- Many software package do not support it and their install will fail. Among them, Microsoft Internet Explorer, OnNet 32, Novell's Protected-mode client (which is MUCH more secure than Microsoft Client for Netware).
- It cannot be used with the Microsoft Network client over TCP/IP, since Microsoft provides no real-mode driver for TCP/IP compatibe with Windows 95. That means, it cannot be used with Samba
- It makes software upgrades almost impossible since every client turned on will lock many DLLs on the server, and thus produce *sharing violations* if you try to upgrade them.

Consequently, we throwed away of this document all the informations and bug-workaround collected during months (you can still find them as a HTML document at http://cuiwww.unige.ch/info/pc/remote-boot/win95old/win95old.html) and turned to our new disk-based remote-boot concept. Basically, the configuration for Windows 95 is now almost as easy the configuration for DOS.

Setting up a Stand-Alone Client

Setup a regular Windows 95 client, either starting from scratch as explained in the configuration of a DOS client, starting from the DOS client and installing over the network (that is what we did). You can also start with a preconfigured Windows machine, but you will probably have less knowledge of what stuff is on the hard disk.

Proceed as described above for a DOS client. It is usually NOT necessary to use EMM386 with Windows 95. If you are using Windows 95 OSR2 (alias MSWIN 4.1, alias Windows 95 service pack 1, alias Windows 95 with Internet Explorer),

you should add the following line in the `[Options]` section of `MSDOS.SYS` (yes, it is a text file):

```
AUTOSCAN=0
```

This will let Windows know that you do not want ScanDisk to be runned automatically at boot time.

If you want to reduce network and server load (which will improve your system performances) while keeping all softwares on the server, you should consider installing the excellent Shared LAN Cache, from Measurement Techniques, Inc (see http://www.lancache.com). This software runs on each client computer, and caches to the local hard disk every data obtained from the network. Even MS-Office starts much faster the second time you run it... You need one license per client computer, but it is not very expensive, and the firm make special prices for universities and colleges. The best thing to do is to go to their Web site and download the free evaluation copy.

Building the Disk Image

Your MrZip script will be named `zipwin95.mrz` and contain:

```
showlog
filter -"temp/*"
filter -"*.swp"
fullzip "c:/" "L:/tftpboot/win95.imz"
```

To build the image, mount the `admin` volume on drive `L:`, go to your `utils` directory and type the following command:

```
mrzip -b zipwin95
```

A few minutes later, you will have a new file if the server `/tftpboot` subdirectory called `win95.imz`, which is a compressed image of your hard disk. If your compressed image was bigger than 87 MB, it has probably been splitted in two or more fragments. These fragments will automatically loaded one after the other when needed. Note that an image bigger than 87 MB will usually take More than one minute to uncompress and may irritate your users. Our Windows 95 image is only 70 MB big, because most software (except Office and Explorer) completely reside on the server. Only 45 seconds are needed to uncompress the image and restore the full disk.

Copy the following batch file to `/tftpboot/win95.bpb`:

```
hidelog
setpartitions "bigdos:1024"
setbootpart 1
fullunzip "win95.imz" 1
hidebootprom
hdboot :1
```

Your remote-boot Windows 95 configuration is ready ! You can now either set the BOOTP-option-155 to `"win95"`, or type `include "win95.bpb"` from within BpBatch to test it.

Adapting the configuration for other Machines

The big difference between Windows 3.1 and Windows 95 is that the later includes code for Plug-and-play , ie. automatic detection of your hardware. This not a bad thing in itself, but the trouble is that it is often too sensible, and that it sometimes fails.

If you try to start another client with exactly the same boot image, you will probably get several messages during start telling that Windows has detected new hardware: a new sound card, a new hard-disk, a new network card, and even a mouse... There can be two reasons for that:

- the devices may not use the same ressources (for instance the mouse is not connected on the same port, or the sound card is not connected in the same slot - yes, that is detected)
- the devices may tell to Windows 95 their personal serial number (for instance, every Windows 95 differenciate every network card on the basis of its world-wide unique ethernet address)

The fact that Windows 95 discover that the hardware has changed may not be a problem if the plug-and-play works as-is, but it become a problem when the plug-and-play does not work. For instance, Windows 95 plug-and-play for our Logitech PS2/aux mouse does not work, and result in no mouse at all. To solve such kind of problems, arrange to have all computers as similar as possible, or make different images for different hardware. Later, you will discover that you can simply use the same image and just have several copies of the registery, that you can copy after having restoring the disk image but before booting.

The thing you cannot avoid to differ between computers is the network card. PCI cards usually do not mind, but ISA Plug and Play do. Bad luck for us, the plug-and-play code for our SMC EtherEZ card hangs the computer. The only solution is to let Windows 95 believe that it already know the network card, and that it is not necessary to trigger plug-and-play. The trick for doing that is to automatically insert an entry for the network card in Windows 95 registery, before starting it. Note that this trick is not any more needed with most PCI cards.

Move the `autoexec.bat` to the server as described above for DOS. Edit it (on the server) and add the following lines:

```
rem --- Patch Windows registery in order to avoid plug-and-play detection
regedit /L:c:\windows\system.dat /R:c:\windows\user.dat c:\temp\patch.reg
```

`regedit` is a standard Windows 95 program that let you browse the registery if you start it from within Windows 95, or do simple operations on the registery if you call it from DOS. Run `regedit` under Windows 95, search for your network card, usually under

```
HKEY_LOCAL_MACHINE\Enum\ISAPNP
```

and export the branch using the *File* menu. This will create a text file, that you should same as `patch.ref` in the server `/tftpboot` diretory. Edit this file and find out where the card ethernet address is stored (do that on two different machines and compare the files if you can't find it by yourself). Replace it by a pettern in the form `${MACID}`. Then add lines to the `win95.bpb` script like this:

```
set macid = "$BOOTP-Client-ID"
patch "patch.ref" "{:1}temp/patch.reg"
```

(do any necessary string manipulation for setting `MACID` if it is not exactly the client Ethernet address). That's all, your clients should not any more try to autodect the network card.

Once again, this whole trick is not necessary when using PCI network adapters. Incidentally, we can use the same mechanism for automatically configuring the hostname, which Windows 95 does not seem to take into account when configuring through DHCP. We just add the following line to our `patch.ref` file:

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP]
"ComputerName"="${BOOTP-Host-Name}"

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\MSTCP]
"HostName"="${BOOTP-Host-Name}"

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\control\ComputerName\ComputerName]
"ComputerName"="${BOOTP-Host-Name}"
```

Using this small registery trick, your configuration should normally be portable for all machines with similar configura
If you cannot avoid that Windows detect some hardware as new on one machine, try to rebuild the disk image from this machine. This will include the registery configuration specific to this machine into the image, and hopefully supress the problem.

System Maintenance and Upgrades

If you want later to upgrade software, install bug fixes and security fixes, proceed as follow:

- Remote-boot a client computer to get a fresh install
- Make your changes
- Redo the disk image
- Copy the new image in place of the old one on the server

That means, you can upgrade software on your server-based configuration as if it were a purely local install.

4.7 Setting up Windows NT

We do not use Windows NT for remote-boot client computers but we have tested our system to ensure that it work as well. And it works.

As our utilities currently have no support for NTFS (we neither have the documentation nor the time to do that, but I would be happy to help anyone who is interested in doing it), you will have to install NT on FAT16 (simply do not convert your partitions to NTFS during the setup).

Copy your `win95.bpb` boot script to `winnt.bpb`. Change the `setpartitions` line in `winnt.bpb` to the following:

```
setpartitions "BIGDOS:512 BIGDOS:512"
```

Then boot Windows 95 using this script, and install your NT client on drive C. Do not worry about the second partition for now. Do not install too much stuff, or you will get a really large and slow-to-uncompress image. Remove Windows 95 from the disk disk C, you do not need it in a Windows NT image (the boot menu is handled by the bootprom, not by NT boot loader).

Reboot your computer in without overwriting the hard disk, ie. do not execute the `winnt` script but just

```
hidebootprom
hdboot
```

Your NT station should start-up correctly. Make any necessary customization.

Building the Disk Image

The trouble with Windows NT is that direct disk access is prohibed by the kernel. That means, `MrZip` will not even be able to read the boot sectors. The best way to do an image is then to boot Windows 95 and to run `MrZip` from a DOS window. To do that, change the `winnt.bpb` script so that the Windows 95 image is not restored on the first but on the second partition:

```
hidelog
setpartitions "BIGDOS:512 BIGDOS:512"
setbootpart 2
fullunzip "win95.imz" 2
hidebootprom
hdboot :2
```

(if you have any supplementary patch, change the `"{:1}"` to `"{:2}"`). Boot with this script; you should have Windows 95 running, but a new drive `D:` should be available, with Windows NT inside.

Make your disk image as usual (but on `D:`, of course), and save it as `winnt.imz` on the server `/tftpboot` directory. Edit one last time the `winnt.bpb` script like this:

```
hidelog
setpartitions "BIGDOS:512 BIGDOS:512"
setbootpart 1
fullunzip "winnt.imz" 1
clean 2
#fullunzip "win95.imz" 2
hidebootprom
hdboot :1
```

Your Windows NT remote-boot configuration is ready. Of course, if you do not like to have two partitions, you can setup a single partition instead. But when you have to rebuild the image, you will have to setup the second partition again for booting Windows 95.

System Maintenance and Upgrades

If you want later to upgrade software, install bug fixes and security fixes, proceed as follow:

- Remote-boot a client computer to get a fresh install
- Make your changes
- Edit `winnt.bpb`: comment the `clean` and winnt `fullunzip`, uncomment win95 `fullunzip`
- Redo the disk image
- Copy the new image in place of the old one on the server

That's all, folks !

4.8 Troubleshooting (FAQ)

This section lists most frequently encountered problems.

The image download never ends

You are probably using a standard TFTP server, and it cannot handle more than 65535 packets of 512 bytes (or even 32767 packets for the Solaris server). That is, your image must be fragmented in pieces of no more than 30 MB (or 15 MB for Solaris). See under *CopyArchive* for instructions on fragmenting an existing image. But you should seriously thing about using InCom's extended TFTP server, as it is much more efficient (it uses packets of 1408 bytes instead of 512 bytes).

The archive decompression fails immediately

There are three possibilities. Either the image is really corrupted on the server (try use MrZip to see if it is the case), or the file transfer has failed because of TFTP timeout, or because of incompatible protocol.

TFTP timeout occurs when the network is too heavily loaded (for instance if you try to download a huge image with more than four clients at a time). In this case, `BpBatch` does not retry indefinitely because it would not help. Shut down a few computers and retry with no more than four computers (or maybe even three). If you often need to download images for a lot of computers, you can try our special Broadcast TFTP server (see the section dedicated to it).

Incompatible protocol is caused by using a standard TFTP server (typically the one built-in in your UNIX server)

while asking BpBatch to work with enhanced TFTP. If you use a standard TFTP server, **you should remove the extension** (see the explanation in the next question).

The computer hangs instead of downloading/unzipping (1)

If you are using Incom's TFTP server, try to add -s 1408 59 to the command line. If you are not using an enhanced TFTP server, remove the `.P` extension from BpBatch filename on the server and in `bootptab`.

Detailed explanation : this problem occurs if you did not setup an extended TFTP server but you used `bpbatch.P` as the bootfilename DHCP/BOOTP tag. BpBatch will indeed try to connect to an extended TFTP server when the bootfilename ends with a `.P` extension. To solve this problem, you can either remove the `.P` extension at the end of the bootfilename (it will tell BpBatch to use standard TFTP) or install an extended TFTP server. The only supported extended TFTP server today is the one provided by Incom. You can find compiled binaries on their web site, or on our distribution directory. For Incom's TFTP server to properly work with the extended TFTP feature, you must add `-s 1408 59` to the command line.

The computer hangs instead of downloading/unzipping (2)

May be your computer has a bad VESA support. Try giving the `-v` command-line argument or setting the VESA variable to `"OFF"`.

VESA scrolling is broken

We use a VESA 1.1 function for scrolling. If your video adapter does not support VESA 1.1, forget it. If the scrolling works for one page, but then produces a strange strippled pattern, do not worry. This is a known bug, I will fix it as soon as I have time for it (VESA scrolling is not really essential...)

There is a corrupted file in the cache

When a file in the cache is corrupted by an external program, it is automatically removed from the cache. When a file in the cache is not fully written (because the computer is turned off during the file transfer), it is also automatically removed. But if the server transmits a corrupted file or if the transfer aborts from the server side, it is possible that this file stays in the cache. You can clean-up the cache simply by holding both shift down while `BpBatch` access it for the first time. Alternatively, you can evaluate `clean -1` in interactive mode.

The EXIT command does not work in a batch file

This is not a bug. Exit is not a command. There is no exit or quit command because it does not make any sense to exit from a boot script without booting. And MrBatch is really the same program as BpBatch. What you can do instead is calling HdBoot. This makes sense, and the DOS version will cleanly exit instead of rebooting. Note that you can exit from the DOS version at any time by pressing Ctrl-Break. This will restore all hooked interrupts before leaving.

The Print command does not print

If you try to print something and immediately enter interactive mode, you may not see your text. This is because your text was written on the *runtime* screen and the `Interact` command has switched the display to the *Log* screen. Just put a `GetKey` after the print commands and you will see the text output.

MrZip says `Malloc failed`

MrZip needs a lot of conventional memory to run. If you encounter this problem, first ensure that you have unloaded the bootprom either using `HideBootprom` or using InCom's `bputil`. If you run MrZip from bare MS-DOS (not within Windows 95 DOS box), you should use EMM386 to load the network drivers high in order to get as much conventional memory as possible. From a Windows 95 DOS box, there is usually no problem (as long as you have not left your old 16-bit stuff in your `autoexec.bat` when you installed Windows 95).

MrZip aborts while reading directories

This bug has already been fixed once. Get the latest release of `MrZip`. If the problem persists, try to build your image with `Trace` set to `"ON"` (and usually `PauseLog` set to `"OFF"`); this will let you discover which file causes the problem. Send a detailled bug report.

MrZip cannot access some file

MrZip is probably trying to read a locked, open or special file, such as Windows swap file. Such files should usually not be included in the image and should be filtered out (using the `filter` command). It is also possible that the operating system is playing you a trick. If MrZip does not tell you what file causes the problem, try to build your image with `Trace` set to `"ON"` (and usually `PauseLog` set to `"OFF"`). You can also try to use direct disk access (that is, do not refer the source partition as `"C:"` or `"/"` but as `"{:1}"` or whatever partition it is). Using direct disk access is usually slower because we have less buffers than the operating system, but it may be sometimes more reliable.

Disk images are always reloaded from the server

Disk images are stored in the special cache area and should not be reloaded if they have not changed on the server. However, as the cache area always starts after the last used partition, changing the total size of partitions will move the location of the cache and thus destroy its content. Another possible reason for a file disappearing from the cache is that the previous file has grown more than one-and-an-half times its initial size. The file would then have been overwritten and need to be downloaded once again. This should almost never occurs. A third possible reason is a too small cache area. If the free space left outside the partitions is less than one-and-an-half times the sum of all compressed image sizes, only the most recently used images will be present in the cache and the other will have to be reloaded on demand.

Red Hat Linux 5.1 does not boot properly

This distribution assumes Linux was booted using `lilo` and checks for the `BOOT_IMAGE` command line argument (in `/etc/rc.d/rc.sysinit`). Simply add it in the `linuxboot` call, or change your `rc.sysinit`.

The broadcast TFTP ramdisk hangs (*Got in bound state*)

Linux dhcp client is a program that dynamically changes the IP address of the client according to DHCP offers. If the address is offered forever (infinite lease time), the DHCP client just set the address and returns (this is what we expect). However, if the lease time is limited, the DHCP client must remain loaded and ask for new addresses every few minutes. And if the DHCP client does not return, MrBatch will never be loaded... The solution is to give an infinite lease time (sometimes encoded as -1).

File access hangs under BpBatch, but not under MrBatch

This problem occured on an AMI BIOS dated 94/07/25. We investigated a little bit, and found no solution. It seems that this problem is due to a bug in this BIOS (some register or memory location must be destroyed).

Unzip of a fragmented archive fails (Malloc failed)

This problem was introduced with PXE compatibility, but has now been fixed. Please get the latest version.

MrBatch and MrZip complain about the terminal under RedHat 5.x

This problem has been fixed in the 9th of August version of MrBatch/MrZip. There was a problem with a new version of ncurses which has been released with RedHat 5.1.

"libncurses.so.3.0: cannot open shared object file" under Linux

MrZip has been linked to the version 3.0 of libncurses. You can use other versions of libncurses only if they are newer than version 3.0. To use a newer libncurses, all you have to do is to create a soft link from libncurses.so.3.0 to your libncurses.so.xx file. With RedHat 5.1, you can use the following command : `cd /usr/lib ; ln -s libncurses.4.2 libncurses.3.0` You can also download a version recent version of mrzip/mrbatch. Starting from the 10/25/98, mrbatch is now compiled under RedHat 5.1.

MrBatch and MrZip do not start under Linux (file not found)

This problem is the reverse of the previous one. Now that the distribution is libc6 ready, it cannot be used any more with libc5. If you encounter this problem, simply upgrade your Linux box (Well, if we hear too much complaints, we might try to keep two distributions...).

I can not access other mode than the default 800x600 VESA mode

You should first display the contents of the `VESA-Modes` variable, to see if your hardware support the mode you would like to use. Then, try one of the two ways to select a special VESA mode :

- `InitGraph "mode"`: Try InitGraph "1024x768", and then run the graphical primitive you are interested (e.g `DrawGif`).
- `VESA-Modes`: The first field of the `VESA-Modes` variable is the name of the default mode. If you change the VESA-Modes variable, all graphical primitive will use the mode you specified.

BpBatch prints a "Malloc failed" message when restoring multiple fragments images

We corrected a bug in the memory allocation functions of BpBatch. You should make sure that you have a version of BpBatch which has been released after september the 22nd 1998.

Fullunzip using the Linux version of MrBatch always fails

We corrected this problem in the 09/22/1998 release.

Scandisk says my disk is corrupted

The 10/25/98 release did correct a problem with large images. Try to download a recent version of BpBatch.

My RedHat boot floppydisk does not work with FloppyBoot

This bug has been corrected in the 10/25/98 release.

My FAT32 disk image does not boot properly

This bug has been corrected in the 02/09/99 release.

---

5. Remote-Boot Tools Reference Manual

This section provides detailled informations on the use of the tools we developped at the CUI, University of Geneva for this remote-boot configuration.

5.1 BpBatch, MrBatch and MrZip

These three names stand for three variants of the same program, with the following characteristics:

- `BpBatch` is a special program that can be started from the BootProm before the operating system is loaded. It is made of two parts: `bpbatch.P`, the dynamic loader, and `bpbatch.ovl`, the program itself. `BpBatch` has full disk I/O capabilities through our own implementation of FAT16, FAT32 and Ext2fs, as well as remote network I/O capabilities through the BootProm TFTP API. `BpBatch` was compiled under DOS using Borland C 5.0 and Turbo Assembler 3.2.
- `MrBatch` is the DOS/Linux version of `BpBatch`. All commands recognized by `BpBatch` are recognized by `MrBatch` and vice versa. This is very usefull if you want to test your batch scripts from a DOS/Linux session. Under DOS, `MrBatch` emulates remote I/O by OS-based file access if the bootprom is not available. Under Linux, the bootprom cannot be seen anymore but `MrBatch` can emulate it using Linux IP support, or use OS-based file access. `MrBatch` was compiled under Linux using GCC 2.7.2.1 and under DOS using Borland C 5.0 and Turbo Assembler 3.2.
- `MrZip` is an interpreter that recognizes a superset of `MrBatch` language, and that serves to build disk images. In `MrZip`, the limited remote file I/O is replaced by a full-featured OS-based file access. `MrZip` does not include VESA support. `MrZip` was compiled under Linux using GCC 2.7.2.1 and under DOS using Borland C 5.0 and Turbo Assembler 3.2.

Command Line Arguments

All programs accept the same syntax of arguments. `MrBatch` and `MrZip` take them from the command line, while `BpBatch` look for them in the BOOTP option 155 (decimal). Here is the syntax of the arguments:

```
[-x] [-l] [-b] [-v] [-w] [-i] [script-basename]
```

where:

- `-x` disable the use of extended memory
- `-l` disable the use of ISO-latin-8859-1 as default character set
- `-b` cancel the bootprom detection (which cause a floppy seek under DOS)
- `-v` cancel the VESA detection (which cause a switch to full screen under Windows 95)
- `-w` enable direct disk write access (disabled by default under DOS and Linux)
- `-i` enable interactive mode even if a script name is provided

The `script-basename` is optional. If provided, `MrBatch` and `BpBatch` load the file with the `.bpb` extension, and `MrZip` loads the file with the `.mrz` extension. If not provided, `MrBatch` and `MrZip` run in interactive mode while `BpBatch` loads the file with the same basename as the BOOTP Boot file and a `.bpb` extension.

TOC

Syntax rules

The following rules apply when `BpBatch` parses an input line.

- Commands are parsed line by line. Lines are separated by CR and/or LF.
- The maximal line length is currently 255 characters.
- Keywords and variable names are case-insensitive.
- " is interpreted as the special string delimiter
- When ${variable} or $variable is encountred, it is substituted by the value of the variable, or by an empty string if the variable is undefined. The substitution also occurs within a string. Moreover, the resulting substituted value must be explicitely enclosed between double quotes if used as a string value (ie. one should merely speak of macro expansion than of a variables).
  - ❍ `\a` is substituted by the audible-bell character (ASCII 7)
  - ❍ `\b` is substituted by the backspace character (ASCII 8)
  - ❍ `\n` is substituted by the newline character (ASCII 10)
  - ❍ `\r` is substituted by the return character (ASCII 13)
  - ❍ `\t` is substituted by the tabulation character (ASCII 9)
  - ❍ `\v` is substituted by the vertical-tab character (ASCII ...)
  - ❍ `\nnn` where n is a 3-digit octal number between 000 and 377 is substituted by the character with ascii code specified
  - ❍ `\X` where X is any other character not listed above is substituted by X itself. In particular,
    - ■ `\"` is substituted by a regular double-quote (not a string-delimiter)
    - ■ `\$` is substituted by a regular dollar sign (not variable substitution)
    - ■ `\\` is substituted by a regular backslash (not a special character)
- The character "end of string" (ASCII code 0) CANNOT be used anywhere as it is used internally as end-of-string delimiter
- The character "floating diaeresis" (ASCII code dec 249, hex F9, octal 371) CANNOT be used in any string as it is used internally as string delimiter in the input parsing routine.
- The character "block space" (ASCII code dec 255, hex FF, octal 377) CANNOT be used in any variable value as it is used internally as variable delimiter.

Empty lines are ignored. Lines starting with a sharp (#) are treated as comments and are not interpreted. Lines starting with a column (`:`) are treated as labels and are not interpreted.

String expressions

Strings are delimited by opening and closing double-quotes:

```
"Hello world"
```

To include double-quotes within a string, quote them using a backslash:

```
"I said: \"Hello world\""
```

Strings can be postfixed with a few operators.

- ❍ The character substitution operator:

```
"Hello world"/o=u/      ==      "Hellu wurld"
"198.76.54.32"/.= /     ==      "198 76 54 32"
```

- ❍ The word selection operator (zero-based):

```
"Hello world"{0}        ==      "Hello"
"198 76 54 32"{1-3}     ==      "76 54 32"
```

- The substring selection operator (zero-based):

```
"Hello world"[4]        ==      "o"
"Hello world"[4-7]      ==      "o wo"
```

Operators can be chained by postfixing one after the other. For informations about the string length and word count operators, see under "Numerical expressions".

Numerical expressions

Numerical expressions work on 32-bits integer numbers (from -2,147,483,646 to 2,147,483,647). Hexadecimal octal and binary numbers are not understood. Whenever a numerical expression is expected, the following are recognized:

- A positive or negative integer number
- An expression in the form (*expr1 op expr2*) where *op* can be either +, -, * (multiply), / (divide) or % (modulo) and `expr` is a numerical expression. Note that EACH operation MUST be enclosed between parenthesis :

```
((3 * 5)+2)             == 17
```

- The string-length operator (@), followed by a string :

```
@"Hello world"          == 11
```

- The word-count operator (#) followed by a string :

```
#"Hello world"          == 2
```

Durations

A few commands expect durations as arguments. Durations are measured in seconds, with a precision of up to a tenth of second:

```
Delay 3                 waits for 3 seconds
Delay 0.3               waits for 3/10 seconds
```

Colors

Whenever a color is expected, you can either use the numeric value of the color or its symbolic name (case-insensitive). The following colors are recognized

```
Black          0
Blue           1
Green          2
Cyan           3
Red            4
Magenta        5
Brown          6
LightGray      7
DarkGray       8
LightBlue      9
LightGreen    10
```

```
LightCyan      11
LightRed       12
LightMagenta   13
Yellow         14
White          15
```

File References

File names are strings. They must therefore always be enclosed between double-quotes. File names are case-sensitive on case-sensitive filesystems, case-insensitive on case-insensitive filesystems. Slash and backslash can be freely used one in place of the other. Do not forget to double backslash since a single backslash is an escape character.

There are two kinds of file references:

- Direct disk files
- Foreign files

Direct disk files are referenced using the following notation:

```
"{disk:partition}/absolute/filename"
```

The disk number can be omitted and defaults to zero. For instance, `"{:1}/usr/bin"` points to `/usr/bin` assuming there is such a directory on the first partition. Direct file I/O is solely based on our own file access routines (we do not use the operating system).

There are two *special* partitions. Partition zero corresponds to the hard disk master boot record (MBR) and has a pseudo file-system which let you access the boot code. Partition minus-one (-1) corresponds to the cache filesystem (see below).

Under BpBatch/MrBatch, foreign files correspond to remote files on the TFTP server when the BootProm is available:

```
"help.bpb"           is the file help.bpb in the /tftpboot directory
"gifs/MyImage.gif"   is a file in /tftpboot/gifs
```

Other TFTP servers can be referenced :

```
"198.76.54.32:help.bpb"
```

If the other server is behind a gateway :

```
"198.70.0.1/198.76.54.31:help.bpb"
```

One can also specify a specific port for the TFTP connection :

```
"198.76.54.32@89:getpasswd/smith"
```

There can be only one open remote file at a time. If the BootProm is not available, remote files are emulated using the operating system file I/O, but the same restriction apply.

Under MrZip, foreign files correspond to files as seen by the operating system. There is no limitation, and foreign files can be used wherever direct disk files can be. Foreign files are usually faster than direct disk files, because the operating system has more buffers. Foreign files can refer to network files if supported by the operating system.

```
"C:\\autoexec.bat"
"C:/config.sys"
```

```
"/mnt/net/usr"
```

The Cache Filesystem

In order to reduce network load and to fasten the boot process, disk archives, linux kernels and possibly other files are cached on the hard disk. This disk cache is located at the end of the hard disk, between the last cylinder allocated in the partition table and the last physical cylinder of the disk (out of any allocated partition). There MUST be room between the last partition and the end of the disk if you want the cache filesystem to work. The cache filesystem MUST work if you want to restore a disk image.

The disk cache is organised in a volatile, CRC-validated filesystem : Each directory entry and each 32 KB data block is validated by a 32-bits CRC. Whenever a directory entry or a data block unexpectedly changes, the file is automatically removed from the cache and downloaded again upon the next request.

You can freely access the cache filesystem from within BpBatch, MrBatch and MrZip using direct disk access on the special partition `"{:-1}"`. To see the content of the cache, just type :

```
logdir "{:-1}"
```

If the cache ever gets corrupted and is not automatically cleaned (which should never occurs), you can either type :

```
clean -1
```

(in interactive mode) or hold both shifts down when BpBatch access the cache for the first time.

Special variables

Some variable are initially set and/or have special meanings. Some of them exist within all programs, other are only available under MrZip and other are only available when a BOOTP/DHCP reply has been received.

General variables

- `$Program` is set to "BpBatch" within BpBatch, "MrBatch" within MrBatch and "MrZip" within MrZip
- `$Basename` is set to the basename of the script on which the batch interpreter was started
- `$HelpFile` is the name of the file loaded when `Help` is invoked. Default: `"${Basename}.hlp"`
- `$BOOTP-...` are variables set from the BOOTP/DHCP reply (see the paragraph on BOOTP/DHCP variables for more details)
- `$DHCP-...` are variables set from the DHCP reply (see the paragraph on BOOTP/DHCP variables for more details)
- `$Disks` is set to the space-separated list of sizes for each disk. That means, `#"$Disks"` represent the number of disks and `"$Disks"{0}` is the size of the first disk
- `$Keypressed` is set to the next ready-to-read key available in the keyboard buffer (if available)
- `$LBA` controls the use of LBA to access disks > 2Gb. Default: "ON"
- `$FDA` controls the use of fast disk access (write accross cylinders). Default: "ON"
- `$VESA` controls the use of VESA graphics. Default: "ON" if available
- `$VESA-Modes` gives the list of all available VESA modes. The first entry of the list is the default mode, which is used when no parameter is given to InitGraph. Note: if VESA="OFF", this variable is blank
- `$APM` is set to "ON" if your computer supports Avanced Power Management. If $APM is "ON", you can use the command PowerOff to turn your computer off. Default: depends on your hardware
- `$Trace` controls the display of each command before execution. It also controls the display of file names when creating new archives. Default: "OFF"
- `$AutoShowLog` controls the automatic switch to the text log whenever the ESC key is pressed. Default:

"ON"

- `$PauseLog` controls the pause between each page of log when the log is visible. Default: "ON"
- `$CacheDisk` is set to the disk used for caching remote files. Default: empty == 0, the first hard disk
- `$CacheAlways` controls the automatic caching of remote files copied, patched or drawn as GIF. Default: "OFF"
- `$CacheNever` prevents any file from being cached. Turn this variable on for diskless Linux boot. Default: "OFF"
- `$CacheReserve` controls the preventive allocation of 25 percent more space than necessary in the cache partition, to let the files grow. Turn this variable off if you are short of disk space. Default: "ON"
- `$ExtMemory` controls the use of Extended Memory (or XMS). Once deactivated, extended memory cannot be reactivated. Default: "ON" if available
- `$IsoLatin` controls the interpretation of upper ASCII codes in included and patched files. The IsoLatin settings are processed at the time the file is loaded, not at the time the file is processed. Default: "ON"
- `$ProgressX` and `$ProgressY` controls the position of the progress window displayed in VESA graphics during archive download and decompression. Default: 200 200
- `$EXT2-Backup` controls the update of superblock backups in Linux ext2 filesystem. Superblock backups take a few seconds to do and are never used by current kernels (only by e2fsck).
- `$Security-Gateway` controls the gateway-server used for user authentication. Our special authentication gateway must be running on the target computer. Default: `"${BOOTP-Server-IP}@89"` (ie. the TFTP server, on port 89)
- `$Security-Check` contains the answer of the security server for the last check performed, either PASSED or FAILED. Default: "FAILED"
- `$Security-Passwd, $HelpTopic, $OnExit, $OnKey-...` are used internally.

See also BOOTP variables and MrZip-specific variables.

MrZip-specific variables

The following variables are only used within MrZip.

- `$TempPath` controls the directory where temporary files will be stored. Default: <empty> == current directory
- `$DumpFormat` controls the way archives are dumped to the log when requested. It is a string containing
  - "h"/"H" to display the archive header
  - "b"/"B" to summarize/dump boot sectors
  - "s"/"S" to display a short/long allocation summary
  - "d"/"D" to display a short/long directory listing
  - "f"/"F" to summarize/dump files

  Default: "hbD"
- `$FragmentSize` controls the size of archive pieces. If you do not use InCom's extended TFTP server, you should set this to "30 MB". Default: "87 MB"
- `$SourceArchive, $DestArchive, $Filter...` are used internally.

BOOTP variables

The following BOOTP-... and DHCP-... variables are recognized, as long as a BOOTP/DHCP reply has been received (TCP/IP Bootprom must be reported as detected):

```
$BOOTP-Client-ID
$BOOTP-Your-IP
```

```
$BOOTP-Server-IP
$BOOTP-Gateway-IP
$BOOTP-Bootfile
$BOOTP-Server-Name
$BOOTP-Subnet-Mask
$BOOTP-Time-Offset
$BOOTP-Routers
$BOOTP-Time-Servers
$BOOTP-Name-Servers
$BOOTP-Domain-name-Servers
$BOOTP-BOOTP-Log-Servers
$BOOTP-Cookie-Servers
$BOOTP-Lpr-Servers
$BOOTP-Impress-Servers
$BOOTP-Resource-Location-Servers
$BOOTP-Host-Name
$BOOTP-Boot-Size
$BOOTP-Merit-Dump
$BOOTP-Domain-Name
$BOOTP-Swap-Servers
$BOOTP-Root-Path
$BOOTP-Extensions-Path
$BOOTP-IP-Forwarding
$BOOTP-Interface-MTU
$BOOTP-All-Subnets-Are-Local
$BOOTP-Broadcast-Address
$BOOTP-NIS-Domain
$BOOTP-NIS-Servers
$BOOTP-NTP-Servers
$BOOTP-Font-Servers
$BOOTP-X-Display-Manager
$DHCP-IP-Address-Lease-Time
$DHCP-Message-Type
$DHCP-Server-Identifier
$DHCP-Message
$DHCP-Renewal-Time
$DHCP-Rebinding-Time
$BOOTP-NIS+-Domain
$BOOTP-NIS+-Servers
$BOOTP-Server-Name
$BOOTP-Bootfile
$BOOTP-Mobile-IP-Agent
$BOOTP-SMTP-Servers
$BOOTP-POP3-Servers
$BOOTP-NNTP-Servers
$BOOTP-WWW-Servers
$BOOTP-Finger-Servers
$BOOTP-IRC-Servers
$BOOTP-StreetTalk-Servers
$BOOTP-STDA-Servers
```

Other BOOTP/DHCP parameters can be used under the name

```
$BOOTP-Option-n
```

where n is the decimal representation of the BOOTP option number.

Do not mix-up `BOOTP-Gateway-IP`, which is the gateway to use for TFTP and should be 0.0.0.0 if the TFTP server is in the same subnet, and `BOOTP-Routers`, which contains the default IP gateway(s). The TCP/IP Bootprom sometimes seems to set the value of `BOOTP-Gateway-IP` from the value in `BOOTP-Routers`, causing each TFTP ack packet to be sent to the router first. To avoid such behaviour, if your TFTP server is in the same subnet as the client, force `BOOTP-Gateway-IP` to `0.0.0.0` (thanks to Maciek Uhlig for having pointed out this problem).

Monitoring commands

This section lists commands for monitoring the system state. Optional arguments are listed between parenthesis (I would have prefered square brackets, but LaTeX do not like them at this place...)

Interact

Show the log and turn to interactive mode until QUIT or EXIT is entered. Type HideLog before quitting if you want to avoid disturbing log messages during batch execution.

Help (topic)

Load the on-line help file (`bpbatch.hlp`) and display the description of the given topic. If no topic is provided, or if the topic is unknown, display the help index.

Log "text"

Display the string on the log. No return/linefeed is implicitely added.

Echo "text"

Display the string on the log and go to the next line. Equivalent to

```
Log "text\r\n".
```

LogVars ("pattern")

Log (ie. display on the log) all variables matching the given pattern. The pattern can contain wildcards (? and *).

```
Example: LogVars "BOOTP-*"              list all BootP variables
```

LogDir "path/pattern"

Log (ie. display on the log) all files from the given path that match the pattern. The pattern can contain wildcards (? and *).

```
Example: LogDir "/usr/g*p"              list files names like g...p
```

LogTree "path"

Log the directory tree starting with the given path as root.

LogFile "filename"

Log the content of the file. The file must be no more than 64 KB big.

ShowLog

Make the log visible if it was hidden. Automatically performed when ESC is pressed with "$AutoShowLog" == "ON" and when entering interactive mode.

HideLog

Prevent log messages to appear on the screen. Default state when BpBatch, MrBatch and MrZip are started on a script file.

CaptureLog

Record all log output to a 64 KB buffer until EndCapture is issued. Wrap around buffer if the log output is more than 64 KB big. This command can be used to create a text file with an arbitrary content. The EndCapture MUST occurs within the same batch file.

EndCapture ("filename")

End up the capture of the log. If a filename is given, store the captured text to a file. Otherwise, discard it.

Beep

Make a sound. This command is equivalent to Echo "\007".

Control commands

This section lists commands that control the batch execution. Optional arguments are listed between parenthesis.

Include "filename"

Load the given file and start up the parser on it. Go back to the current point when the include file processing is done. The interpretation of characters above ASCII 127 within the include file depends on the value of $IsoLatin at the time the file is included.

OnExit *command*

Setup an exit-handler that will automatically be evaluated at the end of current batch file.

Goto *label*

Move the execution cursor to the given label (ie. the line starting with :*label*)

Eval "command"

Perform all substitutions on the "command" and run the parser on it.

If ...

```
If (not) <expr1> (==|!=|<|>|>=|<=|=>|=<|<>) <expr2> <command>
If (not) (ci) "str1" (==|!=|<|>|>=|<=|=>|=<|<>) "str2" <command>
If (not) (ci) "str1" Match-Expr "pattern" <command>
If (not) (ci) "str1" Match-Passwd "unix-passwd" <command>
If (not) (ci) "str1" in "wordlist" <command>
If (not) (ci) "str1" in-file "filename" <command>
If (not) exist "filename" <command>
If (not) valid <disk>:<partition> <command>
```

These commands execute *command;* if the test succeeds. The 1st form compares two numerical expressions. The 2nd form compares two strings, optionally case-insensitive. The 3rd form tests if "str1" matches the given pattern (wildcards allowed). The 4th form tests if the clear password "str1" matches the Unix-crypted password. The 5th form tests if "str1" is included in the word list. The 6th form tests if "str1" is included in the word file. The 7th form tests if the given file exists. The 8th form tests if the given partition is valid (i.e. formatted). This form is only supported by BpBatch versions after February 1999.

Set ...

```
Set variable = "string-value"
Set variable = <expr>
```

Setup a value for the given variable. If the given value is a numerical expresison, it will be implicitely converted to a string. A variable can be used anywhere by refering it as $variable or ${variable}. If the resulting reference is to be interpreted as a string, it should be enclosed between double quotes: "$variable" or "${variable}".

Delay *duration*

Waits until the specified duration (expressed in seconds) expired. See also the paragraph on the format of durations.

GetTime *variable*, GetDate *variable*

Get the CMOS time and store it into *variable*in the form HH:MM:SS. Get the CMOS date and store it into *variable*in the form YY/MM/DD. This can be used to customize the behavior of your boot scripts depending on the time of day or on the date.

SetTime "HH:MM:SS", SetDate "YY/MM/DD"

Set the computer CMOS time or date to the given value. If you have a security gateway (our special TFTP server) running, you can automatically adjust the CMOS time and date of the client computers at each boot by evaluating the following command:

```
include "$Security-Gateway:gettime"
```

If you want to understand what this command does, just type:

```
logfile "$Security-Gateway:gettime"
```

Poweroff

Turn off the computer. This command only works if the computer is Advanced Power Management (APM) compatible.

Keyboard-related commands

This section lists commands that let you monitor the keyboard input. Optional arguments are listed between parenthesis. See also under *National Language Support*.

GetKey (*variable*)

Indefinitely wait until a key is pressed and store it in the *variable*.

WaitForKey *duration* (*command*)

Wait until a key is pressed for no more than *duration* seconds. If no key has been pressed after the given time, evaluate the *command*. Otherwise, leave the key in the keyboard buffer. See also the paragraph on the format of durations.

Input (*variable* (*max-length*))

Read a return-terminated string from the keyboard and store the result string in *variable* (without the terminating return). If *max-length* is given, do not allow the user to enter more than this number of characters.

See also `GetPasswd` under *Security-related commands*.

OnKey "c" *command*

Setup a key handler that will automatically evaluate the given *command* when the key "c" is pressed (except is explicitely waited by a GetChar or an Input command). If the string `"default"` is used instead of a single character, the command is executed if any other key is pressed.

Text output commands

This section lists commands used to perform regular text output. All these commands can be used in graphic mode also, with the same behaviour (except that text mode provides 80x25 characters while graphic mode provides 100x37, because graphic mode characters are of size 8x16). Optional arguments are listed between parenthesis. See also under *National Language Support*.

Print *"text"/expr*

Print the specified string/expression at current cursor position and using current text attributes, then move the cursor. Add "\r\n" to the end of the string to go to the next line.

TextAttr *fg-color bg-color*

Setup the text attributes. One can also put a single numeric value representing both colors and defined as 16**bg-color+fg-color*.

If you need more fantasy, you can use `LoadFont`. See under *National Language Support*.

At *line,col* (*command*)

Move the cursor position to the specified position and evaluate the command if provided.

```
Example: At 10,20 Print "Gnats and rats !"
```

Clear (*color* (*pattern-char* (*top,left,bottom,right*)))

Fill the given text area with the given *pattern-char* (either a string or the decimal ascii code). The area defaults to the full screen, the pattern char defaults to the full block (ASCII dec 219) and the color defaults to black (clear screen). Move the cursor to the upper left corner of the cleared area.

BpMenu backward compatibility commands

```
.ATT (<attribute>)
.CLS (<attribute>)
.DEF <key> (<timeout_val>)
.KEY <key> <filename>
.POS ((<x>) <y>)
.PWD <key> <cpasswd>
.WLN (<text>)
.WRT <text>
```

See InCom's manual for more infos. We wrote some time ago a program program for editing menu files using this syntax, but it is preferable to make your menus using the new explicit syntax. Note that the .PWD command is not implemented because we do not now the password crypting algorithm used by InCom GmbH.

Graphics output commands

This section lists commands used to perform graphic-mode output. For the functions listed in this section, coordinates are given in pixels. You can also use all text output commands (see above) in graphic mode. Optional arguments are listed between parenthesis.

Note that the graphic mode is automatically turned on whenever a graphic command is used, unless the variable `VESA` is set to `"OFF"`.

InitGraph ("mode")

Turn on VESA graphics. The origin is on the upper-left corner of the screen (0 0). VESA graphics may hang

some computers under Windows 95. Run MrBatch with the -v option to avoid such problems.

You can request a specific video mode if you use the parameter "mode" This parameter is optional: if you do not specify any value, the video mode will be taken from the first field of the VESA-Modes variable.

Valid modes are :

- 640x480 => 640 by 480 pixels, 256 colors
- 800x600 => 800 by 600 pixels, 256 colors (default mode)
- 1024x768 => 1024 by 768 pixels, 256 colors
- 1280x1024 => 1280 by 1024 pixels, 256 colors

The VESA-Modes variable lists the video modes supported by your hardware.

Example: `InitGraph "640x480"`

CloseGraph

Close VESA graphic mode and go back to text mode.

DrawBar ***x-pos y-pos width height color***

VESA graphics. Draw a filled bar of the given size and colors.

DrawWindow ***x-pos y-pos width height*** **(*bg-color* (*bar-color*)) ("title" (*title-color*))**

VESA graphics. Draw a window of the given size and colors. The background color defaults to LightGray and the title-bar color defaults to Blue. If you include a title string and a color, this text will be displayed in the title bar.

Drawtext ***x-pos y-pos*** **"text" (*fg-color*)**

VESA graphics. Draw the text string at the given position with a transparent background. The color defaults to text foreground color.

DrawGif "gif-filename" (*x-pos y-pos* (*color-strategy*))

VESA graphics. Load the given GIF-87a file and draw it on the screen. The file can be interlaced, but must be in GIF-87a (not GIF-89a). The image size should fit in the selected video mode. You cannot load a 1024x768 GIF file when you selected a 640x480 mode. The GIF position defaults to the top left corner of the screen (0 0).

The *color-strategy* defines the allocation of colors in the palette when more than 256 colors are needed (for instance when two 256 colors GIF files are displayed simultaneously):

- `Best-Colors` use best possible colors for the most recent GIF
- `Spare-Colors` try to avoid allocating colors, change existing colors
- `Share-Colors` try to avoid allocating colors, use existing colors
- `Reuse-Colors` allocate no new color, only use existing colors

The default strategy is `Best-Colors`.

Security-related commands

This section lists commands that help you authenticate a user. Optional arguments are listed between parenthesis.

Some of these functions cooperate with a *Security gateway*, that you should first install. See the section on *Special TFTP servers* for more infos.

GetPasswd (*variable* (*max-length*))

Same as Input, but echo stars instead of the typed characters.

Crypt "text" "salt" *variable*

Apply the Unix crypt function to the given 8-chars text and store the resulting crypted string into *variable*. The "salt" is usually a two-character string that will be found as the first two characters of the crypted string.

Note that Unix crypt is a one-way function. It is not possible to decode the crypted string. One can only try to crypt another string with the same salt and compre the resulting crypted string.

DESCrypt "text" "key" *variable*

Crypt the given text using the given 8-chars key and store the result as an hexadecimal string in *variable*.

DESDecrypt "hexcode" "key" *variable*

Decrypt the given hexadecimal string using the given 8-chars key and store the result in *variable*.

MD5 "text" *variable*

Compute the MD5 checksum of the given text and store it as an hexadecimal string in *variable*. Can be used as an alternative to the Unix crypt function to check for passwords bigger than 8 characters.

CheckUser "user" "password" "domain"

Connect to the $Security-Gateway and check if the given user exist in the given radius domain and uses the specified password. If the domain is `"Unix"`, use the Unix user/password definition on the security gateway. For any other domain, use the security gateway domain definition file to determine the real Radius or NT domain to check.

Set the value of $Security-Check to "PASSED" or "FAILED". The password do not transit in clear on the network.

Disk-related commands

This section lists commands for preparing the hard-disk. Optional arguments are listed between parenthesis.

GetPartitions *variable* (*disk*)

Read the partition table(s) for the given disk and store it as a string into the given *variable*. The result string is a space-separated list of *Type:Size*, where

- *Type* is FAT16, EXT, BIGDOS, NTFS, FAT32, FAT32-LBA, BIGDOS-LBA, EXT-LBA, LINUX-SWAP, LINUX-EXT2 or the decimal filesystem id for unknown types.
- *Size* is the size of the partition in megabytes.

See SetPartitions for more informations about partitions.

SetPartitions "partitions" (*disk*)

Setup the partition table(s) to the content of the string. The format used is the same that for GetPartitions. This command also reset all boot flags (hint: use SetBootPart).

The main partition table in the master boot record (MBR) has only four entries. Moreover, DOS and Windows accept only ONE FAT partition (called the Primary partition, C:) in the main partition table. Any supplemental FAT partition should be nested in an extended partition (and is thus called a Logical partition). If we give numbers 1-4 to the partitions described in the MBR partition table and numbers 5-8 to the partitions described in the first extended partition, the definition of two FAT partitions would work by defining partition 1 as FAT, partition 2 as EXT and partition 5 as FAT. Partitions 3,4,6,7 and 8 should be marked as UNUSED. The same scheme can be used recursively to define more than two FAT partitions: nesting another extended partition in partition 6 and adding a logical FAT partition in partition 9.

In the most strict interpretation of DOS specifications, that means that entries 3 and 4 of the partition tables are never used. In practice, some versions of DOS and some other OS are able to use more than two partitions per

partition table, but there is no clear rule. On this side, BpBatch is rather flexible in its interpretation of partit tables, it can often understands things that OSes cannot.

One universal rule is that there should never be more than one extended partition per partition table, otherwise the partition numbering scheme breaks down.

If you want to try funny configurations, make your own experiments, but don't complain if the OS does not recognize your partitions. The only way it is guarantee to work is to use the primary partition to store the OS boot partition, and to nest all other partitions, one at a time, in extended partitions.

Example of extended partitions :

```
SetPartitions "BIGDOS:100 EXT:400 EMPTY EMPTY BIGDOS:400"
```

GetBootPart *variable* (*disk*)

Get the partition number with the boot flag turned on (DOS says: the activated primary partition) and store it to the *variable*. The first partition is numbered 1. If no partitions has the boot flag turned on, answers zero.

SetBootPart *partition* (*disk*)

Set the boot flag to the given partition. The boot flag let the master boot record (MBR) choose which partition to boot on. The first partition is numbered 1.

Blank *partition* (*disk*)

Fill the given partitions with zeroes. Can take quite a lot of time for big partitions. Do not format the partition for any operating system. See also `Clean`.

Clean *partitions* (*disk*) ("label")

Fast-format the given partition(s) according to the type declared in the partition table. If a label is given and the filesystem supports it, setup the partition label. For a paranoiac full format, call `Blank` on the partition first.

Clean is supported for (FAT16) BIGDOS, FAT32, EXT, LINUX-EXT2 and LINUX-SWAP partitions. To clean the master boot record (MBR), use `Clean 0`.

Clean should be used on data partitions and on MBR/EXT partitions. It is totally useless to clean a partition before unzipping a filesystem on it using `FullUnzip`.

FullUnzip "full-archive" *partition* (*disk*)

Decompress a full disk archive to the given partition, overwriting any existing file (clean-up on the fly).

FullUnzip is supported for (FAT16) BIGDOS, FAT32 and LINUX-EXT2.

This commands turn on VESA graphics to display a progress banner, unless `VESA` has been turned `OFF`.

IncrUnzip "incr-archive" "destpath"

Decompress an incremental disk archive to the given path. Files in the archive replace those with the same name on the target path, but other files are not deleted.

IncrUnzip is supported for (FAT16) BIGDOS, FAT32 and LINUX-EXT2. This command is far less efficient than FullUnzip since the existing filesystem structure must be preserved. However, it avoids multiplying the number of different disk images by storing the differences only.

FileUnzip "source-filename" "dest-filename"

Uncompress a file previously compressed with `MrZip` FileZip command. The file is validated by a 32-bits CRC.

Copy "source-filename" "dest-filename"

Copy the source file to the destinaton file, byte-to-byte. Can be used after a FullUnzip for instance to update

TOC

configuration files from the server without rebuilding the image. Better to use `FileUnzip` for big and easy-to-compress files.

Append "src-filename-1" "src-filename-2" "dest-filename"

Copy the first, then the second file to the destination file, byte-to-byte. Can be used on arbitrary large files. The destination file cannot be one of the two source files.

Patch "source-filename" "dest-filename" ("prefix" ("postfix"))

Read the source file and perform variable substitution before writing it to the destination file. The interpretation of characters above ASCII 127 depends on the value of $IsoLatin.

By default, variables are recognized when prefixed by "${" and postfixed by "}". This can be changed to any other non-empty string. remember that if you want to use a dollar sign within the prefix or suffix, you must escape it or it will get macro-evaluated. For instance, if you want to explicitely use the default prefix and postfix, use:

```
Patch "source-file" "dest-file" "\${" "}"
```

MkDir "path"

Recursively create directories from the root to the given full path. If the path already exists, this command has no effect.

Delete "filename", Del "filaname"

Remove the given file. The file must exist.

DelTree "path"

Recursively remove all files and directories under the given path, and remove the directory itself.

Boot commands

This section lists commands for continuing the boot process. Optional arguments are listed between parenthesis.

HideBootProm

Restore the memory and the interrupt vectors allocated by the bootprom. All attempts to make TFTP transfers will fail after calling this command. It is usually a good idea to call this command before HdBoot, or you might run short of memory under DOS/Windows. This command is implicitely called by FloppyBoot.

Note that although this function restore all vectors "officially" rerouted by the BootProm, it does not seems to restore everything. But it works well enough for DOS and Windows.

LoadRamDisk "ramdisk-filename"

Load a floppy disk image into the extended memory and redirect the BIOS Disk Services to make floppy disk calls use this image instead. This command implicitely calls `HideBootProm`. Call `FloppyBoot` to boot on the ramdisk you just loaded.

This kind of ramdisk may not be as robust as what you get when you use the TFTPBoot command. The only advantage is that it only steals a few hundred bytes of conventional memory instead of the >64 KB reserved by the TCP/IP BootPROM. Warning, nothing secures the extended memory in which the ramdisk resides. There is no way to uninstall such a ramdisk.

LoadZRamDisk "ramdisk-filename"

Do the same as `LoadRamDisk`, but for an image that has been compressed using `MrZip` FileZip command. Compressed ramdisks are protected against data corruption (and uncomplete download) by a byte count and a 32-bits CRC.

TOC

TFTPBoot "remote-bootfile"

Chain to another boot file (for instance a floppy image made with InCom's BpShell program). See the file referencing conventions for accessing a file on another TFTP server.

FloppyBoot

Hide the Boot ROM, load the floppy disk boot sector and boot on it.

HdBoot (*disk*)(:*partition*)

Load the given boot sector and boot from it. The disk default to zero, the first hard disk, and the partition defaults to zero, ie. the master boot record. You can boot from any partition, but be warned that Windows 95 may not let you boot a partition that has not been set as the boot partition (hint: use SetBootPart).

This command does not implicitely call HideBootProm, so you might want to call it before.

LinuxBoot "kernelfile" ("command-line" ("ramdisk-file"))

Load the given kernel and ramdisk into the high memory, setup the command line and boot the kernel. It is a good idea to put at least a minimal command line with the location of the root filesystem (like `"root=`dev/hda1"/). If you are using a linux system that heavily relies on `lilo` (like RedHat Linux 5.1), it may be necessary to add to the command line something like `BOOT_IMAGE=linux`. Note that the kernel can be loaded by TFTP (automatically cached on the hard disk) or directly from the target root partition.

This command works for small and big kernels (`zImage` and `bzImage`).

National language support

This section lists commands related not national language support. Optional arguments are listed between parenthesis.

RemapKeys "original-keys" "remapped-keys"

National keyboard support. Remap given keys to other characters. For instance, to swap the Y and Z keys, use

```
Remapkeys "yzYZ" "zyZY"
```

It is a good idea to use the quoted octal notation when using characters not included in the minimal ASCII character set, in order to avoid a dependency to the iso-latin modal settings.

For international keyboards, there are two keys that produce a backslash in non-remapped (US) mode. Each of them can be independantly remapped, thanks to the fact that `BpBatch` sees one of them as a key answering ASCII code 252 (octal) or ASCII code 335 (octal) when shifted.

If you send me a sample script that does keyboard mapping for your national keyboard, I will make it available under `http://cuiwww.unige.ch/info/pc/remote-boot/soft/sample-scripts` To help you make your own keyboard mapping, I suggest pressing all *special* keys without remapping the keyboard and writing down the character they produce. These will be the `original-keys`. The `remapped-keys` simply are the key you would have liked to see, in the same order. If some keys (either original or remapped) produce characters above ASCII dec 127, use the quoted octal notation. You can easily get the octal code for any given character by looking in the ASCII table of HelpPC for instance (HelpPC is a shareware hypertext on-line help program by David Jurgens).

RemapAltkeys "original-keys" "remapped-keys"

National keyboard support. Remap the given keys when ALT is depressed For instance, to map Alt-2 to the ampersand sign, use

```
RemapAltKeys "2" "@"
```

Note that dead keys are not supported.

LoadCodePage "cpxxx.bin"

Load and activate the given binary Codepage file. Codepages are used for the translation of Unicode characters (present on VFAT valumes for instance) into 8-bits characters. If you do not have the right Codepage loaded, you will get FAT warnings while accessing the filesystem when special characters are encountred.

All binary codepage files are available at
http://cuiwww.unige.ch/info/pc/remote-boot/soft/codepage.zip

The default codepage is 850, a reordered superset of ISO-Latin-1. If you load a more exotic codepage, you should usually turn the variable `$IsoLatin` to `"off"` or you might get meaningless implicit conversions. Moreover, if you want to display exotic characters, you should also load the proper screen font (use `"LoadFont"`).

LoadFont "fontfile"

Load and activate a VGA/VESA font, both in text and graphic mode. The font file must be a binary file of 16 bztes/characters (8x16 bitmap). This command can be used for National Language Support as well as for Fantasy support.

An archive with several fantasy fonts is available at
http://cuiwww.unige.ch/info/pc/remote-boot/soft/fonts.zip. This archive also contains a program to extract fonts for your codepage from the DOS `.CPI` file.

Commands specific to MrZip

Source...

```
Source (i)archive "filename"
Source path "path"
```

Set the source for the archive manipulation to the given (incremental) archive file or disk path.

Dest...

```
Dest (i)archive "filename"
Dest (i)dump
Dest path "path"
```

Set the destination for the archive manipulation to the given (incremental) archive file, dump or disk path. To control the quantity of data displayed during dump, use the $DumpFormat special variable.

FileZip "source-filename" "dest-filename"

Compress a file for further decompression with FileUnzip or for using as ZRamDisk. The file is validated by a 32-bits CRC.

Filter...

```
Filter -"pattern"
Filter +"pattern"
```

Avoid/allow files and directories matching the given pattern (wildcards allowed) to be included in the archive. The pattern is matched agains the full pathname. By default, all files are included in the image. You only need to explicitely allow files that where cancelled by a filter. Each negative filter has its own positive filter (allowed) sublist.

TOC

For DOS/Windows images, you will typically use

```
Filter -"*.swp"
Filter -"temp/*"
```

and for Unix images, you will typically use

```
Filter -"var/log/*"
Filter -"tmp/*"
```

CopyArchive

Start the archive manipulation operation, according to source, destination and filter settings. Except in a few circumstances, you will probably use the shortcut below instead of explicitely calling `CopyArchive`. One circumstance in which you will use `CopyArchive` explicitely is when you want to change the fragmentation of an image, as follow:

```
set FragmentSize="30 MB"
Source archive "original.imz"
Dest archive "refragmented.imz"
CopyArchive
```

FullZip "path" "full-archive"

Shortcut for

```
Source path "path"
Dest archive "full-archive"
CopyArchive
```

You should usually first setup filters.

IncrZip "path" "incr-archive"

Shortcut for

```
Source path "path"
Dest iarchive "incr-archive"
CopyArchive
```

FullDump "full-archive"

Shortcut for

```
Source archive "full-archive"
Dest dump
CopyArchive
```

IncrDump "incr-archive"

Shortcut for

```
Source iarchive "incr-archive"
Dest dump
CopyArchive
```

XCopy "srcpath" "dstpath"

Shortcut for

```
Source path "srcpath"
Dest path "dstpath"
CopyArchive
```

5.2 NoBreak.sys

`Nobreak.sys` is a very small (about 350 bytes only) driver that you include at the beginning of your `config.sys`. Its goal is to secure the boot process, until the user is logged in. DOS provides a setting for this (namely `BREAK=OFF`), but it is not drastic enough, and has almost no effect in the `autoexec.bat`. Our driver works by modifying the scan-code of the key pressed when a break is requested, directly at the BIOS level. This way, no program at all can receive a break until break is enabled again.

The driver must be loaded from the `config.sys` (or using the `devlod` program from *Undocumented DOS*). Afterwards, break can be enabled by sending `Yes` to the `NOBRK` pseudo-device, and disabled again by sending `No` (in fact, only the first character, `Y` or `N` is significant).

As this driver relies on the BIOS, it does only work for DOS and Windows 3.1. Windows 95 has its own low-level keyboard handling routines.

Assembler source code is available.

---

6. Special TFTP Servers

As the only network support available in the TCP/IP BootPROM is TFTP, there is a special interest in enhancing TFTP servers for providing new capabilities.

6.1 Incom Enhanced TFTP Server

InCom GmbH distributes with the TCP/IP BootPROM an enhanced TFTP server that can send packets of up to 1408 bytes instead of the standard 512 bytes. This is a great enhancement that you should use. This server is available on the TCP/IP Bootprom Utility disk for Solaris, Windows and as Netware NLM.

6.2 Linux Enhanced TFTP Server

We built a modified version of Linux TFTP server that acts as InCom enhanced TFTP server. Basically, we simply changed the packet size from 512 to 1408 bytes and the port from 69 to 59. It is available from `http://cuiwww.unige.ch/info/pc/remote-boot/soft/etdtpd.tar.gz`.

6.3 The Security Gateway

We wrote a special TFTP server that serves as security gateway for authenticating users. This server runs under Linux or Solaris, and can authenticate users according to a Unix password database (NIS and shadow passwords are supported), a Windows NT (or Samba) server or a Radius server. It is available from `http://cuiwww.unige.ch/info/pc/remote-boot/soft/stdtpd.tar.gz`, with source and precompiled binaries. The precompiled binaries do not include NT password encryption as we cannot distribute `libdes` but compilation is straightforward.

In order to use the security gateway, you just have to setup a trivial *security domains* configuration file that describes to which authentication server each logical security domains maps (the `Unix` domain implicitely maps to the server Unix password database). This is a sample configuration file:

```
#
# STFTPD configuration file
#
# This file specify the server of the "security domains". Two types of
# authentication servers are supported : radius or winnt (winnt includes
# NT Server and Samba)
#
# Format of radius servers
# radius        <domain>        <serveraddress>         <secret>
#
# secret is the secret word as specified in your /etc/raddb/clients file
#
# Format of SMB servers
# winnt         <domain>        <serveraddress>         <netbiosname>
#
# netbiosname is the NETBIOS name of your server
```

```
#
# Examples
radius          sec-dom-rad     radiusserver    testing123
winnt           sec-dom-nt1     192.168.1.1     NTSERVER1
winnt           sec-dom-smb     samba           SAMBA1
```

Note that if you are using Samba, you must set `security = user`.

You can also provide to the security server a file containing a list of users which are not allowed to log on (for which the check will fail anyways).

6.4 The Broadcast TFTP Server

We wrote a special TFTP server that implements a home-made Broadcast variant of TFTP. Using this server, we were able to download images to 25 clients on a heavily loaded 10 Mb ethernet network at 6 Mb/s (it is more efficient than the regular TFTP because it does not need to acknowledge each packets). This server runs under Linux or Solaris. It is available from `http://cuiwww.unige.ch/info/pc/remote-boot/soft/btdtpd.tar.gz`, with source and precompiled binaries.

As the TCP/IP bootprom does not support this protocol, our solution consist in booting a tiny ramdisk-based linux system using the tools described in this document, and running the Linux version of MrBatch which has built-in support for Broadcast TFTP. A simple batch file can the download all files to the cache in a few minutes, simultaneously on all client computers. You do not need to install Linux yourself to use this package, except if you have exotic hardware and cannot directly use the kernel provided in the package.

The process works as follow. First, you startup the broadcast server manually, giving the number of expected client computers as argument (remember, this procedure is not to be used every day but only when you changed an image and want to ensure it is immediately uploaded to all your client computers). Then, you turn on all client computers, which will run the following BpBatch script:

```
#
# This batch is run by bpbatch to launch a mini-linux using an initial
# ramdisk, which will then run mrbatch under linux.
#
# The broadcast TFTP protocol only works with the Linux implementation of
# mrbatch, because of the lack of broadcast support in the bootprom itself.
#
# 1. Setup a tiny partition, to let a lot of space for the cache
setpartitions "BIGDOS:50"
# 2. Clean the MBR
clean 0
# 3. Run a Linux Kernel with initrd (Initial Ramdisk) supprt, and use
#    bcastrd.gz as the initial ramdisk (will be mounted root and then
#    executed via /linuxrc). See initrd.txt for more details about
#    initial ramdisks. You don't have to specify a root device (second
#    parameter is null) to the kernel, it will use the initial ramdisk.
linuxboot "linux.krn" "" "bcastrd.gz"
# 4. The initial ramdisk will run dhcpcd to setup networking using DHCP.
#    It will then run mrbatch -w bcastlx
```

The initial ramdisk contains:

- `dhcpcd`, a DHCP client used to setup networking
- `mrbatch`
- `linuxrc`, a little wrapper automatically started by initrd and that starts `dhcpcd` then `mrbatch`.
- `usr/lib/terminfo/l/linux`, used by MrBatch
- `dev/*`, devices needed to run Linux and mrbatch

All programs are statically linked and stripped, to avoid `libc.so` which is really huge. The resulting ramdisk is Gzipped and takes less than 300 KB. The kernel itself takes 450 KB (with many network cards and initrd support). When Linux is up and running, MrBatch is called with the following script (that you should edit for your needs):

```
# This file is executed when mrbatch is launched by the initial ramdisk
# bcastrd.gz
# It's main purpose is to "broacast copy" files to the cache
#
# 1. Be verbose
showlog
# 2. Don't want a "press a key"
set pauselog="OFF"
# 3. Set partitions at their final values.
#    Important: Since you will copy files into the cache to be used in future
#    boot, you need to specify the same partitions as in the future boots.
setpartitions "BIGDOS:1024"
# 4. Clean the CACHE partition
clean -1
# 5. And the copy files into the cache, using the Broadcast TFTP protocol
#    (port 99)
#
# You can use the script "as is", but you surely need to modify the following
# line ! In our example, we download the file mblinux.imz, which is the image
# file for our installation of Linux.
copy "$BOOTP-Server-IP@99:mblinux.imz" "{:-1}mblinux.imz"
```

When the transfer is done, you can simply turn off all client computers and change their initial boot script to your favorite menu.

Remote X Apps mini-HOWTO

Vincent Zweije, zweije@xs4all.nl

v, 14 July 1998

This mini-HOWTO describes how to run remote X applications. That is, how to have an X program display on a different computer than the one it's running on. Or conversely: how to make an X program run on a different computer than the one you're sitting at. The focus of this mini-HOWTO is on security.

1. Introduction

2. Related Reading

3. The Scene

4. A Little Theory

5. Telling the Client

6. Telling the Server

- 6.1 Xhost
- 6.2 Xauth
- 6.3 Ssh

7. Troubleshooting

1. Introduction

This mini-HOWTO is a guide how to do remote X applications. It was written for several reasons.

1. Many questions have appeared on usenet on how to run a remote X application.
2. I see many, many hints of ``use `xhost +hostname`'' or even ```xhost +`'' to allow X connections. **This is ridiculously insecure**, and there are better methods.
3. I do not know of a simple document that describes the options you *do* have. Please inform me zweije@xs4all.nl if you know more.

This document has been written with unix-like systems in mind. If either your local or remote operating system are of another flavour, you may find here how things work. However, you will have to translate examples yourself to apply to your own system(s).

The most recent version of this document is always available on WWW at http://www.xs4all.nl/~zweije/xauth.html. It is also available as the Linux Remote X Apps mini-HOWTO at http://sunsite.unc.edu/LDP/HOWTO/mini/Remote-X-Apps. Linux (mini-)HOWTOs are available by http or ftp from sunsite.unc.edu.

This is version 0.5.1. No guarantees, only good intentions. I'm open to suggestions, ideas, additions, useful pointers, (typo) corrections, etc... I want this to remain a simple readable document, though, in the best-meant HOWTO style. Flames to `/dev/null`.

Contents last updated on 14 July 1998 by Vincent Zweije

2. Related Reading

A related document on WWW is ``What to do when Tk says that your display is insecure", http://ce-toolkit.crd.ge.com/tkxauth/. It was written by Kevin Kenny. It suggests a similar solution to X authentication to that in this document (xauth). However, Kevin aims more at using xdm to steer xauth for you.

The X System Window System Vol. 8 X ``Window System Administrator's Guide" from O'Reilly and Associates has also been brought to my attention as a good source of information. Unfortunately, I've not been able to check it out.

Yet another document much like the one you're reading now, titled ``Securing X Windows", is available at http://ciac.llnl.gov/ciac/documents/ciac2316.html.

Also check out usenet newsgroups, such as `comp.windows.x`, `comp.os.linux.x`, and `comp.os.linux.networking`.

3. The Scene

You're using two computers. You're using the X window system of the first to type to and look at. You're using the second to do some important graphical work. You want the second to show its output on the display of the first. The X window system makes this possible.

Of course, you need a network connection for this. Preferably a fast one; the X protocol is a network hog. But with a little patience and suitable protocol compression, you can even run applications over a modem. For X protocol compression, you might want to check out dxpc http://ccwf.cc.utexas.edu/~zvonler/dxpc/ or LBX http://www.ultranet.com/~pauld/faqs/LBX-HOWTO.html (also known as the LBX mini-HOWTO).

You must do two things to achieve all this:

1. Tell the local display (the server) to accept connections from the remote computer.
2. Tell the remote application (the client) to direct its output to your local display.

4. A Little Theory

The magic word is `DISPLAY`. In the X window system, a display consists (simplified) of a keyboard, a mouse and a screen. A display is managed by a server program, known as an X server. The server serves displaying capabilities to other programs that connect to it.

A display is indicated with a name, for instance:

- `DISPLAY=light.uni.verse:0`
- `DISPLAY=localhost:4`
- `DISPLAY=:0`

The display consists of a hostname (such as `light.uni.verse` and `localhost`), a colon (`:`), and a sequence number (such as `0` and `4`). The hostname of the display is the name of the computer where the X server runs. An omitted hostname means the local host. The sequence number is usually 0 -- it can be varied if there are multiple displays connected to one computer.

If you ever come across a display indication with an extra `.n` attached to it, that's the screen number. A display can actually have multiple screens. Usually there's only one screen though, with number `n=0`, so that's the default.

Other forms of `DISPLAY` exist, but the above will do for our purposes.

5. Telling the Client

The client program (for instance, your graphics application) knows which display to connect to by inspecting the `DISPLAY` environment variable. This setting can be overridden, though, by giving the client the command line argument `-display hostname:0` when it's started. Some examples may clarify things.

Our computer is known to the outside as light, and we're in domain uni.verse. If we're running a normal X server, the display is known as `light.uni.verse:0`. We want to run the drawing program xfig on a remote computer, called `dark.matt.er`, and display its output here on light.

Suppose you have already telnetted into the remote computer, `dark.matt.er`.

If you have csh running on the remote computer:

```
dark% setenv DISPLAY light.uni.verse:0
dark% xfig &
```

or alternatively:

```
dark% xfig -display light.uni.verse:0 &
```

If you have sh running on the remote computer:

```
dark$ DISPLAY=light.uni.verse:0
dark$ export DISPLAY
dark$ xfig &
```

or, alternatively:

```
dark$ DISPLAY=light.uni.verse:0 xfig &
```

or, of course, also:

```
dark$ xfig -display light.uni.verse:0 &
```

It seems that some versions of telnet automatically transport the `DISPLAY` variable to the remote host. If you have one of those, you're lucky, and you don't have to set it by hand. If not, most versions of telnet do transport the `TERM` environment variable; with some judicious hacking it is possible to piggyback the `DISPLAY` variable on to the `TERM` variable.

The idea with piggybacking is that you do some scripting to achieve the following: before telnetting, attach the value of `DISPLAY` to `TERM`. Then telnet out. At the remote end, in the applicable `.*shrc` file, read the value of `DISPLAY` from `TERM`.

6. Telling the Server

The server will not accept connections from just anywhere. You don't want everyone to be able to display windows on your screen. Or read what you type -- remember that your keyboard is part of your display!

Too few people seem to realise that allowing access to your display poses a security risk. Someone with access to your display can read and write your screens, read your keystrokes, and read your mouse actions.

Most servers know two ways of authenticating connections to it: the host list mechanism (xhost) and the magic cookie mechanism (xauth). Then there is ssh, the secure shell, that can forward X connections.

6.1 Xhost

Xhost allows access based on hostnames. The server maintains a list of hosts which are allowed to connect to it. It can also disable host checking entirely. Beware: this means no checks are done, so *every* host may connect!

You can control the server's host list with the xhost program. To use this mechanism in the previous example, do:

```
light$ xhost +dark.matt.er
```

This allows all connections from host `dark.matt.er`. As soon as your X client has made its connection and displays a window, for safety, revoke permissions for more connections with:

```
light$ xhost -dark.matt.er
```

You can disable host checking with:

```
light$ xhost +
```

This disables host access checking and thus allows *everyone* to connect. You should *never* do this on a network on which you don't trust *all* users (such as Internet). You can re-enable host checking with:

```
light$ xhost -
```

xhost - by itself does *not* remove all hosts from the access list (that would be quite useless - you wouldn't be able to connect from anywhere, not even your local host).

Xhost is a very insecure mechanism. It does not distinguish between different users on the remote host. Also, hostnames (addresses actually) can be spoofed. This is bad if you're on an untrusted network (for instance already with dialup PPP access to Internet).

6.2 Xauth

TOC

Xauth allows access to anyone who knows the right secret. Such a secret is called an authorization record, or a magic cookie. This authorization scheme is formally called MIT-MAGIC-COOKIE-1.

The cookies for different displays are stored together in `~/.Xauthority`. Your `~/.Xauthority` must be inaccessible for group/other users. The xauth program manages these cookies, hence the nickname xauth for the scheme.

On starting a session, the server reads a cookie from the file that is indicated by the `-auth` argument. After that, the server only allows connections from clients that know the same cookie. When the cookie in `~/.Xauthority` changes, *the server will not pick up the change*.

Newer servers can generate cookies on the fly for clients that ask for it. Cookies are still kept inside the server though; the don't end up in `~/.Xauthority` unless a client puts them there. According to David Wiggins:

> A further wrinkle was added in X11R6.3 that you may be interested in. Via the new SECURITY extension, the X server itself can generate and return new cookies on the fly. Furthermore, the cookies can be designated ``untrusted'' so that applications making connections with such cookies will be restricted in their operation. For example, they won't be able to steal keyboard/mouse input, or window contents, from other trusted clients. There is a new ``generate'' subcommand to xauth to make this facility at least possible to use, if not easy.

Xauth has a clear security advantage over xhost. You can limit access to specific users on specific computers. It does not suffer from spoofed addresses as xhost does. And if you want to, you can still use xhost next to it to allow connections.

Making the Cookie

If you want to use xauth, you must start the X server with the `-auth authfile` argument. If you use the startx script, that's the right place to do it. Create the authorization record as below in your startx script.

Excerpt from `/usr/X11R6/bin/startx`:

```
mcookie|sed -e 's/^/add :0 . /'|xauth -q
xinit -- -auth "$HOME/.Xauthority"
```

Mcookie is a tiny program in the util-linux package, primary site ftp://ftp.math.uio.no/pub/linux/. Alternatively, you can use md5sum to massage some random data (from, for instance, `/dev/urandom` or `ps -axl`) into cookie format:

```
dd if=/dev/urandom count=1|md5sum|sed -e 's/^/add :0 . /'|xauth -q
xinit -- -auth "$HOME/.Xauthority"
```

TOC

If you can't edit the startx script (because you aren't root), get your system administrator to set up sta properly, or let him set up xdm instead. If he can't or won't, you can make a `~/.xserverrc` script. you have this script, it is run by xinit instead of the real X server. Then you can start the real X server from this script with the proper arguments. To do so, have your `~/.xserverrc` use the magic cookie line above to create a cookie and then exec the real X server:

```
#!/bin/sh
mcookie|sed -e 's/^/add :0 . /'|xauth -q
exec /usr/X11R6/bin/X "$@" -auth "$HOME/.Xauthority"
```

If you use xdm to manage your X sessions, you can use xauth easily. Define the DisplayManager.authDir resource in `/etc/X11/xdm/xdm-config`. Xdm will pass the `-auth` argument to the X server when it starts. When you then log in under xdm, xdm puts the cookie in your `~/.Xauthority` for you. See xdm(1) for more information. For instance, my `/etc/X11/xdm/xdm-config` has the following line in it:

```
DisplayManager.authDir: /var/lib/xdm
```

Transporting the Cookie

Now that you have started your X session on the server host `light.uni.verse` and have your cookie in `~/.Xauthority`, you will have to transfer the cookie to the client host, `dark.matt.er`.

The easiest is when your home directories on light and dark are shared. The `~/.Xauthority` files are the same, so the cookie is transported instantaneously. However, there's a catch: when you put a cookie for `:0` in `~/.Xauthority`, dark will think it's a cookie for itself instead of for light. You must use an explicit host name when you create the cookie; you can't leave it out. You can install the same cookie for both `:0` and `light:0` with:

```
#!/bin/sh
cookie=`mcookie`
xauth add :0 . $cookie
xauth add "$HOST:0" . $cookie
exec /usr/X11R6/bin/X "$@" -auth "$HOME/.Xauthority"
```

If the home directories aren't shared, you can transport the cookie by means of rsh, the remote shell:

```
light$ xauth nlist :0 | rsh dark.matt.er xauth nmerge -
```

1. Extract the cookie from your local `~/.Xauthority` (`xauth nlist :0`).
2. Transfer it to dark.matt.er (`| rsh dark.matt.er`).
3. Put it in the `~/.Xauthority` there (`xauth nmerge -`).

It's possible that rsh doesn't work for you. Besides that, rsh also has a security drawback (spoofed host names again, if I remember correctly). If you can't or don't want to use rsh, you can also transfer the cookie manually, like:

```
light$ echo $DISPLAY
:0
light$ xauth list $DISPLAY
light/unix:0 MIT-MAGIC-COOKIE-1 076aaecfd370fd2af6bb9f5550b26926
light$ rlogin dark.matt.er
Password:
dark% setenv DISPLAY light.uni.verse:0
dark% xauth add $DISPLAY . 076aaecfd370fd2af6bb9f5550b26926
dark% xfig &
[15332]
dark% logout
light$
```

See also rsh(1) and xauth(1x) for more information.

It may be possible to piggyback the cookie on the `TERM` or `DISPLAY` variable when you do a telnet to the remote host. This would go the same way as piggybacking the `DISPLAY` variable on the `TERM` variable. See section 5: Telling the Client. You're on own here from my point of view, but I'm interested if anyone can confirm or deny this.

Using the Cookie

An X application on dark.matt.er, such as xfig above, will automatically look in `~/.Xauthority` there for the cookie to authenticate itself with.

6.3 Ssh

Authority records are transmitted with no encryption. If you're even worried someone might snoop on your connections, use ssh, the secure shell. It will do X forwarding over encrypted connections. And besides, it's great in other ways too. It's a good structural improvement to your system. Just visit http://www.cs.hut.fi/ssh/, the ssh home page.

Who knows anything else on authentication schemes or encrypting X connections? Maybe kerberos?

---

7. Troubleshooting

The first time you try to run a remote X application, it usually does not work. Here are a few common error messages, their probable causes, and solutions to help you on your way.

```
xterm Xt error: Can't open display:
```

There is no `DISPLAY` variable in the environment, and you didn't tell the application with the `-display` flag either. The application assumes the empty string, but that is syntactically invalid. To solve this, be sure that you set the `DISPLAY` variable correctly in the environment (with `setenv` or `export` depending on your shell).

```
_X11TransSocketINETConnect: Can't connect: errno = 101
xterm Xt error: Can't open display: love.dial.xs4all.nl:0
```

Errno 101 is ``Network is unreachable''. The application could not make a network connection to the server. Check that you have the correct `DISPLAY` set, and that the server machine is reachable from your client (it should be, after all you're probably logged in to the server and telnetting to the client).

```
_X11TransSocketINETConnect: Can't connect: errno = 111
xterm Xt error: Can't open display: love.dial.xs4all.nl:0
```

Errno 111 is ``Connection refused''. The server machine you're trying to connect to is reachable, but the indicated server does not exist there. Check that you are using the right host name and the right display number.

```
Xlib: connection to ":0.0" refused by server
Xlib: Client is not authorized to connect to Server
xterm Xt error: Can't open display: love.dial.xs4all.nl:0.0
```

The client could make a connection to the server, but the server does not allow the client to use it (not authorized). Make sure that you have transported the correct magic cookie to the client, and that it has not expired (the server uses a new cookie when a new session starts).

SLIP/PPP Emulator HOWTO

TOC

version 3.0 (updated 8/7/97) by Irish

This document will describe how to get your Linux box to connect to a generic site via a SLIP/PPP emulator, such as SLiRP or TIA. I take no responsibility for your use of this information, but corrections are welcome.

I will not attempt to describe all that can be done over this connection once established, but I will refer you to other documents or sites that will contain the relevant information.

That done, let's do it!

- 1.0 How does an emulator work?

  It is important to understand how a SLIP/PPP emulator works in general, to be aware of its limitations. First, you do not need to install the emulator on your machine. It runs on your remote host only. What you need on your machine is the TCP/IP and SLIP or PPP protocols installed in your kernel (more on that in a bit), and some clients.

  Here's what happens: you send network requests from your machine to your remote host over the SLIP/PPP link. The emulator grabs them and sends them out to the Net at large. Then, incoming data is sent back from the Net to your account on the remote host, where the emulator grabs it and sends it back over the SLIP/PPP link to your machine. So, to the Net it appears as if you are working out of your account on the remote host, but to you it looks like you are really connected right to the Net.

  As you can see, this can confuse stuff that is incoming from the Net. For example, talk doesn't work via an emulator, because the incoming talk request tries to start the remote hosts talk daemon, not yours.

  The other big difference between an emulator and real SLIP/PPP is you are NOT assigned your own IP address; remember, you are only converting a dialup account to a SLIP/PPP connection.
- 1.1 What is TIA?

  The Intenet Adaptor was written by the fine folks at marketplace.com. They have quit working on it now, and consider it a 'mature' product. The last version is 2.05; if you are going to use TIA, please get this version. !NOTE! At last report, only site licenses are available now.
- 1.2 What is SLiRP? This is a freeware application, covered under the GNU Public License. It is out of beta, and is real stable. It sports a few more features than TIA, and is (reportedly) easier on host resources. Available at fine Linux FTP sites everywhere.
- 1.3 What about Term?

  If you don't have root access to your machine for one reason or another, and you can't persuade someone who does to install SLIP and dip, then you won't have much choice except to use Term. If you want to know more about Term, please read the HOWTO on Sunsite.
- 1.4 What about real SLIP/PPP?

  Hey, a lot of places are offering real SLIP/PPP at reasonable prices for non-dedicated dialup nowadays. If you can afford it, or feel the need, go for it. The Real Thing is always better than emulation. However, one of the things an emulator provides is security; it acts as an impregnable firewall, and with it you are pretty much as bulletproof as your provider is. You'll pay for it though, emulation is slower, and does not fully support every protocol. Still and all, emulation is good enough for most people, and is a fine way to connect to the Net.

Enough already! How do I set it up?

Alright hotrod, we're getting there, but first there's some info you need to get, and a couple of explanations.

First, you need to get the IP address of your provider, if you don't have it already. You can do this by typing 'who' at a shell

TOC

prompt there, or reading their /etc/hosts, or even just asking them. While your getting that, get the IP address of the POP/SMTP host and the NNTP server (if you don't know what those are, trust me and ask for them anyway, you'll need 'em).

Next, let me explain quickly about hostnames and domain names. The hostname is the first part of a machines name, the domain is the rest of it. Like this: hostname.domain.name (a four parter would be hostname.do.main.name). Together, they make a Fully Qualified Domain Name (FQDN).

The name of your Linux machine is stored in /etc/HOSTNAME, and yes, you can change it. In fact, please do (if you're using Slackware, it'll be "darkstar.frop.org" by default). Just edit the /etc/HOSTNAME file to be whatever you want - remember, if you are using an emulator, you are essentially invisible to the Net at large, so you can be creative. However, if you plan to get a real SLIP/PPP connection down the road sometime, you'll probably want to conform to being a sub-host of your provider. For example, if your provider is "info.superhiway.com", the domain is "superhiway.com" and the hostname is "info". So you could either be "offramp.info.superhiway.com" or just "offramp.superhiway.com". Either would work, and in either case your hostname is "offramp".

- 2.0 Preparing your Linux machine

  In a nutshell, here's what's required. I'll explain the first three in detail.

  - -> Compile your kernel to include SLIP and TCP/IP
  - -> Edit some files in /etc
  - -> Install and configure dip
  - -> Install pppd (optional, for PPP only)

  You will also need to get an emulator binary appropriate for your remote host and install it. SLiRP is available at Sunsite and other fine Linux FTP sites. All of the relevant info for TIA is available at TIAs home site.

That's it! Let's get started.

- 2.1 Compiling your kernel

  If you have never done this, you should. And you should read the FAQ first, but don't worry, it's easy. If you want, you can e-mail me and I'll help.

  I'll assume that you have here, for the sake of brevity. When you 'make config', look for "Network Devices". Say 'y', of course, then say 'y' to SLIP, CSLIP, (or PPP) and TCP/IP and 'n' to everything else, unless you have ethercards or need some other protocol for something else. **If you don't do this, it won't work!** Finish compiling and installing the new kernel, then,

- 2.2 Edit some files in /etc

  These files will set up your routes to your remote host. /etc is the directory for system configurations. Replace everything in double quotes with the appropriate values, naturally (but don't include the quotes - they are there for reference). There are three files you need to edit, they are:

  - /etc/hosts:

```
#The next line is required to be EXACTLY as below.
127.0.0.1       localhost

#Note at the end of the next two entries the hostname is repeated.
#This abbreviation is an alias, and is required for the first entry.
#If you're on a LAN, you'll need to substitute your IP address for
#the first one below.
192.0.2.1       "yourhostname.domain.name  yourhostname"
"XXX.XXX.XX.XX" "remotehostname.domain.name  remotehostname"
```

  - /etc/host.conf:

```
order hosts, bind
multi on
```

- /etc/resolv.conf:

```
domain "yourdomain.name"
#The next line usually uses the same remote IP address that's in /etc/hosts
nameserver "XXX.XXX.XX.XX"
```

- To use an NNTPserver, put this line in your /etc/profile:

  export NNTPSERVER="remote.hosts.nntpservername"

- 2.3.1 DIP

DIP (Dialup Internet Protocol) is what you will use to dial up the remote host, start the emulator, and convert the line to SLIP/PPP. It comes in the "N" set of Slackware, along with a bunch of clients and utilities, some of which you may want to install also =). It is also available at Sunsite in an individual tar file

Once you have it installed, you will need to have a dip script, Here's a sample, just plug in the appropriate stuff where the double quotes are (but don't include the quotes, they are there for reference).

----------CUT HERE----------

```
main:
  get $local "yourhostname.domain.name"
  get $remote "remotehostname.domain.name"

#Your port here
  port cua"?"

#Use 115200 for 28.8 modems
  speed 57400
  reset

#don't use spaces in your AT command string!
  init AT "string of commands"
  wait OK 5

# This will redial. If it doesn't work, play with the wait time (listen
# to your modem). If it still doesn't work, mail me.
# See also; the note at the bottom of the script re: error codes.

dial:
  dial "phonenumber"
  print Dialing...
  if $errlvl != 0 goto error

#You may need to change this wait time to suit your modem
  wait BUSY 20
  if $errlvl == 0 goto dial

login:
  print Connected and Logging in...
```

TOC

```
#This wait and send gets me past my hosts Annex. Change for your site!
  wait ==> 60
  send 4\n
  wait ogin: 60
  if $errlvl != 0 goto login_error1
  send "LOGIN"\n
  wait assword: 60
  if $errlvl != 0 goto login_error2
  send "PASSWORD"\n

loggedin:
  wait "SYSTEM PROMPT" 60
  if $errlvl != 0 goto shell_error
  print Logged in!
  send "emulator startup command"\n
  wait "emulator startup response" 60
  if $errlvl != 0 goto emu_error

#The recommended mtu setting is 1500, but this is faster interactively.
#Ftp may be slower, so adjust to taste.
  get $mtu 296
  default

done:
  print CONNECTED to $remote with address $rmtip
  mode CSLIP
  goto exit

error:
  print Dialing Error

login_error1:
  print No Login

login_error2:
  print No Password prompt

shell_error:
  print No shell prompt

emu_error:
  There was a problem starting the emulator

exit:

#This will error out when the -v flag is used, but work when run normally
#(ends dip at a local prompt).
  \r
```

----------CUT HERE----------

- 2.3.2 Dip notes Newer versions of dip don't return modem status codes (BUSY, NO CONNECT, etc.), they use numbers instead. Here's a table:
  - 0 = OK
  - 1 = CONNECT

TOC

- 2 = ERROR
- 3 = BUSY
- 4 = NO CARRIER

This would make your dial section look like this:

```
dial:
  dial "phonenumber"
  print Dialing...
  if $errlvl != 0 goto error
  wait 1 20
  if $errlvl != 1 goto dial
```

Thanks to Lee Olds (lee@eskimo.com) for that bit.

\n = newline, \r = carriage return. You may require one or the other (or both) in the appropriate places. If the ones provided don't work, experiment.

A word about the form of the wait and sends. If it isn't clear by the example, here's what's happening:

```
  wait ogin: 60
        (the script will wait for the remote to send 'ogin:' for 60
         seconds. When it gets it, the script advances. If it doesn't get
         it, it will advance after the timeout of 60 seconds.)
  if $errlvl != 0 goto login_error1 (this whole line is optional)
        (if the script is advancing because it got what it was waiting for,
         this gets skipped. if the script is advancing because it timed out,
o        it will go to where you tell it to ('login_error1' in this case)
  send "LOGIN"\n
        (DIP will send 'LOGIN' and a newline)
```

You can put in sleep statements if you need to, like this: "sleep 10" This will make the script pause for 10 seconds.

After editing this file, rename it, say, remote.dip and put it in /root. Then, as root, run 'dip remote'. Use the -v flag the first time to debug it ('dip -v remote'), this will show you all the steps dip takes.

Dip will only run as root, but there is a way to make it run from a user account. For now, if you need this info, ask. I may include it in this file later if enough people want it.

If dip errors out right away, try removing the comments from the script.

- 2.4 PPP

The above gets you rolling with CSLIP, which is really fine most of the time. It truly is easy to get working, since the protocol is supported at the kernel level - just make sure it's in there and it works.

However, some of you will want PPP, for what reasons, only you can say. I'm not going to tell you how to set it up here, there's a whole other HOWTO written just for that. What I will do is tell you what the gotchas are when using PPP with an emulator.

First, the latest version of DIP says you can use it to start PPP, and you can, BUT it only starts the PPP daemon - no flags, nothing. So if you want to use DIP to start PPP, be sure to put all of your startup info in the /etc/ppp/options file, or it won't work.

Speaking of the options file, one of the things that MUST be in there no matter how you start pppd is this: '192.0.2.1:XXX.XXX.XX.XX'. What this is is 'localIPaddress:remoteIPaddress'. You need it there because normally pppd can fill in the blank itself, but fails when connecting to an emulator.

Be sure to get the latest and greatest pppd package. It seems there were a few versions recently that had a bit of

trouble.

Lastly, unless you have a good reason to use PPP, or just want to learn how it works, you really don't need it. I have tried them both, and didn't notice any performance difference with the standard set of clients. Of course, Your Mileage May Vary. :)

- 3.0 Now what?

  Hey, if all went well, you are now connected to the Net! Try a 'telnet remotehostname' (remember that abbreviation in your hosts file?). You should get the remotes telnet login prompt. Go on, try it! You can now telnet/FTP/etc. anywhere on the Net from your local machine. How? Your machine is using the remote as a nameserver to resolve addresses (resolv.conf). Experiment, find out what works and what doesn't.

  You will not be able to use services that are not provided already on your remote host! For example, if your server doesn't allow telnets, chances are you can't do it either. Why? Because the emulator is only re-directing what is already available.

- 3.1 Mail

  The obvious text based solution is Pine, you can get it from The University of Washington.

  My pick for 'Best Mail Reader, X Based' has to go to XFmail. This has now gotten out of beta, and rocks. Easy to install; requires no local mail system be installed (but will work with local spools) because it has POP built in - no more popclient/fetchpop! Easy to use; totally customizable from the GUI. Also supports MIME. Get it from The XFmail Homesite.

- 3.2 News

  The text based News Reader SLRN is fairly simple to install and use. It is a true NNTP newsreader, which means *no local news transport is required* to make it work (you don't have install Cnews or INN). It does require that you have access to an NNTPserver, naturally, but most ISPs have this. It is text based, but has color and mouse support, and runs fine in an rxvt window.

  The X based News Reader KNews is almost perfect. As with SLRN above, no local transport is required, but an NNTP server is. It's at least worth a look.

- 3.3 Mosaic/Netscape

  Obviously, you will need to have X running for this to work, but it's a simple matter of FTPing the binary, unpacking it, and installing it.

- 4.0 Misc.

  If I've left anything out, or you still have questions, I read my mail daily. This information gets updated fairly regularly, so keep checking back every so often (I'll change the version numbers). Naturally, the more you contribute, the more info will be included here, so don't hesitate to tell me about whatever you have to offer.

  If you would like some help, or are having trouble with your setup, I'll help *but you must send me the following info:* 1. A copy of your three /etc files. 2. A copy of your script output, run with the -v flag (**please** remove your password from this!)

  Even if you don't have anything to contribute, but used this info sucsessfully, PLEASE MAIL ME. I want to know how useful this really is.

- 4.1 References

  Sunsite is the FTP site sunsite.unc.edu, a veritable plethora of everything Linux, including almost every FAQ, HOWTO, and README written.

  Marketplace.com is the home of TIA, the only place it is available.

  I am Irish, irish@eskimo.com, available for comment on Linux almost everyday, when I'm not petting my cat or

TOC

hugging my kid.

This HOWTO is available in the following places:

- http://www.eskimo.com/~irish
- ftp://ftp.eskimo.com/u/i/irish
- ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/mini/TIA

sendmail address rewriting mini-HOWTO

Thomas Roessler, roessler@guug.de

v0.0, 6 May 1998

This document is a brief description of how to set up sendmail's configuration file for the home user's dial-up access.

1. Introduction

2. File Roadmap

3. Configuring sendmail

- 3.1 The main configuration file
- 3.2 Address rewriting
- 3.3 Aliases

4. Further reading

1. Introduction

We assume that you have the kind of Internet access which seems to be most common at universities and online services nowadays: You dial into your provider's network using PPP over a serial connection. Your incoming mail is spooled at the provider's POP or IMAP server, while outgoing messages are to be sent via SMTP. You don't have a domain name of your own, so everything has to use *one* address.

We assume that you have already installed a fairly recent version of Eric Allman's sendmail (version 8.8.8 is current at the time of this writing and should work fine).

This document is partially referring to specific properties of Debian GNU/Linux systems; users of different distributions will have to take some care.

Make sure you have the following information at hand:

- Your ISP's mail server
- Your Internet mail address

The configuration we are planning has two main goals:

1. Sending mail between various local users must be possible.
2. The outside world must see the local users' ISP mail addresses, not the local ones.

To achieve this, we will make use of sendmail's `genericstable` feature.

2. File Roadmap

We will put all of sendmail's configuration files in a separate directory under `/etc`: `/etc/mail`. Usually, sendmail will expect these files to reside directly under `/etc`. To avoid problems, `/etc/sendmail.cf` should be a symbolic link to `/etc/mail/sendmail.cf`.

The following files will populate `/etc/mail`:

- =20
- `aliases` - contains additional local addresses
- `genericsdomain` - contains some information on your local host's configuration
- `genericstable` - contains the actual rewriting rules.
- `sendmail.cf` - sendmail's configuration file
- `sendmail.mc` - the source of `sendmail.cf`.

Some of these files will be accompanied by `.db` files. They contain hashed databases for sendmail's direct use.

We assume that the `cf` part of sendmail's source tree resides under a directory named `/usr/lib/sendmail.cf`. This is the case on Debian GNU/Linux systems. Other distributions will put this stuff at different places. Please refer to your distribution's documentation for details.

3. Configuring sendmail

3.1 The main configuration file

Sendmail uses a highly complex rule system for it's configuration. While you can do lots of neat tricks with this stuff, writing a `sendmail.cf` file from scratch is rather unusual and time-consuming. If you are interested in doing so, you should stop reading this document right now and instead read the "Bat Book" from O'Reilly.

Instead of hand-crafting these rules, we will rely on the `m4` macro processor to put together our configuration file from ready-made pieces which are distributed together with sendmail.

Let's look at the first lines of the `sendmail.mc` file:

```
include(/usr/lib/sendmail.cf/m4/cf.m4)
VERSIONID(`sendmail.mc - roessler@guug.de')
OSTYPE(debian)
define(`ALIAS_FILE',`/etc/mail/aliases')
```

In the beginning, `cf.m4` is included. This m4 macro file contains lots of macro definitions for the rest of the file. Be sure that the path you give here is correct - the one we are representing in our example is typical for Debian GNU/Linux. The `OSTYPE` macro is used to give some useful defaults for certain configuration values. If you aren't using a Debian system, you should replace the word "debian" by "linux" here. `ALIAS_FILE` tells sendmail where to look for the list of aliases.

The following lines tell sendmail to use the `genericstable` feature, and where to find the configuration files needed to use it:

```
FEATURE(masquerade_envelope) FEATURE(genericstable, `hash
-o /etc/mail/genericstable')
GENERICS_DOMAIN_FILE(`/etc/mail/genericsdomain')
```

The `masquerade_envelope` feature tells sendmail to apply header rewriting to the *envelope* sender of a message. This is the mail address to which external mail delivery subsystems will direct their delivery failure reports and warning messages. The `generics*` files will be explained below.

Now, we have to define a so-called smart host, that is, a machine which will handle outgoing mail for your system. Note that this machine may be different from your ISP's POP and IMAP servers. If in doubt, contact the hotline. The code in the master configuration file:

```
define(`SMART_HOST',`mail-out.your.provider')
```

Please replace *mail-out.your.provider* by the fully qualified hostname of your internet service provider.

The final two lines include the "mailer" definitions which are needed by sendmail to find out how to handle various types of mail:

```
MAILER(local)
MAILER(smtp)
```

To generate the `sendmail.cf` file from this `sendmail.mc`, type the following commands (as root):

```
# m4 sendmail.mc > _sendmail.cf
# mv -f _sendmail.cf sendmail.cf
```

Note the technique of writing `m4`'s output to a temporary file which is thereafter moved to the proper place. This helps us to prevent sendmail from reading partially written configuration files.

3.2 Address rewriting

First, we have to tell sendmail what addresses are to be considered local (and thus should be subjected to the rewriting). This is quite simple: Just put the fully qualified host name of your machine into the file `/etc/mail/genericsdomain`. To get your host's fully qualified name, type the following command:

```
 $ hostname -f
```

Now, let's come to the rewriting table proper: `/etc/mail/genericstable`. This file consists of two white-space separated columns. The first column contains the local address, the second column contains the e-mail address which should be used instead. The file may look like this:

```
harry   harryx@your.isp
maude   maudey@her.isp
root    fredx@your.isp
news    fredx@your.isp
```

Note that there should be one entry for *each* account on the local machine, so that automatically generated mail which leaks out of the local system carries correct header information.

For performance reasons, sendmail won't use this text file directly, but rely on a "hashed" version instead. To generate it, type the following command:

TOC

```
# makemap -r hash genericstable.db < genericstable
```

Note that the rewriting rules from the `genericstable` will *not* apply to local mail or to messages you receive from outside - the mapping is only used if a message leaves your local system for your ISP's smart host.

3.3 Aliases

The aliases file contains additional local names which are only valid for local messages. This is useful for administrative accounts like `root` which receive automatically generated messages from your system.

A reasonable start for `/etc/mail/aliases` could look like the following file:

```
root: fred
news: root
postmaster: root
mail: root
www: root

nobody: &sol;dev&sol;null
MAILER-DAEMON: nobody
```

This example will forward local mail for the `root`, `news`, `postmaster`, `mail`, and `www` users to `fred`, while messages for `nobody` and `MAILER-DAEMON` will be redirected to `/dev/null`.

Just like the `genericstable`, `aliases` may contain *lots* of entries. Thus, it would once again be inefficient for sendmail to use the text file we just described. The same mechanism as with `genericstable` is used for `aliases`: A hashed database is generated. Instead of using `makemap` directly, you can type in the command `newaliases` this time. It will automatically take care of all what's needed.

4. Further reading

The sendmail source distribution includes quite a bit of documentation. Read it, especially the file `cf/README`.

If you are interested to dive deeper into sendmail's configuration options, you want to get the "Bat Book" from O'Reilly: Bryan Costales, Eric Allman, and Neil Rickert: "sendmail". O'Reilly, 1993.

Sendmail + UUCP HOWTO

Author: Jamal Hadi Salim (jamal@glcom.com or hadi@cyberus.ca)

History:

- Initial edition July/96: works only with non-bind edition of sendmail
- Oct 01 /96: works with the redhat supplied binary
- Oct 07 /96: added/modified a tip on how to compile non-bind sendmail
- Oct 25 /96: added/modified a tip on how to keep sendmail getHostbyAddr() happy via a dummy /etc/hosts entry
- Mar 15 /97: I am amazed at the response/feedback i am receiving so i decided to refine this further. Fixed some small errors; added diagram + references.
- Oct 25/97: Htmlise+ add blurb on DNS capable machines

After finally settling down and properly reading TFM (the Bat Book, to be precise) i have solved the Caramilk secret! The steps below have been tested with the binary supplied by RedHat. Perhaps i should send this tip to the maintainer of the Sendmail FAQ too since there is absolutely no mention of UUCP interworking with sendmail there.

What this Document tries to show you

How to setup a single machine, reffered to as *me.com* in the example, with no direct access to the internet to route mail for you via sendmail and UUCP to a 'smarter host' eg your ISP.
Optionally you can configure this machine to feed other machines i.e you are their 'smart' host or gateway.

What this Document does not tell you

I have been receiving incredible number of questions regarding topics which are beyond the focus of this HOWTO. The purpose of this document is not to describe them so please try not send me any questions on the following:

a) How to setup UUCP.

Refer to many fine references on this, including:

- UUCP HOWTO (Vince Skahan,)
- Using and managing UUCP (Ed Ravin et al -- publisher O'reilly) which i think is the best authority on

UUCP

- Linux Network Administrator Guide (Olaf Kirch)

TOC

b) how to setup DNS.

Refer to the many fine references on this, including:

- Caching named mini howto (Nicolai Langfeldt)
- The bat book (Sendmail;Costalles, Allman, Rickert;publisher O'reilly)
- DNS and BIND (Publisher O'reilly)
- TCP/IP Network Administration (Hunt, Craig;Publisher O'reilly)
- Linux Network Administrator Guide (Olaf Kirch)
- sendmail Theory and Practice (Avolio and Vixie; publisher Digital press)

c) how to setup databases (other than the mailertable) to work with sendmail.

Refer to the many fine references on this, including:

- The bat book (Sendmail;Costalles, Allman, Rickert -- publisher O'reilly)
- TCP/IP administration (Publisher O'reilly)
- Sendmail Theory and Practice (Avolio and Vixie; publisher Digital press)
- Docs bundled with sendmail d) how to write or modify rulesets or mailers.
- All the above references in c)

You can also find help on the following **newsgroups**:
comp.mail.sendmail
comp.mail.uucp
comp.os.linux.networking
IRC channels
#linux, #unix

THE SETUP

This is for a system, hypothecally named *me.com*, whose mail is sent out only via UUCP. *mysmarthost* is the ISPs hostname as set up in the **uucp sys file** and *me.com* is the hostname we use or are known by in the MX records on the internet.
mysmarthost hooks to the internet. We really dont care how it does it, we just know it knows how to get us there.
me.com, whose setup we describe in the example, feeds sites *down.com* and *system1.org* and their

subnodes. *me.com* connects to *down.com* via TCP/IP and connects to *system1.org* via raw UUCP usi
phone line.
The setup description is shown below:

```
        ------------------------
        |                      |
        |     Internet         |
        |                      |
        -----------------------
                    |
                    | PPP/dedicated line running TCP-IP
                    |
                -------------
                |            |
                |mysmarthost | Authoritative name server for
                |            | *.me.com, *.down.com, *.system1.org
                |            |
                --------------
                    |
                    | UUCP via a phone line receive for
                    | system1.org, *.system1.org,*.down.com, down.com
                    | as well as me.com and *.me.com
                 ------------
                | * *****  *  |
   ------------ | * me.com *  | ------
                | *        *  |      |
   |            | * *****  *  |      |
   |             ------------        |
   |UUCP via phone line              | uucp via TCP/IP
   |                                 |
 -----------                  -----------
|           |                |           |
|system1.org|                | down.com  |-------
|           |                |           |      |
 ----------                  ------------       |
     |                                          |
     |                                          |
     |                                          | LAN: smtp to nodes
     | UUCP phone line
     |                                          |
-----------                              -------------
|         |                              |           |
|system1's|                              | down's    |
|subnodes |                              | LAN       |
|         |                              ------------
```

pre-requisites

1) If you are already running DNS or have a pointer to a DNS server (in your /etc/resolv.conf) and are on a live connection to the net then skip this part. You are safe. You dont have to do any 'smart' thing skip to **2)**
If you are on a machine using uucp for mail then you dont really need to compile bind/resolv into sendmail;

How to survive name canonicalization on a stand-alone machine with no DNS

PART I
If you get a pre-compiled sendmail with BIND in it you can still live with it (As is the case with Redhat supplied sendmail).

a) Refer to my solution for this based on the m4 file described (my favorite). Try to use this option unless you really insist on b) or c) below OR

b)If you talk to me i could supply you with a non-BIND compiled version (8.8.5). Perhaps RedHat should consider supplying two sendmail binaries (like slakware) One with BIND and the other with no BIND. OR

c) The quickest solution to get sendmail with no bind is to edit *src/conf.h* lines below to read as i have shown:

 # ifndef NAMED_BIND conf.h:# define NAMED_BIND 0 / use Berkeley Internet Domain Server */*

PART II:
Sendmail would however normally insist on doing name look ups to resolve the host part using *gethostbyaddr()* for each email it sends (even if you tell it not to canonify names).

a) use an *RFC1597* IP address (10.0.0.0-10.255.255.255, 172.16.0.0-172.31.255.255, 192.168.0.0-192.168.255.255)
Edit */etc/hosts* and add the following as a sample for the host myhost
10.0.0.1 myhost.me.com myhost or
b) If you use NIS (and NIS is compiled into sendmail as seems to be the case in Redhat.) make sure the */etc/nsswitch.conf* file contains:
hosts: files dns and in */etc/hosts* and add the following as a sample for the host myhost [again using RFC1597 IP addressing scheme]
10.0.0.1 myhost.me.com myhost

NOTE: ---- In both a) and b) above it is critical that you list the long (fully qualified) in the /etc/hosts file used. i.e the line should read *10.0.0.1 myhost.me.com myhost* and NOT *10.0.0.1 myhost* you will also need to define the sendmail **$w** macro to hold *myhost.me.com*
add the Following line to your */etc/sendmail.cf* after it is generated ***Djmyhost.me.com*** (or you can use

MASQUAREADE_AS(myhost.me.com) in the *.mc* below)

NOTE!!!!! *myhost* as a domain name is given as an example; you dont have to use it as is. Pick your own hostname if you dont have one already.

---

2) Ensure that you have makemap and it is capable of supporting hash and or the more common dbm format or even btree format (The Redhat version does not support dbm). Makemap is normaly distributed with sendmail.

SETUP (the mc file)

I) create your .mc file using your favorite editor; i'll call this file *my.mc*

a) No DNS -- standalone

---

include(`../m4/cf.m4')
VERSIONID(`me.com's setup with uucp created by xxxx --no dns ')
dnl OSTYPE(linux)
FEATURE(nodns)dnl
FEATURE(nocanonify)dnl
FEATURE(always_add_domain)dnl
FEATURE(mailertable, hash /etc/mailertable)dnl
MAILER(local)dnl
MAILER(smtp)dnl
MAILER(uucp)
define(`SMART_HOST', uucp-dom:mysmarthost)

Let's dissect this:

- *include(`../m4/cf.m4')* requests for the m4 macro found in ../m4/cf.m4 to be include to resolve some things
- *VERSIONID(`me.com's setup with uucp created by xxxx')dnl* This is used to distinguish the different versions of .cf file you might end up creating.
- *OSTYPE(linux)* This is used to define/redefine linux specific stuff. It is safe to have it here.
- *FEATURE(nodns)dnl* This says we dont have a DNS server (so in effect we are using uucp only for mail. Sendmail must be compiled not to use bind) This is now obsolete. I will let it stay here just in case you use an older version of sendmail.
- *FEATURE(nocanonify)dnl* This says Don't pass addresses to *$[... $]* for canonification. Normally if you have BIND compiled in sendmail would try and expand the alias/IP address to a canonical name using DNS. You dont wanna do this if all you have are feeds which connect to you via UUCP i.e you are a stand alone.
- *FEATURE(always_add_domain)dnl* This adds the local domain host name even on locally delivered

TOC

mail Not necessary i just like it; ***This might not be a safe feature since spammers, with proper setup conditions, can abuse you***.
me.com is not connected to the net via tcp/ip so it is safe

- *FEATURE(mailertable, hash /etc/mailertable)dnl* the file */etc/mailertable* is going to be a *hash* database where we will store routing information of certain sites. If you dont have anyone who you feed uucp to then you dont need this. More on this later. If you wanted to use the *dbm* format you would have *FEATURE(mailertable, dbm /etc/mailertable)dnl* If you wanted to use the *btree* format you would have *FEATURE(mailertable, btree /etc/mailertable)dnl*
- *MAILER(local)dnl MAILER(smtp)dnl MAILER(uucp)* These are the mailers we use.
- *define(`SMART_HOST', uucp-dom:mysmarthost)* Our smart host is our ISP who is defined in the uucp **sys** file as *system mysmarthost*. Any messages that we cant handle (i.e ones for domains or mailers we dont understand) will be passed on to our smarthost/ISP to figure out. note we use uucp-dom as the uucp mailer. This particular mailer uses smtp rewriting rules.

b) DNS on

---

include(`../m4/cf.m4')
VERSIONID(`me.com's setup with uucp created by xxxx --dns enabled')dnl
OSTYPE(linux)
FEATURE(always_add_domain)dnl
FEATURE(mailertable, hash /etc/mailertable)dnl
MAILER(local)dnl
MAILER(smtp)dnl
MAILER(uucp)
define(`SMART_HOST', uucp-dom:mysmarthost)

II) create your sendmail.cf file

a) backup your old */etc/sendmail.cf* file
b) overwrite the sendmail.cf with the new one: *m4 my.mc > /etc/sendmail.cf*
c) verify that the sendmail.cf file was correctly created based on your specifications.
Some of the gotchas that i came across were: - *missing uucp-dom mailer* This was because i didnt have the smtp mailer entry. Actually, i understand it is important to have it defined before the uucp

III) create your /etc/mailertable

if you dont have uucp sites feeding off you skip this step

a) create/edit the file /etc/mailertable

sample /etc/mailertable

TOC

system1.org uucp-dom:system1
.system1.org uucp-dom:system1
down.com uucp-dom:down
up.down.com error: Host is unknown at me.com
.down.com uucp-dom:down
.me.com error: Host unknown at me.com

This says anything that is addressed to *system1.org* or to **.system1.org* that we receive will be sent using the *uucp-dom* mailer and be delivered to *system1* (defined in the the uucp **sys** file) in other words we are doing the mail routing for **.system1.org ;* similar for *down.com* and its subnodes except for the additional filtering; we bounce back any mail for down's subnode *up.down.com* using sendmail's built-in *error* mailer with the message ***"Host is unknown at me.com"***; This is because we have been asked to do this by the admin at *down.com.* To bounce unknown subdomains to us we use the last line in the mailertable (which is what all unknown nodes default to).
b) create the database with **makemap**

if you use *hash*:
makemap hash /etc/mailertable < /etc/mailertable
if you use *dbm:*
makemap dbm /etc/mailertable </etc/mailertable
if you use *btree:*
makemap btree /etc/mailertable </etc/mailertable

do this every time you change the file.

IV) restart sendmail

V) test sendmail

i) sendmail -bv user@destination
ii) using sendmail -bt and enter various addresses using 3,0
ruleset to see where they end up and the various rulesets and cf walks taken.

You are set!

TODO

1) Add info on setting up system down.com (so as to show how to take care of those smtp nodes)
2) Add info on how to setup news to be delivered via UUCP (preferably INN). This might require changing the title of the doc.

Secure POP via SSH mini-HOWTO

Manish Singh, <yosh@gimp.org>

v1.0, 30 September 1998

This document explains how to set up secure POP connections using ssh.

1. Introduction

2. The Basic Technique

- 2.1 Setting up Port Forwarding
- 2.2 Testing it out

3. Using it With Your Mail Software

- 3.1 Setting up fetchmail
- 3.2 Automating it all
- 3.3 Not using fetchmail

4. Miscellany

- 4.1 Disclaimer
- 4.2 Copyright
- 4.3 Acknowledgements

1. Introduction

Normal POP mail sessions, by their very nature, are insecure. The password goes across the network in cleartext for everyone to see. Now, this may be perfectly acceptable in a trusted or firewalled environment. But on a public network, such as a university or your run-of-the-mill ISP, anyone armed with a simple network sniffer can grab your password right off the wire. This is compounded by the fact that many people set their computers to check for mail at regular intervals, so the password is sent out quite frequently, which makes it easy to sniff.

With this password, an attacker can now access your email account, which may have sensitive or private information. It is also quite common that this password is the same as the user's shell account, so there is the possibility for more damage.

By doing all POP traffic using an encrypted channel, **nothing** goes in cleartext over the network. We can use ssh's diverse methods of authentication, instead of a simple plaintext password. That is the real point of using this method: not because we get encrypted content (which is futile at this point, since it's probably gone unencrypted over several networks already before reaching your mailbox; securing those communications is the job of GNU Privacy Guard or PGP, not ssh), but the secure authentication.

There are other methods of achieving secure authentication already, such as APOP, KPOP, and IMAP. However, using ssh has the advantage that it works with normal POP configurations, without requiring special client (not all mail clients support advanced protocols) or server support (except for sshd running on the server). You mail provider may be unable or unwilling to use a more secure protocol. Besides, by using ssh you can compress the traffic too, which is a nice little extra for people with slow connections.

2. The Basic Technique

This technique relies on a fundamental feature of ssh: *port forwarding*

There are many variations on this theme, which depend on your desired mail setup. They all require ssh, which is available from http://www.ssh.fi/ and mirrors. RPMs are available at ftp://ftp.replay.com/pub/crypto/ and Debian packages are available at ftp://non-us.debian.org/debian-non-US/ (and their respective mirrors).

2.1 Setting up Port Forwarding

To start port forwarding, run the following command:

```
ssh -C -f popserver -L 11110:popserver:110 sleep 5
```

Let's take a closer look at that command:

`ssh`

The ssh binary itself, the magic program that does it all.

`-C`

This enables compression of the datastream. It's optional, but usually useful, especially for dialup users.

`-f`

Once ssh has done authentication and established port forwarding, fork to background so other programs can be run. Since we're just using the port forwarding features of ssh, we don't need a tty attached to it.

`popserver`

The POP server we're connecting to.

`-L 11110:popserver:110`

Forward local port 11110 to port 110 on the remote server `popserver`. We use a high local port (11110) so any user can create forwardings.

`sleep 5`

After ssh has forked itself into the background, it runs a command. We use `sleep` so that the connection is maintained for enough time for our mail client to setup a connection to the server. 5 seconds is usually sufficient time for this to happen.

You can use most other options to ssh when appropriate. A common setting may be a username, since it

might be different on the POP server.

This *requires* sshd running on the remote server `popserver`. However, you do not need to have an active shell account there. The time it takes to print a message ``You cannot telnet here'' is enough to setup a connection.

2.2 Testing it out

Once you've figured out the details command to run to establish port forwarding, you can try it. For example:

```
$ ssh -C -f msingh@popserver -L 11110:popserver:110 sleep 1000
```

`popserver` is the ol' POP server. My username on my local machine is `manish` so I need to explicitly specify the username `msingh`. (If your local and remote usernames are the same the `msingh@` part is unnecessary.

Then it prints:

```
msingh@popserver's password:
```

And I type in my POP password (you may have different shell and POP passwords though, so use your shell one). Now we're done! So we can try:

```
$ telnet localhost 11110
```

which should print something like:

```
QUALCOMM POP v3.33 ready.
```

Woohoo! It works! The data is sent out over the network encrypted, so the only cleartext is over the loopback interfaces of my local box and the POP server.

---

3. Using it With Your Mail Software

This section describes setting up your POP client software to use the ssh forwarded connection. It's primary focus is fetchmail (ESR's excellent mail-retrieval and forwarding utility), since that is the most flexible software I have found for dealing with POP. fetchmail can be found at http://www.tuxedo.org/~esr/fetchmail/. It will do you a great service to read the excellent documentation that comes with fetchmail.

3.1 Setting up fetchmail

The following is my `.fetchmailrc`

```
defaults
        user msingh is manish
        no rewrite

poll localhost with protocol pop3 and port 11110:
        preconnect "ssh -C -f msingh@popserver -L 11110:popserver:110 sleep 5"
        password foobar;
```

Pretty simple, huh? fetchmail has a wealth of commands, but the key ones are the `preconnect` line and the `poll` option.

We're not connecting directly to the POP server, but instead localhost and port 11110. The `preconnect` does the forwarding each time fetchmail is run, leaving open the connection for 5 seconds, so fetchmail can make it's own connect. The rest fetchmail does itself.

So each time you run fetchmail, you're prompted for your ssh password for authentication. If you run fetchmail in the background (like I do), it's inconvenient to have to do that. Which brings us to the next section.

3.2 Automating it all

ssh can authenticate using many methods. One of these is an RSA public/private key pair. You can generate an authentication key for your account using `ssh-keygen`. An authetication key can have a passphrase associated with it, or the passphase can be blank. Whether you want a passphrase depends on how secure you think the account you are using locally is.

If you think your machine is secure, go ahead and have a blank passpharase. Then the above `.fetchmailrc` works just by running fetchmail. You can then run fetchmail in daemon mode when you dial up and mail is fetched automatically. You're done.

However, if you think you need a passphrase, things get more complex. ssh can run under control of an **agent**, which can register keys and authenticate whatever ssh connections are made under it. So I have this script `getmail.sh`:

```
#!/bin/sh
ssh-add
while true; do fetchmail --syslog --invisible; sleep 5m; done
```

When I dialup, I run:

```
$ ssh-agent getmail.sh
```

This prompts me for my passphrase once, then checks mail every 5 minutes. When the dialup connection is closed, I terminate ssh-agent. (This is automated in my ip-up and ip-down scripts)

3.3 Not using fetchmail

What if I can't/don't want to use fetchmail? Pine, Netscape, and some other clients have their own POP mechanisms. First, consider using fetchmail! It's far more flexible, and mail clients shouldn't be doing that kind of stuff anyway. Both Pine and Netscape can be configured to use local mail systems.

But if you must, unless your client has a preconnect feature like fetchmail, you're going to have to keep the ssh port forward active for the entire time you're connected. Which means using `sleep 100000000` to keep the connection alive. This might not go over well with your network admins.

Secondly, some clients (like Netscape) have the port number hardcoded to 110. So you need to be root to do port forwarding from privledged ports. This is also annoying. But it should work.

---

4. Miscellany

4.1 Disclaimer

There is no guarantee that this document lives up to its intended purpose. This is simply provided as a free resource. As such, the author of the information provided within cannot make any guarentee that the information is even accurate. Use at your own risk.

Cryptographic software such as ssh may be subject to certain restrictions, depending on where you live. In some countries, you must have a license to use such software. If you are unsure of your local laws, please consult someone who is familiar with your situation for more information.

The use of the information provided in this document is most likely not anticipated by your mail service provider. The author does not encourage the abuse and misuse of network services, and provides this document for informational purposes only. If you are in doubt about whether the use of these techniques falls within the service agreement of your mail provider, please clear that up beforehand.

4.2 Copyright

This document is copyright © 1998 Manish Singh <yosh@gimp.org>

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copying, provided that this copyright notice is included exactly as in the original, and that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this document into another language, under the above conditions for modified versions.

Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All trademarks used in this document are acknowledged as being owned by their respective owners.

4.3 Acknowledgements

Special thanks goes to Seth David Schoen <schoen@uclink4.berkeley.edu>, who enlightened me in the ways of ssh port forwarding.

Small Memory Mini-HOWTO v0.1

by: Todd Burgess

tburgess@uoguelph.ca

Introduction

The purpose of this mini HOWTO is to describe how to run Linux on a system with a small amount of memory. Assuming buying more memory is out of the question there are many things you can do to tighten up memory usage in Linux.

Many Linux distributions out of the box are quite bloated from a memory perspective. They run more services and offer more features then most of us will ever need. By removing many of these services you can free up several megabytes of real memory.

My own system is a 486DX2-66 with 12MB of physical memory and 12MB of swap space. It has run Linux for the last 3 years quite happily and hopefully it will run Linux for several more years. :)

Linux Kernel

All the Linux kernels which come with distributions are quite bloated and contain more features then any of us will ever need or use. If you have not re-compiled your own kernel it is highly recommended that you do so. How to re-compile a kernel is beyond the scope of this document but many excellent Linux books and guides cover this subject in intimate detail.

If you do re-compile your kernel remember to put in no more features then you need. For instance: how many of you include PLIP support in your kernel? How many of you who include it actually use it? Smaller kernels require less time to load, use less memory and use less CPU cycles.

Another thing is modules. I personally do not use them because I found them to be a cumbersome. If you use them and like them then they can help to relieve "kernel bloat."

Virtual Consoles

VCs are a great way to free up memory. Most Linux distributions run about 6 of them out of the box. On average running 6 VCs requires about 4MB of memory. Removing a couple of them can free up a couple MBs of memory.

Most of users can get away running only 3 or 4 VCs. How many you choose to remove is a matter personal preference. Just remember that the fewer you run the more memory your applications will have to run.

TOC

The file which outlines how many VCs get loaded is in the file /etc/inittab . In order to remove VCs:

1. Load /etc/inittab in a text editor
2. Look for a line which looks like (the key feature being a line which starts with c1):
   c1:12345:respawn:/sbin/getty tty1 38400 linux
   Start at the highest number (ie c6) and comment it out by inserting a '#' in the first row. Repeat this step as many times as needed. Remember every line you comment out is one less VC running.
3. Re-boot the system for your changes to take effect.

Daemons

Many Linux distributions run many daemons most of us will never use. Most of these daemons are loaded by scripts. Where these scripts are and what they are called depends on your Linux distribution. Slackware set-up scripts are buried in /etc/rc.d/rc.* .

Before you proceed a knowledge of Unix shell script programming would be a definite asset. If however you have no experience writing Unix shell scripts what follows is probably the quickest introduction to shell script programming ever written.

Take the following shell script #!/bin/sh echo "hello world" #echo "good bye cruel world"

The following will echo the string "hello world". Shell scripts must contain the the line "#!/bin/sh" at the very top line. After that every line is executed as if you had typed it at the keyboard (think of shell scripts as nothing more then glorified keyboard macros).

Lines which begin with a '#' are said to be commented out because they do not get executed by the shell. Most start up scripts when they load daemons look like:

if somecondition do something fi What you want to do is comment out every line starting with the if statement and ending with the fi statement.

If you want to find where a daemon is loaded what you want to do is search the start up scripts for the name of the daemon. If I wanted to find where inetd is loaded in Slackware I would do the following: 1 cd /etc/rc.d 2 grep -n inetd rc.*

inetd

inetd allows people to do things like telnet, ftp and send talk requests to your machine. If you never use your system as a server or need to access any of its services remotely you can remove inetd.

lpd

lpd is used to print files on your printer using the lpr command. If you never print on your Linux box you can remove lpd. If however you own a HP Deskjet printer and would like to print I highly recommend the package I put together called dj-printcap which is available at:
ftp://sunsite.unc.edu/pub/Linux/system/Printing/dj-printcap.tar.gz

TOC

nfsd and mountd

These two daemons are used to run a NFS server. If you never use your Linux system as a NFS server you can safely remove these two daemons.

portmap

The portmap daemon is used to handle RPC services. If you do not run a NFS server or any other RPC programs you can remove portmap.

sendmail

sendmail is another daemon which requires a fair bit of memory. If you never use your Linux box as a relay for sending e-mail or you never receive mail on your Linux box you can probably remove sendmail. If you do send e-mail from your Linux box most e-mail clients can be set-up to send e-mail from another mail server.

others

There may be other daemons your system starts up which you do not need. Remove what you feel you have to. Two daemons which you must run are syslogd and klogd.

Conclusions

The following illustrates the steps I took to tighten up my memory usage on my Linux box. Hopefully I have provided you with some insight into what you can do with your Linux box to conserve memory.

Good luck and happy hacking!
Todd Burgess
tburgess@uoguelph.ca
http://eddie.cis.uoguelph.ca/~tburgess

Software-RAID HOWTO

Linas Vepstas, linas@linas.org

v0.54, 21 November 1998

RAID stands for "Redundant Array of Inexpensive Disks", and is meant to be a way of creating a fast and reliable disk-drive subsystem out of individual disks. RAID can guard against disk failure, and can also improve performance over that of a single disk drive. This document is a tutorial/HOWTO/FAQ for users of the Linux MD kernel extension, the associated tools, and their use. The MD extension implements RAID-0 (striping), RAID-1 (mirroring), RAID-4 and RAID-5 in software. That is, with MD, no special hardware or disk controllers are required to get many of the benefits of RAID.

Preamble

This document is copyrighted and GPL'ed by Linas Vepstas (linas@linas.org). Permission to use, copy, distribute this document for any purpose is hereby granted, provided that the author's / editor's name and this notice appear in all copies and/or supporting documents; and that an unmodified version of this document is made freely available. This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY, either expressed or implied. While every effort has been taken to ensure the accuracy of the information documented herein, the author / editor / maintainer assumes NO RESPONSIBILITY for any errors, or for any damages, direct or consequential, as a result of the use of the information documented herein.

RAID, although designed to improve system reliability by adding redundancy, can also lead to a false sense of security and confidence when used improperly. This false confidence can lead to even greater disasters. In particular, note that RAID is designed to protect against *disk* failures, and not against *power* failures or *operator* mistakes. Power failures, buggy development kernels, or operator/admin errors can lead to damaged data that it is not recoverable! RAID is *not* a substitute for proper backup of your system. Know what you are doing, test, be knowledgeable and aware!

1. Introduction

2. Understanding RAID

3. Setup & Installation Considerations

4. Error Recovery

5. Troubleshooting Install Problems

6. Supported Hardware & Software

7. Modifying an Existing Installation

8. Performance, Tools & General Bone-headed Questions

9. High Availability RAID

10. Questions Waiting for Answers

11. Wish List of Enhancements to MD and Related Software

1. Introduction

1. **Q**: What is RAID?

   A: RAID stands for "Redundant Array of Inexpensive Disks", and is meant to be a way of creating a fast and reliable disk-drive subsystem out of individual disks. In the PC world, "I" has come to stand for "Independent", where marketing forces continue to differentiate IDE and SCSI. In it's original meaning, "I" meant "Inexpensive as compared to refrigerator-sized mainframe 3380 DASD", monster drives which made nice houses look cheap, and diamond rings look like trinkets.

2. **Q**: What is this document?

   A: This document is a tutorial/HOWTO/FAQ for users of the Linux MD kernel extension, the associated tools, and their use. The MD extension implements RAID-0 (striping), RAID-1 (mirroring), RAID-4 and RAID-5 in software. That is, with MD, no special hardware or disk controllers are required to get many of the benefits of RAID.

   This document is **NOT** an introduction to RAID; you must find this elsewhere.

3. **Q**: What levels of RAID does the Linux kernel implement?

   A: Striping (RAID-0) and linear concatenation are a part of the stock 2.x series of kernels. This code is of production quality; it is well understood and well maintained. It is being used in some very large USENET news servers.

   RAID-1, RAID-4 & RAID-5 are a part of the 2.1.63 and greater kernels. For earlier 2.0.x and 2.1.x kernels, patches exist that will provide this function. Don't feel obligated to upgrade to 2.1.63; upgrading the kernel is hard; it is *much* easier to patch an earlier kernel. Most of the RAID user community is running 2.0.x kernels, and that's where most of the historic RAID development has focused. The current snapshots should be considered near-production quality; that is, there are no known bugs but there are some rough edges and untested system setups. There are a large number of people using Software RAID in a production environment.

   RAID-1 hot reconstruction has been recently introduced (August 1997) and should be considered alpha quality. RAID-5 hot reconstruction will be alpha quality any day now.

   A word of caution about the 2.1.x development kernels: these are less than stable in a variety of ways. Some of the newer disk controllers (e.g. the Promise Ultra's) are supported only in the 2.1.x kernels. However, the 2.1.x kernels have seen frequent changes in the block device driver, in the DMA and interrupt code, in the PCI, IDE and SCSI code, and in the disk controller drivers. The combination of these factors, coupled to cheapo hard drives and/or low-quality ribbon cables can lead to considerable heartbreak. The `ckraid` tool, as well as `fsck` and `mount` put

TOC

considerable stress on the RAID subsystem. This can lead to hard lockups during boot, where even the magic alt-SysReq key sequence won't save the day. Use caution with the 2.1.x kernels, and expect trouble. Or stick to the 2.0.34 kernel.

4. **Q**: I'm running an older kernel. Where do I get patches?

   A: Software RAID-0 and linear mode are a stock part of all current Linux kernels. Patches for Software RAID-1,4,5 are available from http://luthien.nuclecu.unam.mx/~miguel/raid. See also the quasi-mirror ftp://linux.kernel.org/pub/linux/daemons/raid/ for patches, tools and other goodies.

5. **Q**: Are there other Linux RAID references?

   A:

   - Generic RAID overview: http://www.dpt.com/uraiddoc.html.
   - General Linux RAID options: http://linas.org/linux/raid.html.
   - Latest version of this document: http://linas.org/linux/Software-RAID/Software-RAID.html.
   - Linux-RAID mailing list archive: http://www.linuxhq.com/lnxlists/.
   - Linux Software RAID Home Page: http://luthien.nuclecu.unam.mx/~miguel/raid.
   - Linux Software RAID tools: ftp://linux.kernel.org/pub/linux/daemons/raid/.
   - How to setting up linear/stripped Software RAID: http://www.ssc.com/lg/issue17/raid.html.
   - Bootable RAID mini-HOWTO: ftp://ftp.bizsystems.com/pub/raid/bootable-raid.
   - Root RAID HOWTO: ftp://ftp.bizsystems.com/pub/raid/Root-RAID-HOWTO.
   - Linux RAID-Geschichten: http://www.infodrom.north.de/~joey/Linux/raid/.

6. **Q**: Who do I blame for this document?

   A: Linas Vepstas slapped this thing together. However, most of the information, and some of the words were supplied by

   - Bradley Ward Allen < ulmo@Q.Net>
   - Luca Berra < bluca@comedia.it>
   - Brian Candler < B.Candler@pobox.com>
   - Bohumil Chalupa < bochal@apollo.karlov.mff.cuni.cz>
   - Rob Hagopian < hagopiar@vu.union.edu>
   - Anton Hristozov < anton@intransco.com>
   - Miguel de Icaza < miguel@luthien.nuclecu.unam.mx>
   - Marco Meloni < tonno@stud.unipg.it>
   - Ingo Molnar < mingo@pc7537.hil.siemens.at>

- Alvin Oga < alvin@planet.fef.com>
- Gadi Oxman < gadio@netvision.net.il>
- Vaughan Pratt < pratt@cs.Stanford.EDU>
- Steven A. Reisman < sar@pressenter.com>
- Michael Robinton < michael@bzs.org>
- Martin Schulze < joey@finlandia.infodrom.north.de>
- Geoff Thompson < geofft@cs.waikato.ac.nz>
- Edward Welbon < welbon@bga.com>
- Rod Wilkens < rwilkens@border.net>
- Johan Wiltink < j.m.wiltink@pi.net>
- Leonard N. Zubkoff < lnz@dandelion.com>
- Marc ZYNGIER < zyngier@ufr-info-p7.ibp.fr>

Copyrights

- Copyright (C) 1994-96 Marc ZYNGIER
- Copyright (C) 1997 Gadi Oxman, Ingo Molnar, Miguel de Icaza
- Copyright (C) 1997, 1998 Linas Vepstas
- By copyright law, additional copyrights are implicitly held by the contributors listed above.

Thanks all for being there!

---

2. Understanding RAID

1. **Q**: What is RAID? Why would I ever use it?

   A: RAID is a way of combining multiple disk drives into a single entity to improve performance and/or reliability. There are a variety of different types and implementations of RAID, each with its own advantages and disadvantages. For example, by putting a copy of the same data on two disks (called **disk mirroring**, or RAID level 1), read performance can be improved by reading alternately from each disk in the mirror. On average, each disk is less busy, as it is handling only 1/2 the reads (for two disks), or 1/3 (for three disks), etc. In addition, a mirror can improve reliability: if one disk fails, the other disk(s) have a copy of the data. Different ways of combining the disks into one, referred to as **RAID levels**, can provide greater storage efficiency than simple mirroring, or can alter latency (access-time) performance, or throughput (transfer rate) performance, for reading or writing, while still retaining redundancy that is useful for guarding against failures.

   Although RAID can protect against disk failure, it does not protect against operator and administrator (human) error, or against loss due to programming bugs (possibly due to bugs in the RAID software itself). The net abounds with tragic tales of system administrators who have bungled a RAID installation, and have lost all of their data. RAID is not a substitute for frequent, regularly scheduled backup.

   RAID can be implemented in hardware, in the form of special disk controllers, or in software, as a kernel module that is layered in between the low-level disk driver, and the file system which sits above it. RAID hardware is always a "disk controller", that is, a device to which one can cable up the disk drives. Usually it comes in the form of an adapter card that will plug into a ISA/EISA/PCI/S-Bus/MicroChannel slot. However, some RAID controllers are in the form of a box that connects into the cable in between the usual system disk controller, and the disk drives. Small ones may fit into a drive bay; large ones may be built into a storage cabinet with its own drive bays and power supply. The latest RAID hardware used with the latest & fastest CPU will usually provide the best overall performance, although at a significant price. This is because most RAID controllers come with on-board DSP's and memory cache that can off-load a considerable amount of processing from the main CPU, as well as allow high transfer rates into the large controller cache. Old RAID hardware can act as a "de-accelerator" when used with newer CPU's: yesterday's fancy DSP and cache can act as a bottleneck, and it's performance is often beaten by pure-software RAID and new but otherwise plain, run-of-the-mill disk controllers. RAID hardware can offer an advantage over pure-software RAID, if it can makes use of disk-spindle synchronization and its knowledge of the disk-platter position with regard to the disk head, and the desired disk-block. However, most modern (low-cost) disk drives do

TOC

not offer this information and level of control anyway, and thus, most RAID hardware does not take advantage of it. RAID hardware is usually not compatible across different brands, makes and models: if a RAID controller fails, it must be replaced by another controller of the same type. As of this writing (June 1998), a broad variety of hardware controllers will operate under Linux; however, none of them currently come with configuration and management utilities that run under Linux.

Software-RAID is a set of kernel modules, together with management utilities that implement RAID purely in software, and require no extraordinary hardware. The Linux RAID subsystem is implemented as a layer in the kernel that sits above the low-level disk drivers (for IDE, SCSI and Paraport drives), and the block-device interface. The filesystem, be it ext2fs, DOS-FAT, or other, sits above the block-device interface. Software-RAID, by its very software nature, tends to be more flexible than a hardware solution. The downside is that it of course requires more CPU cycles and power to run well than a comparable hardware system. Of course, the cost can't be beat. Software RAID has one further important distinguishing feature: it operates on a partition-by-partition basis, where a number of individual disk partitions are ganged together to create a RAID partition. This is in contrast to most hardware RAID solutions, which gang together entire disk drives into an array. With hardware, the fact that there is a RAID array is transparent to the operating system, which tends to simplify management. With software, there are far more configuration options and choices, tending to complicate matters.

As of this writing (June 1998), the administration of RAID under Linux is far from trivial, and is best attempted by experienced system administrators. The theory of operation is complex. The system tools require modification to startup scripts. And recovery from disk failure is non-trivial, and prone to human error. RAID is not for the novice, and any benefits it may bring to reliability and performance can be easily outweighed by the extra complexity. Indeed, modern disk drives are incredibly reliable and modern CPU's and controllers are quite powerful. You might more easily obtain the desired reliability and performance levels by purchasing higher-quality and/or faster hardware.

2. **Q**: What are RAID levels? Why so many? What distinguishes them?

   A: The different RAID levels have different performance, redundancy, storage capacity, reliability and cost characteristics. Most, but not all levels of RAID offer redundancy against disk failure. Of those that offer redundancy, RAID-1 and RAID-5 are the most popular. RAID-1 offers better performance, while RAID-5 provides for more efficient use of the available storage space. However, tuning for performance is an entirely different matter, as performance depends strongly on a large variety of factors, from the type of application, to the sizes of stripes, blocks, and files. The more difficult aspects of performance tuning are deferred to a later section of this HOWTO.

   The following describes the different RAID levels in the context of the Linux software RAID implementation.

   - **RAID-linear** is a simple concatenation of partitions to create a larger virtual

TOC

partition. It is handy if you have a number small drives, and wish to create a single, large partition. This concatenation offers no redundancy, and in fact decreases the overall reliability: if any one disk fails, the combined partition will fail.

- **RAID-1** is also referred to as "mirroring". Two (or more) partitions, all of the same size, each store an exact copy of all data, disk-block by disk-block. Mirroring gives strong protection against disk failure: if one disk fails, there is another with the an exact copy of the same data. Mirroring can also help improve performance in I/O-laden systems, as read requests can be divided up between several disks. Unfortunately, mirroring is also the least efficient in terms of storage: two mirrored partitions can store no more data than a single partition.
- **Striping** is the underlying concept behind all of the other RAID levels. A stripe is a contiguous sequence of disk blocks. A stripe may be as short as a single disk block, or may consist of thousands. The RAID drivers split up their component disk partitions into stripes; the different RAID levels differ in how they organize the stripes, and what data they put in them. The interplay between the size of the stripes, the typical size of files in the file system, and their location on the disk is what determines the overall performance of the RAID subsystem.
- **RAID-0** is much like RAID-linear, except that the component partitions are divided into stripes and then interleaved. Like RAID-linear, the result is a single larger virtual partition. Also like RAID-linear, it offers no redundancy, and therefore decreases overall reliability: a single disk failure will knock out the whole thing. RAID-0 is often claimed to improve performance over the simpler RAID-linear. However, this may or may not be true, depending on the characteristics to the file system, the typical size of the file as compared to the size of the stripe, and the type of workload. The `ext2fs` file system already scatters files throughout a partition, in an effort to minimize fragmentation. Thus, at the simplest level, any given access may go to one of several disks, and thus, the interleaving of stripes across multiple disks offers no apparent additional advantage. However, there are performance differences, and they are data, workload, and stripe-size dependent.
- **RAID-4** interleaves stripes like RAID-0, but it requires an additional partition to store parity information. The parity is used to offer redundancy: if any one of the disks fail, the data on the remaining disks can be used to reconstruct the data that was on the failed disk. Given N data disks, and one parity disk, the parity stripe is computed by taking one stripe from each of the data disks, and XOR'ing them together. Thus, the storage capacity of a an (N+1)-disk RAID-4 array is N, which is a lot better than mirroring (N+1) drives, and is almost as good as a RAID-0 setup for large N. Note that for N=1, where there is one data drive, and one parity drive, RAID-4 is a lot like mirroring, in that each of the two disks is a copy of each other. However, RAID-4 does **NOT** offer the read-performance of mirroring, and offers considerably degraded write

performance. In brief, this is because updating the parity requires a read of the old parity, before the new parity can be calculated and written out. In an environment with lots of writes, the parity disk can become a bottleneck, as each write must access the parity disk.

- **RAID-5** avoids the write-bottleneck of RAID-4 by alternately storing the parity stripe on each of the drives. However, write performance is still not as good as for mirroring, as the parity stripe must still be read and XOR'ed before it is written. Read performance is also not as good as it is for mirroring, as, after all, there is only one copy of the data, not two or more. RAID-5's principle advantage over mirroring is that it offers redundancy and protection against single-drive failure, while offering far more storage capacity when used with three or more drives.
- **RAID-2 and RAID-3** are seldom used anymore, and to some degree are have been made obsolete by modern disk technology. RAID-2 is similar to RAID-4, but stores ECC information instead of parity. Since all modern disk drives incorporate ECC under the covers, this offers little additional protection. RAID-2 can offer greater data consistency if power is lost during a write; however, battery backup and a clean shutdown can offer the same benefits. RAID-3 is similar to RAID-4, except that it uses the smallest possible stripe size. As a result, any given read will involve all disks, making overlapping I/O requests difficult/impossible. In order to avoid delay due to rotational latency, RAID-3 requires that all disk drive spindles be synchronized. Most modern disk drives lack spindle-synchronization ability, or, if capable of it, lack the needed connectors, cables, and manufacturer documentation. Neither RAID-2 nor RAID-3 are supported by the Linux Software-RAID drivers.
- **Other RAID levels** have been defined by various researchers and vendors. Many of these represent the layering of one type of raid on top of another. Some require special hardware, and others are protected by patent. There is no commonly accepted naming scheme for these other levels. Sometime the advantages of these other systems are minor, or at least not apparent until the system is highly stressed. Except for the layering of RAID-1 over RAID-0/linear, Linux Software RAID does not support any of the other variations.

---

3. Setup & Installation Considerations

1. **Q**: What is the best way to configure Software RAID?

   A: I keep rediscovering that file-system planning is one of the more difficult Unix configuration tasks. To answer your question, I can describe what we did. We planned the following setup:

   - two EIDE disks, 2.1.gig each.

```
disk partition mount pt.  size    device
  1      1       /        300M   /dev/hda1
  1      2       swap      64M   /dev/hda2
  1      3       /home    800M   /dev/hda3
  1      4       /var     900M   /dev/hda4

  2      1       /root    300M   /dev/hdc1
  2      2       swap      64M   /dev/hdc2
  2      3       /home    800M   /dev/hdc3
  2      4       /var     900M   /dev/hdc4
```

   - Each disk is on a separate controller (& ribbon cable). The theory is that a controller failure and/or ribbon failure won't disable both disks. Also, we might possibly get a performance boost from parallel operations over two controllers/cables.
   - Install the Linux kernel on the root (`/`) partition `/dev/hda1`. Mark this partition as bootable.
   - `/dev/hdc1` will contain a ``cold'' copy of `/dev/hda1`. This is NOT a raid copy, just a plain old copy-copy. It's there just in case the first disk fails; we can use a rescue disk, mark `/dev/hdc1` as bootable, and use that to keep going without having to reinstall the system. You may even want to put `/dev/hdc1`'s copy of the kernel into LILO to simplify booting in case of failure. The theory here is that in case of severe failure, I can still boot the system without worrying about raid superblock-corruption or other raid failure modes & gotchas that I don't understand.
   - `/dev/hda3` and `/dev/hdc3` will be mirrors `/dev/md0`.
   - `/dev/hda4` and `/dev/hdc4` will be mirrors `/dev/md1`.
   - we picked `/var` and `/home` to be mirrored, and in separate partitions, using the following logic:
     - `/` (the root partition) will contain relatively static, non-changing data: for all practical purposes, it will be read-only without actually being marked & mounted read-only.

TOC

- `/home` will contain "slowly" changing data.
- `/var` will contain rapidly changing data, including mail spools, database contents and web server logs.

The idea behind using multiple, distinct partitions is that **if**, for some bizarre reason, whether it is human error, power loss, or an operating system gone wild, corruption is limited to one partition. In one typical case, power is lost while the system is writing to disk. This will almost certainly lead to a corrupted filesystem, which will be repaired by `fsck` during the next boot. Although `fsck` does it's best to make the repairs without creating additional damage during those repairs, it can be comforting to know that any such damage has been limited to one partition. In another typical case, the sysadmin makes a mistake during rescue operations, leading to erased or destroyed data. Partitions can help limit the repercussions of the operator's errors.

- Other reasonable choices for partitions might be `/usr` or `/opt`. In fact, `/opt` and `/home` make great choices for RAID-5 partitions, if we had more disks. A word of caution: **DO NOT** put `/usr` in a RAID-5 partition. If a serious fault occurs, you may find that you cannot mount `/usr`, and that you want some of the tools on it (e.g. the networking tools, or the compiler.) With RAID-1, if a fault has occurred, and you can't get RAID to work, you can at least mount one of the two mirrors. You can't do this with any of the other RAID levels (RAID-5, striping, or linear append).

So, to complete the answer to the question:

- install the OS on disk 1, partition 1. do NOT mount any of the other partitions.
- install RAID per instructions.
- configure `md0` and `md1`.
- convince yourself that you know what to do in case of a disk failure! Discover sysadmin mistakes now, and not during an actual crisis. Experiment! (we turned off power during disk activity — this proved to be ugly but informative).
- do some ugly mount/copy/unmount/rename/reboot scheme to move `/var` over to the `/dev/md1`. Done carefully, this is not dangerous.
- enjoy!

2. **Q**: What is the difference between the `mdadd`, `mdrun`, *etc.* commands, and the `raidadd`, `raidrun` commands?

   A: The names of the tools have changed as of the 0.5 release of the raidtools package. The `md` naming convention was used in the 0.43 and older versions, while `raid` is used in 0.5 and newer versions.

3. **Q**: I want to run RAID-linear/RAID-0 in the stock 2.0.34 kernel. I don't want to apply the raid patches, since these are not needed for RAID-0/linear. Where can I get the raid-tools to manage this?

   A: This is a tough question, indeed, as the newest raid tools package needs to have the RAID-1,4,5 kernel patches installed in order to compile. I am not aware of any pre-compiled, binary version of the raid tools that is available at this time. However, experiments show that the raid-tools binaries, when compiled against kernel 2.1.100, seem to work just fine in creating a RAID-0/linear partition under 2.0.34. A brave soul

TOC

has asked for these, and I've **temporarily** placed the binaries mdadd, mdcreate, etc. at http://linas.org/linux/Software-RAID/ You must get the man pages, etc. from the usual raid-tools package.

4. **Q**: Can I strip/mirror the root partition (`/`)? Why can't I boot Linux directly from the `md` disks?

   A: Both LILO and Loadlin need an non-stripped/mirrored partition to read the kernel image from. If you want to strip/mirror the root partition (`/`), then you'll want to create an unstriped/mirrored partition to hold the kernel(s). Typically, this partition is named `/boot`. Then you either use the initial ramdisk support (initrd), or patches from Harald Hoyer < HarryH@Royal.Net> that allow a stripped partition to be used as the root device. (These patches are now a standard part of recent 2.1.x kernels)

   There are several approaches that can be used. One approach is documented in detail in the Bootable RAID mini-HOWTO: ftp://ftp.bizsystems.com/pub/raid/bootable-raid.

   Alternately, use `mkinitrd` to build the ramdisk image, see below.

   Edward Welbon < welbon@bga.com> writes:

   - ... all that is needed is a script to manage the boot setup. To mount an `md` filesystem as root, the main thing is to build an initial file system image that has the needed modules and md tools to start `md`. I have a simple script that does this.
   - For boot media, I have a small **cheap** SCSI disk (170MB I got it used for $20). This disk runs on a AHA1452, but it could just as well be an inexpensive IDE disk on the native IDE. The disk need not be very fast since it is mainly for boot.
   - This disk has a small file system which contains the kernel and the file system image for `initrd`. The initial file system image has just enough stuff to allow me to load the raid SCSI device driver module and start the raid partition that will become root. I then do an

     ```
     echo 0x900 > /proc/sys/kernel/real-root-dev
     ```

     (`0x900` is for `/dev/md0`) and exit `linuxrc`. The boot proceeds normally from there.
   - I have built most support as a module except for the AHA1452 driver that brings in the `initrd` filesystem. So I have a fairly small kernel. The method is perfectly reliable, I have been doing this since before 2.1.26 and have never had a problem that I could not easily recover from. The file systems even survived several 2.1.4[45] hard crashes with no real problems.
   - At one time I had partitioned the raid disks so that the initial cylinders of the first raid disk held the kernel and the initial cylinders of the second raid disk hold the initial file system image, instead I made the initial cylinders of the raid disks swap since they are the fastest cylinders (why waste them on boot?).
   - The nice thing about having an inexpensive device dedicated to boot is that it is easy to boot from and can also serve as a rescue disk if necessary. If you are interested, you can take a look at the script that builds my initial ram disk image and then runs `LILO`.

http://www.realtime.net/~welbon/initrd.md.tar.gz

It is current enough to show the picture. It isn't especially pretty and it could certainly build a much smaller filesystem image for the initial ram disk. It would be easy to a make it more efficient. But it uses `LILO` as is. If you make any improvements, please forward a copy to me. 8-)

5. **Q**: I have heard that I can run mirroring over striping. Is this true? Can I run mirroring over the loopback device?

   A: Yes, but not the reverse. That is, you can put a stripe over several disks, and then build a mirror on top of this. However, striping cannot be put on top of mirroring.

   A brief technical explanation is that the linear and stripe personalities use the `ll_rw_blk` routine for access. The `ll_rw_blk` routine maps disk devices and sectors, not blocks. Block devices can be layered one on top of the other; but devices that do raw, low-level disk accesses, such as `ll_rw_blk`, cannot.

   Currently (November 1997) RAID cannot be run over the loopback devices, although this should be fixed shortly.

6. **Q**: I have two small disks and three larger disks. Can I concatenate the two smaller disks with RAID-0, and then create a RAID-5 out of that and the larger disks?

   A: Currently (November 1997), for a RAID-5 array, no. Currently, one can do this only for a RAID-1 on top of the concatenated drives.

7. **Q**: What is the difference between RAID-1 and RAID-5 for a two-disk configuration (i.e. the difference between a RAID-1 array built out of two disks, and a RAID-5 array built out of two disks)?

   A: There is no difference in storage capacity. Nor can disks be added to either array to increase capacity (see the question below for details).

   RAID-1 offers a performance advantage for reads: the RAID-1 driver uses distributed-read technology to simultaneously read two sectors, one from each drive, thus doubling read performance.

   The RAID-5 driver, although it contains many optimizations, does not currently (September 1997) realize that the parity disk is actually a mirrored copy of the data disk. Thus, it serializes data reads.

8. **Q**: How can I guard against a two-disk failure?

   A: Some of the RAID algorithms do guard against multiple disk failures, but these are not currently implemented for Linux. However, the Linux Software RAID can guard against multiple disk failures by layering an array on top of an array. For example, nine disks can be used to create three raid-5 arrays. Then these three arrays can in turn be hooked together into a single RAID-5 array on top. In fact, this kind of a configuration will guard against a three-disk failure. Note that a large amount of disk space is "wasted" on the redundancy information.

```
For an NxN raid-5 array,
N=3, 5 out of 9 disks are used for parity (=55%)
N=4, 7 out of 16 disks
N=5, 9 out of 25 disks
```

```
...
N=9, 17 out of 81 disks (=~20%)
```

In general, an MxN array will use M+N-1 disks for parity. The least amount of space is "wasted" when M=N.

Another alternative is to create a RAID-1 array with three disks. Note that since all three disks contain identical data, that 2/3's of the space is "wasted".

9. **Q**: I'd like to understand how it'd be possible to have something like `fsck`: if the partition hasn't been cleanly unmounted, `fsck` runs and fixes the filesystem by itself more than 90% of the time. Since the machine is capable of fixing it by itself with `ckraid --fix`, why not make it automatic?

   A: This can be done by adding lines like the following to `/etc/rc.d/rc.sysinit`:

```
mdadd /dev/md0 /dev/hda1 /dev/hdc1 || {
    ckraid --fix /etc/raid.usr.conf
    mdadd /dev/md0 /dev/hda1 /dev/hdc1
}
```

   or

```
mdrun -p1 /dev/md0
if [ $? -gt 0 ] ; then
        ckraid --fix /etc/raid1.conf
        mdrun -p1 /dev/md0
fi
```

   Before presenting a more complete and reliable script, lets review the theory of operation. Gadi Oxman writes: In an unclean shutdown, Linux might be in one of the following states:

   - The in-memory disk cache was in sync with the RAID set when the unclean shutdown occurred; no data was lost.
   - The in-memory disk cache was newer than the RAID set contents when the crash occurred; this results in a corrupted filesystem and potentially in data loss. This state can be further divided to the following two states:
     - Linux was writing data when the unclean shutdown occurred.
     - Linux was not writing data when the crash occurred.

   Suppose we were using a RAID-1 array. In (2a), it might happen that before the crash, a small number of data blocks were successfully written only to some of the mirrors, so that on the next reboot, the mirrors will no longer contain the same data. If we were to ignore the mirror differences, the raidtools-0.36.3 read-balancing code might choose to read the above data blocks from any of the mirrors, which will result in inconsistent behavior (for example, the output of `e2fsck -n /dev/md0` can differ from run to run).

   Since RAID doesn't protect against unclean shutdowns, usually there isn't any "obviously

TOC

correct" way to fix the mirror differences and the filesystem corruption.

For example, by default `ckraid --fix` will choose the first operational mirror and update the other mirrors with its contents. However, depending on the exact timing at the crash, the data on another mirror might be more recent, and we might want to use it as the source mirror instead, or perhaps use another method for recovery.

The following script provides one of the more robust boot-up sequences. In particular, it guards against long, repeated `ckraid`'s in the presence of uncooperative disks, controllers, or controller device drivers. Modify it to reflect your config, and copy it to `rc.raid.init`. Then invoke `rc.raid.init` after the root partition has been fsck'ed and mounted rw, but before the remaining partitions are fsck'ed. Make sure the current directory is in the search path.

```
mdadd /dev/md0 /dev/hda1 /dev/hdc1 || {
    rm -f /fastboot             # force an fsck to occur
    ckraid --fix /etc/raid.usr.conf
    mdadd /dev/md0 /dev/hda1 /dev/hdc1
}
# if a crash occurs later in the boot process,
# we at least want to leave this md in a clean state.
/sbin/mdstop /dev/md0

mdadd /dev/md1 /dev/hda2 /dev/hdc2 || {
    rm -f /fastboot             # force an fsck to occur
    ckraid --fix /etc/raid.home.conf
    mdadd /dev/md1 /dev/hda2 /dev/hdc2
}
# if a crash occurs later in the boot process,
# we at least want to leave this md in a clean state.
/sbin/mdstop /dev/md1

mdadd /dev/md0 /dev/hda1 /dev/hdc1
mdrun -p1 /dev/md0
if [ $? -gt 0 ] ; then
    rm -f /fastboot             # force an fsck to occur
    ckraid --fix /etc/raid.usr.conf
    mdrun -p1 /dev/md0
fi
# if a crash occurs later in the boot process,
# we at least want to leave this md in a clean state.
/sbin/mdstop /dev/md0

mdadd /dev/md1 /dev/hda2 /dev/hdc2
mdrun -p1 /dev/md1
if [ $? -gt 0 ] ; then
    rm -f /fastboot             # force an fsck to occur
```

```
        ckraid --fix /etc/raid.home.conf
        mdrun -p1 /dev/md1
    fi
    # if a crash occurs later in the boot process,
    # we at least want to leave this md in a clean state.
    /sbin/mdstop /dev/md1

    # OK, just blast through the md commands now.  If there were
    # errors, the above checks should have fixed things up.
    /sbin/mdadd /dev/md0 /dev/hda1 /dev/hdc1
    /sbin/mdrun -p1 /dev/md0

    /sbin/mdadd /dev/md12 /dev/hda2 /dev/hdc2
    /sbin/mdrun -p1 /dev/md1
```

In addition to the above, you'll want to create a `rc.raid.halt` which should look like the following:

```
/sbin/mdstop /dev/md0
/sbin/mdstop /dev/md1
```

Be sure to modify both `rc.sysinit` and `init.d/halt` to include this everywhere that filesystems get unmounted before a halt/reboot. (Note that `rc.sysinit` unmounts and reboots if `fsck` returned with an error.)

10. **Q**: Can I set up one-half of a RAID-1 mirror with the one disk I have now, and then later get the other disk and just drop it in?

    A: With the current tools, no, not in any easy way. In particular, you cannot just copy the contents of one disk onto another, and then pair them up. This is because the RAID drivers use glob of space at the end of the partition to store the superblock. This decreases the amount of space available to the file system slightly; if you just naively try to force a RAID-1 arrangement onto a partition with an existing filesystem, the raid superblock will overwrite a portion of the file system and mangle data. Since the ext2fs filesystem scatters files randomly throughput the partition (in order to avoid fragmentation), there is a very good chance that some file will land at the very end of a partition long before the disk is full.

    If you are clever, I suppose you can calculate how much room the RAID superblock will need, and make your filesystem slightly smaller, leaving room for it when you add it later. But then, if you are this clever, you should also be able to modify the tools to do this automatically for you. (The tools are not terribly complex).

    Note:A careful reader has pointed out that the following trick may work; I have not tried or verified this: Do the `mkraid` with `/dev/null` as one of the devices. Then `mdadd -r` with only the single, true disk (do not mdadd `/dev/null`). The `mkraid` should have successfully built the raid array, while the mdadd step just forces the system to run

in "degraded" mode, as if one of the disks had failed.

4. Error Recovery

1. **Q**: I have a RAID-1 (mirroring) setup, and lost power while there was disk activity. Now what do I do?

   A: The redundancy of RAID levels is designed to protect against a **disk** failure, not against a **power** failure. There are several ways to recover from this situation.

   - Method (1): Use the raid tools. These can be used to sync the raid arrays. They do not fix file-system damage; after the raid arrays are sync'ed, then the file-system still has to be fixed with fsck. Raid arrays can be checked with `ckraid /etc/raid1.conf` (for RAID-1, else, `/etc/raid5.conf`, etc.) Calling `ckraid /etc/raid1.conf --fix` will pick one of the disks in the array (usually the first), and use that as the master copy, and copy its blocks to the others in the mirror. To designate which of the disks should be used as the master, you can use the `--force-source` flag: for example, `ckraid /etc/raid1.conf --fix --force-source /dev/hdc3` The ckraid command can be safely run without the `--fix` option to verify the inactive RAID array without making any changes. When you are comfortable with the proposed changes, supply the `--fix` option.
   - Method (2): Paranoid, time-consuming, not much better than the first way. Lets assume a two-disk RAID-1 array, consisting of partitions `/dev/hda3` and `/dev/hdc3`. You can try the following:
     1. `fsck /dev/hda3`
     2. `fsck /dev/hdc3`
     3. decide which of the two partitions had fewer errors, or were more easily recovered, or recovered the data that you wanted. Pick one, either one, to be your new ``master'' copy. Say you picked `/dev/hdc3`.
     4. `dd if=/dev/hdc3 of=/dev/hda3`
     5. `mkraid raid1.conf -f --only-superblock`

     Instead of the last two steps, you can instead run `ckraid /etc/raid1.conf --fix --force-source /dev/hdc3` which should be a bit faster.
   - Method (3): Lazy man's version of above. If you don't want to wait for long fsck's to complete, it is perfectly fine to skip the first three steps above, and move directly to the last two steps. Just be sure to run `fsck /dev/md0` after you are done. Method (3) is actually just method (1) in disguise.

   In any case, the above steps will only sync up the raid arrays. The file system probably needs fixing as well: for this, fsck needs to be run on the active, unmounted md device.

TOC

With a three-disk RAID-1 array, there are more possibilities, such as using two disks to "vote" a majority answer. Tools to automate this do not currently (September 97) exist.

2. **Q**: I have a RAID-4 or a RAID-5 (parity) setup, and lost power while there was disk activity. Now what do I do?

   A: The redundancy of RAID levels is designed to protect against a **disk** failure, not against a **power** failure. Since the disks in a RAID-4 or RAID-5 array do not contain a file system that fsck can read, there are fewer repair options. You cannot use fsck to do preliminary checking and/or repair; you must use ckraid first.

   The `ckraid` command can be safely run without the `--fix` option to verify the inactive RAID array without making any changes. When you are comfortable with the proposed changes, supply the `--fix` option.

   If you wish, you can try designating one of the disks as a "failed disk". Do this with the `--suggest-failed-disk-mask` flag.

   Only one bit should be set in the flag: RAID-5 cannot recover two failed disks. The mask is a binary bit mask: thus:

   ```
   0x1 == first disk
   0x2 == second disk
   0x4 == third disk
   0x8 == fourth disk, etc.
   ```

   Alternately, you can choose to modify the parity sectors, by using the `--suggest-fix-parity` flag. This will recompute the parity from the other sectors.

   The flags `--suggest-failed-dsk-mask` and `--suggest-fix-parity` can be safely used for verification. No changes are made if the `--fix` flag is not specified. Thus, you can experiment with different possible repair schemes.

3. **Q**: My RAID-1 device, `/dev/md0` consists of two hard drive partitions: `/dev/hda3` and `/dev/hdc3`. Recently, the disk with `/dev/hdc3` failed, and was replaced with a new disk. My best friend, who doesn't understand RAID, said that the correct thing to do now is to "`dd if=/dev/hda3 of=/dev/hdc3`". I tried this, but things still don't work.

   A: You should keep your best friend away from you computer. Fortunately, no serious damage has been done. You can recover from this by running:

   ```
   mkraid raid1.conf -f --only-superblock
   ```

   By using `dd`, two identical copies of the partition were created. This is almost correct, except that the RAID-1 kernel extension expects the RAID superblocks to be different. Thus, when you try to reactivate RAID, the software will notice the problem, and deactivate one of the two partitions. By re-creating the superblock, you

TOC

should have a fully usable system.

4. **Q**: My version of `mkraid` doesn't have a `--only-superblock` flag. What do I do?

   A: The newer tools drop support for this flag, replacing it with the `--force-resync` flag. It has been reported that the following sequence appears to work with the latest tools and software:

   ```
   umount /web (where /dev/md0 was mounted on)
   raidstop /dev/md0
   mkraid /dev/md0 --force-resync --really-force
   raidstart /dev/md0
   ```

   After doing this, a `cat /proc/mdstat` should report `resync in progress`, and one should be able to `mount /dev/md0` at this point.

5. **Q**: My RAID-1 device, `/dev/md0` consists of two hard drive partitions: `/dev/hda3` and `/dev/hdc3`. My best (girl?)friend, who doesn't understand RAID, ran `fsck` on `/dev/hda3` while I wasn't looking, and now the RAID won't work. What should I do?

   A: You should re-examine your concept of ``best friend''. In general, `fsck` should never be run on the individual partitions that compose a RAID array. Assuming that neither of the partitions are/were heavily damaged, no data loss has occurred, and the RAID-1 device can be recovered as follows:

   1. make a backup of the file system on `/dev/hda3`
   2. `dd if=/dev/hda3 of=/dev/hdc3`
   3. `mkraid raid1.conf -f --only-superblock`

   This should leave you with a working disk mirror.

6. **Q**: Why does the above work as a recovery procedure?

   A: Because each of the component partitions in a RAID-1 mirror is a perfectly valid copy of the file system. In a pinch, mirroring can be disabled, and one of the partitions can be mounted and safely run as an ordinary, non-RAID file system. When you are ready to restart using RAID-1, then unmount the partition, and follow the above instructions to restore the mirror. Note that the above works ONLY for RAID-1, and not for any of the other levels.

   It may make you feel more comfortable to reverse the direction of the copy above: copy **from** the disk that was untouched **to** the one that was. Just be sure to fsck the final md.

7. **Q**: I am confused by the above questions, but am not yet bailing out. Is it safe to run `fsck /dev/md0` ?

   A: Yes, it is safe to run `fsck` on the `md` devices. In fact, this is the **only** safe place to run `fsck`.

8. **Q**: If a disk is slowly failing, will it be obvious which one it is? I am concerned that it won't be, and this confusion could lead to some dangerous decisions by a sysadmin.

   A: Once a disk fails, an error code will be returned from the low level driver to the

TOC

RAID driver. The RAID driver will mark it as ``bad'' in the RAID superblocks of the ``good'' disks (so we will later know which mirrors are good and which aren't), and continue RAID operation on the remaining operational mirrors.

This, of course, assumes that the disk and the low level driver can detect a read/write error, and will not silently corrupt data, for example. This is true of current drives (error detection schemes are being used internally), and is the basis of RAID operation.

9. **Q**: What about hot-repair?

   A: Work is underway to complete ``hot reconstruction''. With this feature, one can add several ``spare'' disks to the RAID set (be it level 1 or 4/5), and once a disk fails, it will be reconstructed on one of the spare disks in run time, without ever needing to shut down the array.

   However, to use this feature, the spare disk must have been declared at boot time, or it must be hot-added, which requires the use of special cabinets and connectors that allow a disk to be added while the electrical power is on.

   As of October 97, there is a beta version of MD that allows:

   - RAID 1 and 5 reconstruction on spare drives
   - RAID-5 parity reconstruction after an unclean shutdown
   - spare disk to be hot-added to an already running RAID 1 or 4/5 array

   By default, automatic reconstruction is (Dec 97) currently disabled by default, due to the preliminary nature of this work. It can be enabled by changing the value of `SUPPORT_RECONSTRUCTION` in `include/linux/md.h`.

   If spare drives were configured into the array when it was created and kernel-based reconstruction is enabled, the spare drive will already contain a RAID superblock (written by `mkraid`), and the kernel will reconstruct its contents automatically (without needing the usual `mdstop`, replace drive, `ckraid`, `mdrun` steps).

   If you are not running automatic reconstruction, and have not configured a hot-spare disk, the procedure described by Gadi Oxman < gadio@netvision.net.il> is recommended:

   - Currently, once the first disk is removed, the RAID set will be running in degraded mode. To restore full operation mode, you need to:
     - stop the array (`mdstop /dev/md0`)
     - replace the failed drive
     - run `ckraid raid.conf` to reconstruct its contents
     - run the array again (`mdadd`, `mdrun`).

     At this point, the array will be running with all the drives, and again protects against a failure of a single drive.

   Currently, it is not possible to assign single hot-spare disk to several arrays. Each array requires it's own hot-spare.

TOC

10. **Q**: I would like to have an audible alarm for ``you schmuck, one disk in the mirror is down'', so that the novice sysadmin knows that there is a problem.

    A: The kernel is logging the event with a ```KERN_ALERT`'' priority in syslog. There are several software packages that will monitor the syslog files, and beep the PC speaker, call a pager, send e-mail, etc. automatically.

11. **Q**: How do I run RAID-5 in degraded mode (with one disk failed, and not yet replaced)?

    A: Gadi Oxman < gadio@netvision.net.il> writes: Normally, to run a RAID-5 set of n drives you have to:

    ```
    mdadd /dev/md0 /dev/disk1 ... /dev/disk(n)
    mdrun -p5 /dev/md0
    ```

    Even if one of the disks has failed, you still have to `mdadd` it as you would in a normal setup. (?? try using /dev/null in place of the failed disk ??? watch out) Then, The array will be active in degraded mode with (n - 1) drives. If ```mdrun`'' fails, the kernel has noticed an error (for example, several faulty drives, or an unclean shutdown). Use ```dmesg`'' to display the kernel error messages from ```mdrun`''. If the raid-5 set is corrupted due to a power loss, rather than a disk crash, one can try to recover by creating a new RAID superblock:

    ```
    mkraid -f --only-superblock raid5.conf
    ```

    A RAID array doesn't provide protection against a power failure or a kernel crash, and can't guarantee correct recovery. Rebuilding the superblock will simply cause the system to ignore the condition by marking all the drives as ``OK'', as if nothing happened.

12. **Q**: How does RAID-5 work when a disk fails?

    A: The typical operating scenario is as follows:

    - A RAID-5 array is active.
    - One drive fails while the array is active.
    - The drive firmware and the low-level Linux disk/controller drivers detect the failure and report an error code to the MD driver.
    - The MD driver continues to provide an error-free `/dev/md0` device to the higher levels of the kernel (with a performance degradation) by using the remaining operational drives.
    - The sysadmin can `umount /dev/md0` and `mdstop /dev/md0` as usual.
    - If the failed drive is not replaced, the sysadmin can still start the array in degraded mode as usual, by running `mdadd` and `mdrun`.

13. **Q**:

    A:

14. **Q**: Why is there no question 13?

TOC

A: If you are concerned about RAID, High Availability, and UPS, then its probably a good idea to be superstitious as well. It can't hurt, can it?

15. **Q**: I just replaced a failed disk in a RAID-5 array. After rebuilding the array, `fsck` is reporting many, many errors. Is this normal?

    A: No. And, unless you ran fsck in "verify only; do not update" mode, its quite possible that you have corrupted your data. Unfortunately, a not-uncommon scenario is one of accidentally changing the disk order in a RAID-5 array, after replacing a hard drive. Although the RAID superblock stores the proper order, not all tools use this information. In particular, the current version of `ckraid` will use the information specified with the `-f` flag (typically, the file `/etc/raid5.conf`) instead of the data in the superblock. If the specified order is incorrect, then the replaced disk will be reconstructed incorrectly. The symptom of this kind of mistake seems to be heavy & numerous `fsck` errors.

    And, in case you are wondering, **yes**, someone lost **all** of their data by making this mistake. Making a tape backup of **all** data before reconfiguring a RAID array is **strongly recommended**.

16. **Q**: The QuickStart says that `mdstop` is just to make sure that the disks are sync'ed. Is this REALLY necessary? Isn't unmounting the file systems enough?

    A: The command `mdstop /dev/md0` will:

    - mark it "clean". This allows us to detect unclean shutdowns, for example due to a power failure or a kernel crash.
    - sync the array. This is less important after unmounting a filesystem, but is important if the `/dev/md0` is accessed directly rather than through a filesystem (for example, by `e2fsck`).

---

5. Troubleshooting Install Problems

1. **Q**: What is the current best known-stable patch for RAID in the 2.0.x series kernels?

   A: As of 18 Sept 1997, it is "2.0.30 + pre-9 2.0.31 + Werner Fink's swapping patch + the alpha RAID patch". As of November 1997, it is 2.0.31 + ... !?

2. **Q**: The RAID patches will not install cleanly for me. What's wrong?

   A: Make sure that `/usr/include/linux` is a symbolic link to `/usr/src/linux/include/linux`. Make sure that the new files `raid5.c`, etc. have been copied to their correct locations. Sometimes the patch command will not create new files. Try the `-f` flag on `patch`.

3. **Q**: While compiling raidtools 0.42, compilation stops trying to include <pthread.h> but it doesn't exist in my system. How do I fix this?

   A: raidtools-0.42 requires linuxthreads-0.6 from: ftp://ftp.inria.fr/INRIA/Projects/cristal/Xavier.Leroy Alternately, use glibc v2.0.

4. **Q**: I get the message: `mdrun -a /dev/md0: Invalid argument`

   A: Use `mkraid` to initialize the RAID set prior to the first use. `mkraid` ensures that the RAID array is initially in a consistent state by erasing the RAID partitions. In addition, `mkraid` will create the RAID superblocks.

5. **Q**: I get the message: `mdrun -a /dev/md0: Invalid argument` The setup was:
   - raid build as a kernel module
   - normal install procedure followed ... mdcreate, mdadd, etc.
   - `cat /proc/mdstat` shows

     ```
     Personalities :
     read_ahead not set
     md0 : inactive sda1 sdb1 6313482 blocks
     md1 : inactive
     md2 : inactive
     md3 : inactive
     ```

   - `mdrun -a` generates the error message `/dev/md0: Invalid argument`

     A: Try `lsmod` (or, alternately, `cat /proc/modules`) to see if the raid modules are loaded. If they are not, you can load them explicitly with the `modprobe raid1` or `modprobe raid5` command. Alternately, if you are using the autoloader, and expected `kerneld` to load them and it didn't this is probably because your loader is missing the info to load the modules. Edit `/etc/conf.modules` and add the following lines:

```
alias md-personality-3 raid1
alias md-personality-4 raid5
```

TOC

6. **Q**: While doing `mdadd -a` I get the error: `/dev/md0: No such file or directory`. Indeed, there seems to be no `/dev/md0` anywhere. Now what do I do?

   A: The raid-tools package will create these devices when you run `make install` as root. Alternately, you can do the following:

   ```
   cd /dev
   ./MAKEDEV md
   ```

7. **Q**: After creating a raid array on `/dev/md0`, I try to mount it and get the following error: `mount: wrong fs type, bad option, bad superblock on /dev/md0, or too many mounted file systems`. What's wrong?

   A: You need to create a file system on `/dev/md0` before you can mount it. Use `mke2fs`.

8. **Q**: Truxton Fulton wrote:

   On my Linux 2.0.30 system, while doing a `mkraid` for a RAID-1 device, during the clearing of the two individual partitions, I got `"Cannot allocate free page"` errors appearing on the console, and `"Unable to handle kernel paging request at virtual address ..."` errors in the system log. At this time, the system became quite unusable, but it appears to recover after a while. The operation appears to have completed with no other errors, and I am successfully using my RAID-1 device. The errors are disconcerting though. Any ideas?

   A: This was a well-known bug in the 2.0.30 kernels. It is fixed in the 2.0.31 kernel; alternately, fall back to 2.0.29.

9. **Q**: I'm not able to `mdrun` a RAID-1, RAID-4 or RAID-5 device. If I try to `mdrun` a `mdadd`'ed device I get the message "`invalid raid superblock magic`".

   A: Make sure that you've run the `mkraid` part of the install procedure.

10. **Q**: When I access `/dev/md0`, the kernel spits out a lot of errors like `md0: device not running, giving up !` and `I/O error...`. I've successfully added my devices to the virtual device.

    A: To be usable, the device must be running. Use `mdrun -px /dev/md0` where x is l for linear, 0 for RAID-0 or 1 for RAID-1, etc.

11. **Q**: I've created a linear md-dev with 2 devices. `cat /proc/mdstat` shows the total size of the device, but `df` only shows the size of the first physical device.

    A: You must `mkfs` your new md-dev before using it the first time, so that the filesystem will cover the whole device.

12. **Q**: I've set up `/etc/mdtab` using mdcreate, I've `mdadd`'ed, `mdrun` and `fsck`'ed my two `/dev/mdX` partitions. Everything looks okay before a reboot. As soon as I reboot, I get an `fsck` error on both partitions: `fsck.ext2: Attempt to read block from filesystem resulted in short read while trying too open /dev/md0`. Why?! How do I

fix it?!

A: During the boot process, the RAID partitions must be started before they can be `fsck`'ed. This must be done in one of the boot scripts. For some distributions, `fsck` is called from `/etc/rc.d/rc.S`, for others, it is called from `/etc/rc.d/rc.sysinit`. Change this file to `mdadd -ar` *before* `fsck -A` is executed. Better yet, it is suggested that `ckraid` be run if `mdadd` returns with an error. How do do this is discussed in greater detail in question 14 of the section "Error Recovery".

13. **Q**: I get the message `invalid raid superblock magic` while trying to run an array which consists of partitions which are bigger than 4GB.

    A: This bug is now fixed. (September 97) Make sure you have the latest raid code.

14. **Q**: I get the message `Warning: could not write 8 blocks in inode table starting at 2097175` while trying to run mke2fs on a partition which is larger than 2GB.

    A: This seems to be a problem with `mke2fs` (November 97). A temporary work-around is to get the mke2fs code, and add `#undef HAVE_LLSEEK` to `e2fsprogs-1.10/lib/ext2fs/llseek.c` just before the first `#ifdef HAVE_LLSEEK` and recompile mke2fs.

15. **Q**: `ckraid` currently isn't able to read `/etc/mdtab`

    A: The RAID0/linear configuration file format used in `/etc/mdtab` is obsolete, although it will be supported for a while more. The current, up-to-date config files are currently named `/etc/raid1.conf`, etc.

16. **Q**: The personality modules (`raid1.o`) are not loaded automatically; they have to be manually modprobe'd before mdrun. How can this be fixed?

    A: To autoload the modules, we can add the following to `/etc/conf.modules`:

```
alias md-personality-3 raid1
alias md-personality-4 raid5
```

17. **Q**: I've `mdadd`'ed 13 devices, and now I'm trying to `mdrun -p5 /dev/md0` and get the message: `/dev/md0: Invalid argument`

    A: The default configuration for software RAID is 8 real devices. Edit `linux/md.h`, change `#define MAX_REAL=8` to a larger number, and rebuild the kernel.

18. **Q**: I can't make `md` work with partitions on our latest SPARCstation 5. I suspect that this has something to do with disk-labels.

    A: Sun disk-labels sit in the first 1K of a partition. For RAID-1, the Sun disk-label is not an issue since `ext2fs` will skip the label on every mirror. For other raid levels (0, linear and 4/5), this appears to be a problem; it has not yet (Dec 97) been addressed.

6. Supported Hardware & Software

1. **Q**: I have SCSI adapter brand XYZ (with or without several channels), and disk brand(s) PQR and LMN, will these work with md to create a linear/stripped/mirrored personality?

   A: Yes! Software RAID will work with any disk controller (IDE or SCSI) and any disks. The disks do not have to be identical, nor do the controllers. For example, a RAID mirror can be created with one half the mirror being a SCSI disk, and the other an IDE disk. The disks do not even have to be the same size. There are no restrictions on the mixing & matching of disks and controllers.

   This is because Software RAID works with disk partitions, not with the raw disks themselves. The only recommendation is that for RAID levels 1 and 5, the disk partitions that are used as part of the same set be the same size. If the partitions used to make up the RAID 1 or 5 array are not the same size, then the excess space in the larger partitions is wasted (not used).

2. **Q**: I have a twin channel BT-952, and the box states that it supports hardware RAID 0, 1 and 0+1. I have made a RAID set with two drives, the card apparently recognizes them when it's doing it's BIOS startup routine. I've been reading in the driver source code, but found no reference to the hardware RAID support. Anybody out there working on that?

   A: The Mylex/BusLogic FlashPoint boards with RAIDPlus are actually software RAID, not hardware RAID at all. RAIDPlus is only supported on Windows 95 and Windows NT, not on Netware or any of the Unix platforms. Aside from booting and configuration, the RAID support is actually in the OS drivers.

   While in theory Linux support for RAIDPlus is possible, the implementation of RAID-0/1/4/5 in the Linux kernel is much more flexible and should have superior performance, so there's little reason to support RAIDPlus directly.

3. **Q**: I want to run RAID with an SMP box. Is RAID SMP-safe?

   A: "I think so" is the best answer available at the time I write this (April 98). A number of users report that they have been using RAID with SMP for nearly a year, without problems. However, as of April 98 (circa kernel 2.1.9x), the following problems have been noted on the mailing list:

   - Adaptec AIC7xxx SCSI drivers are not SMP safe (General note: Adaptec adapters have a long & lengthly history of problems & flakiness in general. Although they seem to be the most easily available, widespread and cheapest SCSI adapters, they should be avoided. After factoring for time lost, frustration, and corrupted data, Adaptec's will prove to be the costliest mistake you'll ever make. That said, if you have SMP problems with 2.1.88, try the patch ftp://ftp.bero-online.ml.org/pub/linux/aic7xxx-5.0.7-linux21.tar.gz I am not sure if this patch has been pulled into later 2.1.x kernels. For further info, take a look at the mail archives for March 98 at

http://www.linuxhq.com/lnxlists/linux-raid/lr_9803_01/ As usual, due to the rapidly changing nature of the latest experimental 2.1.x kernels, the problems described in these mailing lists may or may not have been fixed by the time your read this. Caveat Emptor.)

- IO-APIC with RAID-0 on SMP has been reported to crash in 2.1.90

---

7. Modifying an Existing Installation

1. **Q**: Are linear MD's expandable? Can a new hard-drive/partition be added, and the size of the existing file system expanded?

   A: Miguel de Icaza < miguel@luthien.nuclecu.unam.mx> writes:

   > I changed the ext2fs code to be aware of multiple-devices instead of the regular one device per file system assumption.
   >
   > So, when you want to extend a file system, you run a utility program that makes the appropriate changes on the new device (your extra partition) and then you just tell the system to extend the fs using the specified device.
   >
   > You can extend a file system with new devices at system operation time, no need to bring the system down (and whenever I get some extra time, you will be able to remove devices from the ext2 volume set, again without even having to go to single-user mode or any hack like that).
   >
   > You can get the patch for 2.1.x kernel from my web page:
   >
   > `http://www.nuclecu.unam.mx/~miguel/ext2-volume`

2. **Q**: Can I add disks to a RAID-5 array?

   A: Currently, (September 1997) no, not without erasing all data. A conversion utility to allow this does not yet exist. The problem is that the actual structure and layout of a RAID-5 array depends on the number of disks in the array. Of course, one can add drives by backing up the array to tape, deleting all data, creating a new array, and restoring from tape.

3. **Q**: What would happen to my RAID1/RAID0 sets if I shift one of the drives from being `/dev/hdb` to `/dev/hdc`? Because of cabling/case size/stupidity issues, I had to make my RAID sets on the same IDE controller (`/dev/hda` and `/dev/hdb`). Now that I've fixed some stuff, I want to move `/dev/hdb` to `/dev/hdc`. What would happen if I just change the `/etc/mdtab` and `/etc/raid1.conf` files to reflect the new location?

   A: For RAID-0/linear, one must be careful to specify the drives in exactly the same order. Thus, in the above example, if the original config is

   ```
   mdadd /dev/md0 /dev/hda /dev/hdb
   ```

   Then the new config *must* be

   ```
   mdadd /dev/md0 /dev/hda /dev/hdc
   ```

   For RAID-1/4/5, the drive's "RAID number" is stored in its RAID superblock, and therefore the order in which the disks are specified is not important. RAID-0/linear does not have a superblock due to it's older design, and the desire to maintain backwards compatibility with this older design.

4. **Q**: Can I convert a two-disk RAID-1 mirror to a three-disk RAID-5 array?

   A: Yes. Michael at BizSystems has come up with a clever, sneaky way of doing this. However, like virtually all manipulations of RAID arrays once they have data on them, it is dangerous and prone to human error. **Make a backup before you start**.

TOC

```
I will make the following assumptions:
----------------------------------------------
disks
original: hda - hdc
raid1 partitions hda3 - hdc3
array name /dev/md0

new hda - hdc - hdd
raid5 partitions hda3 - hdc3 - hdd3
array name: /dev/md1

You must substitute the appropriate disk and partition numbers for
you system configuration. This will hold true for all config file
examples.
----------------------------------------------
DO A BACKUP BEFORE YOU DO ANYTHING
1) recompile kernel to include both raid1 and raid5
2) install new kernel and verify that raid personalities are present
3) disable the redundant partition on the raid 1 array. If this is a
 root mounted partition (mine was) you must be more careful.

 Reboot the kernel without starting raid devices or boot from rescue
 system ( raid tools must be available )

 start non-redundant raid1
mdadd -r -p1 /dev/md0 /dev/hda3

4) configure raid5 but with 'funny' config file, note that there is
  no hda3 entry and hdc3 is repeated. This is needed since the
  raid tools don't want you to do this.
--------------------------------
# raid-5 configuration
raiddev                 /dev/md1
raid-level              5
nr-raid-disks           3
chunk-size              32

# Parity placement algorithm
parity-algorithm        left-symmetric

# Spare disks for hot reconstruction
nr-spare-disks          0

device                  /dev/hdc3
raid-disk               0

device                  /dev/hdc3
raid-disk               1
```

```
device                  /dev/hdd3
raid-disk               2
----------------------------------------
 mkraid /etc/raid5.conf
5) activate the raid5 array in non-redundant mode

mdadd -r -p5 -c32k /dev/md1 /dev/hdc3 /dev/hdd3

6) make a file system on the array

mke2fs -b {blocksize} /dev/md1

recommended blocksize by some is 4096 rather than the default 1024.
this improves the memory utilization for the kernel raid routines and
matches the blocksize to the page size. I compromised and used 2048
since I have a relatively high number of small files on my system.

7) mount the two raid devices somewhere

mount -t ext2 /dev/md0 mnt0
mount -t ext2 /dev/md1 mnt1

8) move the data

cp -a mnt0 mnt1

9) verify that the data sets are identical
10) stop both arrays
11) correct the information for the raid5.conf file
  change /dev/md1 to /dev/md0
  change the first disk to read /dev/hda3

12) upgrade the new array to full redundant status
 (THIS DESTROYS REMAINING raid1 INFORMATION)

ckraid --fix /etc/raid5.conf
```

8. Performance, Tools & General Bone-headed Questions

1. **Q**: I've created a RAID-0 device on `/dev/sda2` and `/dev/sda3`. The device is a lot slower than a single partition. Isn't md a pile of junk?

   A: To have a RAID-0 device running a full speed, you must have partitions from different disks. Besides, putting the two halves of the mirror on the same disk fails to give you any protection whatsoever against disk failure.

2. **Q**: What's the use of having RAID-linear when RAID-0 will do the same thing, but provide higher performance?

   A: It's not obvious that RAID-0 will always provide better performance; in fact, in some cases, it could make things worse. The ext2fs file system scatters files all over a partition, and it attempts to keep all of the blocks of a file contiguous, basically in an attempt to prevent fragmentation. Thus, ext2fs behaves "as if" there were a (variable-sized) stripe per file. If there are several disks concatenated into a single RAID-linear, this will result files being statistically distributed on each of the disks. Thus, at least for ext2fs, RAID-linear will behave a lot like RAID-0 with large stripe sizes. Conversely, RAID-0 with small stripe sizes can cause excessive disk activity leading to severely degraded performance if several large files are accessed simultaneously.

   In many cases, RAID-0 can be an obvious win. For example, imagine a large database file. Since ext2fs attempts to cluster together all of the blocks of a file, chances are good that it will end up on only one drive if RAID-linear is used, but will get chopped into lots of stripes if RAID-0 is used. Now imagine a number of (kernel) threads all trying to random access to this database. Under RAID-linear, all accesses would go to one disk, which would not be as efficient as the parallel accesses that RAID-0 entails.

3. **Q**: How does RAID-0 handle a situation where the different stripe partitions are different sizes? Are the stripes uniformly distributed?

   A: To understand this, lets look at an example with three partitions; one that is 50MB, one 90MB and one 125MB. Lets call D0 the 50MB disk, D1 the 90MB disk and D2 the 125MB disk. When you start the device, the driver calculates 'strip zones'. In this case, it finds 3 zones, defined like this:

```
Z0 : (D0/D1/D2) 3 x 50 = 150MB  total in this zone
Z1 : (D1/D2)  2 x 40 = 80MB total in this zone
Z2 : (D2) 125-50-40 = 35MB total in this zone.
```

   You can see that the total size of the zones is the size of the virtual device, but, depending on the zone, the striping is different. Z2 is rather inefficient, since there's

TOC

only one disk. Since `ext2fs` and most other Unix file systems distribute files all over the disk, you have a 35/265 = 13% chance that a fill will end up on Z2, and not get any of the benefits of striping. (DOS tries to fill a disk from beginning to end, and thus, the oldest files would end up on Z0. However, this strategy leads to severe filesystem fragmentation, which is why no one besides DOS does it this way.)

4. **Q**: I have some Brand X hard disks and a Brand Y controller. and am considering using `md`. Does it significantly increase the throughput? Is the performance really noticeable?

   A: The answer depends on the configuration that you use.

   Linux MD RAID-0 and RAID-linear performance:

   If the system is heavily loaded with lots of I/O, statistically, some of it will go to one disk, and some to the others. Thus, performance will improve over a single large disk. The actual improvement depends a lot on the actual data, stripe sizes, and other factors. In a system with low I/O usage, the performance is equal to that of a single disk.

   Linux MD RAID-1 (mirroring) read performance:

   MD implements read balancing. That is, the RAID-1 code will alternate between each of the (two or more) disks in the mirror, making alternate reads to each. In a low-I/O situation, this won't change performance at all: you will have to wait for one disk to complete the read. But, with two disks in a high-I/O environment, this could as much as double the read performance, since reads can be issued to each of the disks in parallel. For N disks in the mirror, this could improve performance N-fold.

   Linux MD RAID-1 (mirroring) write performance:

   Must wait for the write to occur to all of the disks in the mirror. This is because a copy of the data must be written to each of the disks in the mirror. Thus, performance will be roughly equal to the write performance to a single disk.

   Linux MD RAID-4/5 read performance:

   Statistically, a given block can be on any one of a number of disk drives, and thus RAID-4/5 read performance is a lot like that for RAID-0. It will depend on the data, the stripe size, and the application. It will not be as good as the read performance of a mirrored array.

   Linux MD RAID-4/5 write performance:

   This will in general be considerably slower than that for a single disk. This is because the parity must be written out to one drive as well as the data to another. However, in order to compute the new parity, the old parity and the old data must be read first. The old data, new data and old parity must all be XOR'ed together to determine the new parity: this requires considerable CPU cycles in addition to the numerous disk accesses.

5. **Q**: What RAID configuration should I use for optimal performance?

   A: Is the goal to maximize throughput, or to minimize latency? There is no easy

TOC

answer, as there are many factors that affect performance:

- operating system - will one process/thread, or many be performing disk access?
- application - is it accessing data in a sequential fashion, or random access?
- file system - clusters files or spreads them out (the ext2fs clusters together the blocks of a file, and spreads out files)
- disk driver - number of blocks to read ahead (this is a tunable parameter)
- CEC hardware - one drive controller, or many?
- hd controller - able to queue multiple requests or not? Does it provide a cache?
- hard drive - buffer cache memory size -- is it big enough to handle the write sizes and rate you want?
- physical platters - blocks per cylinder -- accessing blocks on different cylinders will lead to seeks.

6. **Q**: What is the optimal RAID-5 configuration for performance?

   A: Since RAID-5 experiences an I/O load that is equally distributed across several drives, the best performance will be obtained when the RAID set is balanced by using identical drives, identical controllers, and the same (low) number of drives on each controller. Note, however, that using identical components will raise the probability of multiple simultaneous failures, for example due to a sudden jolt or drop, overheating, or a power surge during an electrical storm. Mixing brands and models helps reduce this risk.

7. **Q**: What is the optimal block size for a RAID-4/5 array?

   A: When using the current (November 1997) RAID-4/5 implementation, it is strongly recommended that the file system be created with `mke2fs -b 4096` instead of the default 1024 byte filesystem block size.

   This is because the current RAID-5 implementation allocates one 4K memory page per disk block; if a disk block were just 1K in size, then 75% of the memory which RAID-5 is allocating for pending I/O would not be used. If the disk block size matches the memory page size, then the driver can (potentially) use all of the page. Thus, for a filesystem with a 4096 block size as opposed to a 1024 byte block size, the RAID driver will potentially queue 4 times as much pending I/O to the low level drivers without allocating additional memory.

   Note: the above remarks do NOT apply to Software RAID-0/1/linear driver.

   Note: the statements about 4K memory page size apply to the Intel x86 architecture. The page size on Alpha, Sparc, and other CPUS are different; I believe they're 8K on Alpha/Sparc (????). Adjust the above figures accordingly.

   Note: if your file system has a lot of small files (files less than 10KBytes in size), a considerable fraction of the disk space might be wasted. This is because the file system allocates disk space in multiples of the block size. Allocating large blocks for small files clearly results in a waste of disk space: thus, you may want to stick to small block sizes, get a larger effective storage capacity, and not worry about the

TOC

"wasted" memory due to the block-size/page-size mismatch.

Note: most "typical" systems do not have that many small files. That is, although there might be thousands of small files, this would lead to only some 10 to 100MB wasted space, which is probably an acceptable tradeoff for performance on a multi-gigabyte disk.

However, for news servers, there might be tens or hundreds of thousands of small files. In such cases, the smaller block size, and thus the improved storage capacity, may be more important than the more efficient I/O scheduling.

Note: there exists an experimental file system for Linux which packs small files and file chunks onto a single block. It apparently has some very positive performance implications when the average file size is much smaller than the block size.

Note: Future versions may implement schemes that obsolete the above discussion. However, this is difficult to implement, since dynamic run-time allocation can lead to dead-locks; the current implementation performs a static pre-allocation.

8. **Q**: How does the chunk size (stripe size) influence the speed of my RAID-0, RAID-4 or RAID-5 device?

   A: The chunk size is the amount of data contiguous on the virtual device that is also contiguous on the physical device. In this HOWTO, "chunk" and "stripe" refer to the same thing: what is commonly called the "stripe" in other RAID documentation is called the "chunk" in the MD man pages. Stripes or chunks apply only to RAID 0, 4 and 5, since stripes are not used in mirroring (RAID-1) and simple concatenation (RAID-linear). The stripe size affects both read and write latency (delay), throughput (bandwidth), and contention between independent operations (ability to simultaneously service overlapping I/O requests).

   Assuming the use of the ext2fs file system, and the current kernel policies about read-ahead, large stripe sizes are almost always better than small stripe sizes, and stripe sizes from about a fourth to a full disk cylinder in size may be best. To understand this claim, let us consider the effects of large stripes on small files, and small stripes on large files. The stripe size does not affect the read performance of small files: For an array of N drives, the file has a 1/N probability of being entirely within one stripe on any one of the drives. Thus, both the read latency and bandwidth will be comparable to that of a single drive. Assuming that the small files are statistically well distributed around the filesystem, (and, with the ext2fs file system, they should be), roughly N times more overlapping, concurrent reads should be possible without significant collision between them. Conversely, if very small stripes are used, and a large file is read sequentially, then a read will issued to all of the disks in the array. For a the read of a single large file, the latency will almost double, as the probability of a block being 3/4'ths of a revolution or farther away will increase. Note, however, the trade-off: the bandwidth could improve almost N-fold for reading a single, large file, as N drives can be reading simultaneously (that is, if read-ahead is used so that all of the disks are kept active). But there is another, counter-acting trade-off: if all of the drives are already busy reading one file, then attempting to read

TOC

a second or third file at the same time will cause significant contention, ruining performance as the disk ladder algorithms lead to seeks all over the platter. Thus, large stripes will almost always lead to the best performance. The sole exception is the case where one is streaming a single, large file at a time, and one requires the top possible bandwidth, and one is also using a good read-ahead algorithm, in which case small stripes are desired.

Note that this HOWTO previously recommended small stripe sizes for news spools or other systems with lots of small files. This was bad advice, and here's why: news spools contain not only many small files, but also large summary files, as well as large directories. If the summary file is larger than the stripe size, reading it will cause many disks to be accessed, slowing things down as each disk performs a seek. Similarly, the current ext2fs file system searches directories in a linear, sequential fashion. Thus, to find a given file or inode, on average half of the directory will be read. If this directory is spread across several stripes (several disks), the directory read (e.g. due to the ls command) could get very slow. Thanks to Steven A. Reisman < sar@pressenter.com> for this correction. Steve also adds:

> I found that using a 256k stripe gives much better performance. I suspect that the optimum size would be the size of a disk cylinder (or maybe the size of the disk drive's sector cache). However, disks nowadays have recording zones with different sector counts (and sector caches vary among different disk models). There's no way to guarantee stripes won't cross a cylinder boundary.

The tools accept the stripe size specified in KBytes. You'll want to specify a multiple of if the page size for your CPU (4KB on the x86).

9. **Q**: What is the correct stride factor to use when creating the ext2fs file system on the RAID partition? By stride, I mean the -R flag on the `mke2fs` command:

```
mke2fs -b 4096 -R stride=nnn  ...
```

What should the value of nnn be?

A: The `-R stride` flag is used to tell the file system about the size of the RAID stripes. Since only RAID-0,4 and 5 use stripes, and RAID-1 (mirroring) and RAID-linear do not, this flag is applicable only for RAID-0,4,5. Knowledge of the size of a stripe allows `mke2fs` to allocate the block and inode bitmaps so that they don't all end up on the same physical drive. An unknown contributor wrote:

> I noticed last spring that one drive in a pair always had a larger I/O count, and tracked it down to the these meta-data blocks. Ted added the `-R stride=` option in response to my explanation and request for a workaround.

For a 4KB block file system, with stripe size 256KB, one would use `-R stride=64`.

If you don't trust the `-R` flag, you can get a similar effect in a different way. Steven A.

TOC

Reisman < sar@pressenter.com> writes:

> Another consideration is the filesystem used on the RAID-0 device. The ext2 filesystem allocates 8192 blocks per group. Each group has its own set of inodes. If there are 2, 4 or 8 drives, these inodes cluster on the first disk. I've distributed the inodes across all drives by telling mke2fs to allocate only 7932 blocks per group.

Some mke2fs pages do not describe the `[-g blocks-per-group]` flag used in this operation.

10. **Q**: Where can I put the `md` commands in the startup scripts, so that everything will start automatically at boot time?

    A: Rod Wilkens < rwilkens@border.net> writes:

    > What I did is put ``mdadd -ar'' in the ``/etc/rc.d/rc.sysinit'' right after the kernel loads the modules, and before the ``fsck'' disk check. This way, you can put the ``/dev/md?'' device in the ``/etc/fstab''. Then I put the ``mdstop -a'' right after the ``umount -a'' unmounting the disks, in the ``/etc/rc.d/init.d/halt'' file.

    For raid-5, you will want to look at the return code for `mdadd`, and if it failed, do a

    ```
    ckraid --fix /etc/raid5.conf
    ```

    to repair any damage.

11. **Q**: I was wondering if it's possible to setup striping with more than 2 devices in `md0`? This is for a news server, and I have 9 drives... Needless to say I need much more than two. Is this possible?

    A: Yes. (describe how to do this)

12. **Q**: When is Software RAID superior to Hardware RAID?

    A: Normally, Hardware RAID is considered superior to Software RAID, because hardware controllers often have a large cache, and can do a better job of scheduling operations in parallel. However, integrated Software RAID can (and does) gain certain advantages from being close to the operating system.

    For example, ... ummm. Opaque description of caching of reconstructed blocks in buffer cache elided ...

    On a dual PPro SMP system, it has been reported that Software-RAID performance exceeds the performance of a well-known hardware-RAID board vendor by a factor of 2 to 5.

    Software RAID is also a very interesting option for high-availability redundant server systems. In such a configuration, two CPU's are attached to one set or SCSI disks. If one server crashes or fails to respond, then the other server can `mdadd`, `mdrun` and `mount` the software RAID array, and take over operations. This sort of dual-ended operation is not always possible with many hardware RAID controllers, because of

TOC

the state configuration that the hardware controllers maintain.

13. **Q**: If I upgrade my version of raidtools, will it have trouble manipulating older raid arrays? In short, should I recreate my RAID arrays when upgrading the raid utilities?

    A: No, not unless the major version number changes. An MD version x.y.z consists of three sub-versions:

    ```
    x:      Major version.
    y:      Minor version.
    z:      Patchlevel version.
    ```

    Version x1.y1.z1 of the RAID driver supports a RAID array with version x2.y2.z2 in case (x1 == x2) and (y1 >= y2). Different patchlevel (z) versions for the same (x.y) version are designed to be mostly compatible.

    The minor version number is increased whenever the RAID array layout is changed in a way which is incompatible with older versions of the driver. New versions of the driver will maintain compatibility with older RAID arrays.

    The major version number will be increased if it will no longer make sense to support old RAID arrays in the new kernel code.

    For RAID-1, it's not likely that the disk layout nor the superblock structure will change anytime soon. Most all Any optimization and new features (reconstruction, multithreaded tools, hot-plug, etc.) doesn't affect the physical layout.

14. **Q**: The command `mdstop /dev/md0` says that the device is busy.

    A: There's a process that has a file open on `/dev/md0`, or `/dev/md0` is still mounted. Terminate the process or `umount /dev/md0`.

15. **Q**: Are there performance tools?

    A: There is also a new utility called `iotrace` in the `linux/iotrace` directory. It reads `/proc/io-trace` and analyses/plots it's output. If you feel your system's block IO performance is too low, just look at the iotrace output.

16. **Q**: I was reading the RAID source, and saw the value `SPEED_LIMIT` defined as 1024K/sec. What does this mean? Does this limit performance?

    A: `SPEED_LIMIT` is used to limit RAID reconstruction speed during automatic reconstruction. Basically, automatic reconstruction allows you to `e2fsck` and `mount` immediately after an unclean shutdown, without first running `ckraid`. Automatic reconstruction is also used after a failed hard drive has been replaced.

    In order to avoid overwhelming the system while reconstruction is occurring, the reconstruction thread monitors the reconstruction speed and slows it down if its too fast. The 1M/sec limit was arbitrarily chosen as a reasonable rate which allows the reconstruction to finish reasonably rapidly, while creating only a light load on the system so that other processes are not interfered with.

17. **Q**: What about "spindle synchronization" or "disk synchronization"?

TOC

A: Spindle synchronization is used to keep multiple hard drives spinning at exactly the same speed, so that their disk platters are always perfectly aligned. This is used by some hardware controllers to better organize disk writes. However, for software RAID, this information is not used, and spindle synchronization might even hurt performance.

18. **Q**: How can I set up swap spaces using raid 0? Wouldn't striped swap ares over 4+ drives be really fast?

    A: Leonard N. Zubkoff replies: It is really fast, but you don't need to use MD to get striped swap. The kernel automatically stripes across equal priority swap spaces. For example, the following entries from `/etc/fstab` stripe swap space across five drives in three groups:

```
/dev/sdg1       swap    swap    pri=3
/dev/sdk1       swap    swap    pri=3
/dev/sdd1       swap    swap    pri=3
/dev/sdh1       swap    swap    pri=3
/dev/sdl1       swap    swap    pri=3
/dev/sdg2       swap    swap    pri=2
/dev/sdk2       swap    swap    pri=2
/dev/sdd2       swap    swap    pri=2
/dev/sdh2       swap    swap    pri=2
/dev/sdl2       swap    swap    pri=2
/dev/sdg3       swap    swap    pri=1
/dev/sdk3       swap    swap    pri=1
/dev/sdd3       swap    swap    pri=1
/dev/sdh3       swap    swap    pri=1
/dev/sdl3       swap    swap    pri=1
```

19. **Q**: I want to maximize performance. Should I use multiple controllers?

    A: In many cases, the answer is yes. Using several controllers to perform disk access in parallel will improve performance. However, the actual improvement depends on your actual configuration. For example, it has been reported (Vaughan Pratt, January 98) that a single 4.3GB Cheetah attached to an Adaptec 2940UW can achieve a rate of 14MB/sec (without using RAID). Installing two disks on one controller, and using a RAID-0 configuration results in a measured performance of 27 MB/sec.

    Note that the 2940UW controller is an "Ultra-Wide" SCSI controller, capable of a theoretical burst rate of 40MB/sec, and so the above measurements are not surprising. However, a slower controller attached to two fast disks would be the bottleneck. Note also, that most out-board SCSI enclosures (e.g. the kind with hot-pluggable trays) cannot be run at the 40MB/sec rate, due to cabling and electrical noise problems.

    If you are designing a multiple controller system, remember that most disks and controllers typically run at 70-85% of their rated max speeds.

    Note also that using one controller per disk can reduce the likelihood of system outage

due to a controller or cable failure (In theory -- only if the device driver for the controller can gracefully handle a broken controller. Not all SCSI device drivers seem to be able to handle such a situation without panicking or otherwise locking up).

9. High Availability RAID

1. **Q**: RAID can help protect me against data loss. But how can I also ensure that the system is up as long as possible, and not prone to breakdown? Ideally, I want a system that is up 24 hours a day, 7 days a week, 365 days a year.

   A: High-Availability is difficult and expensive. The harder you try to make a system be fault tolerant, the harder and more expensive it gets. The following hints, tips, ideas and unsubstantiated rumors may help you with this quest.

   - IDE disks can fail in such a way that the failed disk on an IDE ribbon can also prevent the good disk on the same ribbon from responding, thus making it look as if two disks have failed. Since RAID does not protect against two-disk failures, one should either put only one disk on an IDE cable, or if there are two disks, they should belong to different RAID sets.
   - SCSI disks can fail in such a way that the failed disk on a SCSI chain can prevent any device on the chain from being accessed. The failure mode involves a short of the common (shared) device ready pin; since this pin is shared, no arbitration can occur until the short is removed. Thus, no two disks on the same SCSI chain should belong to the same RAID array.
   - Similar remarks apply to the disk controllers. Don't load up the channels on one controller; use multiple controllers.
   - Don't use the same brand or model number for all of the disks. It is not uncommon for severe electrical storms to take out two or more disks. (Yes, we all use surge suppressors, but these are not perfect either). Heat & poor ventilation of the disk enclosure are other disk killers. Cheap disks often run hot. Using different brands of disk & controller decreases the likelihood that whatever took out one disk (heat, physical shock, vibration, electrical surge) will also damage the others on the same date.
   - To guard against controller or CPU failure, it should be possible to build a SCSI disk enclosure that is "twin-tailed": i.e. is connected to two computers. One computer will mount the file-systems read-write, while the second computer will mount them read-only, and act as a hot spare. When the hot-spare is able to determine that the master has failed (e.g. through a watchdog), it will cut the power to the master (to make sure that it's really off), and then fsck & remount read-write. If anyone gets this working, let me know.
   - Always use an UPS, and perform clean shutdowns. Although an unclean shutdown may not damage the disks, running ckraid on even small-ish arrays is painfully slow. You want to avoid running ckraid as much as possible. Or you can hack on the kernel and get the hot-reconstruction code debugged ...
   - SCSI cables are well-known to be very temperamental creatures, and prone to cause all sorts of problems. Use the highest quality cabling that you can find for

TOC

sale. Use e.g. bubble-wrap to make sure that ribbon cables to not get too close to one another and cross-talk. Rigorously observe cable-length restrictions.

- Take a look at SSI (Serial Storage Architecture). Although it is rather expensive, it is rumored to be less prone to the failure modes that SCSI exhibits.
- Enjoy yourself, its later than you think.

---

10. Questions Waiting for Answers

1. **Q**: If, for cost reasons, I try to mirror a slow disk with a fast disk, is the S/W smart enough to balance the reads accordingly or will it all slow down to the speed of the slowest?
2. **Q**: For testing the raw disk thru put... is there a character device for raw read/raw writes instead of `/dev/sdaxx` that we can use to measure performance on the raid drives?? is there a GUI based tool to use to watch the disk thru-put??

11. Wish List of Enhancements to MD and Related Software

Bradley Ward Allen < ulmo@Q.Net> wrote:

Ideas include:

- Boot-up parameters to tell the kernel which devices are to be MD devices (no more ``mdadd")
- Making MD transparent to ``mount"/``umount" such that there is no ``mdrun" and ``mdstop"
- Integrating ``ckraid" entirely into the kernel, and letting it run as needed

(So far, all I've done is suggest getting rid of the tools and putting them into the kernel; that's how I feel about it, this is a filesystem, not a toy.)

- Deal with arrays that can easily survive N disks going out simultaneously or at separate moments, where N is a whole number > 0 settable by the administrator
- Handle kernel freezes, power outages, and other abrupt shutdowns better
- Don't disable a whole disk if only parts of it have failed, e.g., if the sector errors are confined to less than 50% of access over the attempts of 20 dissimilar requests, then it continues just ignoring those sectors of that particular disk.
- Bad sectors:
  - A mechanism for saving which sectors are bad, someplace onto the disk.
  - If there is a generalized mechanism for marking degraded bad blocks that upper filesystem levels can recognize, use that. Program it if not.
  - Perhaps alternatively a mechanism for telling the upper layer that the size of the disk got smaller, even arranging for the upper layer to move out stuff from the areas being eliminated. This would help with a degraded blocks as well.
  - Failing the above ideas, keeping a small (admin settable) amount of space aside for bad blocks (distributed evenly across disk?), and using them (nearby if possible) instead of the bad blocks when it does happen. Of course, this is inefficient. Furthermore, the kernel ought to log every time the RAID array starts each bad sector and what is being done about it with a ``crit" level warning, just to get the administrator to realize that his disk has a piece of dust burrowing into it (or a head with platter sickness).
- Software-switchable disks:

  ``disable this disk''

  would block until kernel has completed making sure there is no data on the disk being shut down that is needed (e.g., to complete an XOR/ECC/other error

TOC

correction), then release the disk from use (so it could be removed, etc.);

``enable this disk''

would `mkraid` a new disk if appropriate and then start using it for ECC/whatever operations, enlarging the RAID5 array as it goes;

``resize array''

would respecify the total number of disks and the number of redundant disks, and the result would often be to resize the size of the array; where no data loss would result, doing this as needed would be nice, but I have a hard time figuring out how it would do that; in any case, a mode where it would block (for possibly hours (kernel ought to log something every ten seconds if so)) would be necessary;

``enable this disk while saving data''

which would save the data on a disk as-is and move it to the RAID5 system as needed, so that a horrific save and restore would not have to happen every time someone brings up a RAID5 system (instead, it may be simpler to only save one partition instead of two, it might fit onto the first as a gzip'd file even); finally,

``re-enable disk''

would be an operator's hint to the OS to try out a previously failed disk (it would simply call disable then enable, I suppose).

Other ideas off the net:

- finalrd analog to initrd, to simplify root raid.
- a read-only raid mode, to simplify the above
- Mark the RAID set as clean whenever there are no "half writes" done. -- That is, whenever there are no write transactions that were committed on one disk but still unfinished on another disk. Add a "write inactivity" timeout (to avoid frequent seeks to the RAID superblock when the RAID set is relatively busy).

Sound Blaster AWE 32/64 HOWTO

by Marcus Brinkmann < Marcus.Brinkmann@ruhr-uni-bochum.de>

v1.2, 11 January 1998

This document describes how to install and configure a Sound Blaster 32 (SB AWE 32, SB AWE 64) card from Creative Labs in a Linux System using the AWE Sound Driver Extension written by Takashi Iwai. It also covers some special tools and players for the SB AWE series. Reference system is a Debian GNU/Linux System, but every other Linux Distribution should also work.

1. Introduction

- 1.1 Acknowledgments
- 1.2 Revision History
- 1.3 New versions of this document
- 1.4 Feedback
- 1.5 Distribution Policy

2. Before you start

- 2.1 Introduction
- 2.2 Some general notes about the SB AWE cards
- 2.3 Some general notes about the Plug and Play cards
- 2.4 Some general notes about loadable kernel modules
- 2.5 Some general notes about the kernel sound driver

3. How to install SB AWE sound support

- 3.1 Things you will need
- 3.2 Getting started

- 3.3 Compiling the kernel
- 3.4 Reboot

4. Testing the Sound Driver

- 4.1 `/proc/devices, /dev/sndstat`
- 4.2 Output - The Raw Audio Device
- 4.3 Output - The OPL-2/OPL-3 Synthesis
- 4.4 Output - The Wave Table Synthesis
- 4.5 Mixing
- 4.6 Input - Sampling with the Raw Audio Device
- 4.7 The MIDI Port

5. AWE Driver Software

- 5.1 `sfxload`
- 5.2 `drvmidi`

6. Appendix

- 6.1 Additional Information
- 6.2 Sources
- 6.3 Sample `isapnp.conf`

---

1. Introduction

This is the Sound Blaster AWE HOWTO. It gives you detailed information about getting the most out of your Sound Blaster 32 or better, including Wave Table synthesis. This document covers all SB cards up from the SB 32.

The Sound Blaster series is well-known in the DOS and Windows community, and a lot of Linux users want to use this sound card under Linux as well. Unfortunately, Creative Labs only provides a sound driver for Windows and DOS, so it is not trivial to install and use a SB card under Linux. This document tries to describe how one can use the features of the SB AWE series under a Linux environment.

1.1 Acknowledgments

This documents contains information I got from the AWE Driver FAQ and the ISA PnP FAQ. See section Sources for author and location of this documents. Thanks to the authors, the SB AWE support is possible.

A lot of essential work was done by Hannu Savolainen, who developed the sound driver that comes with the Linux kernel. Thank you!

I want to thank Nicola Bernardelli for testing the AWE64 stuff. Without him, many errors would have remained undetected.

Thanks to the SGML Tools package, this HOWTO is available in several formats, all generated from a common source file.

1.2 Revision History

Version 1.0

first version

Version 1.1

corrected spelling (thanks Curt!), added version requirement for `isapnp`, now available via sunsite and lots of mirrors

Version 1.2

French version now available!, minor corrections, lots of spelling fixed (thanks to ispell)

TOC

1.3 New versions of this document

The latest version can be found on my Home Page, namely at Sound Blaster AWE HOWTO.

New versions of this document will be uploaded to various anonymous ftp sites that archive such information including ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/mini.

Hypertext versions of this and other Linux HOWTOs are available on many World-Wide-Web sites, including http://sunsite.unc.edu/LDP/. Most Linux CD-ROM distributions include the HOWTOs, often under the `/usr/doc` directory.

Thanks to Arnaud Launay zoro@lsol.tm.fr, a French version of this document is now available! The latest version can be found via http or anonymous ftp.

If you make a translation of this document into another language, let me know and I'll include a reference to it here.

1.4 Feedback

I rely on you, the reader, to make this HOWTO useful. If you have any suggestions, corrections, or comments, please send them to me, Marcus.Brinkmann@ruhr-uni-bochum.de, and I will try to incorporate them in the next revision.

i would especially welcome information about the startup procedure (boot scripts etc.) of other famous Linux distributions, e. g. Red Hat or SuSE Linux.

If you publish this document on a CD-ROM or in hardcopy form, a complimentary copy would be appreciated. Mail me for my postal address. Also consider making a donation to the Linux Documentation Project to help support free documentation for Linux. Contact the Linux HOWTO coordinator, Tim Bynum linux-howto@sunsite.unc.edu, for more information.

1.5 Distribution Policy

Copyright 1997 Marcus Brinkmann.

This HOWTO is free documentation; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but **without any warranty**; without even the implied warranty of **merchantability** or **fitness for a particular purpose**. See the GNU General Public License for more details.

You can obtain a copy of the GNU General Public License by writing to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

---

2. Before you start

2.1 Introduction

This document tries to help you install and use a Sound Blaster AWE 32 or Sound Blaster AWE 64 from Creative Labs in your Linux system. The reference system is a Debian GNU/Linux system on an Intel i586 platform, but it should work with any other Linux distribution as well as on every platform that is supported by the Linux sound driver (differences are mentioned where they appear, if any).

Be sure to read the Linux Sound HOWTO (see section Additional Information) carefully. I consider my document a supplement to the Sound HOWTO, and often you can find more information about things I have left out there.

2.2 Some general notes about the SB AWE cards

The SB AWE 32 sound card provides a raw audio device, standard OPL-2/OPL-3 synthesis, a MPU-401 MIDI port and 32 voices EMU 8000 Wave Table synthesis (for an explanation on these and other terms see the Linux Sound HOWTO). One goal of this document is to help you get all these features to work properly.

The SB AWE 64 has the capabilities of the SB AWE 32 and an additional Wave Guide synthesis Creative Labs is especially proud of. The problem for Linux users is, that the additional 32 voices are software generated and output via the raw wave devices. Because Creative Labs sees no market in Linux drivers, a Wave Guide synthesis sound driver is only available for Windows 3.1 and Windows 95.

This means that, from a Linux user's point of view, the SB AWE 32 and SB AWE 64 are almost identical. From now on I will only refer to the SB AWE in general and will only mention differences where they appear (if any).

2.3 Some general notes about the Plug and Play cards

Most modern cards for the Intel platform are ISA PnP cards, which is an abbreviation for ``Plug and Play''. This means, that the card has to be configured by the operation system, and this has to be done through an initialization routine at boot time. In general, there are at least three possible ways to do this:

1. You have a PnP Bios, which means that your Bios is aware of PnP cards and can configure them. If you think you can use all the features of your SB AWE PnP just because you have a PnP Bios, you are out of luck. Even if the Bios claims to support PnP cards it only initializes a subset of the ports and addresses used by your sound card. You will probably be able to play raw wave data, but you won't be able to play MIDI music, for example. For this reason, a PnP Bios is not an option.

2. You have an operating system that supports PnP cards. The current stable Linux kernel (2.0.x) is not such an operating system, so we have to wait for future kernel releases, that will support PnP devices.
3. You have a special program, started at boot time, that initializes your PnP cards. This is the way we do it.

The most commonly used software to initialize PnP cards under Linux are the ``isapnptools'' (see section Sources and section Getting Started). They provide a predictable way to configure all ISA PnP cards in your machine, not only your sound card.

2.4 Some general notes about loadable kernel modules

Some device drivers can be built as modules instead of compiling them into the kernel. You can find more information about modules in the Kernel HOWTO and the Module HOWTO (see section Additional Information).

If you have a PnP card, you *must* install sound support as a loadable kernel module. This means, that you can't build the sound driver into the kernel, but you will have to build it as a module that can be loaded into the kernel at runtime. This is because the kernel will be installed before your ISA PnP card can be configured, and your sound driver has to be loaded after your ISA PnP card is configured.

The sound kernel module can be loaded manually via `insmod sound` or `modprobe -a sound` or in the appropriate boot script of your Linux system (in Debian, it is sufficient to append a single line containing `sound` to `/etc/modules`). Another approach is to launch `kerneld`, a daemon that installs and removes kernel modules as needed.

Note that `kerneld` may not be the best solution for the AWE sound driver module, because it takes time to load the module in the kernel, especially if you want to use Wave Table synthesis and load big Sound Font banks, which you have to do each time after inserting the module. Because `kerneld` removes unused modules after one minute by default, it is perhaps better to insert the sound module manually or at boot time. Note that inserting the sound module manually or at boot time does prevent `kerneld` from removing it when it is idle. By the way: You can manually insert the sound module and use `kerneld` at the same time. The two methods don't conflict, but `kerneld` does not care about the sound module anymore.

This is especially useful if your mixer settings get disturbed after removing and reinstalling the module. A solution for this problem will be given in section Mixing (there is described how you can start your mixer automatically when the sound module gets inserted). However, it takes time for `kerneld` to load the module, to load the sound font bank and to start your mixer, and for this and other reasons it is better to install the sound module at boot time and not let `kerneld` remove it.

2.5 Some general notes about the kernel sound driver

TOC

You can install sound support in the kernel as a built-in or as a loadable module. If you have a PnP card, you have to install sound support as a module, because the PnP card needs to be initialized via the isapnptools before the module gets installed.

As you can imagine, you will have to recompile the kernel. I will give you a few hints about it below. For now, let's talk about the sound support in the kernel source. The kernel ships with the Free (Lite) Version of the OSS (USS) sound driver. The current version of this driver (3.5.4) does not support the SB AWE in full, but the SB 16 part of it. So you can have a raw audio device and OPL-2/OPL-3 synthesis if you use it, but you will not be able to play midi music with Wave Table synthesis.

If you want to use the Wave Table device, you can either buy the commercial sound driver from 4Front Technologies (someone please confirm me that it can do it, please), or patch your kernel with the AWE 32 Sound Driver Extension by Takashi Iwai. The former is beyond the scope of this document, I assume you want to use the latter.

The AWE32 Sound Driver Extension (see section Sources) is published under the GNU copyright license and ships with a number of tools to make use of the EMU 8000 wave synthesis of the SB AWE cards.

The `awedrv` software is already included in newer kernel source trees (starting from some 2.1.x kernels, but perhaps you want to upgrade the sources in your kernel tree, they may be old.

---

3. How to install SB AWE sound support

3.1 Things you will need

These are the requirements for SB AWE support under Linux:

- a functional Linux system (e.g. the Debian GNU/Linux distribution),
- a SB AWE 32 or compatible card (e.g. SB32, SB32 PnP, SB AWE64, ...),
- the sources of the Linux kernel, including the sound driver OSS/Free (normally included, check in `/usr/src/Linux/drivers/sound/Readme`),
- the AWE 32 sound driver extensions.

If you have a PnP card, you will also need:

- the isapnptools software package.

Look in section Sources for information where you can obtain these programs.

If you have a Debian GNU/Linux system, this means that you will need the packages `kernel-source-<version>`, `awe-drv` and perhaps the package `isapnptools`. You will want some of the other `awe-*` packages although they are not essential. I strongly recommend the `kernel-package` for easy kernel compiling and installing.

3.2 Getting started

Let's assume you have properly installed your card in a socket on your motherboard and perhaps already tested it under a DOS or Windows environment.

The next step is to initialize the card at boot time if and only if it is a PnP card. Follow the documentation in the source package to compile and install `isapnptools` (or just use the Debian binary package), and make a

```
# pnpdump > /etc/isapnp.conf
```

as root. This will generate a hopefully proper configuration file for your PnP cards, but with all devices commented out. Please edit this file carefully, and compare the values for DMA channels, IO bases and interrupts with the configuration of the cards under an DOS or Window environment if possible (with Win95, look at the resources used by the card under `resources` in the `device manager`). If you have problems, read the documentation that ships with the `isapnptools` carefully.

CAUTION: `isapnptools` often fails to detect all three I/O ports of the SB AWE Wave Table device. Please check carefully the `WaveTable` entry in your `isapnp.conf` against the example at the end of this HOWTO. Adjust the I/O base addresses if necessary.

CAUTION: According to the `isapnp-faq`, sometimes `isapnp` fails to program the number of the logical device. If you encounter error messages like the following:

```
Error occurred executing request 'LD 2' on or around line...
```

then try adding `(VERIFYLD N)` at the top of `isapnp.conf`. You have to use at least version 1.10 for this work. If you can't use version 1.10 or newer, you can also POKE the logical device numbers directly. Please r to the `isapnp-faq` for more information on this approach. If it doesn't work for you, please contact me (and the `isapnp` people will be interested, too, I would think).

CAUTION: Make sure that the last line is `(WAITFORKEY)`, that will sometimes be omitted by older versions of `pnpdump`.

A sample `isapnp.conf` for only one PnP card (the sound card) can be found at the end of the document (see section Sample <tt>isapnp.conf</tt>).

If you have a Debian system, no further adjustments are necessary. `isapnp` will be started at boot time in `/etc/init.d/boot` with this snippet of script code, that you may want to include in your boot scripts:

```
# Configure the isa plug and play boards before loading
# modules. Need to do this before loading modules to get
# a chance of configuring and starting PnP boards before
# the drivers mess all this up.
#
if [ -x /etc/init.d/isapnp ]
then
  /etc/init.d/isapnp start
fi
```

where `/etc/init.d/isapnp` is

```
#! /bin/sh
# /etc/init.d/isapnp: configure Plug and Play boards
test -x /sbin/isapnp || exit 0
/sbin/isapnp /etc/isapnp.conf
exit 0
```

If you have another Linux distribution, you are on your own. I don't know what to do (anybody willing to submit more specific information?). Just make sure that isapnp gets started *before* the modules will be loaded (see below).

3.3 Compiling the kernel

Before recompiling the kernel, you have to apply the AWE Driver Extension to the sound driver. Even if your kernel source tree already includes the `awedrv` extension (check `/drivers/sound/lowlevel/` for that), you may want to upgrade the `awedrv` software. Follow the installation instructions in the `awedrv` source directory. In brief, you have to run an installation script that applies the patches to the kernel sources.

Be careful if you have upgraded the kernel source tree after running the install script. The script just checks if a certain file exists - if it exists, it does not apply the necessary patch. You are well advised to remove the file `drivers/sound/lowlevel/awe_wav.c` before running the script after upgrading the kernel source.

Next you have to configure the kernel for sound support. I hope you know a bit about kernel compiling; see the Sound HOWTO and the Kernel HOWTO for details. Go in the source directory of your kernel sources (`/usr/src/linux` for example), and do

```
# make config
```

or `make menuconfig` or `make xconfig`. Then you have to configure your kernel in the standard way. Use this opportunity to make a small and powerful kernel, especially designed for your system. Be sure to answer `Enable loadable module support` with `Y`, if you want to install the sound driver as a loadable module (a *must* if you have a PnP card), but I'm sure you want to do it anyway.

At one point, you will be asked if you want sound card support. You are free to answer with `Y` or with `M` if you have *not* a PnP card. You *must* answer with `M`, for module, if you have a PnP card. You have to compile sound card support as a module if you have a PnP card because the PnP card has to be initialized before the module gets loaded.

The following questions you should answer with `Y`, all other with `N`:

```
Sound Blaster (SB, SBPro, SB16, clones) support (CONFIG_SB) [Y/n/?]
Generic OPL2/OPL3 FM synthesizer support (CONFIG_ADLIB) [Y/n/?]
/dev/dsp and /dev/audio support (CONFIG_AUDIO) [Y/n/?]
MIDI interface support (CONFIG_MIDI) [Y/n/?]
FM synthesizer (YM3812/OPL-3) support (CONFIG_YM3812) [Y/n/?]
lowlevel sound driver support [Y/n/?]
AWE32 support (CONFIG_AWE32_SYNTH) [Y/n/?]
```

Only the latter is actually for the Wave Table synthesis. The others are SB 16 options from the OSS/Free sound driver.

In addition, you have to configure the sound card I/O port. Look at the `isapnp.conf` file for hints, if you have one. For me, the following default values are sufficient. Note that the default value from the kernel configuration script may be wrong (especially the values `SBC_IRQ` and `SB_MPU_BASE` seem to be incorrect for most cards).

```
I/O base for SB Check from manual of the card (SBC_BASE) [220]
Sound Blaster IRQ Check from manual of the card (SBC_IRQ) [5]
Sound Blaster DMA 0, 1 or 3 (SBC_DMA) [1]
Sound Blaster 16 bit DMA (_REQUIRED_for SB16, Jazz16, SMW) 5, 6 or 7
(use 1 for 8 bit cards) (SB_DMA2) [5]
MPU401 I/O base of SB16, Jazz16 and ES1688 Check from manual of the card
(SB_MPU_BASE) [330]
SB MPU401 IRQ (Jazz16, SM Wave and ES1688) Use -1 with SB16 (SB_MPU_IRQ) [-1]
```

Now recompile the kernel. Debian users should use the `kernel-package`. This package makes the kernel compile as easy as installing a debian package. Look at the documentation in `/usr/doc/kernel-package/`. Here is a hint:

```
# make-kpkg clean
# make-kpkg -revision custom.1.0 kernel_image
```

and then `dpkg -i /usr/src/kernel-image-2.0.29_custom.1.0_i386.deb`.

If you have another Linux distribution, follow the standard way for compiling a new kernel. Don't forget `make modules` and `make modules_install`. Look at the Sound HOWTO and perhaps the Kernel HOWTO for more information.

TOC

3.4 Reboot

After installing the new kernel, you should do a reboot (be sure to have a functional boot disk at hand). Cross you fingers.

If you have a PnP card be sure to launch isapnp either in a boot script (as described above) or manually:

```
# /sbin/isapnp /etc/isapnp.conf
Board 1 has Identity 74 00 00 e3 10 48 00 8c 0e:  CTL0048 Serial No 58128
[checksum 74]
```

Now you can install the sound driver, if you have compiled it as a module:

```
# modprobe -a sound
AWE32 Sound Driver v0.3.3e (DRAM 2048k)
```

If you think the memory detection was not correct (I have a report of one who has a AWE64 with 4096k, and ``detected'' have been 28672k), try either to upgrade the `awedrv` software or to specify the amount of memory in the file `/usr/src/linux/drivers/sound/lowlevel/awe_config.h`, for example:

```
#define AWE_DEFAULT_MEM_SIZE  4096   /* kbytes */
```

Sorry, you have to recompile the kernel then (perhaps compiling the modules will be sufficient, but I don't know for sure).

If it works, you may want to have the sound module loaded automatically. You can use `kerneld` (why this is a bad idea is explained in section 1.4) or append a single line containing `sound` to your `/etc/modules` (in Debian) or add `/sbin/modprobe -a sound` to your start-up script.

---

4. Testing the Sound Driver

4.1 /proc/devices, /dev/sndstat

If you have built-in sound support, you will get some useful information at boot time. If you have sound support installed as a loadable module, you can get the same information (perhaps after removing the sound module with `modprobe -r sound` first) with

```
# modprobe -a sound trace_init=1
Sound initialization started

<Sound Blaster 16 (4.13)> at 0x220 irq 5 dma 1,5
<Sound Blaster 16> at 0x330 irq 5 dma 0
<Yamaha OPL3 FM> at 0x388
Sound initialization complete
AWE32 Sound Driver v0.3.3e (DRAM 2048k)
```

If you have a `/proc` virtual file system, you can look for the sound device with

```
# cat /proc/devices
Character devices:
[...]
14 sound
[...]
```

Next make sure you have the correct devices installed under `/dev/`. Please look at the Sound HOWTO for details. Then ask `/dev/sndstat` about the status of the sound module:

```
# cat /dev/sndstat
Sound Driver:3.5.4-960630 (Sat Oct 11 19:35:14 CEST 1997 root,
Linux flora 2.0.29 #1 Sat Oct 11 19:12:56 CEST 1997 i586 unknown)
Kernel: Linux flora 2.0.29 #1 Sat Oct 11 19:36:23 CEST 1997 i586
Config options: 0

Installed drivers:
Type 1: OPL-2/OPL-3 FM
Type 2: Sound Blaster
Type 7: SB MPU-401

Card config:
Sound Blaster at 0x220 irq 5 drq 1,5
```

```
SB MPU-401 at 0x330 irq 5 drq 0
OPL-2/OPL-3 FM at 0x388 drq 0

Audio devices:
0: Sound Blaster 16 (4.13)

Synth devices:
0: Yamaha OPL-3
1: AWE32 Driver v0.3.3e (DRAM 2048k)

Midi devices:
0: Sound Blaster 16

Timers:
0: System clock

Mixers:
0: Sound Blaster
1: AWE32 Equalizer
```

If you don't have an output like this, perhaps there is an error in your configuration. Go on and see what doesn't work, then go back to step Getting Started, checking everything.

4.2 Output - The Raw Audio Device

Try to get an `.au` file (Sun workstation) or a raw sample file, and do

```
# cat bell.au > /dev/audio
```

resp.

```
# cat sample > /dev/dsp
```

You should hear the content of the file via the `Audio Device 0: Sound Blaster 16 (4.3)`.

4.3 Output - The OPL-2/OPL-3 Synthesis

If you want to use the OPL-2/OPL-3 FM synthesis to play MIDI-Files with your sound card, try the program `playmidi` (see Appendix B). Start it with

```
# playmidi -f dance.mid
```

or

```
# playmidi -4 dance.mid
```

TOC

The former will give you OPL-2, the latter OPL-3 MIDI music. If you are embarrassed about the sou don't blame playmidi: It is the FM synthesis that sounds bad.

Imagine you would only have the OSS/Free Sound Driver: Than this is the quality of MIDI Music you can achieve (apart from software synthesis). Fortunately, you have a SB AWE, and you can use the Wave Table capabilities with the AWE Sound Driver Extension.

4.4 Output - The Wave Table Synthesis

The AWE Driver Extension comes with special tools (`awesfx`) to make use of the EMU 8000 Wave Table synthesis. First, you have to load a Sound Font Bank on your card - even if you want to use the ROM samples! You can get the files from your Windows installation - look for files ending in `*.sfb` or `*.sf2`.

The ROM samples can be loaded with `SYNTHGM.SBK`, real samples are in `SYNTHGS.SBK` and `SYNTHMT.SBK`, as well as in `SAMPLE.SBK`. You can get other Sound Font Banks via ftp or www, try the EMU Homepage, the web site from Creative Labs, and look out for the Chaos samples, they are really good (be sure to check out the AWE Driver Web Site).

Try to load the standard GM (ROM) set with

```
# sfxload -i synthgm.sbk
```

and then play a midi file that comes with your SB AWE:

```
# drvmidi dance.mid
```

4.5 Mixing

Get your favorite mixer program and start it. Start a raw audio sample and two MIDI files at the same time, and try out the mixer settings. Play a bit, it should be very easy. Here a list of the devices and how they are called:

Yamaha OPL-3

Synth or FM

AWE32 Driver

Synth or FM (does anyone know a mixer were the last two are separated from each other?)

Sound Blaster 16 (4.13)

PCM or DSP

PC Speaker (still alive)

Spkr

TOC

Other mixer settings refer to the CD ROM, possibly connected to the sound card, to Master Volume, Bass, Treble and Recording Level of the different input lines. You can specify which lines should be recorded.

4.6 Input - Sampling with the Raw Audio Device

You can record different sources: a connected CD ROM, a microphone connected to `mic`, and any that you can connect to `line in`. Set the mixer in the appropriate position. Play a sound and record into a file, reading from the Raw Audio Device, for example:

```
# cdplay
# dd bs=8k count=5 </dev/dsp >music.au
5+0 records in
5+0 records out
# cat music.au >/dev/dsp
```

records and plays five seconds of audio from the input device.

4.7 The MIDI Port

Sorry, no information about MIDI port yet!

---

5. AWE Driver Software

5.1 sfxload

You can load samples in your DRAM on the sound card with the `sfxload` tool. Note that you can only load one sample file per bank, with bank 0 as default. So, after `sfxload synthgs.sbk`, the only samples on your sound card are the GS samples. If you want to load additional Sound Font Banks, you have to use the `-b` option, for example:

```
# sfxload synthgs.sbk
# sfxload -b1 sample.sbk
# drvmidi sfx.mid
```

You can load a default Sound Font Bank automatically with installing the module. Just append a line like

```
post-install sound /usr/bin/sfxload synthgm.sbk
```

to your `/etc/conf.modules` file.

See the documentation for more details about `sfxload`.

5.2 drvmidi

With the `drvmidi` program, you can use your AWE Driver to play MIDI files. Just specify the name of your midi file after the command:

```
# drvmidi waltz.mid
```

See the documentation for more details about `drvmidi`

6. Appendix

6.1 Additional Information

The Linux Sound HOWTO

Author: Jeff Tranter, < jeff_tranter@pobox.com>

Last Version: v1.17, 4 August 1997

Please take a good look at the Sound HOWTO (available from sunsite.unc.edu and other sites). It contains a bulk of data about compiling the kernel with sound support, and explains a lot about the sound devices, applications etc.

I understand this HOWTO as an addition to the Sound HOWTO.

The Linux Sound Playing HOWTO

Author: Yoo C. Chung, < wacko@laplace.snu.ac.kr>

Last Version: v1.4, 13 May 1997

You should also look at the Sound Playing HOWTO. It tells you about the various sound formats and applications to play them.

The AWE Driver FAQ

Author: Takashi Iwai < iwai@dragon.mm.t.u-tokyo.ac.jp>

Source: http://bahamut.mm.t.u-tokyo.ac.jp/~iwai/awedrv/awedrv-faq.html

If you have problems installing the AWE Driver or using the Wave Table synthesis of your Sound Blaster card, then look here.

The ISA PnP FAQ

Author: Peter Fox < fox@roestock.demon.co.uk>

Source: http://www.roestock.demon.co.uk/isapnptools/isapnpfaq.html

If you have problems configuring your ISA PnP card, then this is your book of wisdom.

6.2 Sources

`isapnptools`

Author: Peter Fox < fox@roestock.demon.co.uk>

Last Version: 1.11

Source: http://www.roestock.demon.co.uk/isapnptools/index.html

If mailing patches, bug reports or comments, please put 'isapnp' somewhere in the subject line, and mail to isapnp@roestock.demon.co.uk.

`awedrv`

Author: Takashi Iwai < `iwai@dragon.mm.t.u-tokyo.ac.jp`>

Last Version: 0.4.2c

Source: http://bahamut.mm.t.u-tokyo.ac.jp/~iwai/awedrv/

Linux kernel

Author: Linus Torvald and many other

Last Version: At the moment, you should use 2.0.29

Source: everywhere, where you can obtain Linux `:)`

OSS/Free

Author: Hannu Savolainen (Please check http://www.4front-tech.com/usslite before mailing me).

Last Version: I have 3.5.5-beta1

Source: with the Linux kernel source package

Information: http://www.4front-tech.com/usslite or European mirror.

6.3 Sample `isapnp.conf`

In the ISA PnP configuration file created by pnpdump, the single devices of your PnP cards appear as sections. In the output below, one card was detected (the sound card), with four logical devices:

- `LD 0`: Audio Device
- `LD 1`: IDE Device
- `LD 2`: WaveTable Device
- `LD 3`: Gameport Device

I left `LD 1` unconfigured, because I have no CD ROM attached to my sound card IDE port. If you have no IDE port on your SB, then `LD 1` will be the Gameport Device and `LD 3` will not appear.

Please refer to section Getting started for further (important!) information about this file.

```
# $Id: pnpdump.c,v 1.9 1997/06/10 21:37:32 fox Exp $
# This is free software, see the sources for details.
```

TOC

```
# This software has NO WARRANTY, use at your OWN RISK
#
# For details of this file format, see isapnp.conf(5)
#
# For latest information on isapnp and pnpdump see:
# http://www.roestock.demon.co.uk/isapnptools/
#
# Compiler flags: -DREALTIME -DNEEDSETSCHEDULER

(READPORT 0x0203)
(ISOLATE)
(IDENTIFY *)

# Try the following if you get error messages like
# Error occurred executing request 'LD 2' on or around line...

#(VERIFYLD N)

# ANSI string -->Creative SB32 PnP<--

(CONFIGURE CTL0048/58128 (LD 0
#     ANSI string -->Audio<--

  (INT 0 (IRQ 5 (MODE +E)))
  (DMA 0 (CHANNEL 1))
  (DMA 1 (CHANNEL 5))
  (IO 0 (BASE 0x0220))
  (IO 1 (BASE 0x0330))
  (IO 2 (BASE 0x0388))

  (ACT Y)
))

(CONFIGURE CTL0048/58128 (LD 1
#     ANSI string -->IDE<--

# (INT 0 (IRQ 10 (MODE +E)))
# (IO 0 (BASE 0x0168))
# (IO 1 (BASE 0x036e))

# (ACT Y)
))

(CONFIGURE CTL0048/58128 (LD 2
#     ANSI string -->WaveTable<--
```

```
  (IO 0 (BASE 0x0620))
  (IO 1 (BASE 0x0A20))
  (IO 2 (BASE 0x0E20))

  (ACT Y)
))

(CONFIGURE CTL0048/58128 (LD 3
#     ANSI string -->Game<--

  (IO 0 (BASE 0x0200))
  (ACT Y)
))

# Returns all cards to the "Wait for Key" state
(WAITFORKEY)
```

StarOffice 3.1 Mini-HOWTO

Matthew Borowski (mkb@poboxes.com) (http://mkb.home.ml.org/)

v1.0, 2 June 1998

Information on installing the `StarOffice 3.1` Office Suite by StarDivision.

1. Introduction

- 1.1 Motif info
- 1.2 History of this document

2. Obtaining the StarOffice Office Suite

3. Installation of `StarOffice`

- 3.1 Libc Issues and Fixes
- 3.2 Installing the tar Files
- 3.3 Setup und Configuration
- 3.4 COL (Caldera OpenLinux) setup bug
- 3.5 Fixing the LANG variable problem

4. Running StarOffice

5. StarOffice Tips

TOC

6. More Information

7. Contributions

8. Disclaimer and Copyright

---

1. Introduction

This document deals with installing StarOffice 3.1. The current version of StarOffice is 4.0. Version 4.0 is considerably easier to install and less buggy than version 3.1. In fact, in my opinion there is no need for a HOWTO for version 4.0, since StarDivision has written a fairly complete installation guide. The StarOffice mini-HOWTO was useful while it lasted, but it is now being archived only for the few who still wish to install StarOffice 3.1. For information about StarOffice 4.0, check out my website at http://mkb.home.ml.org/linux/sohowto.html.

The StarOffice Office Suite is a collection of office tools for Linux, written in C using the Motif toolkit.

StarOffice includes:

- StarWriter, a word processor
- StarCalc, a spreadsheet
- StarImage, an image editor
- StarDraw, a draw program similar to Corel DRAW
- StarChart, a chart-making program
- StarMath, a math editor

The StarOffice Office Suite for Linux is free for non-commercial use. Commercial use requires a license. The cost for a commercial license seperate from Caldera's OpenLinux system is currently undecided.

1.1 Motif info

StarOffice is written using the commercial Motif 2.0 GUI toolkit. The dynamically linked version will *not* work with Motif 1.2/LessTif.

The static version (with Motif 2.0 linked in) will work regardless of whether or not you have Motif. However, performance drops.

I run StarOffice on a Pentium 100 with 48 megs RAM, running Redhat 4.1 and Motif 2.0. Dynamically linked StarOffice takes about a minute to load. However, it is noticable that, once it loads up, StarOffice dynamically linked takes up much less resources than the static version.

1.2 History of this document

- Version 1.0 980602 - added information on StarOffice 4.0 and about how this document is now outdated
- Version 0.8 971109 - added tips, installation w/o libc upgrade, etc.
- Version 0.7 970707 - changed section 3.1.1, changed license

TOC

- Version 0.6 970609 - added fix for the LANG variable problem
- Version 0.5 970608 - updated info to StarOffice 3.1 final
- Version 0.1 970505 - the initial release.

---

2. Obtaining the StarOffice Office Suite

The current release of StarOffice for Linux is 3.1 final. This release is available for FTP at

`ftp://ftp.gwdg.de/pub/linux/staroffice`

The file `README.StarOffice` contains information on downloading StarOffice 3.1 final. You will need to get three files:

- `StarOffice31-english.tar.gz` (english) templates/demo docs or
- `StarOffice31-german.tar.gz` (german) templates/demo docs
- `StarOffice31-dynbin.tar.tgz` (dynamically linked) or
- `StarOffice31-statbin.tar.gz` (statically linked) binaries
- `StarOffice31-common.tar.gz`

There is also version of StarOffice, available from Caldera. Check their website at

`http://www.caldera.com`

for details on this. Caldera also makes a StarOffice CD-ROM available for $ 7.99 US.

3. Installation of `StarOffice`

Installation of StarOffice consists of:

- Untarring the distribution files as root in `/usr/local`
- running the `setup` program as a user
- sourcing the `.sd.sh` or `.sd.csh`
- reading section 3.5!!! (do this)

3.1 Libc Issues and Fixes

StarOffice is linked with `libc 5.4.4`.

StarOffice 3.1 *will* work on `Libc 5.3.x`. Unfortunately, the setup program requires 5.4.4 or higher. If you have `libc 5.3.x`, you might be able to get around this by acquiring a copy of `libc 5.4.4+` and adding it to your LD_LIBRARY_PATH environment variable before executing the `setup` script. I haven't tried this, however, so you're on your own.

If you attempt to run the `setup` script on a libc older than 5.4.4, you will get the following error message:

```
line 1: Syntax error at token 'I' expected declarator; i.e.
File ...
```

To upgrade your libc, FTP to `sunsite.unc.edu` and look in the directory `/pub/Linux/GCC` for the file `libc-5.4.33.bin.tar.gz` (or whatever the latest libc is). Extract the contents of this file in a temporary directory. A new lib/ directory will be created. Su to root and copy the file `libc.so.5.4.33` from this directory to your /lib directory. Now, make the symlink from `libc.so.5` to `libc.5.4.33` with the command:

```
ln -sf /lib/libc.so.5.4.33 /lib/libc.so.5
```

then run the `ldconfig` command.

Dr. Romano Giannetti (`romano@iet.unipi.it`) says:

> ...I want only to add that I could install (like you suggested) StarWriter in a redhat 4.2 system which has a libc5.3, without doing the upgrade.
>
> The exact steps are:
>
> 1. get a libc.so.5.4.x. If you have a redhat rpm package (as the one you find in the contrib directory), you can extract the library by going in a scratch directory and doing: `rpm2cpio libc.so.5.4.x-y.rpm | cpio --extract --make-directories` The library will appear in ./lib subdirectory

TOC

2. move libc.so.5.4.x in your home directory. Then (assuming a sh-like shell): `ln -s libc.so.5.4.x libc.so.5` `export LD_LIBRARY_PATH=$HOME:/lib:/usr/lib`

3. Now you can run setup.

3.2 Installing the tar Files

After downloading StarOffice, su or login as root and place the archives in `/usr/local/`. Change directory to `/usr/local/` and extract the files. An example command to decompress a gzipped tar file would be:

```
tar -xzvf filename.tar
```

Older systems may require you first use the `gzip -d` command to unzip the file, then use the `tar -xvf` command to untar it.

The files will extract to their locations within the newly created `usr/local/StarOffice-3.1` tree.

3.3 Setup und Configuration

After you have extracted the StarOffice files as root, you will need to login with your userid. Change directory to `/usr/local/StarOffice-3.1` and execute the `setup` program. This program will install non-shared files and symlinks needed for each individual user. The standard installation is recommended. There *could* be problems if you do not accept the default installation path.

StarOffice makes use of environment variables. The files `.sd.sh` (formatted for the Bourne Shell) and `.sd.csh` (formatted for the C Shell) provide the environment variable settings for StarOffice. These files are located in your home directory.

If you use `bash`, edit your `.bashrc` and add the line:

```
source  /.sd.sh
```

After doing this, restart `bash` to bring the environment variables into effect.

If you use a different shell, consult that shell's man page for information on sourcing a file.

3.4 COL (Caldera OpenLinux) setup bug

Phil Reardon (pcr@busprod.com) says:

> " I found a bug in the setup script for StarOffice that came with my Caldera COL standard release. It produces // in a path where there should only be one /. To fix it, remove the first slash from this line:
>
> ```
> exec ${pfad:='.'}/linux-x86/bin/$name;;
> ```
>
> There should be no / before linux-x86."

3.5 Fixing the LANG variable problem

TOC

The `.sd.sh` and `.sd.csh` files set the LANG variable. This causes problems with perl and man. Man will give the error message

> " Failed to open the message catalog man on the path NLSPATH="

Perl will give the error message

> " warning: setlocale(LC_CTYPE, "")..."

The `.sd.sh` file contains a line that sets LANG=us and another that exports a bunch of variables, including LANG. Remove the LANG=us line and remove LANG from the list of variables, and this will be fixed.

In the `.sd.csh` file (which is formatted for the c shell), you need to remove the line that says "setenv LANG us".

Thanks to Adam L. Klein (`alklein@adelphia.net`) for informing me of this fix.

---

4. Running StarOffice

The StarOffice binaries are located in the directory:
`/usr/local/StarOffice-3.1/linux-x86/bin/`

The applications are: `sdraw3`, `swriter3`, `scalc3`, `smath3`, `schart3`, `simage3`

The program `svdaemon` must be executed in order to use online help. The program `svportmap` must be executed to enable application communication features between StarWriter, StarCalc and StarDraw.

5. StarOffice Tips

Contributed by Patrick D'Cruze (pdcruze@netpal.com.au)

1. Instead of sourcing the .sd.sh (or .sd.csh) file in my .bash_profile script, I instead copied the .sd.sh script, renamed it to swriter and copied it to /usr/local/bin. I then modified it by adding a:

```
#! /bin/bash
```

at the start of the script and a:

```
exec swriter3 $*
```

at the end of the script. Now all I have to do is run swriter and it automatically sets up the appropriate environment variables and then runs StarWriter. Saves having those environment variables pollute other applications/environments.

2. I noticed that for a lot of people, StarOffice takes quite a while to load (upwards of 60 seconds). The consensus on usenet was that a large chunk of this time was due to the symbol relocations that the dynamic linker has to do. ie, for each new symbol the dynamic linker has to locate the appropriate library. StarOffice dynamically links against quite a few libraries so the dynamic linker spends quite a bit of time searching through lots of libraries.

There is a solution to this. I run StarOffice in a chroot'd jail. In the jail, I just put the binaries and libraries that StarOffice uses (all the libs out of /usr/X11R6/lib, libc/libm and libg++/libstdc++). StarWriter takes approximately 15 seconds to come up on my P133/32MB. This is due to the fact that the only libraries present are the ones needed by StarOffice and hence the dynamic linker spends proportionately less time searching through all the libraries on the system (ie, it doesn't search through all the useless libs in /usr/lib etc looking to resolve symbols).

6. More Information

More information on StarOffice for Linux is available directly from StarDivision. StarDivision maintains a website at http://www.stardivision.com

StarDivision runs a news server with support and user groups related to StarOffice for linux. These newsgroups are the best place to find up-to-date information on StarOffice and to ask for help. Using a newsreader, connect to news://starnews.stardivision.com or follow the support link on the StarDivision homepage.

7. Contributions

Thank you to Werner Klaus (wklaus@metronet.de) for translating this document to sgml.

If you can contribute any additional information for this mini-HOWTO, please contact me via internet e-mail. My address is located at the top of this document. You may also wish to take a look at my web page, located at http://mkb.home.ml.org/linux/ or http://www.gslink.com/~mkb/linux/

8. Disclaimer and Copyright

This document is (C) 1997 by Matthew Borowski.

Redistribution of this document in electronic form is permitted under the condition that the document remains unchanged. If you wish to include this document in a CD-ROM or book, please obtain permission from me beforehand (I prefer requests to be sent via e-mail).

The author claims no responsibility in anything that may occur directly or indirectly as a result of using this document. The author is not an employee of StarDivision, GmbH or Caldera, Inc.

Comments and questions may be directed to the author at mkb@poboxes.com.

Using Term to Pierce an Internet Firewall

Barak Pearlmutter, bap@cs.unm.edu

v, 15 July 1996

Directions for using ``term'' to do network stuff through a TCP firewall that you're not supposed to be able to.

1. Disclaimer
2. Copyright
3. Introduction
4. The basic procedure
5. Detailed directions
6. Multiple term sockets
7. The `~/.term/termrc.telnet` init file
8. Direction
9. Security
10. Telnet mode

11. Bugs and term wish list

12. Tricks that don't seem to work

13. Related resources

14. Acknowledgments

1. Disclaimer

!!! READ THIS IMPORTANT SECTION !!!

I hereby disclaim all responsibility for this hack. If it backfires on you in any way whatsoever, that's the breaks. Not my fault. If you don't understand the risks inherent in doing this, don't do it. If you use this hack and it allows vicious hackers to break into your company's computers and costs you your job and your company millions of dollars, well that's just tough nuggies. Don't come crying to me.

2. Copyright

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu via email.

3. Introduction

The "term" program is normally used over a modem or serial line, to allow various host-to-host services to flow along this simple serial connection. However, sometimes it is useful to establish a term connection between two machines that communicate via telnet. The most interesting instance of this is for connecting two hosts which are separated by ethernet firewalls or SOCKS servers. Such firewalls provides facilities for establishing a telnet connection through the firewall, typically by using the SOCKS protocol to allow inside machines to get connections out, and requiring outside users to telnet first to a gateway machine which requires a one-time password. These firewalls make it impossible to, for instance, have X clients on an inside machine communicate with an X server on an outside machine. But, by setting up a term connection, these restrictions can all be bypassed quite conveniently, at the user level.

4. The basic procedure

Setting up a term connection over a telnet substrate is a two-phase process. First your usual telnet client is used to set up a telnet connection and log in. Next, the telnet client is paused and control of the established telnet connection is given to term.

5. Detailed directions

In detail, the process goes like this.

First, from a machine inside the firewall, telnet to a target machine outside the firewall and log in.

Unless you are under linux and will be using the proc filesystem (see below) make sure your shell is an sh style shell. Ie if your default shell is a csh variant, invoke telnet by

```
(setenv SHELL /bin/sh; telnet machine.outside)
```

After logging in, on the remote (outside) machine invoke the command

```
term -r -n off telnet
```

Now break back to the telnet prompt on the local (inside) machine, using `^]` or whatever, and use the telnet shell escape command `!` to invoke term,

```
telnet> ! term -n on telnet >&3 <&3
```

Et voila!!!

(If you have a variant telnet, you might have to use some other file descriptor than 3; easy to check using strace. But three seems to work on all bsd descendent telnet clients I've tried, under both SunOS 4.x and the usual linux distributions.)

Some telnet clients do not have the ! shell escape command. Eg the telnet client distributed with Slackware 3.0 is one such client. The sources that the Slackware telnet client is supposedly built from,

```
ftp://ftp.cdrom.com:/pub/linux/slackware-3.0/source/n/tcpip/NetKit-B-0.05.tar.gz
```

have the shell escape command. A simple solution is therefore to obtain these sources and recompile them. This unfortunately is a task I have had no luck with. Plus, if you are running from inside a SOCKS firewall, you will need a SOCKSified telnet client anyway. To that end, I was able to compile a SOCKSified telnet client from

```
ftp://ftp.nec.com/pub/security/socks.cstc/socks.cstc.4.2.tar.gz
```

or if you're outside the USA,

```
ftp://ftp.nec.com/pub/security/socks.cstc/export.socks.cstc.4.2.tar.gz
```

Alternatively, under linux kernels up to 1.2.13, you can pause the telnet with `^]^z`, figure out its pid, and invoke

```
term -n on -v /proc/<telnetpid>/fd/3 telnet
```

This doesn't work with newer 1.3.x kernels, which closed some mysterious security hole by preventing access to these fd's by processes other than the owner process and its children.

6. Multiple term sockets

It is a good idea to give the term socket an explicit name. This is the "`telnet`" argument in the invocations of term above. Unless you have the TERMSERVER environment variable set to telnet as appropriate, you invoke term clients with the -t switch, e.g. "`trsh -t telnet`".

7. The `~/.term/termrc.telnet` init file

I have checked line clarity using linecheck over this medium. I expected it to be completely transparent, but it is not. However, the only bad character seems to be 255. The `~/.term/termrc.telnet` I use (the `.telnet` is the name of the term connection, see above) contains:

```
baudrate off
escape 255
ignore 255
timeout 600
```

Perhaps it could be improved by diddling, I am getting a throughput of only about 30k cps over a long-haul connection through a slow firewall. Ftp can move about 100k cps over the same route. A realistic baudrate might avoid some timeouts.

8. Direction

Obviously, if you are starting from outside the firewall and zitching in using a SecureID card or something, you will want to reverse the roles of the remote vs local servers given above. (If you don't understand what this means, perhaps you are not familiar enough with term to use the trick described in this file responsibly.)

9. Security

This is not much more of a vulnerability than the current possibility of having a telnet connection hijacked on an unsecured outside machine. The primary additional risk comes from people being able to use the term socket you set up without you even being aware of it. So be careful out there. (Personally, I do this with an outside machine I know to be pretty secure, namely a linux laptop I maintain myself that does not accept any incoming connections.)

Another possibility is to add "socket off" to the remote `~/.term/termrc.telnet`, or `add "-u off"` to invocation of term. This prevents the socket from being hijacked from the remote end, with only a minor loss of functionality.

10. Telnet mode

Be sure the remote telnetd is not in some nasty seven-bit mode. Or if it is, you have to tell term about it when you invoke term, by adding the `-a` switch at both ends. (I sometimes use "`^] telnet> set outbin`" or "`set bin`" or invoke telnet with a `-8` switch to put the connection into eight-bit mode.)

11. Bugs and term wish list

The linecheck program has some problems checking telnet connections sometimes. This is sometimes because it doesn't check the return code of the `read()` call it makes. For network connections, this call to `read()` can return `-1` with an `EINTR` (interrupted) or `EAGAIN` (try again) error code. Obviously this should be checked for.

There are a number of features that could ease the use of term over telnet. These primarily relate to an assumption that influenced the design of term, namely that the connection is low bandwidth, low latency, and somewhat noisy.

A telnet connection is in general high bandwidth, high latency, and error free. This means that the connection could be better utilized if (a) the maximum window size was raised, well above the limit imposed by term's `N_PACKETS/2=16`, (b) there was an option to turn off sending and checking packet checksums, and (c) larger packets were permitted when appropriate.

Also, to enhance security, it would be nice to have a term option to log all connections through the socket it monitors to a log file, or to stderr, or both. This would allow one to see if one's term connection is being subverted by nasty hackers on the outside insecure machine.

12. Tricks that don't seem to work

Some telnet clients and servers agree to encrypt their communications, to prevent evesdropping on the connection. Unfortunately, the hack used above (using the network connection that the telnet client has set up while the telnet client is idle) won't work in that case. Instead, one really must go through the telnet client itself, so it can do its encryption. It seems like that requires a simple hack to the telnet client itself, to add a command that runs a process with its stdin and stdout are connected to the live telnet connection. This would also be useful for various 'bots, so perhaps someone has already hacked it up.

13. Related resources

A vaguely related trick is to SOCKSify one's Term library. Details, including patches to SOCKS, are available from Steven Danz <danz@wv.mentorg.com>.

14. Acknowledgments

Thanks for valuable suggestions from:

- Gary Flake <flake@scr.siemens.com>
- Bill Riemers <bcr@physics.purdue.edu>
- Greg Louis <glouis@dynamicro.on.ca>

Extra copy of IMPORTANT DISCLAIMER --- BELIEVE IT!!!

I hereby disclaim all responsibility for this hack. If it backfires on you in any way whatsoever, that's the breaks. Not my fault. If you don't understand the risks inherent in doing this, don't do it. If you use this hack and it allows vicious hackers to break into your company's computers and costs you your job and your company millions of dollars, well that's just tough nuggies. Don't come crying to me.

TkRat mini-HOWTO

Dave Whitinger, dave@whitinger.net

v1.1, 2 February 1998

This is the TkRat mini-HOWTO. This document is geared toward anyone interested in using their Linux computer to send and receive Internet E-Mail.

1. Introduction

2. Installing TkRat

3. Downloading your E-Mail

4. Configuring TkRat

5. Using TkRat

6. Automatically sorting mail into folders

7. Copyright

8. Acknowledgements

1. Introduction

This document is maintained by Dave Whitinger (dave@whitinger.net). Please mail me with any questions or ideas you have regarding this HOWTO.

This document will attempt to describe the following four things:

1. Installing the TkRat E-Mail program.
2. Downloading mail from an Internet Service Provider.
3. Reading, storing, and sending mail
4. Sorting the mail into folders

The document assumes that you are running Linux with the X Window System, have already established a connection to an Internet Service Provider, and have a POP account with that provider.

Disclaimer: All usual disclaimers apply. The author cannot be held responsible for any (mis)use of the information given here. This document does not attempt to be authoritative in any way. The author voluntarily creates this document seperately from any entity with which he may be involved with. By reading this document, you agree to cause the author no grief or trouble. :)

2. Installing TkRat

I would like to stress that there are many excellent E-Mail programs available for Linux. It is always a matter of personal preference when selecting which one you will use. That said, I will tell you that I have been using a program called "TkRat" for almost a year now - and I love it. It's a graphical program that has all the features you would expect from a quality mail program, it's easy to learn how to use, it's powerful, and it's free. For these reasons, I'll only be covering TkRat in this HOWTO.

The first thing you need to do is download TkRat. You can either obtain the sources and compile them, or you can get the binaries.

Here are some URLs for the software:

- The Red Hat 5.0 binary is available at http://www.ntlug.org/~dave/sw/tkrat/rh5/tkrat-1.1-1.i386.rpm
- The Red Hat 4.x binary is available at http://www.ntlug.org/~dave/tkrat/sw/rh4/tkrat-1.1-1.i386.rpm
- The Source RPM is available at http://www.ntlug.org/~dave/sw/src/tkrat-1.1-1.src.rpm
- Finally, the tar.gz source is availabe at http://www.ntlug.org/~dave/sw/src/tkrat-1.1.tar.gz

When you have one of the above, you are ready to install it on your system. Follow one of the two instructions below.

- For Red Hat Linux (and other RPM based distributions):

  ```
  rpm -Uvh tkrat-1.1-1.rpm
  ```

- To install the tar.gz source, use the commands:

  ```
  tar -zcvpf tkrat-1.1.tar.gz
  cd tkrat-1.1
  ./configure
  make
  make install
  ```

  If anything fails, consult the INSTALL and README files located in the tkrat-1.1 directory.

Congratulations - TkRat should now be installed on your system.

TOC

3. Downloading your E-Mail

In order to read E-Mail, it is usually best to download it using a program called "fetchmail". The following instructions are known to work perfectly for version 2.2 of fetchmail - although they should be applicable to any version.

(Note: Fetchmail is almost certainly already installed on your system, as it is included in all Linux distributions that I know of - if anyone knows this to be untrue, please let me know!).

Perform the following steps:

```
echo "server pop.com protocol pop3 username dave password foo" > ~/.fetchmailrc

(Make sure to replace "pop.com" with the hostname to your ISP's pop server.  Replace
"dave" with your username, and replace "foo" with your username's password.)
```

Now, whenever you want to check your E-Mail, just type "fetchmail" at the prompt. Fetchmail will go off and download all your mail and store it in your local mail folder.

---

4. Configuring TkRat

From the prompt, type "tkrat" and press enter. Click on "Create". Read the message it gives you, then click on "Continue". At this point, you are brought to the main TkRat program. The first thing you should do is customize it to your local system.

1. Go the Admin menu, and select Preferences.
2. Click on "Compose"
3. Enter in your E-Mail address in the "Default Reply-To" field.
4. Click on "Apply".
5. Click on "Sending".
6. In the "Use From Address" field, enter your E-Mail address, followed by your real name in paranthesis. For example, mine looks like "dave@whitinger.net (Dave Whitinger)"
7. Enter your domain name in the "Domain" field. If your E-Mail address is "user@xyz.com" you'll enter "xyz.com".
8. For "Send Via" select "SMPT".
9. In the SMTP host: field, enter the SMPT server address of your ISP (they should be able to provide you with this information).
10. Click "Apply", then "Dismiss".

5. Using TkRat

Upon starting TkRat, You'll notice that you are in a folder called "INBOX". Use fetchmail to download any new mail. If you have any, it will appear in this INBOX folder.

TkRat has an intuitive interface. The top half of the screen is the list of messages you have (Each line is a seperate E-Mail). The bottom half of the screen will be the actual content of the E-Mail.

You can read E-Mails by simply clicking on any message in the list.

When you are finished reading a message, you can move on the next message, reply to that message, compose a new message, or move the message to a different "Folder".

Folders are a way of storing E-Mails into seperate "Folders". This way, you can keep several folders for different topics you may receive E-Mail about.

Perform the following steps to create folders:

1. Click on Admin, then New/Edit Folders
2. You will see a box with a "INBOX" folder. Left-Click in that box.
3. You are then presented with a menu, so select "New File Folder".
4. You'll be given a dialogue box asking for information.
5. In the first textbox - enter a name for the folder.
6. In the second textbox - enter the filename you will use to store the messages in (Hint: Use something like /home/username/mail/name_of_folder. So, if the first field says "Personal", the second field would say something like "/home/dave/mail/personal" (assuming your username is dave).
7. Click OK. You'll notice your new folder is now in the list of folders. So, go to "Window" and then "Close".

You can now Move messages to the new folder - and switch to that folder by selecting the folder's name under the "Folders" menu.

Sending mail is also easy. Click "Compose", type in the recipient's E-Mail address, type in a subject, and then write the message in the body.

When you are finished, click on Send.

6. Automatically sorting mail into folders

If you're like me, you get about 1,500 messages a day. Each time you fetch your mail, you have to wade through all the messages looking for particular ones that interest you.

This is not a problem anymore, thanks to Procmail.

Procmail works like this: Procmail examines each message as it is downloaded, and will perform a series of tasks based upon certain rules that you've specified.

Let's say, for example, that I get about 200 messages every day from a certain mailing list (in this example, I'm using the linux-foo list). Rather than have all those messages go directly into my main inbox, I would rather that they were automatically filtered into a box called "foo-list".

The first step is to take an example mail message that came from that mailing list and examine the mail headers.

I begin to notice a pattern. Every message that comes from that mailing list has a line that says:

Sender: owner-linux-foo@bar.foogers.com

I can now tell procmail to place every message that contains this line into a certain folder.

The way I tell this to procmail is by way of a ".procmailrc" file.

I will use my favorite text editor to create a text file in my home directory called ".procmailrc".

The file will look something like this:

```
LOGFILE=$HOME/.pmlog
MAILDIR=$HOME/mail
VERBOSE

# linux-foo list
:0 Hw
* ^.*[Ss]ender: owner-linux-foo@bar.foogers.edu
foo-list

# if it got to this point, put it in my new mail folder
:0 Hw
newmail
```

Examine the "Ssender" line. You'll notice that that line is the one thing that all messages from that

mailing list have in common.

That section is telling procmail that when it sees a message come through with a header that looks like the above, to put it into the "foo-list" folder.

The next section is saying that if the message matched nothing above, to just place it into newmail.

Now, let's say that anytime I get a message from my good friend, EJ, I want it to go into a folder called "EJ".

I'll just create a new section of my procmailrc file. See below:

```
LOGFILE=$HOME/.pmlog
MAILDIR=$HOME/mail
VERBOSE

# linux-foo list
:0 Hw
* ^.*[Ss]ender: owner-linux-foo@bar.foogers.edu
foo-list

# Message from EJ!
:0 Hw
* ^.*[Ff]rom: ej@mypal.com
ej

# if it got to this point, put it in my new mail folder
:0 Hw
newmail
```

Notice the new section for EJ. When a message comes in with his E-Mail address in the "From" field, it will automatically place it into my "ej" folder.

Now, let's say that there's some lamer out there who keeps on E-Mailing me. I don't want to hear from him, but he's persistent. Once again - procmail to the rescue.

So, let's say I don't ever want to see any mail from Bill Gates. I can setup a recipe to delete any mail that comes from him. Look at my new procmailrc:

```
LOGFILE=$HOME/.pmlog
MAILDIR=$HOME/mail
VERBOSE
```

TOC

```
# Is it coming from Bill Gates?  If yes, DELETE IT!!!
:0 Hw
* ^.*[Ff]rom: bgates@microsoft.com
/dev/null

# linux-foo list
:0 Hw
* ^.*[Ss]ender: owner-linux-foo@bar.foogers.edu
foo-list

# Message from EJ!
:0 Hw
* ^.*[Ff]rom: ej@mypal.com
ej

# if it got to this point, put it in my new mail folder
:0 Hw
newmail
```

Now I have a rather nice procmailrc file. Let's examine what procmail will do to each message as it comes in.

At first, it checks the message to see if it is from "bgates@microsoft.com". If it is, delete it, and it's done.

If the message made it through the first check, it will see if it has the "owner-linux-foo@bar.foogers.edu" in the headers. If it does, it will put it into the "foo-list" folder, and it's done.

If the message still makes it past that, then it checks to see if it's from EJ. If it is, it places it into the "ej" folder.

Now, if the message passes all those tests, then it should just place it into my newmail folder.

Once you have these folders in place and procmail is properly filtering the mail, you can just go into TkRat, "Admin - New/Edit Folder" and create the folders for each filename. They will be located under /home/username/mail/foldername.

---

7. Copyright

This HOWTO is copyright 1998 by Dave Whitinger, and is a free document. You can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version.

8. Acknowledgements

Very special thanks go out to Martin Forsse'n (maf@dtek.chalmers.se) for authoring the TkRat mail user agent.

Red Hat Software - for making all our lives so much easier.

Greetings go out to the North Texas Linux User's Group in Dallas, TX, of which I am a member, and for whom I wrote this document.

Token-Ring mini-HOWTO

Mike Eckhoff, mike.e@emissary.aus-etc.com

v4.1, 7 January 1998

This howto is designed to help you install the kernel patch and also try to point out some things to look for. I suggest that you at least browse through all of this document before attempting to install any part of the Token Ring driver for Linux.

Special Thanks

to Mark Swanson, Peter De Schrijver, David Morris, Paul Norton and everyone else I may have missed who put in their time to write and maintain this driver. Also to packrat for his support of the linux-tr listserv.

1. Copyright and other Jazz

2. Hardware requirements

3. Software needed

4. Installation and setup

5. NetTools installation

6. Known problems

7. Questions and comments

8. Distribution Specific Installations

- 8.1 Slackware 96
- 8.2 RedHat 4.0+

1. Copyright and other Jazz

We do not guarantee that this howto will be accurate for your system. Most people who have used it have had very good results in installing Linux on a Token Ring network.

USE THIS HOWTO AT YOUR OWN RISK!!! ... We are not responsible for any problems caused by using this howto.

If you have any problems with the driver that are not talked about in this howto, feel free to email me at...

mike.e@emissary.aus-etc.com

You may also wish to join the Linux on Token Ring Listserv by mailing majordomo@emissary.aus-etc.com with the body containing:

```
subscribe linux-tr
```

This document is copyright (c) 1995-1998 by Michael Eckhoff. You may make copies of this document in whole or in part, in any medium physical or electronic, as long as the copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, please notify me so that I can make sure that you have the most current information available.

2. Hardware requirements

Make sure that you have a Token Ring card that is supported by this driver. Currently the only cards that are supported are those that use the Tropic chipset.

Cards that I personally know to work are:

- 3Com 3C619B Token Link
- 3Com 3C619C Token Link
- HyperRing Classic 16/4
- IBM Turbo 16/4 ISA adapter**
- IBM Token Ring Auto 16/4 ISA adapter
- IBM Token Ring Auto 16/4 adapter /A
- IBM Token Ring 16/4 adapter /A
- IBM Token Ring adapter /A
- IBM Token Ring adapter II (4 Megabit only)
- IBM 16/4 ISA Token Ring card (16bit)
- IBM 16/4 ISA Token Ring card (8bit)
- Madge Blue (100% IBM compatable)

All other 100% IBM compatable shared-ram adapters should also work fine. Please let us know if you find differently.

It is recommended that you use 16KB Shared RAM for the time being.

Cards that may cause problems:

IBM Turbo 16/4 ISA adapter

This adapter will, in fact, work fine with the Linux token ring driver. However, you MUST run the card in Auto 16/4 compatability mode. The simplest way to set this is to use the LANAID disks sent with the card and run the command:

```
LANAIDC /FAST=AUTO16
```

You should then use `LANAIDC` or `LANAID` to configure the card according to documentation.

Token-Ring Network 16/4 Adapter II

This adapter will **NOT** work. Do not confuse this card with the IBM Token Ring adapter II (4mbit) which does. It is a DMA/Busmaster adapter for ISA.

3Com TokenLink Velocity ISA

TOC

You may or may not get this one to work. I have had reports of people running it without problems, and others who get errors left and right.

PCI adapters

Currently, none of the IBM PCI adapters are supported.

IBM Auto LanStreamer 16/4 Token-Ring PCI Adapter

Currently not supported, but being worked on. This driver should be Full Duplex as well when completed.

---

3. Software needed

NOTE:

> If you are running a 2.0 distribution of Linux, please jump to the distriubtion specific section of this document. The following is mostly for 1.2 kernels.

This assumes you already have Linux up and running.

Obtain the Token Ring patch from:

> ftp://ftp.wayne.esu1.k12.ne.us/pub/Linux/Token-Ring/TokenRing.patch-1.2.0.gz

Obtain the NetTools patched source from:

> ftp://ftp.wayne.esu1.k12.ne.us/pub/Linux/Token-Ring/net-tools-1.2.0.patched.tar.gz

Create a directory for the patches (such as `/usr/src/patches`) and place the patches there.

```
mkdir /usr/src/patches              central directory for patch storage
mkdir /usr/src/patches/token        place TokenRing patch here
```

4. Installation and setup

NOTE:

These instructions are for patching a 1.2 kernel for token ring support. If you have a 2.0 kernel, you only need to recompile the current source and say ``yes'' or ``module'' when asked for token ring support.

1. Install the Token Ring card into the system and configure it for the settings that you want to use. It is a good idea to see if you can use the card through DOS before trying to use it through Linux. If it works in DOS, chances are, it will work in Linux with the same settings. If you have a Plug and Play adapter, if possible, lock the settings once you get them where they work.
2. Make a backup of your linux directory. This is very important in case you need to totally remove the source of the patch from your kernel and go back to your original code.

```
cd /usr/src
tar cvzhf linuxbak.tar.gz linux
```

3. Uncompress the TokenRing patch.

```
cd /usr/src/patches/token
gzip -d TokenRing.patch-1.2.0.gz
```

4. Modify your kernel with the TokenRing patch.

```
cd /usr/src/linux
patch -p1 < /usr/src/patches/token/TokenRing.patch-1.2.0
```

-or-

```
patch -p1 < <directory-of-patchfile>/TokenRing.patch-1.2.0
```

5. Search your kernel for any rejects from the patch and make changes as necessary.

```
find . -name \*.rej -print
```

6. Search your kernel for the orig files and remove them.

```
find . -name \*.orig -print | xargs rm
```

TOC

7. Configure your kernel and remake.

   NOTE:

   Make sure your swap space is active if you have one.

   ```
   cd /usr/src/linux
   make config
   ```

   (The patch should have added two lines to your `config.in` file for the following options)

   ```
   Token Ring support (CONFIG_TR) [y]
   ```

   (and further down the list...)

   ```
   IBM Tropic chipset based adaptor support (CONFIG_IBMTR) [y]
   ```

   ```
   make dep
   make clean
   make zImage
   ```

8. Setup LILO. First rename your `/vmlinuz` kernel to `vmlinuz.old` then copy the kernel to `/vmlinuz`. On my system this would consist of copying

   `/usr/src/linux/arch/i386/boot/zImage` to `/vmlinuz` and editing `/etc/lilo.conf` to boot that kernel.

   Now from the prompt run ``lilo''.

9. You should now be able to reboot your system and use the Token Ring card in your computer. Please check the distribution specifc section for any extra configuration information.

---

5. NetTools installation

The NetTools package contains a lot of the basic utils that you will use to communicate with network devices. This includes programs like `arp`, `rarp`, `route`, `ifconfig` and `netstat`. Since these programs do not know about Token Ring by default, you will need to add the NetTools patch so these utilities can work more efficiently with the Token Ring driver.

NOTICE:

> The current version of NetTools for 1.2.x kernels is 1.2.0. If you are running a 2.0 kernel, your nettools is most likely already up to date. However, you can get the latest source from:
>
> `ftp://tapac.inka.de/pub/comp/Linux/networking/NetTools/`

1. (1) Copy and Untar the NetTools source into your source directory.

```
cp net-tools-1.2.0.patched.tar.gz  /usr/src
tar -zxvof /usr/src/net-tools-1.2.0.tar.gz
```

2. (5) Make the net-tools files.

```
cd /usr/src/net
make install
```

6. Known problems

I personally have had very few problems with this driver. It has been working perfectly for me for quite some time.

If you have any problems with the driver on a 1.2.x kernel, please update to Linux 2.0. The current version, as of this writing, is 2.0.33. There have been many improvements to the token ring driver since the 1.2 patches and most of them have not been ported back.

Also, you will not have to patch a Linux 2.0 kernel for token ring. The source is already included. However, there are some test patches of the code going into 2.1 available if you would like to use it.

7. Questions and comments

Q:

Can the token ring driver be compiled as a module?

A:

Yes, it can be — and it works rather well. There is an extra parameter that you can use when it is compiled as a module. If you ever need to ``spoof'' software install programs, such as the redhat boot disks, into configuring your token ring card as an ethernet device (for NFS/FTP installs, etc), you can use the ```device`'' parameter to force a device name other then `tr0`. You will probably have problems if you try this with multiple adapters. It is mostly there to get around a few incompatabiliites. Ex.:

```
/sbin/insmod ibmtr device=eth0
```

Q:

I keep getting an error code ``0011''. Whats up?

A:

Make sure that your connection to the network is good or that you have a loopback connector on your token ring card. This message just means that it could not open the ring. 99.99% of the times, it is just not plugged into one.

C:

If you have a sound card in your machine, and it sits at IO 0x220, you may end up with a conflict with your token ring adapter at 0xa20. If you notice that a supported adapter does not seem to be working and you have a sound card, please try to either set your token ring card to 0xa24 or move/remove your sound card.

Also...

Here are some email messages that I have received about Token Ring and Linux. In some of the messages, I have removed parts that were not important to save space.

Q:

```
From: "Mr. Chuck Rickard" <chuck@gl.umbc.edu>
Subject: Re: Token Ring Kernel patch

I d/l'd the patch, applied it, and re-compiled. When booting it said,
"tr0: Can't assign device to adapter" and again for tr1.  Any ideas?

Thanks!

Chuck Rickard
(chuck@umbc8.umbc.edu)
```

A:

```
From: David Morris <dwm@shell.portal.com>
```

When this is the only message issued, it means that the PIO request for adapter information (see `segment = inb(PIOaddr)` in `ibmtr.c`) was so out of range that there is no TR card at that IO address.

Q:

```
From: Mike Glover <glover@credit.erin.utoronto.ca>
Subject: Token ring problems.

Thanks for responding. I was starting to thing that I posted incorrectly,
```

TOC

```
and I was about to post again. Anyway, the following  clip is part of my
/var/adm/messages file from when I boot up.

May  2 10:03:14 linux kernel: tr0: Unable to assign adapter to device.
May  2 10:03:14 linux kernel: tr1: Unable to assign adapter to device.

The section, tr0: Unable to assign adapter device. is what kinda confuses
me. The documentation is slim at best so I didn't know what to do with
the message. I know I didn't assign any token ring information,
and I didn't know where to do it.

The hardware:
            Its a PC clone (Dell OMNIPLEX 560 to be exact)
            The token ring card is a Olicom 16/4 Adapter.

I have DOS token ring drivers and it snaps into the network, so there is
nothing wrong with the card. I think, I am just missing something really
small in the config on the Linux side.
```

A:

The Olicom 16/4 Adapter does not use the Tropic Chipset. Try using one of the cards that are listed at the top of the HOWTO.

Q:

```
From: Mike Glover <glover@credit.erin.utoronto.ca>
Subject: Found an IBM card...

I found myself an IBM token ring card and I got a little further,
but still not luck. (I did change /etc/rc.d/rc.inet1 ifconfig entry
from eth0 to tr0)

Anyway, here is part of the /var/adm/messages file:
May  2 16:23:07 linux kernel: IPX Portions Copyright (c) 1995 Caldera, Inc.
May  2 16:23:07 linux kernel: tr0: PIOaddr:  a20 seg/intr: b8 mmio base:
   000dc000 intr: 0
May  2 16:23:07 linux kernel: tr0: Channel ID string not found for PIOaddr: a20
May  2 16:23:07 linux kernel: tr0: Expected for ISA: 5049434f3631313039393020
May  2 16:23:07 linux kernel: tr0:            found: 000902003021111000182000
May  2 16:23:07 linux kernel: tr0: Expected for MCA: 4d415253363358343531382 0
May  2 16:23:07 linux kernel: tr0: Unable to assign adapter to device.
May  2 16:23:07 linux kernel: tr1: Unable to assign adapter to device.
May  2 16:23:07 linux kernel: PPP: version 0.2.7 (4 channels) NEW_TTY_DRIVERS
   OPTIMIZE_FLAGS


Does this tell you anything. It tells me little.
```

A:

```
From: David Morris <dwm@shell.portal.com>
```

This message means either a memory conflict with the MMIO area or a TR card which is not compatible with the driver (at least the signature isn't known).

Q:

```
From: Mike Glover <glover@credit.erin.utoronto.ca>
Subject: Almost there...

The following sample is what I'm getting on my messages file:
```

TOC

```
May  3 14:50:24 linux kernel: tr0: now opening the board...
May  3 14:50:24 linux kernel: tr0: board opened...
May  3 15:11:47 linux kernel: tr0: Arrg. Transmitter busy for more than 50 msec.
   Donald resets adapter, but resetting
May  3 15:11:47 linux kernel:  the IBM tokenring adapter takes a long time.
   It might not even help when the
May  3 15:11:47 linux kernel:  ring is very busy, so we just wait a little longer
   and hope for the best.
May  3 15:11:47 linux kernel: tr0: Arrg. Transmitter busy for more than 50 msec.
   Donald resets adapter, but resetting
May  3 15:11:47 linux kernel:  the IBM tokenring adapter takes a long time.
   It might not even help when the
May  3 15:11:47 linux kernel:  ring is very busy, so we just wait a little longer
   and hope for the best.
May  3 15:11:47 linux kernel: tr0: Arrg. Transmitter busy for more than 50 msec.
   Donald resets adapter, but resetting

The Arrg entry only comes when I telnet, rlogin or ftp to a remote
machine. Once that happens, the connection hangs big time. I took out all
the other hardware (which was 1 SCSI card) and tried running the card all
by itself. Same thing, so it eliminates hardware IRQ's from getting into
a yelling match. The following are the DIP switches on the card:

1       Up      |
2       Down    |
3       Down    |       According to the manual this sets the
4       Up      |       base address to CC000 Which is fine for
5       Up      |       my machine.
6       Down    |
7       Down    +       This sets the IRQ to 2. Which is also fine
8       Down    +
9       Up      X       Primary Lan adapter. Which it is.
10      Up      =       16 KB shared RAM size. This OK?
11      Down    =
12      Up      #       16 Mbps Data rate.

I was wondering if it is in fact the dip switches, or the way I have
configured my route table and other network info.

ttfn,
Mike
```

A:

The Arrg problem is pretty much taken care of in the 2.0 kernels. All of the patches have not been ported back to 1.2, and I doubt that they will be. If you get excessive Arrg messages, do yourself a favour and move to Linux 2.0.

8. Distribution Specific Installations

8.1 Slackware 96

If you have not already ran ``netconfig'' on your system, do so now. Setup your machine just as if it were on Ethernet.

Edit your `/etc/rc.d/rc.inet1` scripts to point to the `tr0` device rather than the `eth0` device.

You should have a line that looks like

```
/sbin/ifconfig eth0 ${IPADDR} broadcast ${BROADCAST} netmask ${NETMASK}
```

Change this line to read

```
/sbin/ifconfig tr0 ${IPADDR} broadcast ${BROADCAST} netmask ${NETMASK}
```

8.2 RedHat 4.0+

The RedHat distribution of Linux has a wonderful Xwindows control panel for configuring modules and devices. If you have added a token ring adapter to a standard RedHat installation, there should already be token ring support compiled in as a module. Try to:

```
modprobe ibmtr
```

and see what you come up with. Chances are, if you have a supported adapter, you will be ready to go.

The easiest way to configure your interface would be to use the control-panel. You will first want to go to the kernel module control and instruct it to load a new network device for token ring. It should then start to autoload the ibmtr driver on boot. You will then want to jump over to network configuration and add a new device, `tr`, and give it an ip address, etc.

You should then be able to restart and go with Linux and Token Ring on RedHat 4.0+.

Please note that these RedHat instructions came from the top of my head. I havn't had to modify any of my RedHat systems for quite some time and am writing this from memory. Either way, they should be able to get you where you need to go.

The Linux Ultra-DMA Mini-Howto

Brion Vibber, brion@pobox.com

v2.1, 27 May 1999

This document is intended to explain how to use Ultra-DMA aka Ultra-ATA aka Ultra33 and Ultra66 hard drives and interfaces with Linux. The most recent version of this mini-Howto can be obtained in HTML format at http://pobox.com/~brion/linux/Ultra-DMA.html.

1. Introduction

- 1.1 Disclaimer
- 1.2 Credits
- 1.3 Document History
- 1.4 Copying

2. What is Ultra-DMA and why do I want it?

- 2.1 IDE, EIDE, & ATAPI
- 2.2 Bus Master DMA
- 2.3 Ultra-DMA aka Ultra-ATA aka Ultra33 aka...
- 2.4 Just how ``Ultra'' is it anyway?
- 2.5 How does UDMA compare to SCSI?

3. Using your UDMA hard drive with an EIDE interface

4. Using your hard drives with a UDMA interface

TOC

5. Offboard PCI UDMA interfaces

- 5.1 Promise Ultra33
- 5.2 Promise Ultra66
- 5.3 Artop ATP850UF
- 5.4 Adding device files

6. Onboard UDMA interfaces

- 6.1 Intel FX, HX, VX, TX, and LX
- 6.2 The VIA VP2 and Related Chipsets
- 6.3 TX Pro and other ``Pro'' boards

7. UDMA-Generic

8. Activating and Deactivating UDMA

- 8.1 Using kernel boot parameters
- 8.2 Using hdparm

9. Problems

- 9.1 The UDMA Blacklist
- 9.2 Are you overclocking?
- 9.3 Is your BIOS current?
- 9.4 If you still can't get it to work!

10. If you have some information about UDMA stuff that's not in this mini-howto...

---

1. Introduction

This document is intended to explain how to use Ultra-DMA aka Ultra-ATA aka Ultra33 and Ultra66 hard drives and interfaces with Linux. In some cases there is no difficulty in using them, but some tweaking can increase performance. In other cases, you need to go to extraordinary lengths simply to access your hard drives.

1.1 Disclaimer

The information in this is document is, to the best of my knowledge, correct, and should work. However, there may be typos, there may be mysterious transmission errors, and there may be strange incompatibilities within your own system that prevent the techniques described herein from working properly. So... before you go fiddling around with you hard drive, **BACK UP ANY DATA YOU WANT TO KEEP!** If you are not already performing regular backups, please start doing so for your own good.

1.2 Credits

Michel Aubry - UDMA-enabled VIA-related patch for <=2.0.33 & more info, grand unified UDMA patch for 2.0.34+

Andrew Balsa - Provided some general UDMA info and the udma-generic patch for Intel TX, SiS, and VP1 on <=2.0.33; also the grand unified UDMA patch for 2.0.34+

Maxime Baudin - French translation

Bokonon - ``Controller'' vs. ``interface''

John G. - VIA VP2 patch for <=2.0.33 & info

Martin Gaitan - Promise Ultra33 ide0/ide1 installation workaround

Andre M. Hedrick - Grand unified UDMA patch for 2.0.34+, Artop ATP850UF

Norman Jacobowitz - Bugged me to add info on the VP3

John Levon - Info on TX Pro mobos

Peter Monta - Info on using two Ultra33 cards

Masayoshi Nakano - Japanese translation

Gadi Oxman - The Promise Ultra33 patch for <=2.0.34 & finding the secret numbers for the workaround

Andy Pearce - Suggested adding info on the additional device files for hde-h

Andrei Pitis - LILO patch

Brion Vibber - The document itself

1.3 Document History

v2.1, 27 May 1999: Corrects some minor omissions and errors from 2.0 and adds information on the Promise Ultra66 and 2.2/2.3 kernels.

v2.0, 7 August 1998: Major updates and almost total restructuring of the document into onboard (motherboard) and offboard (add-in cards) interfaces; the Grand Unified UDMA patch(a part of the Jumbo patch) for 2.0.35. Put credits in alphabetical order by last name. Changed ``controller'' to ``interface'' in many cases to be more technically correct. Added info on enabling/disabling UDMA, the blacklist, and more!

v1.45, 6 July 1998: Minor updates - Red Hat 5.1 and 2.0.34 patch for Promise Ultra33, LILO patch for booting off of PCI interfaces such as the Promise Ultra33

v1.41, 3 May 1998: Fixed a couple of typos, added translators to credits.

v1.4, 28 April 1998: UDMA-Generic patch, some more general info. Copying section added.

v1.3, 5 March 1998: VIA VP3 info, better patching instructions, pointer to more recent Promise patch.

v1.2, 27 January 1998: Additional Promise workaround info.

v1.1, 21 January 1998: New info about VIA chipset, installing *around* the Promise Ultra33, and enabling Bus Master & UDMA transfer modes.

v1.0, 19 January 1998: More or less complete, first version done in SGML.

1.4 Copying

This document may be freely copied and distributed for informational purposes. It may not be modified, except for reformatting, without the permission of the author. If you wish to translate this document into another language you may do so, however you should contact the author first so that updated versions of this document can be sent out to translators as well as directly to the Linux Documentation Project.

2. What is Ultra-DMA and why do I want it?

Here's a brief overview of IDE-based drive technologies:

2.1 IDE, EIDE, & ATAPI

These are older drive technologies. Most non-SCSI hard drives and drive interfaces that you can buy today or are likely to be using are EIDE, although many of the larger drives now available are UDMA.

2.2 Bus Master DMA

Bus Master DMA is a technology for increasing the speed of hard disk data transfers which requires support from the motherboard and the BIOS, and at least some support from the drive.

You can learn more at http://developer.intel.com/design/pcisets/busmastr/FAQs.htm.

2.3 Ultra-DMA aka Ultra-ATA aka Ultra33 aka...

Ultra-DMA has many names, but we'll just call it UDMA in here.

UDMA is a more advanced technology which provides for even faster throughput, up to 33.3 MB/s in UDMA mode 2 and 66.7 MB/s in UDMA mode 4, twice to four times that of EIDE, for much lower prices than SCSI. Many new computers come with large UDMA drives and UDMA interfaces, and it's possible to add a UDMA interface card (such as the Promise Ultra33 or Ultra66) to an existing system to boost speed, even on older non-UDMA drives.

You can learn great details about UDMA at http://www.quantum.com/src/whitepapers/ultraata/

Note that cable length should be kept shorter for UDMA, compared to plain DMA, preferably less than 30 cm (12") maximum length. 66 MB/s requires a special cable and must be kept even shorter.

2.4 Just how ``Ultra'' is it anyway?

Before we get any farther, let's clear up a misconception. That 33 or 66 MB/sec figure is the **burst transfer rate**, and it's not something you're going to see very often. To explain, here is a clip from udma-generic's UDMA.txt:

```
Burst (instantaneous) transfer rates are supposed to go from 16.6MB/s (PIO
mode 4) to 16.6MB/s (DMA mode 2) up to 33MB/s (UDMA). In his patch against
kernel 2.1.55, Kim-Hoe Pang actually checked the UDMA burst transfer rate
with a logic analiser: 60ns/word, which translates into 33MB/s.

Note that burst transfer rates only affect data transfers to/from the EIDE
drive cache (476kB for the IBM 6.4GB drive), and IMHO are not particularly
relevant for most Linux users.

The Linux kernel uses as much RAM as possible to cache hard disk data
accesses, and so if data is not in the kernel cache there is little chance
that it will be in the (much smaller) hard disk cache.
```

TOC

Much more relevant is the **sustained transfer rate**, the speed at which data can be transferred from the drive to m memory where it can be used. An easy way to measure the sustained transfer rate is to use `hdparm`, for instance ``hdparm -Tt /dev/hda'' to measure the rate of the first IDE device.

```
Here is some data gathered after extensive testing, using the hdparm utility
(also written by Mark Lord):

PIO mode 4 transfer rates under Linux:   +/- 5.2MB/s

DMA mode 2 transfer rates under Linux:   +/- 7.2MB/s

UDMA mode 2 transfer rates under Linux:  +/- 9.8MB/s
```

As you can see, UDMA is still almost twice as fast as plain EIDE and significantly faster than plain bus mastering DMA.

2.5 How does UDMA compare to SCSI?

I don't have any hard numbers to give you, but the general consensus is that high-end SCSI can give better performance than UDMA. However if you've looked at the price tags on any hard drives lately you'll notice that UDMA drives tend to be much less expensive. The performance/price ratio favors UDMA in most cases.

---

3. Using your UDMA hard drive with an EIDE interface

This is easy to do. Since all UDMA drives are fully EIDE backward-compatible, just plunk your drive on your EIDE interface like it was any old hard drive and Linux should have no problems detecting or using it. However, you will of course be limited to the slower speed of EIDE.

4. Using your hard drives with a UDMA interface

Well, there is good news and there is bad news. The good news is that a UDMA interface can be used with both UDMA hard drives and legacy EIDE hard drives, and will be a lot faster than an EIDE interface.

The bad news is that the old stock kernels (2.0.x) do not currently support UDMA very well. The new 2.2.x kernels do support UDMA33, however, and there are a number of kernel patches available for currently existing kernels. In addition, certain UDMA interfaces that are add-in cards rather than built into the motherboard require either a patch or some trickery to use on older kernels. That is why this document exists - to explain how to get the patches and work the trickery.

5. Offboard PCI UDMA interfaces

These are UDMA interfaces on PCI cards that can be used to add UDMA support to an existing computer without replacing the motherboard, or for adding support for an additional four drives to a machine which has had its onboard interfaces filled. They can also be found preinstalled in some computers, especially Gateway 2000 and Dell machines.

Most of them are not supported by the old stable kernels (2.0.x), but should work with a 2.2.x kernel - the Red Hat 6.0 and SuSE 6.1 distributions are based on 2.2.x kernels, as are the most recent versions of most other distros. However some of the latest cards (the Promise Ultra66 for instance) won't work even with the current 2.2.x kernels, if you have this or can't get a newer distribution then you must apply a kernel patch or upgrade to a newer kernel version. If you need to install Linux onto a hard drive on one of these interfaces in this case, you will need to use a few odd tricks.

5.1 Promise Ultra33

This is a PCI card that has two UDMA channels on it, supporting up to four drives total. You can look up specifications & pricing at http://www.promise.com.

This card shipped in early model Gateway 2000 Pentium II systems, and may or may not be in more recent models.

Kernels 2.0.35 and later and all 2.2.x kernels support the Ultra33 and you should have no trouble installing a distribution that uses these kernels. However, the older stable kernels (2.0.34 and below) do not, and since most older Linux distributions include these older kernels it can be a little difficult to get Linux installed if you can't or don't want to use a newer version (for instance if you are standardized on a particular version of a distribution throughout your organization).

Installing Linux with the Ultra33

Although there is a patch for the Ultra33 interface, it is not very easy to apply a patch and recompile your kernel if you have not installed Linux yet! So, here is a workaround which allows you to install. Thanks to Gadi Oxman for the following information on getting the interface settings:

```
If we can access the console with the installation disk, we can also
use "cat /proc/pci" to display the Promise interface settings:

    RAID bus interface: Promise Technology Unknown device (rev 1).
      Vendor id=105a. Device id=4d33.
      Medium devsel.  IRQ 12.  Master Capable.  Latency=32.
      I/O at 0xe000.   (a)
      I/O at 0xd804.   (b)
      I/O at 0xd400.   (c)
      I/O at 0xd004.   (d)
      I/O at 0xc800.   (e)

and pass "ide2=a,b+2 ide3=c,d+2" as a command line parameter to the kernel.
```

Note that the numbers probably are not the same as what you will have. Just as an example, the parameters to use for the above set of numbers would be ```ide2=0xe000,0xd806 ide3=0xd400,0xd006`". If you are only using the first channel on the Ultra33 (for instance, if you only have one drive, or two if they are master and slave on the same channel), then you won't need to specify `ide3`.

Red Hat 5.1: Boot with the boot diskette and press enter when prompted. The kernel will load, and then you will be asked for a language, keyboard type, and installation method. You may be prompted for additional information about the source media; it doesn't matter right now what you tell it as long as you can get to the next step. Next you should see a screen titled ``Select Installation Path''; press Alt-F2 now to get to a command prompt. Run ```cat /proc/pci`'', write down the numbers as above, and reboot from the boot disk. This time, type ```linux ide2=` *(this is where you put the numbers like shown above)* `ide3=`*(more numbers)*''. It should now be able to install onto your hard disk without difficulty, however LILO will probably not be able to install; instead make a boot floppy and boot it with the same parameters until you can patch LILO and and the kernel.

Red Hat 5.0 and **Slackware 3.4:** These are similar, but with the wrinkle that the setup programs ignore `/dev/hde-h` (the drives on `ide2` and `ide3`). In order to install to or from these drives it is necessary to override one or both of the onboard interface's channels. However be sure not to override a device that you need to install; for instance if you are installing from a CD-ROM drive on `/dev/hdd` (`ide1` - onboard interface) to a hard drive on `/dev/hde` (`ide2` - the Ultra33), you should override the non-essential `ide0` with `ide2` and leave `ide1` intact. Assuming the numbers above you would boot with ```ide0=0xe000,0xd806`''. Red Hat 5.0 will give you a shell prompt if you use the rescue disk capability, and Slackware includes a shell in the regular installation process. However Red Hat 5.0 is difficult to boot after installation; if you have problems you could try downloading a Slackware boot disk from ftp://ftp.cdrom.com/pub/linux/slackware-3.5/bootdsks.144/ and using that to boot.

With another Linux distribution you will have to improvise a bit, but the process should be about the same as the above.

IMPORTANT: Without the patch (discussed in the section UDMA-Generic), the kernel **needs** these boot parameters in order to access your hard disk! Therefore it is very important that when you configure LILO, either on the hard disk or on a boot floppy, that you give it the **exact same parameters** that you gave when installing. Otherwise your system won't boot! It should be possible to give them to LILO when you boot (ie, press Shift, type in ```linux ide2=`.....'' each time you boot), but only if you kept the numbers! It is recommended that you patch your kernel as soon as possible so you will not have to worry about that anymore; once you are booting with a patched kernel, you can get rid of the boot parameters. Also, as far as I know there is no way to pass boot parameters to a plain kernel boot floppy (as made with ```make zdisk`''), you **must** use LILO or another loader (such as LOADLIN) that lets you pass boot parameters.

However, unpatched kernels and installation programs often have a difficult time actually using ide2 and ide3, even if the drives are detected properly. So if you can't get Linux to install using the above technique, try specifying ide0 or ide1 instead of ide2 or ide3 (thanks to Martin Gaitan for this technique). This essentially replaces the on-board interface with the Promise Ultra33 as far as the kernel is concerned, and you can follow the directions in the next section as if you had physically moved it. Note that if you're using an IDE CD-ROM drive connected to your on-board interface to install from, you will want to make sure that you do not take over the interface that the CD is on or you will not be able to install! If the CD is hda or hdb, use ide1 for your hard drive, and if it is hdc or hdd, then use ide0.

Installing Linux Around the Ultra33

If you cannot get the software workaround to work, you will have to try a more brute force approach. Here's an alternative method that is virtually guaranteed to work, but will require you to open up your computer and mess about in it. **NOTE:** If you are not familiar with the process of connecting and disconnecting IDE drives, **read the manuals** that came with your computer, your hard drive, and/or the Promise Ultra33 before attempting this! If you screw something up and don't know how to put it back, you could end up being sorry!

That being said, it's all really quite simple. Most motherboards these days have built-in EIDE interfaces. Disconnect your hard drive from the Ultra33 and connect it to the onboard interface. If you have other IDE devices, such as a CD-ROM, tape, or ZIP drive, on your oboard interface, it is easiest if you either add the hard drive on an unused channel (the secondary instead of the primary) or temporarily displace a device that you don not need immediately (such as ZIP or tape). Install Linux. Download and apply the Promise UDMA patch (see next section).

TOC

Now you are ready to move the drive back onto the Promise... almost. To be safe, make a kernel-image boot flop (`cd /usr/src/linux ; make zdisk`), which you will be able to use to boot your system in case LILO doesn't work. Actually, to be *very* safe, make two and put one away for now.

Okay, now it is time to think a little... if you have just one hard drive and it is going to be on the Promise, then it will most likely be `/dev/hde` (`a` and `b` are for the primary onboard interface, `c` and `d` for the secondary onboard interface). If you are going to put any other drives on it, then the slave of the Promise's first channel will be `/dev/hdf`, the master of the second will be `/dev/hdg`, and the slave of the second will be `/dev/hdh`.

Edit `/etc/fstab`, and change all the partitions of the hard drives you are moving from the onboard drives (`/dev/hda`, `hdb`, etc) to their new locations on the Promise (`/dev/hde`, `hdf`, etc). If you had to displace any devices (such as a CD-ROM or ZIP drive) that you want to leave on the onboard interface, then change them to their new locations as well. For instance, if your CD-ROM was originally the master on the primary channel (`/dev/hda`), but you put your hard disk there and had to bump the CD to the slave (`/dev/hdb`) or to the secondary channel (`/dev/hdc`), and now you want to put it back, then change it to `/dev/hda`.

If you are using LILO, reconfigure LILO to use the new location of the drive (LILO configuration is beyond the scope of this document, if you do not know how, read the LILO mini-HOWTO), or else it probably will not be able to boot unless you use that boot floppy I had you make, which you will also want to configure to boot off the new partition. This is done using the `rdev` command. Put the floppy in the drive and type ``rdev /dev/fd0 /dev/hde1''. Of course that's assuming your root partition is the first on your first UDMA drive. If not (mine is `/dev/hde7`, for instance), then obviously use the appropriate partition number!

Reboot. Your system should now work fine.

Patching for the Ultra33

Kernels 2.0.35 and later support the Promise Ultra33 natively; download an upgrade from your Linux distribution or from http://www.kernel.org.

For instructions on how to compile the kernel, read the Kernel HOWTO.

Using two Ultra33 cards in one machine

This is currently not working correctly... don't do it right now unless you're willing to fiddle with the kernel to try to get things to work.

5.2 Promise Ultra66

This is essentially the same as the Ultra33 with support for the new UDMA mode 4 66 MB/sec transfer speed. Unfortunately it is not yet supported by 2.2.x kernels.

There is a patch for 2.0.x and 2.2.x kernels availabe at http://www.dyer.vanderbilt.edu/server/udma/, and support is included in the 2.3.x development kernel series at least as of 2.3.3.

However to get far enough to patch or upgrade the kernel you'll have to pull the same dirty tricks as for the Ultra33 as in the section above.

5.3 Artop ATP850UF

This card is supported by udma-generic. Installation of Linux onto a system with one of these as the interface for the target disk may be similar to the workarounds for the Promise Ultra33.

5.4 Adding device files

The tertiary and quaternary IDE interfaces (ide2 and ide3) use device files of the form `/dev/hde*` through `/dev/hdh*`. On older kernels these devices were not automatically created, so you may need to add them manually for things to work properly.

This can be done easily if you have a current copy of the Linux kernel source installed; simply run `/usr/src/linux/scripts/MAKEDEV.ide` and it will create all relevant device files.

---

6. Onboard UDMA interfaces

These are UDMA-capable drive interfaces built into motherboards. They use the standard IDE I/O ports and so are fully usable at the slower non-UDMA speeds on an unpatched 2.0.x kernel such as are used when installing Linux. Thus they should not cause any difficulties during installation, and patching for UDMA speed is a welcome luxury instead of a necessary step. Some UDMA support is in the latest 2.0.x kernels I believe, and is built into current 2.2.x kernels for the Intel chipsets.

6.1 Intel FX, HX, VX, TX, and LX

Thanks again to Gadi for this info:

```
Bus mastering DMA support for the Intel TX chipset is available in 2.0.31
and above.
```

In older kernels (such as Slackware 3.4's 2.0.30), the interface will be used in the slower EIDE mode. In either case the interface will be automatically detected by the kernel and you should have no trouble using it.

Full UDMA support for these chipsets is included in the udma-generic patch; see UDMA-Generic.

6.2 The VIA VP2 and Related Chipsets

This interface also can be autodetected and used in EIDE mode by an unpatched kernel, but if you have one of these, you will want to grab a patch so you can get faster throughput and do away with annoying "unkown PCI device" messages.

One is available at http://www.ipass.net/~prefect/; it is designed for the VIA VP2/97 chipset, found on FIC's PA-2007 and PA-2011 motherboards, but may work on related chipsets. It has been reported that it functions on the newer VIA VP3 chipset, your mileage may vary.

Note that this patch only supports Bus Mastering mode, not full UDMA mode, but it's still better than plain-vanilla EIDE mode. Follow the directions at the patch's site for enabling BMDMA mode.

There is another patch that supports full UDMA mode at http://www.pyreneesweb.com/Udma/udma.html, designed for the VIA VT82C586B, and it ought to work on the VP2, VP3, VPX, P6 and AGP Apollo chipsets. Follow the directions for installation and UDMA enabling there, but it is recommended that you back up any data you want to keep, as there are potential problems with incompatible motherboards. But, if it does work, it should work without problems.

Note that the VP1 chipset is not known to work with these patches, but is supported by the UDMA-Generic patch.

6.3 TX Pro and other ``Pro'' boards

UDMA is not currently supported for the TX Pro motherboards. They are not the same as a TX mobo, and apparently misreport their DMA capabilities hence the problem. Someone is working on this I hear, so a patch may appear some time in the future but not yet.

7. UDMA-Generic

The UDMA-Generic patch, modified by André Balsa, Andre Hedrick, and Michel Aubry from Mark Lord's original Triton DMA driver, provides UDMA support for the following chipsets (as of version 0.3):

- All Intel chipsets: FX, HX, VX, TX, LX
- All SiS chipsets (only SiS5598 tested, but this entire family of chipsets has the same bult-in 5513 interface device).
- VIA chipsets (only 82C586B tested, but again this family of chipsets has the same interface structure). Special diagnostics support is available for the VIA interfaces.
- Promise and Artop PCI UDMA interface cards support. Triones PCI UDMA interface card not included yet.

It is also designed to be easy to extend to support other chipsets.

Udma-generic, also known as the Grand Unified UDMA Patch (GUUP - pronounced ``goop''), has been folded into the Jumbo patch which includes other useful features such as automatic detection of memory over 64 MB and CPU clock speed detection. Unfortunately this patch has vanished and I'm not entirely sure where to find that patch now, so instead one should go to http://www.dyer.vanderbilt.edu/server/udma/ where IDE/UDMA patches for current 2.0.x and and 2.2.x kernels can be found, now including support for the HPT343 and ALi15x3 chipsets.)

Here are a few notes from the author:

```
Performance with IBM UDMA drives on a good motherboard approches the
maximum head transfer rates: about 10 Mb/s (measured with hdparm -t -T).

The Intel TX chipset has a single FIFO for hard disk data shared by
its two IDE interfaces, so using 2 UDMA drives will not yield such a
great improvement over a single UDMA drive.
However, the SiS5598 has two completely separate interfaces, each with
its own FIFO. Theoretically, one could approach 66Mb/s burt transfer
rates on motherboards with the SiS5598 chip, using the md driver and
data striping over two drives. The SiS5571 has the same interface
architecture, I think. I don't have the VIA chipsets datasheets, so I
can't say anything about those.

The Linux IDE (U)DMA kernel driver by Mark Lord has a particularly
low setup time (i.e. latency for data transfers). It is ideal for
frequent, small data transfers (such as those in Linux news servers),
and might be in some cases superior to its SCSI counterparts.
```

8. Activating and Deactivating UDMA

Normally, a UDMA-aware kernel will automatically enable UDMA support for drives and interfaces that support it. However starting with kernel 2.1.113 DMA drive transfers are disabled by default unless you have configured your kernel specifically to automatically activate it. This may seem inconvenient, but some drives and some interfaces do not function properly with DMA enabled; see the UDMA Blacklist below.

8.1 Using kernel boot parameters

On kernels 2.1.113 and up, you can enable DMA for both drives on a given IDE interface using the `ideX=dma` kernel parameter, where X is the number of the interface (the first is 0).

Kernel boot parameters can be set using LILO, LOADLIN, or most Linux boot loaders. For more information see the Bootdisk HOWTO.

8.2 Using hdparm

`hdparm` is a program used to tweak the parameters of hard drives under Linux. Among other things you can use it to enable or disable UDMA for a drive and test its sustained transfer rate.

The current version of `hdparm`, 3.5i as of this writing, is available at http://www.dyer.vanderbilt.edu/server/udma/.

Compile and install it something like this:

```
cd /usr/src
tar zxvf /tmp/download/hdparm-3.5.tar.gz
cd hdparm-3.3
cat /tmp/download/hdparm-3.5i.patch | patch -p0
make
if [ -f /sbin/hdparm ]; then rm -f /sbin/hdparm ; fi
make install
cp /usr/local/sbin/hdparm /sbin/hdparm
```

To enable DMA for a hard drive: `hdparm -d1 /dev/hda`

To disable DMA for a hard drive: `hdparm -d0 /dev/hda`

To measure transfer rate of a hard drive: `hdparm -Tt /dev/hda`

To see what options are enabled for a hard drive: `hdparm /dev/hda`

To see more info on your drive than you wanted to know: `hdparm -i /dev/hda`

For more detailed info (such as how to choose which UDMA mode to use) read the man page (``man 8 hdparm'').

9. Problems

9.1 The UDMA Blacklist

The following drives are ``blacklisted''. You should **not** use UDMA with these drives as it may cause corruption of data.

- Western Digital WDC AC22100H

The UDMA portion of the Jumbo-2.0.35-9 patch automatically disables DMA for this drive.

9.2 Are you overclocking?

If you are, beware! Here is a quote from the udma-generic documentation:

```
DON'T OVERCLOCK the PCI bus. 37.5MHz is the maximum supported speed for
the PCI bus. Some (supposedly compatible) UDMA drives will not even take
37.5MHz, but should be OK at 33.3MHz.

In any case, NEVER, NEVER set the PCI bus to 41.5MHz.

The RECOMMENDED safe setting is 33MHz.
```

9.3 Is your BIOS current?

Here is another clip from the udma-generic docs:

```
The real work involved in setting up the chips for DMA transfers is done
mostly by the BIOS of each motherboard. Now of course one hopes that the
BIOS has been correctly programmed...

For example, the ASUS SP-97V motherboard with its original BIOS (Rev. 1.03)
would malfunction with the modified Linux driver in both DMA mode 2 and UDMA
modes; it would work well using PIO mode 4, or under Windows 95 in all
modes. I downloaded the latest BIOS image (Rev. 1.06) from the ASUS Web site
and flashed the BIOS EPROM with the latest BIOS revision. It has been
working perfectly ever since (at 66 MHz bus speeds).

What this tells us is that the BIOS sets up the DMA controller with specific
timing parameters (active pulse and recovery clock cycles). My initial BIOS
revision probably had bad timings. Since the Windows 95 driver sets up those
timings by itself (i.e. it does not depend on the BIOS to setup the hard
disk controller timing parameters), I initially had problems only with the
Linux driver, while Windows 95 worked well.

So, let me state this again: this Linux (U)DMA driver depends on the BIOS for
correct (U)DMA controller setup. If you have problems, first check that you
have the latest BIOS revision for your specific motherboard.
```

TOC

```
...

New BIOS revisions can be downloaded from your motherboard manufacturer's
Web site. Flashing a new BIOS image is a simple operation but one must
strictly follow the steps explained on the motherboard manual.

Late Award BIOS revisions seem stable with respect to UDMA. Anything with a
date of 1998 should be fine.
```

9.4 If you still can't get it to work!

If nothing in this document proved helpful, or at least not helpful enough to get your machine working, your best bet is to write up a message that fully describes your difficulty, what type of UDMA interface you have, whether it is onboard or on a card, if your drive is actually UDMA or plain EIDE, exactly what configuration of drives you have, what version (distribution & kernel versions if possible) of Linux you are using, and anything else that sounds useful, and post it to the newsgroup comp.os.linux.hardware. You will probably get some helpful suggestions soon.

---

10. If you have some information about UDMA stuff that's not in this mini-howto...

Great! If you know something I don't, by all means send it to me (brion@pobox.com) and I will put it in this document and update it fairly soon.

HOWTO: How to stay updated

Kjetil Bakkeskaug, Kjell Sundby and Stein Gjoen, sgjoen@nyx.net

v0.6, 3 February 1998

This document describes how to stay updated and abreast of the development that takes place in the Linux world of development. Although most of this text is Linux specific there is also a lot of general information on searching efficiently for specific information that can be useful for a wider audience.

1. Introduction

- 1.1 Copyright
- 1.2 Disclaimer
- 1.3 News

2. Information

3. Subscription

- 3.1 Usenet News
- 3.2 Mailing Lists
- 3.3 Magazines

4. Meetings

5. Searching

6. Conclusion

1. Introduction

Development in the world of Linux takes place at an incredible speed and it can be difficult to keep abreast with the latest development. This HOWTO gives you a few guidelines on how to get the information you need, fast and efficiently. Most are quite familiar with using the World Wide Web (WWW) and Usenet News but as will be shown here there are many other methods that can be as good or even better.

1.1 Copyright

This HOWTO is copyrighted 1997 Kjell Sundby, Kjetil Bakkeskaug and Stein Gjoen.

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu via email.

1.2 Disclaimer

Use the information in this document at your own risk. We disavow any potential liability for the contents of this document. Use of the concepts, examples, and/or other content of this document is entirely at your own risk.

All copyrights are owned by their owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

You are strongly recommended to take a backup of your system before major installation and backups at regular intervals.

1.3 News

Added more information on the workings of mailing lists.

2. Information

When you start out with a Linux installation you will normally get quite a bit of information along, not just the installation pamphlet but also substantial online help and information files as well as HOWTO files. This gives you a good starting point but after a while you will find yourself interested in knowing more, updating your system and basically staying informed. For simplicity this kind of information is here divided into several types, the type you subscribe to, information you search for as well as a bit on getting more specific help efficiently.

Even if you don't get printed information of some kind with your Linux packages you will along with any self respecting distribution get a number of directories with documentation of some kind, ranging from the tersest *README* files for most software packages to the more in depth *HOWTOs*, of which this is one.

Have a look in the document archive where most packages store their main documentation and README files etc. Also you will here find the HOWTO archive of ready formatted HOWTOs and also the mini-HOWTO archive of plain text documents.

The kernel source is, of course, the ultimate documentation. In other words, *use the source, Luke*. It should also be pointed out that the kernel comes not only with source code which is even commented (well, partially at least) but also an informative documentation directory. If you are about to ask any questions about the kernel you should read this first, it will save you and many others a lot of time and possibly embarrassment.

The online documentation is excellent for browsing and searching but don't dismiss the printed version altogether; if you cannot even get the machine to boot, how are you going to be able to read that piece of information you need to get the system going again?

3. Subscription

This basically means you set up a subscription of some sort and then follow the news as they come in. be careful not to bite over more than you can chew, *information overload* is more than a buzzword. There is also rather more junk out there than is just annoying, it is a real problem these days. Read critically and be prepared to unsubscribe.

There are two distinct medias for getting continuous updates: news and mailing lists, though sometimes news is gatewayed to mail and vice versa. In general news is a larger volume, larger noise source compared to mailing lists. Trying to follow too many newsgroups is like drinking from a fire hose.

3.1 Usenet News

Getting access to Usenet News is outside the scope of this HOWTO, there are others that will help you with getting and reading News directly from a Linux system. If you have never used News before you should be careful to read the introductory information thoroughly. In spite of looking like an anarchy it does have its own distinct culture, follow a newsgroup for some time before posting yourself. Most importantly, look out for postings called *Frequently Asked Questions* or *FAQ* as they will show you the ropes for the group it is posted to, and most likely give you the answer to what you are looking for. Asking an FAQ will earn you severe negative credibility points as well as a place in many killfiles.

FAQs should be posted regularly but if you cannot find it you can always find it at the main FAQ archive at MIT.

These are also available as web pages.

Still, there is a lot of noise, spam and junk in News and this is where killfiles come in. You will need a news reader with killfile capability and when properly set up it will scan through a newsgroup according to a search key of your own design and mark all flagged postings as already read so you don't have to be bothered by the noise. This improves the signal-to-noise ratio and lets you concentrate on the important parts. Similarly, if you make noise in News others will killfile you so if you later were to ask for help they will never see your post.

Now to business: the following is a list of useful newsgroups:

- announcements
- answers
- development of applications
- development of the system
- hardware
- misc

TOC

- networking
- setting up linux
- X11 on linux

Many national hierarchies also have Linux groups, such as the Norwegian Linux groups. If you cannot find your national or local group you might be able to use Dejanews to find the names for you.

3.2 Mailing Lists

Unlike Usenet News a mailing list is centralised, someone sends a mail to the server and the server in return mails everyone that is subscribed to that particular list. These lists are generally low volume but also very low noise. Any breaches of the charter will be looked harshly upon. Equally seriously it will delay the development or the project that the list is dedicated to. When you subscribe you will normally get an introductory mail describing the charter, again you are strongly recommended to read this very carefully.

There are many types of mail servers that can handle a list and you will need some information on how and where you can subscribe.

One of the most common list servers is *Majordomo* which is what the list server at vger.rutgers.edu. is running. To learn how it works you send a mail message with the word `help` in the body. If you send it something it cannot parse you will get this help message anyway. If you instead mail it the word `lists` you will be returned a list of all mailing lists it serves, and that can be a considerable number.

Other mailing lists use several addresses, one where you send your requests such as `subscribe` and `unsubscribe`, and one where you send your contributions to the list which is usually also the address from which the list is also redistributed to you. Again, sending it the message `help` or something it cannot parse will give you the help information. An example: you send the word `subscribe` to the address `corned-beef-requests@somelistserver.org` and then you get mail from and contribute to the list address `corned-beef-list@somelistserver.org` until you unsubscribe.

A few tips before you start sending in to mailing lists:

- Do not send `subscribe` etc. to the list itself, only to the server address, otherwise you will look silly and you will annoy people. There can be several thousand subscribers to a list and if such errors were to pour in the noise would be too much.
- When you subscribe you will often get an introductory message sent to you automatically. Read it carefully as this should answer most of the initial questions.
- Do not gateway mailing lists to news without asking first as this can cause mailing loops as well as spam.

As mentioned above, vger.rutgers.edu. is one of the main mailing list servers and here is an abbreviated index of what is available for the Linux community:

- linux-8086
- linux-admin

TOC

- linux-alpha
- linux-apps
- linux-arm
- linux-bbs
- linux-c-programming
- linux-config
- linux-console
- linux-diald
- linux-doc
- linux-fido
- linux-fsf
- linux-ftp
- linux-gcc
- linux-gcc-digest
- linux-hams Amateur Radio and Linux discussions
- linux-hppa
- linux-ibcs2
- linux-ipx
- linux-isdn
- linux-japanese
- linux-kernel
- linux-kernel-announce
- linux-kernel-digest
- linux-kernel-patch
- linux-laptop
- linux-linuxss
- linux-lugnuts
- linux-mca
- linux-mips
- linux-msdos
- linux-msdos-devel
- linux-msdos-digest
- linux-net
- linux-new-lists
- linux-newbie
- linux-newbiew
- linux-nys

TOC

- linux-oasg
- linux-oi
- linux-opengl
- linux-pkg
- linux-ppp
- linux-pro
- linux-qag
- linux-raid
- linux-scsi
- linux-serial
- linux-seyon
- linux-smp
- linux-sound
- linux-standards
- linux-svgalib
- linux-tape
- linux-term
- linux-userfs
- linux-word
- linux-x11
- linux-x25
- sparclinux
- ultralinux

There are of course a number of other lists on other server. As this is in a constant state of flux there is little point in naming all but the most important here. Instead you could check out a web page that maintains such a list of lists on various servers of interest to Linux users. It also offers an user friendly interface to subscribe or unsubscribe to the various lists directly.

There is also a web page listing a huge number of lists concerning much more than Linux at list index.

3.3 Magazines

Many have been disappointed at the lack of information on Linux in the trade press. This is probably because certain commercial products would not stand up for any comparison and the advertisers would not stand it at all. Fortunately there is one Linux specific journal, called the *Linux Journal*. More information on subscription etc. can be found at the SSC home page. A table of contents is usually also available online.

There is also an e-zine called Linux Gazette.

4. Meetings

Linux has been created through a massive networked effort, mostly by heavy use of the Internet. Still, there is the chance of meeting real people, face to face, in Linux user groups (LUG) that are all over the world. Search the lists that are published regularly, there could be one near you.

Conferences, install fests, creating new user groups and more is regularly announced on Usenet News announcements. Such events are excellent venues for staying on top of events and also for getting help.

5. Searching

There are many avenues open when searching for something particular. Remember you can also use the web search engines and that some, like

- Altavista
- Excite
- Hotbot

can also search usenet news.

Also remember that Dejanews is a dedicated news searcher that keeps a news spool from early 1995 and onwards.

Even though more and more things take place of the web these days, do not forget that there is a lot of information available on the various ftp servers around the world. Some web search engines also index ftp servers but the tool of choice is still the archie servers, systems that regularly scan major ftp servers around the world and keep lists of files. These can be accessed in many ways, either by archie clients like `archie` or the X11 version `xarchie` which should be available on any well maintained linux system. Failing that you can access archie servers using telnet to any of the servers listed

- Australia
- Austria
- Belgium
- Finland
- Germany
- Korea
- Italy
- Japan
- Poland
- Sweden
- Spain
- United Kingdom
- United States
- United States

Of course you should try to use the server closest to you, and to see the list of current server you can either start the archie client with no arguments or, if telnetting, by querying the server. Online help is available. Unfortunately not all servers are synchronised, so you might have to search a few before

finding what you are looking for.

Recently a more user friendly ftp index server entered the net, the ftpsearch engine, featuring many options and with a rather stark interface, in the best Unix tradition.

Most of these offer help on efficient searching techniques, reading this can speed up your searches enormously. Investing a little time here will pay off in the long run.

If you have trouble getting onto the Internet but have mail then you should have a look at the *access via mail* FAQ. Naturally you can get it over e-mail using the US, Canada and South America server entering only this line in the BODY of the note:

```
/send usenet/news.answers/internet-services/access-via-email
```

or Europe, Asia etc server entering only this line in the BODY of the note:

```
send lis-iis e-access-inet.txt
```

or look through your news spool if you have one locally in news.answers newsgroup.

Finally, you might wish to get more information from a person, say an author of a software package. usually you can find the e-mail address in the accompanying documentation which normally would be in the documentation subdirectory but failing that and also if the given address is no longer valid you could find help in the FAQ for finding e-mail addresses.

---

6. Conclusion

Finding information fast and efficiently is more of an art than a science and we still have not touched on the really difficult part: how do you determine the actual *quality* of the information? It is outside the scope of this HOWTO to tell you that but it is still something you should keep in mind. You should at least check the information is recent enough to be current to your problem.

There are a number of FAQs available that deals with more serious research method topics and you can also see a comprehensive on-line version.

Quoted from someone's signature:

Be alert! The world needs more lerts.

Upgrading Your linux Distribution mini-HOWTO

Greg Louis, glouis@dynamicro.on.ca

v1.11, 6 June 1996

Hints and tips on upgrading from one linux distribution to another.

1. IMPORTANT!!! Disclaimer and Copyright

2. Changes since version 1.1

3. Introduction

- 3.1 How to slay and reincarnate your linux box!
- 3.2 Why would anyone want to do that?
- 3.3 Do you have to ``destroy and reinstall?''
- 3.4 How long will it take?

4. Write down everything you do.

5. Make a full backup of the existing system.

6. Back up /etc and its subdirectories on one or more floppies.

TOC

7. Make separate backups of each group of files you want to preserve.

8. Prepare root and boot floppies for the new installation.

9. Format floppies for the temporary kernel and the final build.

10. Inhibit logins and back up the /root and /home trees.

11. Boot from the new installation's boot and root floppies.

12. Delete the linux partitions with fdisk and recreate them.

13. Run the new linux installation.

14. With the new linux system booted from the hard disk, edit /etc/fstab

15. Restore configuration data to the /etc directory and subdirectories.

16. Configure and rebuild the linux kernel.

17. Restore the stuff from the backups you made earlier.

18. Review security.

19. Enable logins.

20. Sorry, but once again:

21. Acknowledgements

1. IMPORTANT!!! Disclaimer and Copyright

The procedure to which this document attempts to be a guide is inherently dangerous to the programs and data stored in your computer. You carry out any such procedure entirely at your own risk. The steps described in this document worked for the author; there is no guarantee that they will work for you, nor that you can attempt to follow them without serious damage to your computer's programs and/or data. You are entirely on your own in any use you may make of the information presented herein, and the author shall not be liable in any way whatsoever for any damage or inconvenience of any kind that you may suffer in so doing.

This document is copyright 1996, Dynamicro Consulting Limited, and is released under the terms of the GNU General Public License. This basically means that you may copy and modify it at will, but may not prevent others from doing likewise.

Comments and questions may be directed to the author. Especially welcome, for use in future revisions, are accounts of successful upgrades of complex systems.

2. Changes since version 1.1

- Added this history section
- Added Zoltán Hidvégi's suggestion re mtime and ctime. Thanks, Zoltán!
- Added an Acknowledgements section

3. Introduction

3.1 How to slay and reincarnate your linux box!

The purpose of this document is to offer tips to help you through the destruction and reinstallation of a linux system. It's not a foolproof cookbook by any means; but I hope it will serve as some indication of what you need to think about, and of the order in which to do things. It would have been a help to me, if someone else had written something like this before I did my first upgrade; so I hope it will be a help to you, if you have a linux machine to rebuild.

Don't take it as gospel, though: your mileage will almost certainly vary. Even the directory names in this document may be different from the ones you need to use; some people have /usr/home instead of /home, for example; others call it /u, and some (delicate shudder :) even put all their users directly under /usr itself! I can't be specific about your system, so I've just used the names the way they are in mine.

You'll also notice that I use Slackware distributions, and that I assume you've enough RAM and hard disk space to install linux kernel source and build your own kernel. If your system is different, some of my recommendations won't apply; but I hope you'll still find the general outline to be of assistance in your rebuild project.

3.2 Why would anyone want to do that?

Good question! If it can possibly be avoided, don't do it! (That's the single most important recommendation in this whole guide!!!) But there are times when you may have to.

For example, I installed a 4Gb hard disk and then found out that Slackware 2.0 vintage linux didn't know a hard disk could have more than 2Gb, and it got horribly confused. So I had to upgrade to the then-current Slackware 2.3. That upgrade was a gruelling experience, and it's part of the reason I'm writing these notes. I did just about everything wrong, and only good luck and the fact that I had another running linux box beside me saved me from disaster.

As another example, I found that I just couldn't succeed in building a working a.out linux kernel in the 1.3 series, using an out-of-the-box Slackware 2.3 installation (another machine, not the one I botched before). I took the plunge, bought Slackware 3.0 on CDROM and converted to ELF. This time the reinstallation went better, thanks in part to the previous bitter experience, and it served as the source of most of the ideas I'm offering you here.

3.3 Do you have to ``destroy and reinstall?''

It's safer, oddly enough. If you install over top of an existing linux system, chances are you'll have a mixture of old and new binaries, old and new configuration files, and generally a mess to try to

administer. Wiping the system clean, and then putting back only what you know you need, is a drastic but effective way to get a clean result. (Of course we're talking about installing a whole new linux distribution here, not about upgrading one or two packages! The best way to avoid having to do a full reinstallation is, precisely, to keep the individual bits -- especially gcc and its libraries, and binutils -- current. If the stuff you use is reasonably up-to-date, and you can keep it so by bringing in, and if need be compiling, new code from time to time, then there's no need for a mass upgrade.)

As Patrick Volkerding points out (he too recommends the wipe-it-clean procedure for upgrades), installing ELF on top of a running a.out system is a recipe for disaster; at least, if you know enough to try it, you needn't read this guide!

Even without that complication, though, you're better to build from scratch.

3.4 How long will it take?

Depends, of course, on how complex your system is. But I figure that, for the successful upgrade (the other one? -- don't ask! :) I spent about ten hours making backups, six hours rebuilding the system to the point where I could enable logins, and another half day or thereabouts restoring the less-crucial stuff. As time passes I keep discovering little things that still aren't exactly as I want them -- I fix these as they're encountered -- but in the main, twenty hours' work should suffice for a reasonably complex rebuilding job. Maybe less if you're reinstalling from hard disk (I used CDROM) or more if you need to install from floppies. Maybe less if you've got a fast Pentium, more if it's a 386. You get the idea.

So much for the introduction. Here's how to set about it, once you've decided it must be done. Arm yourself with fortitude and Jolt or whatever, and:

---

4. Write down everything you do.

It's extremely valuable to have a record of what you've done in the process of preparing for, and carrying out, the changeover. Especially important is a list of the backups you'll be making in preparation for the destruction of your existing system.

5. Make a full backup of the existing system.

Generally speaking, backups tend to be written on media that are sequentially accessed. That being so, you won't want to use this complete backup for restoring significant numbers of files; it's got too many files on it that you don't want. It's better to create small backups of individual segments that you know you're going to restore in their entirety. I'll list a bunch of examples later.

Why then should you start with a full backup? Two basic reasons: first, in the event of a catastrophic failure installing the new system, you'll have a way to get back to the starting point with minimum pain. Second, no matter how carefully you prepare for the new installation, there is a very large chance that one or two important files will be overlooked. In that case the clumsiness of restoring those one or two files from the full backup set will be preferable to the inconvenience of doing without them.

To save time and space, if you've still got the distribution medium for your old linux version, you might want to back up only those files the mtime or ctime of which is more recent than the date of the original installation.

6. Back up /etc and its subdirectories on one or more floppies.

This is the other extreme: you won't be restoring these files (for the most part, anyway); you'll be comparing them with the new ones that get created during installation. Why? Because the new ones may have data that the old ones didn't, or may express the old data in new ways. Changes in protocols, addition of new tools, or implementation of new features in existing tools may all dictate changes in the formats of the configuration files and startup scripts that the /etc tree contains, and you'll very likely have to edit your old data into these files so as to preserve the new formats and take advantage of the improvements.

7. Make separate backups of each group of files you want to preserve.

This is the most variable part of the job, and all I can really do to help is to describe what I did in my system, in the hope that it will serve as a rough guide. Basically, you want to look at every directory that contains any

- files that aren't part of your standard linux installation, or
- files that are actually newer than the ones you'll install when you do your new linux installation

and separate out only those files that you want to carry over.

(Another possible strategy is to back up all files with mtime or ctime more recent than the day of the previous linux installation, as mentioned above, and then restore from that. If you do that, you have to take into account that the new linux distribution may contain versions of some files that are newer still than the ones you saved.)

In my case, I ended up making a .tgz file on the backup medium for each of

- /usr/lib/rn
- /usr/lib/smail
- /usr/lib/trn (the rest of /usr/lib would be reinstalled)
- /usr/local/src
- /usr/local/bin
- /usr/local/lib
- /usr/local/lpfont
- /usr/local/man
- /usr/local/sbin
- /usr/local/thot (there were other /usr/local files I didn't need)
- /usr/openwin
- /usr/src/lilo-17 (because my new Slackware still had version 16)
- /usr/src/linux-1.2.13 (because I'd done some customizing)
- /usr/X11R6/lib/X11/app-defaults
- /usr/X11R6/lib/X11/initrc (the rest of Xfree86 was to be reinstalled)
- /var/named
- /var/openwin
- /var/texfonts

My machine was relatively easy in that there were no spool files to worry about. I don't run a news spool

on this box, and since there are only two users, it was easiest just to get all the mail read before shutt down. Otherwise, /var/spool directories would have had to be backed up at the last minute. (And, of course, the news library and site directories!)

8. Prepare root and boot floppies for the new installation.

Details of how to do this will be found in the installation guide for your new distribution.

9. Format floppies for the temporary kernel and the final build.

You'll need two, one floppy for each.

After all that's done, you're ready for the Big Moment. The next step removes the system from production.

10. Inhibit logins and back up the /root and /home trees.

This is the last thing to be done on the old system before you destroy it, so as to carry forward the most current user and root information.

11. Boot from the new installation's boot and root floppies.

12. Delete the linux partitions with fdisk and recreate them.

The installation guide will explain how to set about doing this, which will destroy the old system. From now on you're dependent on the quality of the backups you made in the earlier steps! You have been warned!

13. Run the new linux installation.

There are already several good documents describing how to do this, so I'm not going into any detail. Continue from here when the new system can boot from its hard disk.

Along the way, be sure to make a floppy that you can boot as well, since the kernel that the linux setup installs has to be replaced and accidents can happen during that process. Be sure to install the development packages and the kernel source.

14. With the new linux system booted from the hard disk, edit /etc/fstab

and add your swap partition. Then run the command "swapon -a". I don't know why, but Slackware setup doesn't offer to do this for you if your swap partition exists already. Then, when you boot your new system and the rc.S script tries to turn swapping on, it can't find the partition in the fstab file and swapping doesn't get enabled. This step fixes it.

15. Restore configuration data to the /etc directory and subdirectories.

As described above, you can't just copy all of the old files back into /etc and expect things to work properly afterward. Some files you can do that with; for example, /etc/XF86Config (as long as you're using the same version of Xfree86 -- and the same video hardware -- in the new installation as you did in the old). For the most part, though, it's best to use diff to compare the old and new files before doing any copying. Watch out especially for significant changes in the files in /etc/rc.d, which may require you to reestablish your old configuration by hand editing, instead of by copying the old rc scripts from your backup. Once it's all done, reboot.

16. Configure and rebuild the linux kernel.

Even if you don't absolutely have to do this in order to get a kernel that supports your hardware, it's worth doing it in order to get a kernel that doesn't contain masses of drivers for stuff your machine doesn't have. For details, see the Kernel HOWTO. Install the rebuilt kernel on a floppy at first; once that boots ok, install on the hard disk, run lilo if you're using it, and reboot.

17. Restore the stuff from the backups you made earlier.

Some of the binaries may need to be reinstalled from the source directories; I had to do that with lilo, for example, since my version was newer than the one on the Slackware installation and I hadn't bothered to save the binary from /sbin. You'll want to check through your restored programs and confirm the existence and correctness of configuration files, libraries and so on. In some cases, you may have to restore things in a specific order; you did make notes during backup, didn't you? ;-)

18. Review security.

Check file permissions and directory permissions to be sure that access is neither too restricted nor too easy. I find that Slackware tends to lean toward a more open environment than I like, so I go around changing 755's to 711's for binaries in the .../bin directories and stuff like that. Or even 700's in the .../sbin ones. Especial care is needed if you've carried over an ftp server; but then, if you were running an ftp server, you probably thought of that already. :)

19. Enable logins.

You're up and running. Over the next little while, there'll probably be details to clean up; but the bulk of the work is done. Enjoy!

20. Sorry, but once again:

USE THIS INFORMATION AT YOUR OWN RISK!

(See the disclaimer at the start of this document.)

21. Acknowledgements

Thanks for contributing to the content of this mini-HOWTO are gratefully tendered to Zoltán Hidvégi.

Linux on Sony VAIO mini-HOWTO

Hideki Saito hideki@chatlink.com

v1.1, 16 September 1998

This document explains installation of Linux on Sony VAIO computers.

1. Introduction

- 1.1 Notes
- 1.2 Notes about different systems
- 1.3 Update History
- 1.4 Disclaimer
- 1.5 What this document all about?

2. Hardware compatibilities issue

- 2.1 Hardware that not compatible with Linux
- 2.2 Partially compatible hardwares

3. How to install Linux?

4. After all...

5. Little Legal notes

6. Special thanks

1. Introduction

1.1 Notes

This document first appeared in "Hideki Saito's Homepage" at

```
http://ryoohki.anime.net/~hideki/
```

. This version of HOWTO is rewritten for updated information. HTML version is found in http://ryoohki.anime.net/ hideki/VAIO+Linux.html. You may distribute this document freely. When you publish it, please drop me a note; I just want know how this documents are used :)

1.2 Notes about different systems

I've recently received some mail stating that some of contents in this document is incorrect. It is because some reason, hardware installed on more recent systems are slightly different than older systems. I will start covering those information. But since of course, I don't have every system in here, I would like to ask Sony VAIO owners to tell me more information about it. This would help me a lot if you can put [VAIO Information] in the beginning of the subject, so I can take extra attention to it. Also, please include model number of the system. If it is notebook, please include that information in a content, or in a subject. I would like to apologize to ones that got confused by this document. (Addition: 12/29/98)

1.3 Update History

12/29/98 Added several updates

1.4 Disclaimer

Also most of procedures in this document is tested, I will take no responsibilities for any problem caused by using any information in this document. As like in GPL, please try anything in your own risk. I am not native English writer, and you will find many mistakes in this document. Please excuse me on those. Any corrections and clarifications are welcome; please send those to hideki@chatlink.com.

1.5 What this document all about?

In this HOWTO, I will go over how to use Linux under Sony VAIO computers. The VAIO computers are great, but was not very friendly to Linux. But this isn't mean that you cannot use it, actually you can! Throughout this document, I will assume that you own either PCV 70 or 90. Since almost all VAIO models are similar in their architecture (although newer ones have larger disks, updated video card, more

RAM, etc), many of contents in this document should apply to all VAIO models.

2. Hardware compatibilities issue

2.1 Hardware that not compatible with Linux

Most of hardwares installed in the system are compatible under Linux. However there are exceptions of modem, USB Port and MPEG decoder. Please be aware that you cannot use modem, which should be problem for some of you. Modems installed in the VAIO is called DSVD modem, which is the variant of the "Winmodem."

(Addition: 12/28/98) Newer model seems to have normal modem instead of Winmodem.

2.2 Partially compatible hardwares

The soundcard and the Video card installed are partially compatible.

Soundcard

Soundcard is compatible under Linux with supports of Microsoft Sound System + Sound Blaster, if you want to use MIDI under Linux, you need to compromise your sound input capabilities. For this case, cancel out the Sound Blaster. If you need MIDI, and sound input, you may want consider buying OSS/Linux, which you can use all of its capabilities under Linux. The soundcard equipped with the VAIO is Yamaha OPL3SA, and OSS/Linux has full support to it. It seems like that the wavetable synthesis mentioned in the specification is really the soft synthesizer, so you are not missing hardware capabilities on it. As the time I am writing this, there is the OSS/Free 3.8, which supports this soundcard under Yamaha OPL3-SA1 audio controller. Please note that this is not initially installed on 2.0.x. You will need to download from OSS site. 2.1.x comes with updated version of OSS/Free, but I haven't tested compatibility with it. If you see that it is working with this soundcard, please let me know.

Video Card

Videocard is partially supported under Linux. You will have no problem using it under Linux console mode. If you wish to use it under X Window, get 3.3.1 or later XFree86, which has support to ATI Rage. Other commercial X Window system should support this, too. However, 3D acceleration of this video card will not work under Linux, as there is no driver for this. (Update 12/29/98) For model number 505, ALSA drivers on XFree86 V3.3.3 or later is compatible.

3. How to install Linux?

Installing Linux on the system will not be big problem. However, you need to aware that initially, there are one non-DOS partition on the system. I have no idea what that partition is. It is 400MB partition, and it is visible by running fdisk. I simply deleted this partition to allocate it for Linux. And I don't have any problem, so probably it must be something to do with software installation. (Update 12/29/98) This 400MB partition is only on older models, it really supporsed to be another 400MB DOS partition, but Sony made mistake on it. Please refer to other instruction for general installation of Linux. The VAIO has nice support that you can boot directly from CD-ROM, if CD-ROM is bootable. Since the BIOS supports it, it is what you can use in some distribution. This is nice, because you can perform floppyless installation. Particularly, Redhat could boot from CD-ROM. Other distribution like SUSE also claims that it can boot from CD-ROM. But as I have not checked those yet, I cannot be sure on that. If that this way of booting does not work with yours, try checking boot option in the BIOS. Kernel configuration shouldn't be big problem except for the part to setup soundcard. But as I covered that issue earlier, I will not talk about it in this section. The VAIO computer is equipped with APM BIOS, or Advanced Power Management BIOS, I personally don't recommend that you enable support to it under Linux. I had problem when I enabled it. (Update 12/29/98) I have received report that with 505, CD-ROM install is difficult and network install is much easier.

4. After all...

Everything went well? I hope it did. Now your Sony VAIO computer is one of the best Linux workstation. If you have any problem, I may have answer to your questions. Please just send those to hideki@chatlink.com anytime. I will try to answer those as soon as possible.

5. Little Legal notes

VAIO is the trademark of Sony. All other trademark appeared in this documents are belonged to their respective holders. The copyright of this document is reserved by Hideki Saito.

6. Special thanks

I would like to thank to following people, not in any particular order: Daniel Nunes. If you feel that your name should be on this section, please let me know by mail. (recentlly I lost big portion of my mail spool by accident, so it maybe the case)

Vesafb mini-HOWTO

Alex Buell, alex.buell@tahallah.demon.co.uk

v0.5, 2 August 1998

This document describes how to use the vesafb device in Linux with a VESA 2.0 compliant graphic card on Intel platforms.

1. Contributors
2. What is vesafb?
3. What is a framebuffer device?
4. How do I activate the vesafb drivers?
5. What VESA modes are available to me?
6. Is there a X11 driver for vesafb?
7. Can I make vesafb as a module?
8. Miscellaneous/Caveats

1. Contributors

Thanks go to these people listed below who helped improve the vesafb HOWTO.

- Jeff Noxon `jeff@planetfall.com`
- Francis Devereux `f.devereux@cs.ucl.ac.uk`
- Andreas Ehliar `ehliar@futurniture.se`
- Martin McCarthy `marty@ehabitat.demon.co.uk`
- Simon Kenyon `simon@koala.ie`
- David Ford `david@kalifornia.com`
- Chris Black `cblack@cmpteam4.unil.ch`
- N Becker `nbecker@fred.net`
- Bob Tracy `rct@gherkin.sa.wlk.com`
- Marius Hjelle `marius.hjelle@roman.uib.no`
- Aaron Tiensivu `tiensivu@pilot.msu.edu`

and of course the authors of the framebuffer devices:

- Gerd Knorr `kraxel@cs.tu-berlin.de`
- Geert Uytterhoeven `Geert.Uytterhoeven@cs.kuleuven.ac.be`
- Martin Mares `mj@ucw.cz`
- Anyone else, stand up and be counted. :o)

2. What is vesafb?

Vesafb is a framebuffer driver for Intel architecture that works with VESA 2.0 compliant graphic cards. It is closely related to the framebuffer device drivers in the kernel.

vesafb is a display driver that enables the use of graphical modes on your Intel platform for bitmapped text consoles. It can also display a logo, which is probably the main reason why you'd want to use vesafb :o)

Unfortunately, you can not use vesafb successfully with VESA 1.2 cards. This is because these 1.2 cards do not use framebuffering. It may be that someone will write a vesafb12 device driver for these cards, but this will use up precious kernel memory. :o(

There is however a potential workaround to add VESA 2.0 extensions for your legacy VESA 1.2 card. You may be able to download a TSR type program that will run from DOS, and used in cojunction with loadlin, can help configure the card for the appropriate graphic console modes. Note that this will not always work, as an example some Cirrus Logic cards such as the VLB 54xx series are mapped to a range of memory addresses (for example, within the 15MB-16MB range) for frame buffering which preludes these from being used successfully with systems that have more than 32MB of memory. [There is a way to make this work, i.e. if you have a BIOS option to leave a memory hole at 15MB-16MB range, it might work, but I've been told that Linux doesn't support this] If you wish to experiment with this option, there are plenty of TSR style programs available, a prime example is UNIVBE, which can be found on the Internet.

3. What is a framebuffer device?

A framebuffer device is an abstraction for the graphic hardware. It represents the frame buffer of some video hardware, and allows application software to access the graphic hardware through a well-defined interface, so that the software doesn't need to know anything about the low-level interface stuff [Taken from Geert Uytterhoeven's framebuffer.txt in the linux kernel sources]

4. How do I activate the vesafb drivers?

[Note: The information is based on what I know and have achieved using the Linux kernel v2.1.112 on a P75+ with an ATI VideoExpress 2MB graphic card. Feel free to email me with information/special cases for different graphic cards]

Assuming you are using menuconfig, you will need to do the following steps:

Go into the Code Maturity Level menu, and enable the prompt for development and/or incomplete drivers [note this may change for future kernels - when this happens, this HOWTO will be revised]

Go into the Console Drivers menu, and enable the following:

- VGA Text Console
- Video Selection Support
- Support for frame buffer devices (experimental)
- VESA VGA Graphic console
- Advanced Low Level Drivers
- Select Mono, 2bpp, 4bpp, 8bpp, 16bpp, 24bpp and 32bpp packed
- pixel drivers
- VGA character/attributes support

VGA Chipset Support (text only) - vgafb - used to be part of the list above, but it has been removed as it is now deprecated and no longer supported. It will be removed shortly. Use VGA Text Console instead.

Ensure that the Mac variable bpp packed pixel support is not enabled. [As of 2.1.111, this seems to get enabled if Advanced Low Level Drivers is initally selected]

Make sure these aren't going to be modules. [Not sure if it's possible to build them as modules yet - please correct me on this]

Then rebuild the kernel, modify /etc/lilo.conf to include the VGA=ASK parameter, and run lilo, this is required in order for you to be able to select the modes you wish to use.

Reboot the kernel, and as a simple test, try entering 0301 at the VGA prompt (this will give you 640x480 @ 256), and you should be able to see a cute little Penguin logo.

Once you can see that's working well, you can explore the various VESA modes (see below) and decide on the one that you like the best, and hardwire that into the "VGA=x" parameter in lilo.conf. When you have chosen the one you like the best, look up the decimal equivalent from the tables below and use the corresponding decimal number (i.e. for 1280x1024 @ 256, you just use "VGA=775"), and re-run lilo. That's all there it is to it. For further references, read the LoadLin/LILO HOWTOs.

5. What VESA modes are available to me?

This really depends on the type of VESA 2.0 compliant graphic card that you have in your system, and the amount of video memory available. This is just a matter of testing which modes work best for your graphic card.

The following table shows the mode numbers you can input at the VGA prompt (actually these numbers are plus 0x200 to make it easier to refer to the table)

Colours	640x480	800x600	1024x768	1280x1024	1600x1200
256	0301	0303	0305	0307	031C
32,768	0310	0313	0316	0319	031D
65,536	0311	0314	0317	031A	031E
16.8M	0312	0315	0318	031B	031F

For convienence, here is the same table in decimal terms

Colours	640x480	800x600	1024x768	1280x1024	1600x1200
256	769	771	773	775	796
32,768	784	787	790	793	797
65,536	785	788	791	794	798
16.8M	786	789	792	795	799

[The author would be happy if you could supply him with additional 03xx numbers for additional modes i.e 1152x900?]

6. Is there a X11 driver for vesafb?

Yes, there is, actually. You will need to use the XF86_FBdev driver if for some reason your current X11 driver doesn't like vesafb. Go to http://www.xfree86.org, and download the X332servonly.tgz archive, unpack, configure it by editing xc/config/cf/xf86site.def, and uncomment the #define for XF68FBDevServer. Don't worry about the m68k reference, it supports Intel platforms. Then build the whole thing - it'll take a long time though as it's a large source tree.

There is as of yet, no pre-compiled XF86_FBdev binary available, but I understand that Debian may be working on this; and they will be able to produce both libc5/glibc2 variants.

There have been reports that X11 is non functional on certain graphic cards with this vesafb feature enabled, if this is happening, try the new XF86_FBdev driver for X11.

This driver, along with vesafb can also help run X11 in higher graphic resolutions with certain graphic chipsets which are not supported by any of the current X11 drivers. Examples are MGA G-200 et. al.

Hopefully the X11 problems with supported graphic cards will be fixed in future releases.

7. Can I make vesafb as a module?

As of v2.1.112, vesafb can't be modularised, although at some point in time, the developer of vesafb may decide to modify the sources for modularising. Note that even if modularising is possible, at boot time you will not be able to see any output on the display until vesafb is 'modprobed'. It's probably a lot wiser to leave it in the kernel, for these cases when there are booting problems.

8. Miscellaneous/Caveats

This is current only for v2.1.112 of the Linux kernel. This HOWTO will be constantly updated as kernel development progresses. Feel free to email the author with information/flames :o)

Another gotcha is that scrollback buffering does not work yet. May be fixed in future releases.

The VPN HOWTO

Arpad Magosanyi <mag@bunuel.tii.matav.hu>

v0.2,7 Aug1997

v0.2, 7 August 1997

1. Changes

2. Blurb

- 2.1 Copyright
- 2.2 Disclaimer
- 2.3 Disclaimer
- 2.4 Credits
- 2.5 State of this document
- 2.6 Related documentations

3. Introduction

- 3.1 Naming conventions

4. Doing it

- 4.1 Planning
- 4.2 Gathering the tools
- 4.3 Compile and install
- 4.4 Configure the other subsystems
- 4.5 Set up the accounts for the VPN
- 4.6 Generate an ssh key for your master account
- 4.7 Set up automatic ssh login for the slave account
- 4.8 Tighten ssh security on the bastions.

- 4.9 Enable execution of ppp and route for both accounts.
- 4.10 Do the scripting

5. Look at what's happening:

6. Doing it by hand.

- 6.1 Logging in
- 6.2 Firing up ppp
- 6.3 Together the two
- 6.4 Pty redirecting
- 6.5 Is anything on the device?
- 6.6 Setting up the routes

7. Tuning

- 7.1 Configuration tuning
- 7.2 Bandwith vs. cicles

8. Vulnerability analisis

---

1. Changes

The 'no controlling tty problem' -> -o 'BatchMode yes' by Zot O'Connor <zot@crl.com>

warning about kernel 2.0.30 by mag

2. Blurb

This is the Linux VPN howto, a collection of information on how to set up a Virtual Protected Network in Linux (and other unices in general).

2.1 Copyright

This document is part of the Linux HOWTO project. The copyright notice is the following: Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions. All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below. In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs. If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu via email.

2.2 Disclaimer

As usual: the author not responsible for any damage. For the correct wording, see the relevant part of the GNU GPL 0.1.1

2.3 Disclaimer

We are dealing with security: you are not safe if you haven't got good security policy, and other rather boring things.

2.4 Credits

Thanks to all of who has written the tools used.

Thanks to Zot O'Connor <zot@crl.com> for pointing out the "no controlling tty" problem, and it's solution.

TOC

2.5 State of this document

This is very preliminary. You should have thorough knowledge of administrating IP, at least some knowledge of firewalls, ppp and ssh. You should know them anyway if you want to set up a VPN. I just decided to write down my experiences not to forget them. There are possibly some security holes indeed. To be fair I've tried it on hosts configured as routers not firewalls, saying: It's simple from that point.

2.6 Related documentations

- The Linux Firewall-HOWTO /usr/doc/HOWTO/Firewall-HOWTO
- The Linux PPP-HOWTO /usr/doc/HOWTO/PPP-HOWTO.gz
- The ssh documentations /usr/doc/ssh/*
- The Linux Network Admins' Guide
- NIST Computer Security Special Publications http://csrc.ncsl.nist.gov/nistpubs/
- Firewall list (majordomo@greatcircle.com)

---

3. Introduction

As firewalls are in more and more widely use in internet and intranet security, the ability to do nice VPNs is important. Here are my experiences. Comments are welcome.

3.1 Naming conventions

I will use the terms "master firewall" and "slave firewall", though making a VPN has nothing to do with client-server architecture. I simply refer to them as the active and passive participants of the connection's setup. The host which is starts the setup will be referred as the master, and the passive participant will be the slave.

4. Doing it

4.1 Planning

Before you start to set up your system, you should know the networking details. I assume you have two firewalls protecting one intranet per firewall, and they are both connected to the internet. So now you should have two network interfaces (at least) per firewall. Take a sheet of paper, write down their IP addresses and network mask. You will need one more IP adresses per firewall for the VPN you want to do now. Those addresses should be outside of your existing subnets. I suggest using addresses from the "private" address ranges. They are the followings:

- 10.0.0.0 - 10.255.255.255
- 172.16.0.0 - 172.31.255.255
- 192.168.0.0 - 192.168.255.255

For the sake of example, here's a sample configuration: The two bastions are called fellini and polanski. They have one interface for the internet (-out), one for the intranet (-in), and one for the vpn (-vpn). The addresses and netmasks:

- fellini-out: 193.6.34.12 255.255.255.0
- fellini-in: 193.6.35.12 255.255.255.0
- fellini-vpn: 192.168.0.1 point-to-point
- polanski-out: 193.6.36.12 255.255.255.0
- polanski-in: 193.6.37.12 255.255.255.0
- polanski-vpn: 192.168.0.2 point-to-point

So we have the plan.

4.2 Gathering the tools

You will need a

- Linux firewall
- kernel
- very minimal configuration
- ipfwadm
- fwtk
- Tools for the VPN
- ssh
- pppd
- sudo
- pty-redir

Current versions:

- kernel: 2.0.29 Use a stable kernel, and it must be newer than 2.0.20, because the ping'o'death bug. At the time of writing 2.0.30 is the last "stable" kernel, but it has some bugs. If you want to have the fast and cool networking code introduced in it, try a prepatch. the 3rd is working for me nicely.

- base system: I prefer Debian. YMMV. You absolutely don't want to use any big packages, and you never ev tought of using sendmail, of course. You also definitely don't want to enable telnet, ftp, and the 'r' comman usual in case of any other unix hosts).
- ipfwadm: I've used 2.3.0
- fwtk: I've used 1.3
- ssh: >= 1.2.20. There are problems with the underlying protocol in the older versions.
- pppd: I've used 2.2.0f for the tests, but I'm not sure if is it secure, this is why I turned the setuid bit off, and used sudo to launch it.
- sudo: 1.5.2 the newest I am aware of
- pty-redir: It is written by me. Try ftp://ftp.vein.hu/ssa/contrib/mag/pty-redir-0.1.tar.gz. Its version number is 0.1 now. Tell me it there is any problem with it.

4.3 Compile and install

Compile or otherwise install the gathered tools. Look at every one's documentation (and the firewall-howto) for details. Now we have the tools.

4.4 Configure the other subsystems

Configure your firewall rules, etc. You need to enable ssh traffic between the two firewll hosts. It means a connection to port 22 on the slave from the master. Start sshd on the slave and verify if you can login. This step is untested, please tell me your results.

4.5 Set up the accounts for the VPN

Create an account on the slave firewall use your favourite tool (e.g. vi, mkdir, chown, chmod) you might create an account on the master also, but I think you want to set up the connection at boot time, so your ordinary root account will do. Can anyone point out risks on using the root account on the master?

4.6 Generate an ssh key for your master account

Use the ssh-keygen program. Set empty password for the private key if you want to do automatic setup of the VPN.

4.7 Set up automatic ssh login for the slave account

Copy the newly generated public key in the slave account under .ssh/authorized_keys, and set up file permissions like the following:

```
drwx------ 2 slave slave 1024 Apr 7 23:49 ./
drwx------ 4 slave slave 1024 Apr 24 14:05 ../
-rwx------ 1 slave slave 328 Apr 7 03:04 authorized_keys
-rw------- 1 slave slave 660 Apr 14 15:23 known_hosts
-rw------- 1 slave slave 512 Apr 21 10:03 random_seed
```

The first row being ~slave/.ssh, and the second is ~slave.

TOC

4.8 Tighten ssh security on the bastions.

It means the followings on my setup in sshd_conf:

```
PermitRootLogin no
IgnoreRhosts yes
StrictModes yes
QuietMode no
FascistLogging yes
KeepAlive yes
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
```

Password authentication is turned off, so login is only possible with authorized keys. (You've turned off telnet and the 'r' commands of course).

4.9 Enable execution of ppp and route for both accounts.

As the master account is the root in my case, it has nothing to do. For the slave account, the following lines appear in /etc/sudoers:

```
Cmnd_Alias VPN=/usr/sbin/pppd,/usr/local/vpn/route
slave ALL=NOPASSWD: VPN
```

As you can see, I am using some scripts to set up ppp and the routing tables on the slave host.

4.10 Do the scripting

On the master host there is a full-blown init script I am using:

```
#! /bin/sh
# skeleton      example file to build /etc/init.d/ scripts.
#               This file should be used to construct scripts for /etc/init.d.
#
#               Written by Miquel van Smoorenburg <miquels@cistron.nl>.
#               Modified for Debian GNU/Linux
#               by Ian Murdock <imurdock@gnu.ai.mit.edu>.
#
# Version:      @(#)skeleton  1.6  11-Nov-1996  miquels@cistron.nl
#

PATH=/usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/bin/X11/:
PPPAPP=/home/slave/ppp
ROUTEAPP=/home/slave/route
PPPD=/usr/sbin/pppd
NAME=VPN
REDIR=/usr/local/bin/pty-redir
```

```
SSH=/usr/bin/ssh
MYPPPIP=192.168.0.1
TARGETIP=192.168.0.2
TARGETNET=193.6.37.0
MYNET=193.6.35.0
SLAVEWALL=polanski-out
SLAVEACC=slave

test -f $PPPD || exit 0

set -e

case "$1" in
  start)
        echo setting up vpn
        $REDIR $SSH -o 'Batchmode yes' -t -l $SLAVEACC $SLAVEWALL sudo $PPPAPP
>/tmp/device
        TTYNAME=`cat /tmp/device`
echo tty is $TTYNAME
        sleep 10s
        if [ ! -z $TTYNAME ]
        then
        $PPPD $TTYNAME ${MYPPPIP}:${TARGETIP}
        else
                echo FAILED!
                logger "vpn setup failed"
        fi
        sleep 5s
        route add -net $TARGETNET gw $TARGETIP
        $SSH -o 'Batchmode yes' -l $SLAVEACC $SLAVEWALL sudo $ROUTEAPP
    ;;
  stop)
        ps -ax | grep "ssh -t -l $SLAVEACC " | grep -v grep | awk '{print $1}' |
xargs kill
    ;;
  *)
    # echo "Usage: /etc/init.d/$NAME {start|stop|reload}"
    echo "Usage: /etc/init.d/$NAME {start|stop}"
    exit 1
    ;;
esac

exit 0
```

The slave uses one script for routing setup (/usr/local/vpn/route):

```
#!/bin/bash
/sbin/route add -net 193.6.35.0 gw 192.168.0.1
```

and its .ppprc consists of the following:

passive

5. Look at what's happening:

The master logs in into the slave, starts pppd, and redirects this all thing into a local pty. It consists of the following steps:

- allocating a new pty
- sshing into the slave
- running pppd on the slave
- the master runs pppd in this local pty
- and sets up the routing table on the client.

There are (not very tight) timing considerations involved, this is why that 'sleep 10s'.

6. Doing it by hand.

6.1 Logging in

You've already tried if ssh works well, aren't you? If the slave refuses to log you in, read the logs. Perhaps there are problems with file permissions or the sshd setup.

6.2 Firing up ppp

Log in into slave, and issue:

```
sudo /usr/sbin/pppd passive
```

You should see garbage coming at this point. If it works good, if not, there is some problem either with sudo, either with pppd. Look what the commands had said, and at the logs and at the */etc/ppp/options*, and the *.ppprc* file. If it works, write this 'passive' word into .ppprc, and try again. To get rid off the garbage and continue working, press enter,'~' and '^Z'. You should have the master's prompt now, and kill %1. See the section about tuning if you want to know more of the escape character.

6.3 Together the two

Well, then

```
ssh -l slave polanski sudo /usr/sbin/pppd
```

should work also, and deliver the garbage right into your face.

6.4 Pty redirecting

Try to redirect this whole thing this time:

```
/usr/local/bin/pty-redir /usr/bin/ssh -l slave polanski sudo /usr/sbin/pppd
```

Nice long sentence isn't it? You should use the full path into the ssh executable, as the pty-redir program allows only this form for security reasons. Now you've got a device name from the program. Let's say, you've got */dev/ttyp0* You can use the ps command to look what has happened. Look for 'p0'

6.5 Is anything on the device?

Try

```
/usr/sbin/pppd /dev/ttyp0 local 192.168.0.1:192.168.0.2
```

to establish the connection. Look at the output of the ifconfig command to see if the device has established, and use ping to check your virtual net.

6.6 Setting up the routes

Set up the routes on the master host, and on the slave also. Now you should be able to ping one host in one intranet from other host in the other intranet. Set up the additional firewalling rules. Now as you have the VPN, you can set up the rules concerning the connectivity of the two intranets.

---

7. Tuning

7.1 Configuration tuning

As I said this HOWTO is mainly a quick memo on how I had set up a VPN. There are things in the configuration I didn't experiment yet. These things will go into their place when I try them, or anyone tells me "it works in the following way" The most important thing is that the connection ppp uses is not 8-bit yet. I believe it has something to do either with ssh configuration or the pty setup. In this configuration ssh uses the tilde (~) character as an escape character. It might stop or slow down the communication, as any newline-tilde sequence causes ssh to give a prompt. Ssh documentation said: <On most systems, setting the escape character to ``none'' will also make the session transparent even if a tty is used.> The corresponding flag to ssh is '-*e*', and you can also set it in the configuration file.

7.2 Bandwith vs. cicles

Creating anything virtual comes with utilization of real-world resources. A VPN eats up bandwidth and computing resources. The goal would be to get balance between the two. You can tune it with the '-C' switch or the 'CompressionLevel' option. You might try using another cipher, but I don't recommend it. Also note that the round-trip-time can be longer if you use better compression. Any experiments on it are welcome.

8. Vulnerability analisis

I try to cover here the vulnerability issues arising from this particular setup and VPNs in general. Any comments are warmly welcome.

- sudo: Well, I'm excessively using sudo. I believe it's still safer than using setuid bits. It's still a backdraw of Linux that it hasn't got more fine-grained access control. Waiting for POSIX.6 compatibility <http://www.xarius.demon.co.uk/software/posix6/>. What is worse, there are shell scripts which are getting called through sudo. Bad enough. Any idea out there?
- pppd: It runs suid root also. It can be configured by user's .ppprc. There might be some nice buffer overruns in it. The bottom line: secure your slave account as tightly as you can.
- ssh: Beware that ssh older than 1.2.20 has security holes. What is worse, we made a configuration such when the master account had been compromised, the slave account is also compromised, and wide open to attacks using the two sudoed programs. It is because I've choosen not to have password on the master's secret key to enable automatic setup of the VPN.
- firewall: With inproperly set firewall rules on one bastion, you open both of the intranets. I recommend using IP masquerading (as setting up incorrect routes is a bit less trivial), and doing hard control on the VPN interfaces.

Visible bell mini-Howto

Alessandro Rubini, `rubini@linux.it`

v2.2, 11 November 1997

This document explains how to use termcap to configure a visual bell on one's system and describes how to disable audible bells on demand.

1. Introduction
2. Spekearectomy
3. Per-console Beep Configuration
4. Basic Concepts About Termcap and Terminfo
5. Defining a Visible Bell
6. Disabling the Audible Bell on the Text Console
7. Telling Applications to Avoid Beeping
8. The Dark Side of the Problem

1. Introduction

The Linux console driver beeps the audible bell whenever a BEL char is output (ASCII code 7). Though this is a right choice for the default behaviour, many users don't like their computer to beep. This mini-Howto is meant to explain how to tell applications not to output the BEL code. It also explain how to instruct the kernel and the X Window System to avoid beeping when a BEL is output. Note that most of this document refers to the text console, as configuring the X server is an easy catch-all for any user who works in a graphic environment.

In my opinion the best way to face a fussy computer is fixing the hardware, and my own computer doesn't even carry a loudspeaker.

2. Spekearectomy

Speakerectomy is by far the most brilliant solution to the audible bell problem. As its name implies, it consists in removing the beeps by removing the beeper. The operation is straightforward and you don't even need any anesthetic, but if you want there's room for refinement.

PC's are usually equipped with a silly switch to lower CPU clock. The switch is never used when you work in a multitasking environment, as you don't even need to slow the computer down to run games based on software loops. Unfortunately we can't use the switch to increase processor speed, but we can use it to enable/disable the loudspeaker. Sometimes the speaker is useful even if you enjoy a silent number cruncher, for example to signal the end of a lenghty compilation. To modify the switch functionality, just detatch it from the main board and connect its wires in series with the loudspeaker.

Owners of laptop boxes, unfortunately, don't have easy access to the loudspeaker, and neither they have a spare switch to turn to a different task. The preferred solution for such users is configuring their software to avoid beeping, as described below.

3. Per-console Beep Configuration

As of Linux 1.3.43, Martin Mares added the ability to configure the pitch and duration of the beep, by modifying `console.c`. Each console can be configured to feature a different duration and/or pitch of the bell sound; the task is accomplished by using escape sequences to the console device. You can configure your own `~/.profile` or `~/.login` file to select a different beep sound associated to each console (or no beep at all, if needed).

The escape sequences work as follow:

- ESC-`[10;`xx`]` selects the bell frequency in Hertz. The value should be in the range 21-32766, otherwise the result is undefined. If the `xx' argument is missing, the default value (750Hz) will apply, as in `ESC-`[10]`.
- ESC-`[11;`xx`]` selects the bell duration, in milli-seconds. If you specify more than 2 seconds, the default applies (125ms). Once again, if the `xx' argument is missing (ESC-`[11]`) the default value will be used.

To select, for example, a 50Hz pitch for one-second duration, you can "`echo -e "\\33[10;50]\\33[11;1000]"`" with `bash` (where "-e" means `understand Escape sequences'. If you use `tcsh` the same command spells "`echo "\\033[10;50]\\033[11;1000]"`".

Although I don't know of any version of the `setterm` command that supports such configuration, a future version of the command might well support a command-line option to configure the bell sound.

If you run Linux-1.3.43 or newer, you may be satisfied with the escape sequences and avoid reading further. If you run an older kernel, or if you want the visual bell, you'll enjoy the rest of this document.

4. Basic Concepts About Termcap and Terminfo

The file `/etc/termcap` is a text file that lists the `term`inal `cap`abilities. Several applications use the `termcap` information to move the cursor around the screen and do other screen-oriented tasks. `tcsh`, `bash`, `vi` and all the `curses`-based applications use the `termcap` database.

The database describes several terminal types. The `TERM` environment variable selects the right behaviour at run-time, by naming a `termcap` entry to be used by applications.

Within the database, each capability of the terminal appears as a two-letter code and a representation of the actual escape sequence used to get the desired effect. The separator character between different capabilities is the colon (":"). As an example, the audible bell, with code "bl", usually appears as "`bl=^G`". This sequence tells that the bell sound is obtained by printing the control-G character, the ASCII BEL.

In addition to the `bl` capability, the `vb` capability is recognized. It is used to represent the "visible bell". `vb` is usually missing in the `linux` entry of the `termcap` file.

Most modern applications and libraries use the `terminfo` database instead of `termcap`. This database uses one file per terminal-type and lives in `/usr/lib/terminfo`; to avoid using huge directories, the description of each terminal type is stored in a directory named after its first letter; the `linux` entry, therefore, is `/usr/lib/terminfo/l/linux`. To build a `terminfo` entry you'll ``compile'' the `termcap` description; refer to the `tic` program and its manual page.

5. Defining a Visible Bell

You can add the entry for the `vb` capability in your own termcap file, if it doesn't already define one. Dennis Henriksen (duke@diku.dk) suggested to insert the following line in the `termcap` entry for `linux` (note that the entry is called `console` in old distributions):

```
:vb=\E7\E[?5h\E[?5l\E[?5h\E[?5l\E[?5h\E[?5l\E[?5h\E[?5l\E8:\
```

The trailing backslash is used to escape the newline in the database. Dennis' code does the following (his own words):

- Save the cursor position (uust a safety precaution).
- Change the background color several times between normal and reverse.
- Restore the cursor position.

6. Disabling the Audible Bell on the Text Console

If you want to force the visible bell on your console you can use the "`bl`" entry in `termcap` and define it with the same string suggested for "`vb`" above. This approach is handy if you don't want to customize each application (which is described below, anyway). I use this option on all the machines where I can run Linux and I can't detach the speaker.

7. Telling Applications to Avoid Beeping

This is an incomplete list of applications that can be instrued to use the `vb` entry for the current terminal type (using either the `termcap` information or the `terminfo` one):

- The X server: use the "`xset b`" command to select the bell's behaviour. The command takes three numeric arguments: volume, pitch and duration. "`xset -b`" disables the bell altogether. Configuring the X server affects all the applications running on the display.
- `xterm`: xterm can convert each bell to either a visible or audible signal. If you use the audible bell, the settings of "`xset`" will apply. The bell in `xterm` defualts to be audible, but you can use the "`-vb`" command line option and the "`xterm*visualBell: true`" resource to turn it to a visible flash. You can toggle visible/audible signaling at run-time by using the menu invoked by control--left-mouse-button. If you run X you most likely won't need the following information.
- `tcsh` (6.04 and later): "`set visiblebell`". The instruction can be placed in `.cshrc` or can be issued interactively. To reset the audible bell just "`unset visiblebell`". To disable any notification issue use "`set nobeep`" instead.
- `bash` (any bash, as fas as I know): put "`set bell-style visible`" in your `~/.bashrc`. Possible bell-style's are also "none" or "audible".
- `bash` (with `readline`, as well as other `readline` based applications): put "`set prefer-visible-bell`" in `~/.inputrc`.
- `nvi` and `elvis`: put "`set flash`" in `~/.exrc` or tell "`:set flash`" interactively (note the colon). To disable the visible bell use `noflash` in place of `flash`.
- `emacs`: put "`(setq visible-bell t)`" in your `~/.emacs`. It is disabled by "`(setq visible-bell nil)`".
- `less`: use "`-q`" on command line to use the visual bell, use "`-Q`" to disable any reporting. Default options can be put in your environment variable "`LESS`".
- `screen`: issue the CtrlA-CtrlG command. It changes the behaviour of all the virtual screens. Refer to the man page under "CUSTOMIZATION" for setting the default.

8. The Dark Side of the Problem

The bad news is that not every application uses `termcap` or `terminfo`. Most small programs feature 'backslash-a' (alarm) characters in the C source code. The "alarm" code becomes a literal ASCII BEL in the strings as stored in the executable binary. Real application don't usually fall in this category, but be careful of C newcomers who give you their own programs. Students of computer science are the worst of all, granted.

The only way to make these programs silent applications is spekearectomy, or using the escape sequences by Martin Mares.

Linux Modem sharing mini-HOWTO

TOC

Author: Friedemann Baitinger, baiti@toplink.net
last update:

```
V1.01, 11/02/97 source now in HTML format, added a reference on where
                to get 'cu', also added a reference on where to get
                mserver. '-s' parameter added to 'cu' example.

V1.00, 06/12/97 Initial release
```

---

1. Introduction

This mini-HOWTO describes how to setup a Linux system in order to share a modem attached to this system with other systems over a TCP/IP network.

2. The Server Side

It is assumed that the server is a Linux system with either:

a) a modem attached to a `/dev/ttyS`*x* device

b) an 'isdn4linux'-emulated modem mapped to a `/dev/ttyI`*x* device

The easiest setup I can think of uses a five lines perl script to implement a 'modem demon':

```
$ cat /usr/sbin/modemd

#!/usr/bin/perl
select((select(STDOUT), $| = 1)[$[]);
select((select(STDIN), $| = 1)[$[]);
exec 'cu -s 115200 -l /dev/ttyS1';
die '$0: Cant exec cu: $!\n';
```

The modem demon is started by the `INETD` process if a client connects to the appropriate port as described below. The 'modemd simply connects the socket handle with `STDIN` and `STDOUT` of the `'cu'` command and lets `'cu'` handle the actual modem device. In case you don't have `'cu'` on your system, please install the `'UUCP'` package, `'cu'` is usually part of `'UUCP'`.

The existence of the modem demon must be made known to the `INETD` process by updating its configuration file, usually `/etc/inetd.conf` like:

```
#
# modem daemon
#
```

```
    modem stream tcp nowait root /usr/sbin/tcpd /usr/sbin/modemd
/dev/ttyS1
```

In order to make this work, an entry to `'/etc/services'` needs to be added like:

```
    modem           2006/tcp        modemd
```

This associates a symbolic name with an explicit port, `2006` in the example. The portnumber could be any number not already assigned to an existing service. After these changes have been made, a signal must be sent to the inetd process in order to let inetd re-read and process its configuration file:

```
    $ ps |grep inetd
    194  ?  S     0:00 /usr/sbin/inetd

    kill -HUP 194
```

Now the server side is ready to accept requests from clients. The correct function can be verified by:

```
    $ telnet localhost modem

    Trying 127.0.0.1...
    Connected to localhost.
    Escape character is '^]'.
```

You are now connected to the modem. You can now issue `'AT'` command in order to verify the setup:

```
    atz
    atz
    OK

    ati1
    ati1
    Linux ISDN
    OK

    ^]
    telnet>quit
    $
```

Instead of using the Perl script as a modem server, there is also a program named `'mserver'` available on ftp://ftp.innet.be/pub/staff/carl/ . In case it is not there anymore, you may want to use `'archie'` or any other search engine in order to locate the mserver archive. I haven't had the opportunity yet to install and use mserver.

TOC

3. The Client Side

At this time, only Windows client setups are described here. On the client PC, a COM-port redirector for TCP/IP is required. The best program for this purpose I have found is '`DialOut/IP`' from 'Tactical Software' for Windows 3.1 and Windows 95. (The Windows 3.1 version can be used under Windows NT for 16-bit applications only. A 32-bit version for Windows NT is due late summer 1997.)

`DialOut/IP` presents the shared modem on a new virtual COM port that it adds to Windows. This virtual COM port can be used by Windows programs as if the shared modem is directly connected. Most client applications (including Windows 95 dial-up networking) accept this and work as if there were a real COM port and modem, with the general exception being fax applications or any others that need access to UART control lines. `DialOut/IP` can be configured to provide Telnet protocol processing, but that feature applies to certain modem pool products and not to the Linux setup described in this file. Note that, despite its name, `DialOut/IP` can be used also by applications that wait for incoming calls.

On www.tactical-sw.com there is a page for downloading a fully functional evaluation version that times out in 1-2 weeks. Installation and configuration is handled by a setup program, with installation details in the README.TXT file. When you run `DialOut/IP`, you enter the IP address and port number of the shared modem.

DialOut/IP is a commercial product that is licensed on a per-modem basis, that is, the price depends on the number of modems that you are sharing. The license states that you can install the software on any number of PC's that access the shared modems.

4. Security Considerations

If you have only one modem for all your hosts in your local area network, there is probably no reason to worry about security here. However, if any one or more of the hosts in your LAN are connected to the internet by other means than using the modem we have just setup as a modem server, then security considerations are required, otherwise anybody can do a 'telnet your_host modem' and dial out long distance or even international calls at his will.

I suggest to install and configure tcp-wrappers in order to protect the modem sevrer against unauthorized access.

5. Examples

I am using the setup as described in (2) and (3) to run Quicken on my Windows 95 ThinkPad and do homebanking with the modem attached to my Linux machine. The 'modem' in my case is not even a real modem, it is an emulated modem on an ISDN-So card. Quicken just sees a COM port, it doesn't know that the device attached to the COM port is actually at the other end of my Ethernet LAN, nor does it know that it is not a standard analog modem but an ISDN device which happens to understand 'AT' commands.

WordPerfect Mini-howto

Wade Hampton, whampton@staffnet.com

v2.0, 13 August 1997

This document discusses running WordPerfect on Linux, including a brief discussion on the WordPerfect 7.0 beta. This is an update to the previous mini-howto that described notes on setting up WordPerfect 6.0 for SCO UNIX (ODT) on Linux. This document is not endorsed nor sponsored by Corel Corporation or SDC.

1. Introduction

- 1.1 Overview
- 1.2 History
- 1.3 Future Revisions
- 1.4 Feedback
- 1.5 Copyright
- 1.6 Standard Disclaimer

2. What is WordPerfect

3. WordPerfect 7.0 Beta

- 3.1 Impressions of 7.0
- 3.2 Getting WordPerfect 7.0
- 3.3 Installation of 7.0
- 3.4 Features
- 3.5 Bugs in 7.0
- 3.6 Pricing and Availability

TOC

4. WordPerfect 6.0 (Caldera)

- 4.1 Impressions of 6.0
- 4.2 Getting WordPerfect 6.0
- 4.3 Installation of 6.0
- 4.4 Bugs in 6.0

5. WordPerfect 5.1 and 6.0 (SCO release)

- 5.1 Impressions of WP/SCO
- 5.2 Getting WordPerfect WP/SCO
- 5.3 Installation of WP/SCO
- 5.4 Bugs in WP/SCO

6. The Future

7. For More Information

---

1. Introduction

1.1 Overview

The purpose of this document is to discuss WordPerfect on Linux. Discussed are running WP for SCO, WP 6.0 from Caldera, and a discussion on the WordPerfect 7.0 beta.

This document is an update to the previous mini-howto that described notes on setting up WordPerfect 6.0 for SCO UNIX (ODT) on Linux.

This document is not endorsed nor sponsored by Corel Corporation or Software Development Corporation.

1.2 History

- Version 1.0 -- Description on how to use WordPerfect 5.1 and 6.0 for SCO ODT 3.0 on Linux
- Version 2.0 -- Update and inclusion of information on Caldera's WordPerfect 6.0 distribution and the WordPerfect 7.0 beta.

1.3 Future Revisions

New versions of this document will be periodically posted to comp.os.linux.announce, comp.answers, and news.answers. They will also be added to the various anonymous ftp sites who archive such information including sunsite.unc.edu:/pub/Linux/docs/HOWTO.

In addition, you should be generally able to find this document on the Linux WorldWideWeb home page at http://sunsite.unc.edu/mdw/linux.html.

1.4 Feedback

I welcome any feedback, positive or negative, regarding the content of this document via e-mail. Definitely contact me if you find errors or obvious omissions.

I read, but do not necessarily respond to, all e-mail I receive. Requests for enhancements will be considered and acted upon based on my availability.

Flames will quietly go to /dev/null so don't bother.

TOC

1.5 Copyright

The WordPerfect mini-HOWTO is copyrighted (c)1995-1997 Wade Hampton.

A verbatim copy may be reproduced or distributed in any medium physical or electronic without permission of the author. Translations are similarly permitted without express permission if it includes a notice on who translated it.

Short quotes may be used without prior consent by the author. Derivative work and partial distributions of the WordPerfect mini-HOWTO must be accompanied with either a verbatim copy of this file or a pointer to the verbatim copy.

Commercial redistribution is allowed and encouraged; however, the author would appreciate being notified of any such distributions (as a courtesy).

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents.

We further want that ALL information provided in the HOWTOS is disseminated. If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at linux-howto@sunsite.unc.edu.

1.6 Standard Disclaimer

Of course, I disavow any potential liability for the contents of this document. Use of the concepts, examples, and/or other content of this document is entirely at your own risk.

This document is not endorsed nor sponsored by Corel Corporation or Software Development Corporation.

---

2. What is WordPerfect

WordPerfect is a popular, fully-featured commercial word processor program that is available for multiple platforms including Linux. It was originally written and sold by WordPerfect Corporation, subsequently purchased by Novell, then purchased about a year or so ago by Corel Corporation (Ottawa, Ontario, CA). Corel has outsourced the development of WordPerfect for UNIX (including Linux) to SDC Corporation, a corporation including former WordPerfect developers, and specializing in development for UNIX and X windows.

Until the release of Microsoft Word, WordPerfect was the most popular word processor for personal computers (it still claims to be by volume of product shipped, some reports indicate that more copies were sold than Microsoft Word during recent months). WordPerfect was originally written for DOS and later ported to the UNIX and Windows environments. Initial versions of WordPerfect for Windows were very late to market and tended to crash a lot, hence opening the floodGATES for Microsoft Word. WordPerfect is still a better word processor than Word for large documents or documents that must be moved between Mac, UNIX, and Microsoft platforms.

WordPerfect provides a WYSWYG environment for editing text documents. One of the best features is the ability to check your spelling as you type. Included are grammar checker, spelling checker, graphics import and export, and the ability to edit HTML documents. Documents created by WordPerfect on one platform may be read, edited, and printed on other platforms (I have generated a lot of documents using WP 6.0 (Caldera's release) and distributed them to Windows 95 users using WP 7.0 or 8.0).

WordPerfect runs on the following platforms (release or beta versions):

- Linux 1.2.13 (WordPerfect 6.0 for Linux, sold by Caldera)
- Linux 2.0.X (WordPerfect 6.0 and WordPerfect 7.0)
- Microsoft DOS, Windows 3.1, Windows 95, Windows NT
- Macintosh
- HP-UX 10.01
- IBM AIX 4.1.4
- SCO OpenServer 5.0.0
- Sun SPARC Solaris 2.5
- SUN SPARC SunOS 4.1.x
- SUN X86 Solaris 2.5.1
- Digital UNIX 3.2 G Alpha
- SCO 3.2.4.2

3. WordPerfect 7.0 Beta

3.1 Impressions of 7.0

I have been very impressed with the operation of WordPerfect 7.0 for Linux. I only recently downloaded it and tried it. It looks like WP 7.0 for Windows. The features appear to work properly. I especially like the spell-as-you-go feature!

The HTML publisher works a little better than that on WP 6.0, but still has some bugs. Someone reported that WP 7.0 requires a lot of swap space (I have not tested it at this time).

3.2 Getting WordPerfect 7.0

WP 7.0 is available as several LARGE tarballs from sdcorp's FTP site (or via the WWW). I tried for two days to download the 26MB archive using ISDN but only was able to get part of the file. I eventually had to use a commercial account with a T1 to the Internet to get the files. It would be much easier to download if each tarball was split into several smaller files (1-2 MB in size). Note, when you uncompress the tarballs, the result is a small number of medium sized files -- just what is needed.

The current version that you can download is time limited to 15 days. During this time, you can pay them and they will give you a license for it.

```
For WP 7.0 Beta:

     ftp://ftp.sdcorp.com/pub/linux_betab/wp7linuxbetab.tar.gz

           4887525 graphics_tar.gz
          17652138 manual_tar.gz
            450535 wp60ms01_us.all
            290443 wp60ps01_us.all
          26962514 wp7linuxbetab_tar.gz
```

3.3 Installation of 7.0

To install, you must extract the tarballs, run the **Runme** program, answer some questions and then run xwp to start WP 7.0. Note, you need 58 MB just for WP 7.0, prior to the manuals, graphics, etc.

1. Extract the tarball into a temporary directory, need 31M

```
tar xvzf wp7linuxbetab_tar.gz
  creates LINUX* files and Runme, the install program
```

2. As root and with X windows running, run the **Runme** program and you get a series of screens to select the installation type. Answer the questions:

```
./Runme

License Screen:
        Hit OK
```

TOC

```
Demo Screen:
        Hit OK
          Corel Corporation:  800-772-6735
            UNIX Support:     801-765-4019

Installation Directory:
        Enter directory, e.g., /opt/wp7.0
        Hit OK

Installation Size:
        Full:      71-80M
        Medium:    45-51M
        Minimal:   37-41M

Existing Application:
        If you have WP 6.0 (Caldera installs in
        /opt/wp6.0 by default), you can keep your
        settings for terminal drivers, printer drivers
        and printer ports.  Recommended.

        Enter directory (if present)
        Hit OK

Edit /etc/magic?
        Will add WP files to /etc/magic.  Will backup
          /etc/magic as /etc/magic.bak
        Select YES or NO
        Hit OK

Selection of languages to Install
        Only have English, it is selected
        Hit OK

Select print divers:
        Select your printers, may select more than one.
        Hit OK

Assign Printer Drivers to Destinations:
        Click on a printer
        Click on the destination (name from /etc/printcap)
        Hit OK
        Repeat for each printer you selected in previous step.

        Hit OK

Installation Review List:
        Allows you to go back to any screen and make changes.
        Select screen to make changes, or Hit OK to continue
        Hit OK

Graphicsl Corel WordPerfect(UNIX) installation screen
        includes progress bar
```

```
Readme Review screen:
        Allows you to read any review notes....

        Select item and View to view it, e.g., license

        Note if you entered a license, you need to add the
        license manager startup stuff to /etc/rc.d/rc.local!

        Select Done when finished
```

3. Run xwp:

```
/opt/wp7.0/wpbin/xwp& (or wherever you installed it)
```

   Note, I have WP 6.0 from Caldera, the first time I started, I got a prompt to load defaults from 6.0. I selected YES and WP loaded the defaults and exited. I then restarted xwp and my defaults, including the last files I edited with WP 6.0 were used!

3.4 Features

- WordPerfect word processor
- Internet publisher
- QuickCorrect - fix your typos while typing
- Spell-As-You-Go - spell checking while typing
- File compatibility with DOS, UNIX, and other WordPerfect versions on other platforms

3.5 Bugs in 7.0

- HTML export of a WP document with headers and footers sometimes leaves the bottom of the document either centered or bold.
- HTML export of a WP document with a table of contents does not generate cross references.
- HTML export messes up the table of contents and if sections were outlined, they become HTML list items and no longer match the TOC.
- Installation of the manual failed. I have not had time to try to fix this problem.
- Someone reported a problem converting WPG graphics to JPG.
- WP 7.0 **can not** read files from Office97 (this has been a major pain for me as I have received several such files from others and I do not have Office97).
- Someone reported a possible problem with getting printing to work (I had no problems at all).
- Upon starting xwp, I received a message that some backup files existed. Upon clicking on the first backup file, XWP dumped core.

3.6 Pricing and Availability

Upgrade pricing expected to be $149.99 (US) and retail pricing $199.99.

---

4. WordPerfect 6.0 (Caldera)

Caldera sells both a WordPerfect and Motif Bundle ($130 US) and the Caldera Internet Office Suite ($219 US). Both include WordPerfect 6.0 for Linux.

Overall, WP 6.0 has been good to me. I have generated 20-30 large documents and numerous small ones, probably 1000+ pages if I printed them all out. The only problem that I have is that I use WP Times font under Linux and it is not compatible with the WP True Type Times font under Windows.

4.1 Impressions of 6.0

The author has been using WordPerfect for Linux on Caldera Network Desktop (CND) for the past year. The system is a P6/200, 64M RAM, SCSI, CND 1.0 with RiskyHack patches to the Linux 2.0.30 kernel, updated RPM, and many, many RPMS from Redhat and Caldera's WWW sites.

4.2 Getting WordPerfect 6.0

Visit Caldera's WWW site at http://www.caldera.com

4.3 Installation of 6.0

Installation was easy. I just followed the directions provided....

I did have to add **wp** to my fvwm menu (I am using FVWM95 on CND 1.0 with lots of patches) and to looking glass's menus.

My platform is a P6/200, 64M RAM, Caldera CND 1.0, RiskyHacks to libc 5.4.4, kernel 2.0.30 (Joliet and FAT32 patches), dual SCSI, CD-R, etc. WP uses 74 MB of disk space.

4.4 Bugs in 6.0

- Export to HTML has several bugs with complex documents requiring some reformatting.
- Occasional crashes (very rare).
- Occasionally WP loops where it uses all memory and runs out of swap space, then crashes.
- Some fonts are different between Windows and UNIX versions.
- It has been reported to me that some WP macros for UNIX versions are different than those for Windows versions.

5. WordPerfect 5.1 and 6.0 (SCO release)

I started using WP 5.1 for SCO (which was statically linked) since January, 1995. In the Spring, 1995, I obtained a copy of WP 6.0 for SCO (dynamic linked) and was able to install it with some pain. I subsequently have had some problems with the license manager. I have it running under 1.2.8.

This section is mainly from the original WordPerfect mini-howto. It has not been used in a long time. Note, the more recent versions of WP for SCO included both the dynamic and static linked versions, and hence did NOT require SCO's shared libraries.

This is mainly for background information! If you really want WP for Linux, get the new WP 7.0!

5.1 Impressions of WP/SCO

WP for SCO seemed to work fine (once the installation problems were overcome). I used this distribution for about 6-12 months until I received a copy of WP 6.0 for Linux (Caldera) and CND 1.0.

5.2 Getting WordPerfect WP/SCO

Contact SCO or visit their WWW page at http://www.sco.com

5.3 Installation of WP/SCO

Requirements

- Linux 1.1.x to 1.2.x
- modules (dynamic module loading software)
- iBCS2 (Intel BCS2 and ELF loader module)
- SCO dynamic libs (if required)
- Domainname should be setup
- Hostname should be setup and be less than 8 characters (localhost is 9)

Kernel Requirements

The kernel should be setup for module support and ELF support. If you have already done this, skip this step, else, do the following as root:

- Load kernel in /usr/src/linux.xxx] and make sure that the symbolic link to the /usr/include/linux and /usr/include/asm directories are setup (see /usr/src/linux/README).
- Build module support and install it (see below)

TOC

- Build iBCS2 module and install it (see below)
- Build kernel with module and ELF support (answer Y to BINFMT ELF and MOD VERSIONS)
- Reboot on new kernel
- Note: you have to build the module support file "genksyms" PRIOR to making your new kernel (telling the kernel to export symbols: CONFIG MODVERSIONS=y).

Modules

Loadable module support is required for iBCS2 and hence to run WP 6.0. Get the latest from the ftp sites, build it, and install it. If you have already setup your system for loadable modules, skip this step. Loadable module support (/sbin/insmod, /sbin/rmmod, /sbin/lsmod) is required. With recent kernels, genksyms is also required.

- File: modules-1.1.87.tar.gz (or latest)
- Put tar file in /usr/src/modules and build module support
- Read the README in /usr/src/modules (or wherever you put it)
- Cd to /usr/src/modules/genksyms and make genksysms, install it in /usr/local/bin (or /sbin)

iBCS2 Emulator

The iBCS2 emulator allows you to run SCO binaries such as WordPerfect 6.0. Get the latest from the ftp sites, build it, and install it.

- Build and install in /usr/lib/modules
- Read the HINTS file (/usr/src/ibcs/Doc/HINTS)
- Make sure that iBCS2 sets up /dev/XOR, if not, create a symbolic link from /dev/null to /dev/XOR (cd /dev; ln -s /dev/null /dev/XOR)

The interfaces to some subsystems occur at the device layer and thus you need to create some device files in order to use them:

```
* /dev/socksys and /dev/nfsd - interface for SVr3 STREAMS based
  TCP/IP applications

  # mknod /dev/socksys c 30 0
  # ln -s /dev/socksys /dev/nfsd

[WARNING:  the below may already be setup as /dev/inet,
 /dev/arp, etc., they may have other major/minor numbers,
 if so, skip this step]

* /dev/inet/{arp,icmp,ip,rip,tcp,udp} - more network hooks
```

```
    # mknod /dev/inet/arp c 30 2
    # mknod /dev/inet/icmp c 30 2
    # mknod /dev/inet/ip c 30 2
    # mknod /dev/inet/rip c 30 2
    # mknod /dev/inet/tcp c 30 2
    # mknod /dev/inet/udp c 30 2

  [THIS IS REQUIRED AND WILL USUALLY HAVE TO BE DONE MANUALLY!]
  * /dev/X0R - server side of SVR3 local X interface
    (see comments in Doc/Local-X)

    # ln -s /dev/null /dev/X0R

  * /dev/spx - client side of SVR3 local X interface
    (see comments in Doc/Local-X)

    # mknod /dev/spx c 30 1
```

SCO Shared Libraries

DO NOT VIOLATE SCO's COPYRIGHTS! You should get a copy of SCO's shared libraries and install them in /shlib. As root:

- cd /; mkdev /shlib; chmod 755 /shlib
- if required, make a temporary directory and set TMPDIR to it (export TMPDIR=/mytmpdir)
- Copy the following SCO shared libraries from the SCO machine to Linux and put in /shlib:

```
/shlib/libX11R5_s
/shlib/libXtXm*
```

Loading WP60

Make sure that you have LOTS of space in your /tmp directory. If you do not have lots of space, make a tmp directory and set the environment variable TMPDIR to it. As root first try the following:

- Mount the CD (e.g., to /cdrom)
- cd to the CDROM and run the ./install script

If you get the following error you may not have enough space in your /tmp directory, make a new tmp directory and set the environment variable TMPDIR (see above):

```
ERROR:  not enough space in /tmp, or this is an unsupported machine.
```

If all else fails, you may have to do it manually per these notes:

- get iBCS2 running
- copy the $WPDIR/cdrom/install.wp to $WPDIR and edit it I changed it so I could create

TOC

/tmp/WpTmP on an NFS server (did not have the space locally). I then ran the program and ma changes until I got it to startup.]

- change to TMPDIR and run: ifiles/cdlink.sco -r /usr/local/wp60/cdrom This builds links with the proper names but the permissions are not set properly.
- make a temp directory on the server and copy all files to it:

```
cp -r  WpTmP/wp60/SCO/* wp60
```

- change to the temp wp60 directory and changed permissions. cd wp60; chmod 555 wpinstall install/*
- run the wpinstall program and answered the questions.
- Because I still have the flexlm license manager for WP51 on the system, I had to manually run wplicense -e and create a new license.dat.
- Start WP60.

License Manager

WordPerfect uses a license manager (see the manuals). This is a tricky beast that sometimes does not work properly under Linux (I have had it fail, done something else then I was able to start it). If you have problems call WordPerfect and they can assist you. Please **RTFM** the documentation section on the license manager.

To start the license manager, as root:

```
# cd $WPDIR/shbin10
# ./wplicense -s
```

Starting WP60

- start iBCS2
- start the license manager
- setup your DISPLAY environment variable (e.g., export DISPLAY=gator:0)
- run xwp ($WPDIR/wpbin/xwp filename.wpf &)

If you get the errors:

```
/dev/XOR: No such file or directory
Error: Can't open display: :0.0
```

You need to set up your DISPLAY environment variable (per above) and possibly setup the link to the device /dev/XOR (cd /dev; ln -s /dev/null /dev/XOR).

Note if WP "steals" the colors from your X server (a problem if the depth is only 256 colors), disable the display of the graphics banner page by doing the following:

- Start XWP

TOC

- From the control window, click on Preferences
- from the Preferences window, select ENVIRONMENT
- From the ENVIRONMENT window, turn off the checkbox for "Graphical Banner and Startup Screen"

RC Scripts

You will need to add a line to your /etc/rc.d/rc.local script (or similar) to start iBCS2 and the WP license manager (if desired).

```
# /sbin/insmod /usr/lib/modules/iBCS &
# $WPDIR/shbin10/wplicense -s &
```

Changes to install

This was extracted from my original notes and includes changes to **install.wp** that are reported to work:

The install.wp on the CDROM did not work out of the box. I had to modify it to get it to work by moving all files to a temporary directory and changing this install script. The changes are:

```
48c48,49
<       mkdir $LNK_DIR 2> /dev/null
---
> echo making LNK_DIR=$LNK_DIR
>       mkdir $LNK_DIR
51c52
<
---
> echo removing /tmp/TMPTMP
53a55,57
> echo /tmp/TMPTMP=
> cat /tmp/TMPTMP
> echo /tmp/TMPTMP -- to here
62,63c66,83
<       for i in $myfiles
<       do
---
> echo MYFILES=$myfiles  THISDIR=$thisdir
> ls $myfiles
> echo making LNK_DIR=$LNK_DIR/ifiles
> mkdir $LNK_DIR/ifiles
> chmod 777 $LNK_DIR/ifiles
> cp $thisdir/ifiles/* $LNK_DIR/ifiles
> chmod 777 $LNK_DIR/ifiles/*
> ls -l $LNK_DIR/ifiles
```

TOC

```
>
> # echo making links....
> # $LNK_DIR/ifiles/cdlink.sco -r $thisdir
> # echo links should be done...
> # exit
>
>
> #     for i in $myfiles
>         i=$thisdir/ifiles/cdlink.sco
> #     do
65a86,87
> echo /tmp/TMP=
> cat /tmp/TMP
73c95
<                       rm -f /tmp/TMPTMP
---
>               rm -f /tmp/TMPTMP
77c99
<       done
---
> #     done
368a391,394
> echo THISDIR=$thisdir
> thisdir=/usr/local/wp60/cdrom
> echo THISDIR=$thisdir
>
469a496,498
> echo INIT DONE, THISDIR=$thisdir
> ls $thisdir/ifiles/cdinst.*
>
493a523,526
>
> echo RUN_ME=$RUN_ME
> RUN_ME=$thisdir/ifiles/cdinst.sco
> echo RUN_ME=$RUN_ME
```

5.4 Bugs in WP/SCO

I did not keep a detailed list of bugs for WP/SCO, however:

- The initial version had install problems.
- The initial version used SCO shared libraries, hence required a copy of them on Linux.

6. The Future

Sometime in August, WordPerfect 7.0 for Linux should ship. It should be available from SDC for about $149.99 for a competitive upgrade or $199.99 retail.

Corel is working on porting their entire office suite to Java and releasing it as Corel Office for Java(TM). This currently is in Beta and works on Linux (I tried the first beta -- very slow). Check out http://officeforjava.corel.com/about.htm for more information. This promises to be something to watch.

SDC is currently working on WPLinux 8 with an expected beta mid 1998 and a release following by about 5 months. SDC has indicated that if there is not sufficient demand for WP for Linux, they will probably not ship it (they already stopped shipping for SGI and UNIXWare) -- **so if you like it, buy it!**

7. For More Information

Software Development Corporation is the developer of WordPerfect for UNIX. Corel has outsourced the development of WordPerfect for UNIX to SDC. I have been in touch with the SDC product manager for WP for LINUX and he has been very helpful (I have provided him with a review copy of this document).

```
Software Development Corporation
512 East 1860 South
Provo, Utah 84606 USA

Phone: (801) 370-9600
Fax: (801) 370-9696
Email: info@sdcorp.com
```

WWW: http://www.sdcorp.com/wpunix.html

Corel Corporation is the new owner of WordPerfect, having purchased it from Novell. They develop and market excellent products such as Corel Draw and WordPerfect.

```
Corel Corporation
Ottawa, Ontario, CA

Technical Support (WP UNIX): (801) 765-4019
```

WWW: http://www.corel.com

< **Caldera Corporation** sells WordPerfect 6.0 for Caldera Network Desktop (CND) for $129.99 (WordPerfect and Motif Bundle). Note, at this time, I have no information on if Caldera is planning to sell WP 7.0.

```
  Caldera, Inc.
633 South 550 East
Provo, Utah 84606 USA

  (888) GO LINUX
  (801) 377=7687
```

WWW: http://www.caldera.com

As a last resort, email me at whampton@staffnet.com.

X11-big-cursor MINIHOWTO

How to use enlarged mouse cursors with the X window system

Jörg Schneider

v2, 11 August 1997

This document describes how to use enlarged mouse cursors with the X window system.

1. Introduction

2. About this document

3. How to do it

4. Notes and limitations

5. Technical discussion

6. Other ideas how to make the mouse cursor more visible

7. Related info

- 7.1 How to use a font server
- 7.2 How to get the bdf source for some font

1. Introduction

There are several reasons why the standard X mouse cursors are hard to track for some people:

- when running X on a notebook with low contrast LCD
- on normal screens when using high resolution, 1600x1280 e. g.
- for visually impaired persons even on normal hardware

In all cases it might help to use enlarged mouse cursors. Ideally this job should be done by a single X program that automatically enlarges every mouse cursor.

To my knowledge there is no simple way to write a utility like this, because the X protocol has no provision to query mouse cursors. For more details see section Technical discussion below.

If we aim for a less general goal, though, something can be done:

There is a set of standard mouse cursors that can be found in the cursor font (try `xfd -fn cursor` to look at it). Most programs use these mouse cursors and the key idea is to replace the standard cursor font with an enlarged version.

2. About this document

The motivation for this MINIHOWTO was a visually impaired co-student who asked me how to enlarge the mouse cursor under X. After I found out how this can be done, I wrote an initial version of this document. The knowledge about the method described here does not seem to be common, so I decided to share it and submitted this document as a Linux MINIHOWTO, despite the fact that it is *not specific to Linux at all*. As all other MINIHOWTOs it can be found in the home of of the Linux Documentation Project (LDP).

The master of this document is maintained in the SGML/linuxdoc format. This makes it possible to automatically provide versions in the following formats (which can be found in the same place as the master): html, text, LaTeX, DVI, PostScript, GNU info.

Shinobu Miyata has done a Japanese translation of this MINIHOWTO. It can be found in http://i11www.ira.uka.de/~schneid/jp/X11-big-cursor/.

3. How to do it

Follow the steps detailed below. If you don't want to get and compile the bdfresize package yourself, you can skip to step 3 and download a magnified font instead of creating it.

1. get cursor.bdf, the source of the cursor font, from some X distribution, e. g. from ftp://ftp.x.org/pub/R6.3/xc/fonts/bdf/misc/cursor.bdf (if you don't find it there try an archie search or get it from my copy).
2. get, compile and install the bdfresize package from ftp://ftp.cs.titech.ac.jp/X11/contrib/Local/bdfresize-1.4.tar.Z (or from my copy):

   ```
   zcat bdfresize-1.4.tar.Z  | tar xf -
   cd bdfresize-1.4
   xmkmf
   make
   ```

   On Linux you probably have to use:

   ```
   make CCOPTIONS='-include /usr/include/bsd/bsd.h' clean all
   ```

3. create a directory and install a magnified cursor font in it (magnification factor 2 in this example):

   ```
   mkdir $HOME/fonts
   bdfresize -f 2 cursor.bdf | bdftopcf >$HOME/fonts/cursor2.pcf
   mkfontdir $HOME/fonts
   ```

   I have prepared some cursor fonts with the following magnification factors: 1.5, 2, 2.5, 3, 4, 5, 6, 7, 8 and 16. You can download one of them an copy it to `$HOME/fonts` if you don't want to use bdfresize.
4. modify your `.xinitrc` or `.xsession` file: before any X client (that uses cursors) is started the following commands must be executed:

   ```
   xset +fp $HOME/fonts
   xsetroot -cursor_name X_cursor
   ```

5. leave your X session and restart.

That's it—now all mouse cursors should have doubled in size.

4. Notes and limitations

- X servers may have a limit for the maximum cursor size, especially if they use a hardware implementation for the mouse cursor. Others do not have such a limit. E. g. XF86_S3 3.3 works even with a 512x512 mouse cursor (rather slowly).
- The magnified cursor font must have the same name as the original font (the *font name* must be `cursor`, *file name* does not matter)—that is no problem as `bdfresize` does not change the font name.
- The directory with the new cursor font must be placed before the directory with the standard cursor font in the the font path—this is accomplished with `xset +fp` (as opposed to `xset fp+`).
- Changes in `$HOME/fonts/` will be visible only after the command `mkfontdir $HOME/fonts; xset fp rehash` and only in newly started X clients (more exactly: for newly created cursors).
- `xset +fp path` may not work on a X-Terminal. In this case a font server (see the section How to use a fontserver) can be used if supported by the X-Terminal or some other method to install the font on the X-Terminal (this can generally only be done by your system administrator).
- The same approach can be used for `olcursor` and `decw$cursor` fonts and any other cursor font you may encounter.
- Cursor fonts produced by bdfresize don't look smooth, especially at larger magnification factors. It would be nice if someone could create better looking handcrafted version at some common sizes.

5. Technical discussion

Is it possible to write a X program that enlarges cursors automatically?

(Partial) solution 1

Use the XTestCompareCursor from the XTEST extension. For all windows that the mouse pointer enters compare the cursor of this window with a set of `known' cursors (e. g. from the cursor font). If the cursor is found, replace it with an enlarged version, otherwise either leave it alone or substitute a standard cursor. This will only work where the XTest extension is available.

Solution 2

Write a proxy X server that relays all client requests unchanged to the real X server, except that it intercepts all requests corresponing to the `XCreate*Cursor` Xlib functions. `XCreate*Cursor` requests should be modified to use an enlarged cursor.

This proxy server simulates a new display, e. g. `:1`. All clients that connect to this display (e. g. `xterm -display :1`) are displayed on the real server (normally `:0`) and their mouse cursors are enlarged automatically. The mouse cursors of clients that connect to `:0` will remain *unchanged*.

6. Other ideas how to make the mouse cursor more visible

Here are some ideas for rather simple X programs that might make mouse cursors easier to track.

- When a hot key is pressed display something (big cursor, small window, shaped window) at pointer position for 0.5s.
- use XRecolorCursor to change the mouse cursor color every 0.1s

A more demanding project would be **mouse trails** à la windoze, i. e. when the mouse is moved and the mouse cursor needs to be drawn in a different position, then the old mouse cursor does not disappear at once, but after a short delay. Mouse trails would be probably best implemented in a X server, but it might be feasible to do it as a X client, or better as a proxy server (see section Technical discussion for details).

7. Related info

7.1 How to use a font server

A font server is a net service that provides a set of X11 fonts with a simple protocol. It can be queried which fonts it provides and will supply the font bitmap data on request.

You might want to use a font server to provide the X server with a modified cursor font, instead of telling it where to find the font on the file system.

This method is especially handy if you use several machines that don't share a common file system or if you use X terminals that support the font server protocol.

A font server program and associated tools comes with the X11R5+ distribution (AFAIK).

Setting up a font server

Read the manual pages *fs(1)*, *fslsfonts(1)* (or *xfs(1)*, *xfslsfonts(1)* under X11R6) and try it—it isn't hard. Say, you are running the server on host `some.host.edu` on port 7100. You can test the setup with the command

```
fslsfonts -server some.host.edu:7100
```

To actually use the server issue the command

```
xset +fp tcp/some.host.edu:7100
```

which should return without an error message.

7.2 How to get the bdf source for some font

If you have set up a font server simply use `fstobdf` which comes with the font server.

Alternatively you may try `getbdf` which can dump any installed X11 font to a bdf file.

Linux XFree-to-Xinside mini-HOWTO

by Marco Melgazzi, `marco@techie.com`

v1.3, September 1997

How to convert an XFree86 modeline into an XInside/XiGraphics one

1. Introduction
2. Why should I need it ?
3. Let's go
4. Fixing up things
5. The end...
6. Automating the process
7. Thanks to
8. Copyright/legalese

1. Introduction

During the spring of 1996 I've seen a lot of posts in comp.os.linux.x asking how to convert video modes between XFree86 and one of its commercial alternatives: XInside (now named XiGraphics, note anyway that in this document I'll use the old product name, since I will mainly refer to that version)

I had evaluated before the product and had this evaluation version still floating on my hard disk: since I like problem solving, I've decided to give it a try and, after a couple of hours of fiddling and calculating, I came up with a supposedly informative article that was promptly posted.

The discussions about how to convert suddendly vanished and I received 1 (one) mail thanking me for the article so, since maybe somebody else could need this information in the future, I decided to transform that post in the mini-HOWTO you are reading.

Let me state something first: I do NOT work for XInside and I only had access to the evaluation 1.2 version for Linux. I know that nowadays (May 97) AccelX has reached revision 3.1, but I do think that the information included in this document, if not verbatim, is still applicable.

Due to the fact that this HOWTO has been written with the help of a pretty old Xinside version, it may well happen that some of the information contained here is not completely accurate: as you will read later, thanks to the birth of XFree 3.2, I haven't bought this commercial server, so if you have, and if you notice any incorrect information here please take the time to email me.

Notice anyway that fiddling with monitor timings can be hazardous and, for this reason, I absolutely make no guarantees. If it works for you , fine, if you blow up your computer I shall not be held responsible for it.

As you may have noticed from may name, I'm not a native speaker of English so you will probably find some errors here and there, I apologize for them and I ask you to please avoid flooding my mailbox with language-related flames. Thanks !

2. Why should I need it ?

I think that the Xinside policy of not giving you an utility to tweak your video modes (like xvidtune) and/or to import your existing XFree ones in the evaluation (and AFAIK commercial) version is incomprehensible. I've spent about three hours putting this together (hint: I've compared the VESA 1024x768@70Hz entry in the two formats (and I'm nearly an electronic engineer ;-)) while an Xinside programmer could have written a comparable article in a fraction of this time...

I haven't downloaded any evaluation version from 1.3 onwards and I really hope they have fixed this. Well, if they have, this mini-HOWTO could be considered useless but, alas, if you read it you will learn something more about how everything works...

3. Let's go

Let's suppose that you have your oh-so-tweaked XFree86 mode and you want to evaluate Xinside in the same conditions: follow the steps described below and you should be able to do it; we will use my default video mode as a real-life example and I will explain what you will have to do to convert it.

An Xfree86 entry looks like this:

```
Modeline "blahblah" DOTCLK  A B C D  a b c d
```

Every one of the A-D and a-d numbers has a meaning: if you want you can search for it in the 'The Hitchhiker's Guide to X386/XFree86 Video Timing' (/usr/lib/X11/doc/VideoModes.doc) but you don't need to know the theory behind all this to perform a succesful conversion...

My modeline in /usr/lib/X11/XF86Config is:

```
Modeline "1168x876" 105  1168 1256 1544 1640  876 877 891 900
                     |     |    |    |    |    |   |   |   |
                  DOT_CLK  A    B    C    D    a   b   c   d
```

In Xinside, you have to add an entry in the Xtimings file, which should be located in etc/ (from now on we suppose you are in the top Xaccel directory that should be something like /usr/X11/lib/X11/AcceleratedX)

```
!    Somewhere in the file, put here the name you want

[PREADJUSTED_TIMING]
    PreadjustedTimingName = "1168x876 @ 72Hz";

!
!    These four are obvious
!
    HorPixel          = 1168;         // pixels
    VerPixel          = 876;          // lines
    PixelWidthRatio   = 4;
    PixelHeightRatio  = 3;

!
!   hsync: DOT_CLK / D * 1000 [KHz]
!
!   hsync = 105 / 1640 * 1000 = 64.024 KHz
!
```

TOC

```
!   vsync: ( 1 / (( D / DOT_CLK ) * d) ) * 1,000,000 [Hz]
!
!   vsync: ( 1 / (( 1640 / 105 ) * 900) ) * 1,000,000
!           ( 1 / 14057.1428571 ) * 1,000,000 = 71.138 Hz
!

    HorFrequency      = 64.180;        // kHz
    VerFrequency      = 71.138;        // Hz

!   Obvious

    ScanType          = NONINTERLACED;

!
!   Put here the +/-hsync +/-vsync XFree86 options
!
    HorSyncPolarity   = POSITIVE;
    VerSyncPolarity   = POSITIVE;

!   Shouldn't change

    CharacterWidth    = 8;             // pixels

!   DOT_CLK here

    PixelClock        = 105.000;       // MHz
!
!
!   horizontal timings section: [usec]
!
    HorTotalTime  = D / DOT_CLK                  = 15.619;
    HorAddrTime   = A / DOT_CLK                  = 11.124;
    HorBlankStart = A / DOT_CLK                  = 11.124;
    HorBlankTime  = HorTotalTime - HorBlankStart =  4.495;
    HorSyncStart  = B / DOT_CLK                  = 11.962;
    HorSyncTime   = C / DOT_CLK - HorSyncStart   =  2.743;

 !
 !  vertical timings section:    [msec]
 !

    VerTotalTime  = ( HorTotalTime * d ) / 1000  = 14.057;
    VerAddrTime   = ( HorTotalTime * a ) / 1000  = 13.682;
    VerBlankStart = ( HorTotalTime * a ) / 1000  = 13.682;
    VerBlankTime  = VerTotalTime - VerBlankStart =  0.375;
    VerSyncStart  = ( HorTotalTime * b ) / 1000  = 13.698;
```

```
VerSyncTime   = ( HorTotalTime * ( c - b ) ) / 1000
                                             = 0.219

! Finished !
```

Now you have to put this newly created mode in the files shown below in the appropriate place.

---

4. Fixing up things

In the excerpts shown below the -> sign tells you what was modified: do NOT include it in your files!

Monitor entry (mine is monitors/mfreq/mfreq64.vda)

```
        [ESTABLISHED_TIMINGS]
            "640x480 @ 60Hz",
            "640x480 @ 72Hz",
            "640x480 @ 75Hz",
            "800x600 @ 56Hz",
            "800x600 @ 60Hz",
            "800x600 @ 72Hz",
            "800x600 @ 75Hz",
            "1024x768 Interlaced",
            "1024x768 @ 60Hz",
            "1024x768 @ 70Hz",
            "1024x768 @ 75Hz",
        "1152x900 Interlaced",
            "1152x900 @ 60Hz",
            "1152x900 @ 67Hz",
    ->      "1168x876 @ 72Hz",
            "1280x1024 Interlaced",
            "1280x1024 @ 60Hz",
        "1600x1200 Interlaced";
```

Board info file (mine is boards/s3/764-2.xqa , I wonder why they have nearly all the Hercules boards but not MINE: Terminator 64/Dram)

```
    [VISUAL]
        BitsPerPixel   = 8;
        MemoryModel    = Packed;
        ColorModel     = Indexed;
        BitsRGB        = 6;
        NumberOfColors = 256;

        [RESOLUTIONS]
        640x480,
        800x600,
        1024x768,
    ->  1168x876,
        1152x900,
```

TOC

```
    1280x1024

    [DESKTOPS]
    640x480,
    800x600,
    1024x768,
    1152x900,
->  1168x876,
    1280x1024,
    1600x1200
```

If the dot clock is low enough (NOT in this case for my board) you can put the entry even in the 16bpp and 32bpp sec- tions.

The /etc/Xaccel.ini will look something like this

```
-----------------------------------------------------------------
    Board   = "s3/764-2.xqa";
    Monitor = "mfreq/mfreq64.vda";
    Depth   = 8;
->  Desktop = 1168x876;

    [RESOLUTIONS]
->      1168x876,
        1024x768;
```

The actual Xinside mode entry in etc/Xtimings

```
-----------------------------------------------------------------
[PREADJUSTED_TIMING]
    PreadjustedTimingName = "1168x876 @ 72Hz";

    HorPixel          = 1168;          // pixels
    VerPixel          = 876;           // lines
    PixelWidthRatio   = 4;
    PixelHeightRatio  = 3;
    HorFrequency      = 64.024;        // kHz
    VerFrequency      = 71.138;        // Hz
    ScanType          = NONINTERLACED;
    HorSyncPolarity   = POSITIVE;
    VerSyncPolarity   = POSITIVE;
    CharacterWidth    = 8;             // pixels
    PixelClock        = 105.000;       // MHz
    HorTotalTime      = 15.619;        // (usec) =  205 chars
    HorAddrTime       = 11.124;        // (usec) =  146 chars
    HorBlankStart     = 11.124;        // (usec) =  146 chars
```

TOC

```
HorBlankTime      =  4.495;        // (usec) =   59 chars
HorSyncStart      = 11.962;        // (usec) =  157 chars
HorSyncTime       =  2.743;        // (usec) =   36 chars
VerTotalTime      = 14.057;        // (msec) =  900 lines
VerAddrTime       = 13.682;        // (msec) =  876 lines
VerBlankStart     = 13.682;        // (msec) =  876 lines
VerBlankTime      =  0.375;        // (msec) =   24 lines
VerSyncStart      = 13.698;        // (msec) =  877 lines
VerSyncTime       =  0.219;        // (msec) =   14 lines
```

You can check your conversion by running the vgaset program with no parameters while running the Xinside server: it will output an XFree-like line and, if everything went OK, this line will be equal to the line you started from (except if b and c are equal, I haven't been able to reproduce this situation in Xinside: the best case was c=b+1).

---

5. The end...

That's all folks ! I hope this will be useful to you. I don't think I'll buy the XiGraphics server in the near future for one simple reason: the release of XFree86 3.2 solved all of the text speed problems I was having on my humble Trio 64 video board ;)

It seems anyway that the XiGraphics server supports a much wider array of chipsets and video boards than XFree, so it may well happen that the commercial 'alternative' is the only viable one for you. If this is the case, and you bought the XiGraphics server, I would really like to hear from you to know if the information presented here has been useful to you, or if you found it too complex or whatever.

6. Automating the process

This small script automates most of the work. Be very careful with the ScanType and with the two Polarity lines: the script do not set them and, if you are too lazy to correct them, the risks of blowing up your monitor increase quite a lot.

Notice that I don't know if the 'Doublescan' flag has meaning in XInside: if you try to convert a low-res doublescan mode BE CAREFUL, you can easily kill your monitor since the refresh rate that you get is doubled (in fact my 400x300@72Hz became a 400x300@144Hz !).

```
#!/bin/sh
##########################################################################
# XF2XInside
#
# This script converts modelines from XF86Config format to XInside
# format as needed for the etc/Xtiming file.
#
# This is a quick hack, so don't expect much error checking (not to
# speak of anything like user friendlyness).
#
# If you call it without arguments it should tell you what to do.
#
#                               ( July 1996, hcz@tazlwurm.bb.bawue.de)
#
# Btw: New modes created as described in the HOWTO work, but don't
# show up in Xsetup's menu. Anybody who knows why?
#
##########################################################################
#---------------------------------------------- Here we go:
# Change this if your modeline file lives somewhere else:
XF=/usr/X11/lib/X11/XF86Config
if [ $# -ne 1 ] ; then
  echo "usage: ${0##*/} <mode>"
  echo " example: ${0##*/} 1024x764"
  echo -e " function: converts $XF modeline entry into\n Xinside Format (stdout)"
  exit 1
fi
egrep -i "^[\t ]*modeline.+\"$1\""  /usr/X11/lib/X11/XF86Config |
gawk '
NF < 11  { print "! invalid Modeline:\n! " $0 "\n!"; next }
{
  print "//", $0  ":"
  name = $2
  DOT_CLK = $3;
  A = $4;
  B = $5;
  C = $6;
  D = $7;
  a = $8;
  b = $9;
  c = $10;
  d = $11;
```

```
    VerFrequency =  1000000 / ((D / DOT_CLK) * d)
    print "[PREADJUSTED_TIMING]"
    printf "  PreadjustedTimingName = \"%dx%d @ %.0dHz\";\n", A, a, VerFrequency
    print "  HorPixel\t\t= " A ";"
    print "  VerPixel\t\t= " a ";"
    print "  PixelWidthRatio\t= 4;\n  PixelHeightRatio\t= 3;"
    print "  HorFrequency\t\t= " DOT_CLK / D * 1000 ";\t// kHz"
    print "  VerFrequency\t\t= " VerFrequency  ";\t// Hz"
    print "  ScanType\t\t= NONINTERLACED;\t\t// *CHECK*"
    print "  HorSyncPolarity\t= NEGATIVE;\t\t\t// *CHECK*"
    print "  VerSyncPolarity\t= NEGATIVE;\t\t\t// *CHECK*"
    print "  CharacterWidth\t= 8;"
    print "  PixelClock\t\t= " DOT_CLK ";"
    HorTotalTime = D / DOT_CLK
    print "  HorTotalTime\t\t= " HorTotalTime ";"
    print "  HorAddrTime \t\t= " A / DOT_CLK ";"
    print "  HorBlankStart\t\t= " A / DOT_CLK ";"
    print "  HorBlankTime\t\t= " D / DOT_CLK - A / DOT_CLK ";"
    print "  HorSyncStart\t\t= " B / DOT_CLK ";"
    print "  HorSyncTime\t\t= " C / DOT_CLK - B / DOT_CLK ";"
    VerTotalTime  = ( HorTotalTime * d ) / 1000
    print "  VerTotalTime\t\t= " VerTotalTime ";"
    print "  VerAddrTime\t\t= " ( HorTotalTime * a ) / 1000 ";"
    VerBlankStart = ( HorTotalTime * a ) / 1000
    print "  VerBlankStart\t\t= " VerBlankStart ";"
    print "  VerBlankTime\t\t= " VerTotalTime - VerBlankStart ";"
    print "  VerSyncStart\t\t= " ( HorTotalTime * b ) / 1000 ";"
    print "  VerSyncTime\t\t= " ( HorTotalTime * ( c - b ) ) / 1000
    print ""
  }'
```

7. Thanks to

- Heike Claudia Zimmerer hcz@tazlwurm.bb.bawue.de for pointing out a small inconsistency and for sending me a script that automates most of the work.
- Bartosz Maruszewski B.Maruszewski@zsmeie.torun.pl for translating this mini HOWTO in Polish and for pointing out a small typo.

8. Copyright/legalese

(c)opyright 1996-7 by Marco Melgazzi (marco@techie.com) - the GPL (Gnu Public License) applies. To obtain a copy of the GPL write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Trademarks are owned by their owners. There is no warranty on the accuracy and/or the usefulness of the information given in this document.

How to change the title of an xterm

Richard Lister, ric@giccs.georgetown.edu

$Revision: 1.6 $, Last modified: Wed Jan 7 14:34:31 EST 1998

This document explains how to use escape sequences to dynamically change window and icon titles of an xterm. Examples are given for several shells, and the appendix gives escape sequences for some other terminal types.

- Where to find this document
- Static titles
- Dynamic titles
- Examples for different shells
  - zsh
  - bash
  - tcsh
  - csh
  - ksh
- Appendix: escapes for other terminal types
  - SGI `wsh`, `xwsh` and `winterm`
  - CDE `dtterm`
  - Sun `cmdtool` and `shelltool`
  - Example
- Credits

Where to find this document

This document is now part of the Linux HOWTO Index and can be found at `http://sunsite.unc.edu/LDP/HOWTO/mini/Xterm-Title.html`.

The latest version can always be found at `http://www.giccs.georgetown.edu/~ric/howto/Xterm-Title.html`.

Static titles

A static title may be set for any of the terminals `xterm`, `color-xterm` or `rxvt`, by using the `-T` and `-n` switches:

```
xterm -T "My XTerm's Title" -n "My XTerm's Icon Title"
```

TOC

Dynamic titles

Many people find it useful to set the title of a terminal to reflect dynamic information, such as the name of the host the user is logged into, the current working directory, etc.

This may be done by using XTerm escape sequences. The following sequences are useful in this respect:

```
ESC]0;stringBEL    Set icon name and window title to string
ESC]1;stringBEL    Set icon name to string
ESC]2;stringBEL    Set window title to string
```

where `ESC` is the *escape* character (\033), and `BEL` is the *bell* character (\007).

Note: these sequences apply to most xterm derivatives, such as `nxterm`, `color-xterm` and `rxvt`. Other terminal types often use different escapes; see the appendix for examples. For the full list of xterm escape sequences see the file `ctlseq2.txt`, which comes with the xterm distribution, or `xterm.seq`, which comes with the `rxvt` distribution.

These escapes really need to be applied every time the prompt changes. This way the string is updated with every command you issue and can keep track of information such as current working directory, username, hostname, etc. Some shells provide special functions for this purpose, some don't and we have to insert the title sequences directly into the prompt string.

Examples for different shells

In all the examples below we test the environment variable `TERM` to make sure we only apply the escapes to xterms. We test for `TERM=xterm*`; the wildcard is because some variants (such as rxvt) can set `TERM=xterm-color`.

zsh

zsh provides some functions and expansions, which we will use:

```
precmd ()   a function which is executed just before each prompt
chpwd ()    a function which is executed whenever the directory is changed
\e          escape sequence for escape (ESC)
\a          escape sequence for bell (BEL)
%n          expands to $USERNAME
%m          expands to hostname up to first '.'
%~          expands to directory, replacing $HOME with '~'
```

There are many more expansions available: see `'man zshmisc'`.

Thus, the following inserted in `~/.zshrc` will set the xterm title to "*`username@hostname: directory`*" ...

```
case $TERM in
    xterm*)
```

TOC

```
        precmd () {print -Pn "\e]0;%n@%m: %~\a"}
        ;;
esac
```

This could also be achieved by using `chpwd()` instead of `precmd()`.

bash

`bash` supplies a variable `PROMPT_COMMAND` which contains a command to execute before the prompt. This example (inserted in `~/.bashrc`) sets the title to "*`username@hostname: directory`*":

```
PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME}: ${PWD}\007"'
```

where `\033` is the character code for ESC, and `\007` for BEL. Note that the quoting is important here: variables are expanded in `"..."`, and not expanded in `'...'`. So `PROMPT_COMMAND` is set to an unexpanded value, but the variables inside `"..."` are expanded when `PROMPT_COMMAND` is used.

However, `PWD` produces the full directory path. If we want to use the `'~'` shorthand we need to embed the escape string in the prompt, which allows us to take advantage of the following prompt expansions provided by the shell:

```
\u          expands to $USERNAME
\h          expands to hostname up to first '.'
\w          expands to directory, replacing $HOME with '~'
\[...\]     embeds a sequence of non-printing characters
```

Thus, the following produces a prompt of "`bash$ `", and an xterm title of "*`username@hostname: directory`*" ...

```
case $TERM in
    xterm*)
        PS1="\[\033]0;\u@\h: \w\007\]bash\$ "
        ;;
    *)
        PS1="bash\$ "
        ;;
esac
```

Note the use of `\[...\]`, which tells `bash` to ignore the non-printing control characters when calculating the width of the prompt. Otherwise line editing commands get confused while placing the cursor.

tcsh

`tcsh` has functions and expansions similar to those of `zsh`:

```
precmd ()   a function which is executed just before each prompt
cwdcmd ()   a function which is executed whenever the directory is changed
%n          expands to username
%m          expands to hostname
%~          expands to directory, replacing $HOME with '~'
```

Unfortunately, there is no equivalent to zsh's `print` command allowing us to use prompt escapes in the title string, so the best we can do is to use shell variables (in `~/.tcshrc`):

TOC

```
switch ($TERM)
    case "xterm*":
        alias precmd 'echo -n "\033]0;${HOST}:$cwd\007"'
        breaksw
endsw
```

but this gives the directory's full path instead of using '~'. Instead you can insert the string in the prompt:

```
switch ($TERM)
    case "xterm*":
        set prompt="%{\033]0;%n@%m:%~\007%}tcsh%% "
        breaksw
    default:
        set prompt="tcsh%% "
        breaksw
endsw
```

which sets a prompt of "`tcsh% `", and an xterm title and icon of "*`username@hostname: directory`*". Note that the "`%{...%}`" must be placed around escape sequences (and cannot be the last item in the prompt: see `'man tcsh'` for the details).

csh

This is very difficult indeed in `csh`, and we end up doing something like the following (in `~/.cshrc`):

```
switch ($TERM)
    case "xterm*":
        set host=`hostname`
        set user=`whoami`
        alias cd 'cd \!*; set prompt="^[]0;${user}@${host}: ${cwd}^Gcsh% "'
        breaksw
    default:
        set prompt='csh% '
        breaksw
endsw
```

where we have had to alias the `cd` command to do all the work of setting the prompt. Note that the '`^[`' and '`^G`' in the prompt string are single characters for ESC and BEL (can be entered in emacs using '`C-q ESC`' and '`C-q C-g`').

I strongly recommend abandoning csh in favour of a more advanced shell: zsh, bash or tcsh.

ksh

`ksh` provides little in the way of functions and expansions, so we have to insert the escape string in the prompt to have it updated dynamically. This example produces a title of "*`username@hostname: directory`*" and a prompt of `"ksh$ "`.

```
case $TERM in
    xterm*)
        HOST=`hostname`
```

TOC

```
        PS1='^[]0;${USER}@${HOST}: ${PWD}^Gksh$ '
        ;;
    *)
        PS1='ksh$ '
        ;;
esac
```

However, `PWD` produces the full directory path. We can remove the prefix of `$HOME/` from the directory using the `${...##...}` construct. We can also use `${...%%...}` to truncate the hostname:

```
HOST=`hostname`
HOST=${HOST%%.*}
PS1='^[]0;${USER}@${HOST}: ${PWD##${HOME}/}^Gksh$ '
```

Note that the '`^[`' and '`^G`' in the prompt string are single characters for ESC and BEL (can be entered in emacs using '`C-q ESC`' and '`C-q C-g`').

Appendix: escapes for other terminal types

SGI `wsh`, `xwsh` and `winterm`

These terminals set `TERM=iris-ansi` and use the following escapes:

```
ESC P 1 .y string ESC \\        Set window title to string
ESC P 3 .y string ESC \\        Set icon title to string
```

For the full list of `xwsh` escapes see `xwsh(1G)`.

Sun `cmdtool` and `shelltool`

`cmdtool` and `shelltool` both set `TERM=sun-cmd` and use the following escapes:

```
ESC ] l string ESC \            Set window title to string
ESC ] L string ESC \            Set icon title to string
```

These are truly awful programs: use something else.

CDE `dtterm`

`dtterm` sets `TERM=dtterm`, and appears to recognise both the standard `xterm` escape sequences and the Sun `cmdtool` sequences. (I've only tested this on a Sun ... anyone have HP and DEC versions they can test for me?).

Example

Here's my full setup using `zsh` to handle all these terminal types:

```
case $TERM in

  xterm*|dtterm)
    precmd () {print -Pn "\e]0;%n@%m: %~\a"}  ## window & icon title
```

TOC

```
    ;;

  iris-ansi)
    precmd () {
      print -Pn "\eP1.y%n@%m: %~\e\\"         ## window title
      print -Pn "\eP3.y%n@%m: %~\e\\"         ## icon title
    }
    ;;

  sun-cmd)
    precmd () {
      print -Pn "\e]l%n@%m: %~\e\\"           ## window title
      print -Pn "\e]L%n@%m: %~\e\\"           ## icon title
    }
    ;;

esac
```

Credits

Thanks to the following people.

Paul D. Smith <psmith@BayNetworks.COM> and Christophe Martin <cmartin@ipnl.in2p3.fr>

both pointed out that I had the quotes the wrong way round in the `bash` `PROMPT_COMMAND`. Getting them right means variables *are* expanded dynamically.

Paul D. Smith <psmith@BayNetworks.COM>

suggested the use of `\[...\]` in the `bash` prompt for embedding non-printing characters.

Christophe Martin <cmartin@ipnl.in2p3.fr>

provided the solution for `ksh`.

Keith Turner <keith@silvaco.com>

supplied the escape sequences for Sun `cmdtool` and `shelltool`.

Installing Linux on ZIP disk using ppa ZIP Drive Mini-Howto

John Wiggins, jwiggins@comp.uark.edu

v0.7, 26 January 1998

This document is only useful for those with the printer port version of a ZIP drive who wish to have either a portable or backup Linux system on a ZIP disk.

1. Disclaimer

2. Introduction

- 2.1 What's new
- 2.2 Conventions
- 2.3 Updates
- 2.4 References

3. Setting up the ZIP disk

- 3.1 Partitioning ZIP disk
- 3.2 Formatting and mounting the ZIP disk
- 3.3 Creating the boot disk

4. Red Hat 4.2 installation

- 4.1 Personal setup
- 4.2 Package installation
- 4.3 Problems after installation of packages

TOC

5. Slackware 2.2 installation

- 5.1 Requirements
- 5.2 Installation
- 5.3 What to install

6. Creating /etc/fstab

7. Debian 1.2 Installation

- 7.1 Requirements
- 7.2 Overview
- 7.3 Creating the modified Rescue disk
- 7.4 Install the base system on the ZIP drive.
- 7.5 Creating the boot disk
- 7.6 Reboot the system.
- 7.7 Configure the base system and complete the install process.
- 7.8 Installing the modules you built in step 7.3.2.

8. Afterthoughts

---

1. Disclaimer

NOTE: I have no idea if the IDE drive works the same way as the printer port version does since I don't have one, so please don't ask me.

The Debian install portion was basically just copied with little or no editing by this author. As such, there may be some duplication of instructions.

This document assumes the following:

- You have a printer port ZIP drive (since the ZIP Plus has both, I assume that it will work as well.)
- You already have Linux installed and running; this document is not for a first time install of Linux.
- You have ppa support in your current kernel or if module, the ppa module has been loaded.
- The mount point for the ZIP disk is the /iomega directory.

2. Introduction

This document is divided into four basic sections each describing how to install a bare-bones Linux system on a 100MB ZIP disk using a printer port ZIP drive. The first section describes how to set up the ZIP disk and is common to both Red Hat and Slackware distribution installations. The second, third, and fourth sections describe how to install Red Hat 4.2, Slackware 2.2, and Debian 1.2 distributions, respectively, onto the ZIP disk.

NOTE: I realize that Red Hat 5.0 has been released now, but between classes and work, well let's just say that it may be late May before I can get around to work on it. I also, hopefully, will be testing out the other distributions.

2.1 What's new

I've finally found enough time (although I really should be studying for a sociology test...) to update this document. Thanks for all who emailed me with comments.

New to this document:

- The Debian distribution
- LILO on the floppy
- New version of Red Hat (4.2 Biltmore)
- Network configuration (For Red Hat; Untested)

2.2 Conventions

Indicates the following text are commands:

==> Indicates something noteworthy:

```
NOTE:
```

Indicates a screen shot/capture:

```
    Text here.
```

2.3 Updates

For any updates, however rare they may be, please check: http://comp.uark.edu/~jwiggins/linuxZIP/

2.4 References

- Installation-HOWTO
- SCSI-HOWTO
- NET-3-HOWTO (for section 4.3.3)
- ZIP-Drive (mini-HOWTO)
- ParPort kernel patch (gives access to pass-through printer port) http://www.cyberelk.demon.co.uk/parport/

Acknowledgments/Contributors

Slackware 2.2 section courtesy of Michael Littlejohn mike@mesa7.mesa.colorado.edu

Debian 1.2 section courtesy of John D. Blair jdblair@uab.edu

LILO information and many other helpful insight courtesy of Darcy Boese possum@niagara.com and Javier Rodriguez jrodrigu@nextgeninter.net.mx

---

3. Setting up the ZIP disk

(Common for both Red Hat and Slackware distributions.) Before starting, make sure that you have access to the ZIP drive; either by having ppa in the kernel or by having the ppa module loaded. One easy way to find this out is by checking dmesg:

==> **dmesg**

You may have to pipe this to more as dmesg tends to be rather long. Here's a snip from mine:

```
scsi0 : PPA driver version 0.26 using 4-bit mode on port 0x3bc.
scsi : 1 host.
  Vendor: IOMEGA    Model: ZIP 100           Rev: D.08
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi removable disk sda at scsi0, channel 0, id 6, lun 0
SCSI device sda: hdwr sector= 512 bytes. Sectors= 196608 [96 MB] [0.1 GB]
 sda: Write Protect is off
 sda: sda1 sda2
```

If you only see something like:

```
   scsi : 0 hosts.
   scsi : detected total.
```

then you have SCSI support but the ZIP wasn't found.

3.1 Partitioning ZIP disk

To partition the ZIP disk, run fdisk:

==> **fdisk /dev/sda/**

Here is a snap of the partition table I have setup:

```
Disk /dev/sda: 64 heads, 32 sectors, 96 cylinders
Units = cylinders of 2048 * 512 bytes

   Device Boot    Begin    Start      End   Blocks   Id  System
/dev/sda1            1        1       81    82928   83  Linux native
/dev/sda2           82       82       96    15360   82  Linux swap
```

I decided to use a swap partition since I wanted to be able to use this with any machine.

TOC

3.2 Formatting and mounting the ZIP disk

After running fdisk, format the new partition:

==> **mke2fs -c /dev/sda1**

Then, create the swap partition: (15360 blocks as taken from fdisk)

==> **mkswap -c /dev/sda2 15360**

Last, you'll need to mount the ZIP disk:

==> **mount /dev/sda1 /iomega -t ext2**

3.3 Creating the boot disk

Since the ppa version of the ZIP drive isn't a true SCSI device, it isn't a bootable device and, therefore, requires a boot disk which has ppa included in the kernel and not as a module.

Configuring and making the kernel

First, you'll need to configure and make a kernel that has ppa support enabled and not as a loadable module. In order to get to the ppa option, select SCSI support:

SCSI support (CONFIG_SCSI) Y/m/n/?

Plus, SCSI disk support:

SCSI disk support (CONFIG_BLK_DEV_SD) Y/m/n/?

And finally, under the SCSI low-level drivers, is the ppa support:

IOMEGA Parallel Port ZIP drive SCSI support (CONFIG_SCSI_PPA) Y/m/n/?

Again, be sure not to include ppa as a module, but rather in the kernel. Thus far, without the use of the parport kernel patch (see 1.4), the ppa driver will not allow the passive port of the ZIP drive to be used for a printer, so you may want to say no to parallel printer support:

Parallel printer support (CONFIG_PRINTER) N/y/m/?

```
NOTE: For more information concerning the ppa driver, please
      refer to the ZIP-Drive mini-HOWTO.
```

Once the kernel is configured, make the kernel:

==> **make dep; make clean; make zImage**

The new kernel should be found in arch/i386/boot/zImage.

TOC

Getting the kernel to a floppy

After having to have 4 separate floppies due to different kernels and needing different parameters, (plus the great email I got telling me how to do this) I have included a section on LILO as one of the means of creating a bootable floppy.

LILO installation

For those who have to, or for that matter just want to, have several kernels on one floppy (now adays, mine are too large) or just want to be able to pass arguments (such as single user mode) I received email on how to install LILO on a floppy.

Creating the ext2 filesystem

To create an ext2 filesystem on a floppy, just do the same command for the ZIP disk:

==> **mke2fs -t /dev/fd0**

Copying the essential files

Next, make sure there's a directory for a mount point, and mount the floppy (I used /mnt/floppy):

==> **mount /dev/fd0 /mnt/floppy -t ext2**

is to boot properly, you'll need the same files that LILO uses on your current Linux installation.

```
NOTE: The file locations here are from my machine
      and may not be the same for everyone.
```

==> **cp /boot/boot.b /mnt/floppy**

==> **cp /boot/map /mnt/floppy**

==> **cp /usr/src/linux/arch/i386/boot/zImage /mnt/floppy/vmlinuzDESK**

Now to create the config file for LILO, now I miss the liloconfig program... (Thanks to Javier Rodriguez for this info) First, create the LILO config file, /mnt/floppy/lilo.conf, for the kernel(s) for the ZIP disk. Here's what I used so that I could have different kernels to test with:

```
boot=/dev/fd0
map=/mnt/floppy/map
install=/mnt/floppy/boot.b
prompt
compact
timeout=50
image=/mnt/floppy/vmlinuzLAP
   label=Laptop
   root=/dev/sda1
   read-only
```

```
image=/mnt/floppy/vmlinuzDESK
   label=Desktop
   root=/dev/sda1
   read-only
image=/mnt/floppy/vmlinuzDESK
   label=rescue
   root=/dev/hdc1
   read-only
```

I have two kernels, one for my 486 laptop which required the math-co emulation in the kernel and the other for my desktop. The rescue allows me to make an emergency boot to the hard drive.

Last but not least, with the floppy still mounted, run LILO to install it on the floppy with the command:

==> **lilo -C /mnt/floppy/lilo.conf**

Once LILO has been installed on the floppy, skip the next two steps, unless you enjoy doing this over again :)

Kernel only installation

```
NOTE: This does not pertain for the LILO install.
```

Copy the newly made kernel to a floppy disk:

==> **cp arch/i386/boot/zImage /dev/fd0**

or

==> **cat arch/i386/boot/zImage > /dev/fd0**

Yes, there are many ways to copy the kernel to a floppy, but the last way, my favorite, is a little more encrypted. Try not to forget the > unless you like viewing binary files :)

Setting the root and swap on the floppy

```
NOTE: This does not pertain for the LILO install.
```

Once the kernel is on the floppy, you need to set the root device to the ZIP disk: ==> **rdev /dev/fd0 /dev/sda1** I'm not sure if the next option is needed, but I did it none the less. To set the swap:

==> **rdev -s /dev/fd0 /dev/sda2**

---

4. Red Hat 4.2 installation

With anything computer related, something 3 months old is considered obsolete and is in need of upgrading. Since I'm not always going to have the time to update this document with every update, I'll try my best to at least update it every other version. As for the other distributions, unless the authors wish to send updates, they well remain as they are.

4.1 Personal setup

For my installation, I have and used:

- Kernel 2.0.30
- Iomega ppa disk drive
- Red Hat 4.2

4.2 Package installation

When I first decided to attempt to install Red Hat on a ZIP disk, I figured it would be much easier to just use a Red Hat boot disk. Then I woke up. I came very close to actually getting a boot disk created, even got help from various folks at Red Hat via e-mail but ultimately I began to give up on the whole project when I discovered the --root option with rpm.

What packages to get

I found what packages to install by browsing a file I had found on one of Red Hat's mirrors. This file can be found on any mirror at:

```
redhat/redhat-4.2/i386/RedHat/base/comps
```

For this installation, I wanted to include network support but due to Red Hat's X network configuration, I'm going to have to manually configure, or rather manually attempt to configure, the network setup scripts found in /etc/sysconfig/ (see section 4.3.3.)

I decided against installing any development packages as the ZIP drive, at least an unpatched kernel version, is rather slow to get anything to compile. I also choose not to install X mainly for disk space issues. Later on, I may attempt to mount my hard drive and create a symlink with /usr to see if I can get X to work.

Following is a list of what packages I installed, listed in order of installation. Those marked with a * have updates from Red Hat's errata. In parentheses is the updated package number;

```
e.g. NetKit-B-0.09-6 was updated to NetKit-B-0.09-8 so the entry
    would be: *91) NetKit-B-0.09-6 (-8)
```

```
(List created by the command rpm --root /iomega -qa)

 1) setup-1.7-2                    2) pamconfig-0.51-2
 3) filesystem-1.3-1               4) MAKEDEV-2.2-9
 5) adduser-1.7-1                  6) libc-5.3.12-18
```

TOC

```
  7) SysVinit-2.64-8                 8) ash-0.2-8
  9) at-2.9b-2                      10) libtermcap-2.0.8-4
 11) bash-1.14.7-1                  12) bc-1.03-6
 13) bdflush-1.5-5                  14) cpio-2.4.2-4
 15) cracklib-dicts-2.5-1           16) tmpwatch-1.2-1
 17) crontabs-1.5-1                *18) db-1.85-10 (-11)
 19) dev-2.5.1-1                    20) diffutils-2.7-5
 21) etcskel-1.3-1                  22) file-3.22-5
 23) fileutils-3.16-1               24) findutils-4.1-11
 25) grep-2.0-5                     26) groff-1.10-8
*27) ld.so-1.7.14-4 (-5)            28) getty_ps-2.0.7h-4
 29) gzip-1.2.4-7                   30) mingetty-0.9.4-3
*31) initscripts-2.92-1 (93-1)      32) ed-0.2-5
 33) info-3.9-1                     34) ncurses-1.9.9e-4
 35) libg++-2.7.1.4-5              *36) pwdb-0.54-3 (-4)
 37) rootfiles-1.5-1               *38) pam-0.57-2 (-4)
 39) redhat-release-4.2-1           40) less-321-3
 41) mount-2.5l-2                   42) zlib-1.0.4-1
 43) rpm-2.3.11-1                   44) e2fsprogs-1.10-0
 45) sysklogd-1.3-15                46) tar-1.11.8-11
 47) passwd-0.50-7                  48) gawk-3.0.2-1
 49) gdbm-1.7.3-8                   50) gpm-1.10-8
 51) hdparm-3.1-2                   52) kbd-0.91-9
 53) slang-0.99.37-2                54) newt-0.8-1
 55) kbdconfig-1.4-1                56) ncompress-4.2.4-7
*57) sh-utils-1.16-4 (-5)           58) procinfo-0.9-1
*59) logrotate-2.3-3 (4-1)          60) lilo-0.19-1
 61) losetup-2.5l-2                 62) linuxthreads-0.5-1
*63) mkinitrd-1.6-1 (7-1)           64) mailcap-1.0-3
*65) man-1.4h-5 (j-1)               66) mt-st-0.4-2
 67) modules-2.0.0-5                68) mailx-5.5.kw-6
 69) net-tools-1.32.alpha-2         70) procmail-3.10-10
 71) procps-1.01-11                 72) psmisc-11-4
 73) quota-1.55-4                   74) readline-2.0-10
 75) sed-2.05-6                     76) setconsole-1.0-1
 77) sendmail-8.8.5-4               78) shadow-utils-960530-6
 79) stat-1.5-5                     80) tcsh-6.06-10
 81) termcap-9.12.6-5               82) textutils-1.22-1
 83) time-1.7-1                     84) timeconfig-1.8-1
 85) util-linux-2.5-38              86) vim-4.5-2
 87) vixie-cron-3.0.1-14            88) which-1.0-5
 89) zoneinfo-96i-4                 90) tcp_wrappers-7.5-1
*91) NetKit-B-0.09-6 (-8)          *92) lpr-0.18-1 (19-1)
*93) bind-4.9.5p1-2 (9.6-1)        *94) bind-utils-4.9.5p1-2 (9.6-1)
*95) wu-ftpd-2.4.2b12-6 (b15-1)     96) anonftp-2.3-3
 97) zip-2.1-1                      98) unzip-5.12-5
 99) statserial-1.1-7              100) minicom-1.75-2
101) lrzsz-0.12.14-1               102) dip-3.3.7o-9
103) ppp-2.2.0f-3                  104) portmap-4.0-3
105) perl-5.003-8                 *106) traceroute-1.0.4.4bsd-2 (1.4a5-1)
*107) elm-2.4.25-7 (-8)            108) lynx-2.6-2
109) ncftp-2.3.0-5                 110) pine-3.95-2
```

```
 111) rdate-0.960923-1            112) apache-1.1.3-3
*113) nfs-server-2.2beta16-7     *114) nfs-server-clients-2.2beta16-
      (2.2beta16-8)                    (2.2beta16-8)
```

And with all the above installed, I still have 32MB left!

Updates; errata

As many, I hope, Red Hat users know, some packages may be found to have some security flaws or anything else which may cause issues to arise. For this reason, Red Hat releases updates for such packages. I have updated what packages I had which had updates and are marked in the above list. Please refer to Red Hat's web page concerning the updated packages at:

http://www.redhat.com/support/docs/rhl/rh42-errata-general.html

```
NOTE: Before you can update the packages, you'll have to
      run the ldconfig as described in section 3.3.1.1.
```

How to install packages with rpm; without glint

With rpm, use the --root option to specify the mounted directory as the root for installation. I had discovered that many packages were failing to install becuase of preinstall or postinstall scripts that weren't executing correctly due to the different root directory, thus, use the --noscripts option:

==> **rpm --root /iomega -i --noscripts PACKAGE.i386.rpm**

As I'm sure many will notice, you should get an error message like:

```
failed to open /iomega/var/lib/rpm/packages.rpm

error: cannot open /iomega/var/lib/rpm/packages.rpm
```

So, just create the var/lib/rpm directory :)

==> **mkdir /iomega/var; mkdir /iomega/var/lib; mkdir /iomega/var/lib/rpm**

Which came first, pamconfig or pam?

If anyone's tried to install pamconfig, it'll complain about a failed dependency of pam; and when you go try to install pam, pam complains about a failed dependency of pamconfig! This, being the chicken or the egg issue, puzzled me for a while, but thanks to the --nodeps flag, we can force pamconfig to install; besides pam has more failed dependencies than just pamconfig.

==> **rpm --root /iomega -i --nodeps --noscripts pamconfig-0.51-2**

4.3 Problems after installation of packages

Once everything is all nicely installed, unfortunately, the disk is not fully functionable, if that's a word. What I mean to say is, if you try to boot now with the floppy, you won't get very far. As soon as init tries to start up, you'll get two lovely errors; both of which complain about some files not being found that would have been made had the scripts been

run.

The case of the missing /etc/ld.so.cache and libc.so.5

If you tried booting, you'd get two error messages, the first will be the abscence of /etc/ld.so.cache file. The second complains about a missing libc.so.5.

/etc/ld.so.cache

As mentioned by many readers, my previous instructions didn't quite work as stated. In order to get this file created, you'll have to run ldconfig while the ZIP disk is still mounted:

==> **chroot /iomega /sbin/ldconfig**

Thanks to Javier Rodriguez for this solution.

libc.so.5

To solve the missing lib case, you'll have to create a symlink which would have been created by the installation scripts.

==> **cd /iomega/lib; ln -s libc.so.5.3.12 libc.so.5**

Thanks to Darcy Boese for this solution.

Setting root password

Just as ldconfig was ran in 4.3.1.1, you might as well change/create a root password for this new system:

==> **chroot /iomega passwd root**

What the install program created

```
NOTE: This is just a very brief setup, one which I haven't
      been able to test to see if it works.  In theory it
      should, but please do not send me complaints saying
      that this didn't work.
```

While exploring my Red Had 4.2 CD-ROM, I came across something rather interesting; the source code for the install program. I found it under /misc/src/install and one thing which I found of some use was the net.c file. In this, I found what other files that would be created had the install program been run. Most of these just give network support (hince the name net.c) but even if you don't have a network card, you can still use localhost for networking (plus apache will complain about not being able to determine a hostname.) These files include:

```
/etc/hosts
/etc/HOSTNAME
/etc/resolv.conf
/etc/sysconfig/network
/etc/sysconfig/network-scripts/ifcfg-eth0
(or any other network device you may have.)
```

Networking setup

For this document, I wanted network support for my 3Com 3c595 fast-ethernet card (which for the last several months has been used in another machine.)

First I needed a name, and since I'm running my own name server (which is another long story) I gave myself the name: *dash-dot.wig.org* (I just liked the sound of it.) Having a name without an IP is kind of pointless, so I used a reserved non-internet usable network of 192.168.10.0 which my name server also uses. Even though the hostname is typically stored in /etc/HOSTNAME, Red Hat checks for /etc/sysconfig/network for this name; so let's start there. Sample of my /etc/sysconfig/network:

```
NETWORKING=yes
HOSTNAME=dash-dot
DOMAINNAME=wig.org
GATEWAY=
GATEWAYDEV=eth0
NS1=192.168.10.7
```

Next, bascially duplicate the same info here for /etc/HOSTNAME, /etc/resolv.conf, and /etc/hosts:

/etc/HOSTNAME:

```
dash-dot.wig.org
```

/etc/resolv.conf:

```
search wig.org
nameserver ns.wig.org
```

/etc/hosts:

```
127.0.0.1 localhost
192.168.10.99 dash-dot.wig.org dash-dot
192.168.10.7  ns.wig.org ns
```

Red Hat configures all network devices from scripts found in /etc/sysconfig/network-scripts. The configuration of any network device is usually first created via the install program so I had to create these config files manually. They all begin with ifcfg-XXX where XXX is the network interface which ifconfig brings up; e.g. ppp0, eth0, etc. For this example, I had to create a file called ifcfg-eth0 which contains the following:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
BROADCAST=192.168.10.255
NETWORK=192.168.10.0
NETMASK=255.255.255.0
```

```
IPADDR=192.168.10.99
```

And last but not least, in order to get this going right now, while in the /etc/sysconfig/network-scripts directory, just do:

==> **./ifup ifcfg-eth0 boot**

This will start the script which configures the network interface if that interface was set to start at 'boot' time.

For further information, please refer to the NET-3-HOWTO document.

---

5. Slackware 2.2 installation

```
NOTE: This portion has not been updated.
```

5.1 Requirements

- ZIP Disk and Drive (obviously)
- Kernel with ZIP support
- ZIP Howto (recommended)
- 1.44" HD formated floppy
- 1 to 2 hours of time

5.2 Installation

Okay, now comes the fun part: Figuring out what files need to be on the system disk, and what packages that you want (and can fit) on your ZIP drive.

I decided that the easiest way to get started was install Slackware directly to the ZIP drive. I decided on this approach mostly because Slackware is a smaller distribution then Red Hat, and it would be easier to trim out what I didn't want. That and the fact that I am using the Slackware distribution anyway made it an obvious choice.

Installing Slackware onto the ZIP disk is easy, as root run the setup program, and choose /iomega as the install to partition, set the install from partition to where the Slackware sources are (cdrom, hardrive, etc), select install and follow the prompts.

5.3 What to install

The hardest part is deciding what to add, and what not to add. Obviously, you'll need the 'A' series (Which is about 8 megs), the rest is up to you.

I managed to trim down the Slackware release to a respectable installation of 70 megs, which included gcc/g++, perl, X11R6 (NOT ALL OF IT!), sendmail, online docs (Minus all the development man pages, but including all the howto's), and an assortment of other goodies, while leaving about 10 megs free for user files. YMMV

6. Creating /etc/fstab

(Common to both Red Hat and Slackware distributions)

The last thing that needs to be done before rebooting is to create the fstab file on the soon to be root partion. The following is what you should have as a minimum for /iomega/etc/fstab:

```
/dev/sda1      /         ext2        defaults   1   1
/dev/sda2      none      swap        sw
none           /proc     proc        defaults   1   1
```

Save the file, and reboot with the freshly made boot floppy and enjoy!

(Special thanks to Mike for reminding me about this very important and crucial step. - John)

7. Debian 1.2 Installation

```
NOTE: The author of this section sent this to me June, 11th 1997.
```

7.1 Requirements

- Ppa ZIP drive and disk.
- 2 blank 1.44 floppy disks
- A complete set of Debian install disks (review the Debian install docs if you don't remember how to make these)
- A couple hours of time

7.2 Overview

After spending a few hours wrestling with dpkg I decided it would be simpler to modify the Debian "Rescue" disk so that it would recognize the ppa ZIP drive. This proved to be very easy. You can then use this modified disk to procede through the normal Debian base system install. Once you've completed installing the base system you can use a boot disk to start the new base system and complete the installation using dselect. To use this technique you need to build two kernels - one with ppa and initial RAM disk support, and another without the RAM disk support.

If you want, you can skip all the steps in section 2 and let the Debian install procedure handle formatting the ZIP disk for you.

7.3 Creating the modified Rescue disk

The Debian rescue disk is a SYSLINUX style boot disk, which uses a DOS formatted floppy disk and a special boot loader to avoid loading MS-DOS. These disks are very easy to modify to start your own custom boot configuration. The Debian 'boot-floppies' package contains a set of scripts to automate the process of building boot disks. However, its so simple I found it easier to do the process by hand. This deviates a bit from the Debian philosophy, but I'm over it :). There are brief instructions in the 'readme.txt' file of the Rescue floppy.

Use dd (or RAWRITE under DOS) to create a new Rescue disk.

Review the Debian install docs if you don't remember how to do this.

Build a new kernel with ZIP ppa support

Build a new kernel with ZIP ppa support (as in step 3.3.1), but also configure RAM disk and initial RAM disk support. You also need to configure the msdos, fat, minix, ext2fs, and procfs filesystems.

Also configure any modules that you would like in your final installation on the ZIP disk.

Once the kernel is configured, build with:

==> **make dep; make clean**

==> ***make bzImage***

Build the modules with:

==> **make modules**

You will install these later.

```
NOTE: Make sure that you are using 'make bzImage', and
      not 'make zImage'.
```

Mount the new Rescue disk.

==> **fdmount fd0**

or

==> **mount /dev/fd0 /mnt**

or

==> **whatever :)**

Copy the kernel image

Copy the kernel image (on the i386 platform it will be located at arch/i386/boot/bzImage) to 'linux' on the floppy disk.

Editing the 'rdev.sh' script

Open the 'rdev.sh' script located on the Rescue floppy with your favorite editor. Change the last line: from--:

```
'rdev /mnt/linux /dev/ram0'
```

to--:

```
'rdev /mnt/linux /dev/sda1'
```

You will also have to change all occurences of /mnt/linux to the appropriate path. Since I mount my floppies under /fd0, I had to change /mnt/linux to /fd0/linux.

Run this modified 'rdev.sh' script.

==> **./rdev.sh**

TOC

7.4 Install the base system on the ZIP drive.

Boot your modified Rescue disk. If all goes correctly you will be presented with the familiar Debian menu based install process, except that now it is aware of your ppa ZIP drive. Procede through this process as if you were installing the system on a normal hard drive, but mount /dev/sda1 as root and initialize /dev/sda2 as swap.

There is one deviation from the standard install process-- don't install and/or configure any loadable modules. You will install the modules you built in step 7.3.2 later.

7.5 Creating the boot disk

You can create the boot disk just as described in steps 3.3.1 to 3.3.2, or, if you want, just use the "Create Boot Disk" option during the Debian install. I like this second option because I get another SYSLINUX boot disk, allowing me to edit the greeting message to describe the nature of my custom boot floppy and allow me to enter additional kernel arguments. You can even include help files, accessible via the functions keys. You may still wish to rebuild the kernel and modify this boot disk by hand later to remove the RAM disk support. Your call.

7.6 Reboot the system.

Insert your boot disk and choose the 'Reboot' option from the install menu.

7.7 Configure the base system and complete the install process.

When the system reboots you will have a slow but completely workable Debian base installation running off of your ppa ZIP drive. Proceed normally with the installation at this point. I installed all the normal UNIX utilities, along with documentation sets, make, gcc, libraries, and various useful file manipulation utilities. The result is a very powerful emergency boot system that I can use to rescue any of the systems in our department in an emergency.

```
NOTE: You'll at least need to install 'make' to
      complete the next step.
```

7.8 Installing the modules you built in step 7.3.2.

If you installed make in the last step, you should be able to mount the hard drive partition containing your kernel build, cd to the proper directory and run 'make modules_install'. Here's how I did it:

==> **mount /dev/hda2 /mnt**

==> **cd /mnt/usr/src/linux**

==> **make modules_install**

```
NOTE: You won't need to modify the /etc/fstab file, as expla
      in step 5.  The Debian installation process has alread
      taken care of that.
```

8. Afterthoughts

Whew, and sigh. After amassing 31 ZIP disks and a ZIP unleashed battery pack, my next experiment is to try to get pcmcia working for my laptop.

Zip Drive Mini-HOWTO

Kyle Dansie, dansie@ibm.net

v2.4, 10 January 1999

This Document provides a quick reference quide on setting up and using the Iomega ZIP drive with Linux.

1. Introduction

- 1.1 Home of this document

2. Quick Start

3. Configuring a kernel for the ZIP drive

- 3.1 SCSI version

4. The ZIP drive

- 4.1 Parallel port version Zip100
- 4.2 SCSI external version
- 4.3 The SCSI Internal version
- 4.4 The ZIP Plus - IMM driver
- 4.5 ATAPI version
- 4.6 IDE version
- 4.7 USB version
- 4.8 ZIP250 version

TOC

5. Using the ZIP drive

- 5.1 Fdisk, mke2fs, mount, etc.
- 5.2 An existing DOS formatted disk
- 5.3 Re-format as a native Linux disk
- 5.4 fstab entries
- 5.5 The ZIP Tools disk

6. Performance

7. Frequently asked questions

- 7.1 Can I plug a printer into the parallel ZIP drive ?
- 7.2 Do you plan to support EPP/ECP ports in PPA ?
- 7.3 Can I run Linux from a ZIP drive ?
- 7.4 Can I boot from the ZIP drive ?
- 7.5 Why does Iomega use partition number 4 ?
- 7.6 How can I have the disk mounted at boot time ?
- 7.7 What happens if there is no disk inserted when I boot ?
- 7.8 Can I use the parallel drive as a real SCSI disk ?
- 7.9 Can PPA be used with Iomega's parallel port tape drives ?
- 7.10 Will PPA work with the parallel port SyQuest EZ135 ?

8. Getting more current information

---

1. Introduction

The Iomega ZIP drive is a popular, removable media disk drive. It is avaliable in several versions.

- Parallel Port
- SCSI
- IDE - ATAPI
- Plus

The most popular seems to be the Parallel port version. It has been around a long time and it is low cost. The SCSI version has also been around for a while but it needs a scsi card that many people do not have already, so it it not quite as popular. The ATAPI version now seems to be more available. The Plus is the newer version of the ZIP drive. It has the capability to detect what port its plugged into, scsi or parallel. This document will focus on the Parallel port version, but also offer some tips for other versions.

This document incorporates information collected and published by others, in particular:

- Grant Guenther, grant@torque.net
- Scot Wilcoxon, sewilco@fielday.mn.org
- Joe Mack, mack@ncifcrf.gov
- Byron Jeff, byron@cc.gatech.edu

I thank them for their important contributions, and accept responsibility for any errors that I have introduced.

A special thanks to Grant Guenther who wrote the 0.26 version of the ppa program and also wrote the original version of this HOWTO. If you find any mistakes or oversights in this document, please let me know. dansie@ibm.net

1.1 Home of this document

The most current version of this document will reside at http://www.njtcom.com/dansie/zip-drive.html

2. Quick Start

OK so you have purchased a new zip drive and now you want to use it. Some distributions have the modules already built. You can try insmod ppa for starters. (Make sure you have it hooked up and a disk inserted). If the ppa program is not there, then configure the kernel. My machine has 2 parallel ports, one for the zip and one for the printer. This makes things easy, no cable switching.

Configure the kernel:

- cd /usr/src/linux
- make xconfig
- scsi support = Y
- scsi disk support = Y
- Iomega zip support as a module
- printer support also as a module
- save it and exit
- make dep
- make clean
- make zImage or zlilo or zdisk
- make modules
- make modules_install

Now to use the drive:

- load the module `insmod ppa`
- build a mounting point. `mkdir /zip`
- insert a preformatted windoze type disk into the drive.
- mount the disk. `mount -t vfat /dev/sda4 /zip`
- use any standard file commands as in `l /zip, ls /zip, df, cp,`
- when you are finished `umount /zip`

If you had trouble with any of the previous steps, see the following sections for more detailed instructions.

3. Configuring a kernel for the ZIP drive

To use the ZIP drive with Linux, you must have a kernel configured with support for the SCSI system, support for SCSI disks, and support for the host adapter you are using. If you are not familiar with building a kernel, you should read up on the Linux Kernel HOWTO for information.

You must begin the process of building a kernel with the configuration step. Here, you identify the specific kernel components that you need. First step `cd /usr/src/linux`. There are several ways to actually do the configuration. Under X windows I use **make xconfig**. There is also `make menuconfig` or `make config` for command line prompts. The easiest way is with xconfig.

In the section **SCSI Support** set **SCSI support = Y**. Also set **SCSI disk support = Y**.

In the section **SCSI low-level drivers** you want to set **IOMEGA Parallel Port ZIP drive SCSI support = M**. The M stands for modules.

In the section **Character Devices** find and set **Parallell Printer support = M**

If you are a bit unsure about any of this, use zdisk for the make step. This will build and install the kernel to floppy. If you screw it up somehow, you still have a good bootable system on the hard drive.

Now build the kernel with these steps:

- make dep
- make clean
- make zImage or zlilo or zdisk
- make modules
- make mdoules_install

hint if you want to create an output log of the make zImage step you can use

```
make zImage 2>&1 | tee zImage.out
```

This is really a good thing to do, if you have compile errors or other conflicts in the configuration, now you have a file with the error messages or compile errors. This makes asking for help much easier.

You also need to load the modules somewhere. On my system I just added a couple of lines to the boot.local file.

- insmod ppa
- insmod lp

My system is Suse 5.3 and this file is in /etc/rc.d. The location of this file will vary by distribution, but there will be a file somewhere, for local changes to be made at start up time. You should use the file that is applicable for your distribution.

Now `shutdown and reboot`.

You can also build all or part of the SCSI system as modules. If you do this, be sure to load scsi.o, then sd.o and finally the driver for your host adapter, before you try to access the ZIP drive.

3.1 SCSI version

If you already have a SCSI disk in your system, and you are connecting the ZIP drive to the same controller, there is no additional kernel configuration required. Otherwise, you will most likely have to build a new kernel.

If you are building a kernel to support the SCSI version of the ZIP drive, you should select SCSI support and SCSI disk support. You must also select a driver for the interface card you will use. If you have a ZIP Zoom, select the aha152x driver.

Be sure to read the documentation for your adapter in the SCSI HOWTO and any README files in the drivers/scsi subdirectory of the Linux source tree. Pay attention to command line parameters that you might have to use to help the kernel initialise your adapter.

For instance, if you are using the ZIP Zoom card, you will have to add something like

```
aha152x=0x340,11,7,1
```

to the boot command (or include it in your /etc/lilo.conf file in an append clause). This tells the driver the port address and IRQ of your ZIP Zoom card - be sure to use the numbers that correspond to the way your jumpers are set.

You should also read Paul Gortmaker's BOOTPROMPT HOWTO for information about configuring your kernel with LILO or LOADLIN.

---

4. The ZIP drive

There are at least five or six versions of the Iomega ZIP 100 drive. The Iomega page for these drives is at http://www.iomega.com/product/zip/ They all accept special cartridges resembling a 3.5" floppy disk that hold 100 megabytes of data. The disks actually hold 96 cylinders of 2048 sectors each holding 512 bytes. This would normally be called 96 Megabytes.

The external drive comes in these types:

- Parallel port
- SCSI
- Plus

The internal drive comes in these types:

- SCSI
- IDE
- ATAPI

The internal versions can be mounted in 3.5 or 5.25 drive slots. The external versions are in a small blue plastic case enclosure, powered by an external wall brick.

All the drives have a large pushbutton on the front of the drive. This is used to eject the disk. Linux locks the door while using the drive, but if the button is pressed while the door is locked, the ZIP drive will remember and eject the disk as soon as the software unlocks it.

4.1 Parallel port version Zip100

This drive is described at http://www.iomega.com/product/zip/zippp.html

The Linux driver comes in a couple of different versions. The 0.26 version of the ppa.c program ships standard with the 2.0.x kernels. If your kernel is older than 2.0.35 then you have the **OLD** program. Check out

David Campbell's page for the more current version of this program. At the time of this writing the current version is 1.42.

The auto configuration features of the 1.4x program are very nice. You do NOT have to tell the progam what the base address of the parallel port is. The program will prob the standard address's for the parallel ports and attach the drive when it finds it.

The most common problem is to load the lp printer module before the ppa program. The ppa program must be loaded before the lp module. The lp module will tie up the port that the zip drive needs, this is why the ppa progam must be loaded first.

Also be sure that all cables are firmly attached.

If you did everything right :) you will see messages like this at boot time.

TOC

```
VFS: Mounted root (ext2 filesystem) readonly.
Adding Swap: 102528k swap-space (priority -1)
ppa: Version 1.42
ppa: Probing port 03bc
ppa: Probing port 0378
ppa:     SPP port present
ppa:     EPP 1.9 with hardware direction protocol
ppa: Found device at ID 6, Attempting to use EPP 32 bit
ppa: Communication established with ID 6 using EPP 32 bit
ppa: Probing port 0278
ppa:     SPP port present
scsi0 : Iomega parport ZIP drive
scsi : 1 host.
  Vendor: IOMEGA    Model: ZIP 100           Rev: D.09
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi removable disk sda at scsi0, channel 0, id 6, lun 0
SCSI device sda: hdwr sector= 512 bytes. Sectors= 196608 [96 MB] [0.1 GB]
sda: Write Protect is off
  sda: sda4
lp2 at 0x0278, (polling)
```

You can view these messages with dmesg | more. You can save the messages to a file with dmesg > dmesg.txt. This is often usefull if you are trying to debug the installation.

Also see section Getting more information

4.2 SCSI external version

This drive is described at http://www.iomega.com/product/zip/zipscsi.html

The external SCSI version of the ZIP drive has two DB25F connectors, and two configuration switches. One switch selects the drive's target address. The choice is limited to target 5 or 6. The other enables an internal terminator, in case the drive is the last one on a chain. The 25 pin SCSI connectors use the familiar Macintosh style wiring. The drive is shipped with a Macintosh type cable, but standard cables and converters are easily obtained if you are using a host adapter with a Centronics or high-density connector.

I have not seen an internal SCSI drive, but I would expect it to have a standard 50 pin DIP header SCSI connector and the same two switches.

Make sure that the target address you choose does not conflict with any other SCSI devices you may have on the same bus. Also be sure that the physically last drive in a chain has termination enabled, or an external terminator installed.

If you have an internal SCSI disk or CD-rom, and you connect your ZIP drive to the existing adapter, you should check to see if there are any terminators on the card that must be removed. Only the two extreme ends of the SCSI bus should be terminated. If your bus is partly internal and partly external, there should be one terminator on the last external device and one on the last internal device, but no terminators on the adapter card itself.

Be sure that all cables are firmly attached.

The kernel configuration is much the same as the internel version. See the next section SCSI Internal version for an example configuration.

The ZIP ZOOM host adapter

Iomega markets a SCSI host adapter under the name ZIP Zoom. This is actually based on the design of the **Adaptec AHA1520** family of adapters. It has an external Macintosh type DB25F connector, compatible with the cable that comes with the ZIP drive.

Linux supports this adapter with the **aha152x** driver.

4.3 The SCSI Internal version

Install hardware as described in the "Installation and Reference Guide" noting which SCSI ID, IRQ and I/O Port Address are being used. (You'll need this info later.) Things will go smother if the drive and adapter card use different SCSI ID's.

Beware The zip zoom card is hardwired for scsi ID number 7. The scsi id in the command line below is for the card, not the drive. The card and the drive must use different scsi id numbers. For example if the card uses 7 then use 5 or some other number for the drive.

Recompile the kernel after configuring it to include 'SCSI', 'SCSI disk' and 'AHA152X/2825' support.
INSTALL NEW KERNEL :-(

Determine what your kernel command line is:

```
aha152x=[I/O Port][,IRQ][,SCSI ID][,RECONNECT][,PARITY]
```

For example :

```
aha152x=0x140,10,7,1
```

If your using LILO add your kernel command line to your lilo.conf file using the append command.

(ie. append = "aha152x=0x140,10,7,1")

If your using LOADLIN add your kernel command line to the command you use to initiate loadlin.

```
(ie. loadlin c:\vmlinuz root=/dev/hda3 aha152x=0x140,10,5,1 ro)
```

4.4 The ZIP Plus - IMM driver

This drive is described at http://www.iomega.com/product/zip/zipplus.html

This is a **newer** version of the external Zip drive. The **Zip Plus** does not use the ppa driver, instead it uses the **imm** driver. Philippe Andersson sent in this excellent section on the imm driver for the Zip Plus. Thanks Philippe.

First, rebuild your kernel to include the following items:

TOC

- modules support
- SCSI support
- SCSI Disk support
- lp as a module (if you need it)

You'll notice we didn't select any SCSI low-level driver. That's ok -- we'll build it separately later.

Then, you need to get the driver source code from Dave Campbell's home page (http://www.torque.net/~campbell/imm.tar.gz). Version 0.18 is the current one at the time of writing (Aug. 98). Unzip it and untar it somewhere (under `/usr/src`, for instance). Then just run `make`. You'll get the module you need (`imm.o`). Copy it to

```
/lib/modules/$(uname -r)/scsi
```

.

If your lp module was loaded (check with **lsmod**(1)), unload it (`rmmod lp`), then load `imm.o` (`insmod imm`) and you're all set basically.

If the ZIP drive was not connected and powered on at the time, you'll get the message "init_module: Device or resource busy", and the module won't load. (By the way, it looks like you don't need to switch your PC off to connect the ZIP -- just make sure the module is not loaded and the drive is turned off, then plug it in, turn the drive on, and load the module.)

If the ZIP was connected and powered on, but there was no disk in, the module will load all right, but you'll get the message that it can't read the disk partition table. This is ok, as the partition table will be automatically read when you insert a ZIP disk.

If there was a ZIP disk in, you'll get the full information displayed, including a list of partitions defined on the disk and its Write Protect status. Here is a sample load-time message:

```
vger:~# insmod imm
imm: Version 0.18
imm: Probing port 03bc
imm: Probing port 0378
imm:     SPP port present
imm:     ECP with a 16 byte FIFO present
imm:     PS/2 bidirectional port present
imm:     Passed Intel bug check.
imm: Probing port 0278
scsi0 : Iomega ZIP Plus drive
scsi : 1 host.
  Vendor: IOMEGA    Model: ZIP 100 PLUS      Rev: J.66
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi removable disk sda at scsi0, channel 0, id 6, lun 0
SCSI device sda: hdwr sector= 512 bytes. Sectors= 196608 [96 MB] [0.1 GB]
sda: Write Protect is off
 sda: sda1
vger:~#
```

TOC

One last remark to mention that this version of the driver also locks the drive door while the disk is mounte the eject button is pressed in this situation, nothing happens, but the drive "remembers" and ejects the disk soon as it is dismounted.

4.5 ATAPI version

This drive is described at http://www.iomega.com/product/zip/zipatapi.html

There was an IDE version of the drive produced for a while. I think that for the most part this has been replaced by the ATAPI version.

Donald Stidwell sent in these comments on the ATAPI version. Thanks Don.

I use an ATAPI Zip drive and it works with both 2.0.32 and 2.0.33 kernels. I've used it under both RH 5.0 and OpenLinux 1.2 (my current used distribution). To get it to work under OpenLinux, I just enabled **ATAPI floppy support** in the kernel. OpenLinux does not have this support compiled in by default.

No other drivers are needed. It will mount as an extended partition on partition 4. I.E, mine mounts on HDB4. I mount it under /mnt/zip as noauto, although I don't suppose there would be any real problem with automounting. I just wonder about ejecting disks. I always dismount the drive before ejecting a cartridge.

There are more detailed instructions for the ATAPI install in the Linux Gazette May 1998 issue. See the 2 cent tip section.

```
<url url="http://www.linuxgazette.com/issue28/lg_tips28.html#atapi">
```

4.6 IDE version

I have not used the IDE version. Eric Backus sent in these comments. Thanks Eric.

I have one of these. It came with my Gateway 2000 computer a year ago. I think most of these were shipped by large OEM companies like this, before the ATAPI version of the ZIP drive was available.

The good news about this drive: no kernel modules or modifications are needed to support it. It looks to the kernel like an IDE hard drive. It worked for me with no effort with kernel 2.0.31 and 2.0.32.

The bad news about this drive: because it doesn't use ATAPI, you can't use the SCSI-to-ATAPI translation, which means you can't use mtools to write-protect disks (or to eject them, for that matter).

4.7 USB version

This drive is described at http://www.iomega.com/product/zip/imac.html

These drives look very cool and the USB interface also looks good. The only problem is that the USB interface for Linux is still under development at this time. So for now, this drive is not usable under Linux. Check this address for updates.

http://peloncho.fis.ucm.es/~inaky/USB/news.html

4.8 ZIP250 version

TOC

This drive is desribed at http://www.iomega.com/product/zip/zip250.html

This drive is new as of January 1999. I just bought one of these and it works very well in Linux. Just use the IMM driver from http://www.torque.net/~campbell/imm.tar.gz Set up the driver the same way as the zip plus. Mainly just download the driver, unpack it, and run make. Then copy the imm.o executable to the modules directory. For example /lib/modules/2.0.35/scsi (The 2.0.35 is your kernel version). If you are not sure of the version do uname -r.

---

5. Using the ZIP drive

If you have built in all the required components, the kernel should recognize your adapter and drive at boot time. If you are using a loadable module for your driver, the following discussion applies once that module is loaded.

For the most part, the SCSI and parallel versions of the drive behave identically, except that the parallel version is somewhat slower.

5.1 Fdisk, mke2fs, mount, etc.

Once you know the drive name for your ZIP drive, you are set. You can manipulate the drive with the normal Linux disk management commands. `fdisk` (or perhaps `cfdisk`) is used to manipulate the partition tables on the disk. `mke2fs` can be used to format a partition with the ext2 filesystem - the one most commonly used in Linux. `mount` is used to connect a formatted partition into your directory hierarchy.

You should study the manual pages for these tools if you are not familiar with them. Be warned that there are now several quite different versions of the `fdisk` program - be careful.

I'll describe two common scenarios.

5.2 An existing DOS formatted disk

If you have a ZIP disk with a DOS file structure that was originally created by Iomega's tools, the partition scan should say that the disk has one partition, `/dev/sda4`.

You should make a place to mount the disk, lets say `/zip`, and then mount it as an MS-DOS filesystem:

```
mkdir /zip
mount -t vfat /dev/sda4 /zip
```

You could also use msdos instead of vfat. vfat supports long filenames where msdos does not. Now, the files on the disk should appear in /zip. While the disk is mounted, you will not be able to remove it. When you are finished with the disk you can umount it to release it and detach it from your directory hierarchy.

```
umount /zip
```

Once you've made the /zip mount point - you don't need to do it again, so you could come back later and mount something else there.

5.3 Re-format as a native Linux disk

If you want to erase a ZIP disk and make a Linux native file system on it. You should use fdisk on the entire disk:

```
fdisk /dev/sda
```

and delete any existing partitions (with the `d` command). Then create a new partition with the `n` command, make it primary partition number 1, use `w` to write the partition table to disk, and quit with `q`.

Format the partition

```
mke2fs /dev/sda1
```

(The 1 is the number that you gave this partition in `fdisk`). Now you can mount the disk:

```
mount -t ext2 /dev/sda1 /zip
```

(re-using that mount point we created before).

5.4 fstab entries

The /etc/fstab allows you to configure the mount command. I like to be able to mount and write to the zip disk from any login id on the system. I added 2 lines to the end of the fstab file that look like this.

```
/dev/sda1    /zip          ext2          noauto,rw,user,nosuid,sync
/dev/sda4    /zipdos       vfat          noauto,rw,user,nosuid,sync,mode=0777
```

These entries assume that you have a /zip and a /zipdos directory. If you want all users ids to read and write, make sure you set the permissions. For exmaple **chmod 666 /zip** . You must do the chmod as root. The fstab entries also assume that your dos disks are partitioned as 4 (sda4 in my case), and the ext2(linux) disks are partitoned as 1 (sda1). You can read more about the fstab options with **man mount**

With the fstab entries as above you can mount a dos disk with **mount /zipdos** If you are using a linux ext2 disk then use the command **mount /zip**

5.5 The ZIP Tools disk

There is some extra work to be done if you want to use the disk that comes with the ZIP drive. As shipped, the software controlled write protection is enabled. Most people have unlocked the disk under DOS before ever trying to use it with Linux. Linux cannot access a locked disk, and it must be unlocked with Iomega's tools.

A native Linux program to manage the write protection feature, among other things, is expected to be available soon.

---

6. Performance

With the standard version of the 2.0.x kernels you get the 0.26 version of the ppa.c program. This is a very reliable simple program. It has gotten a lot of use and abuse and held up rather well. The only bad thing about this program is that it's not very fast. On my machine I have version 2.0.30 kernel, and 2 parallel ports. This is also a low-buck machine pentium 133 - 16 meg memory, 5 gig maxtor eide hard drive. The machine cost me 500. U.S. Dollars and the drive was 250. I am cheap but so what.

I had the ZIP drive on the second port and the printer on the first port. The second port was an old add in card, no epp mode. This made the driver run in 4 bit mode.

To base line the performance I copied the netscape tar file from the hard drive to the zip drive. This file is about 10.6 meg. Time 7 minutes 15 seconds. Going from the zip drive to the hard drive was about 7 minutes even. Not exactly setting the world on fire.

I then made 2 changes. First I switched the ZIP drive to the first port and went into the bios and set EPP mode on for the first port. At the same time I hooked up the printer to the 2nd port.

Second change I went out and downloaded the new version of the program from David Campbells page Then I installed this program. Check out the readme file that is included.

Then I ran the same test again of copying the Netscape tar file to and from the ZiP drive. The copy to the C drive was 1:55 seconds. From the hard drive to the Zip drive was 1 minute flat. This compares to 7:15 and 7:00 with the old program. This was a big improvement over the 0.26 program. This was also using the zip drive as a vfat file type.

Next I formated a zip drive in ext2 format. Ran the same test again. The copy going to the hard drive was only 50 seconds. The copy from the hard drive to the Zip drive was only 40 seconds.

Performance Summary

For this test case of moving a 10.6 meg file, the 4 bit standard port mode and program version 0.26 is the slowest 7:00 and 7:15

The new program (1.41) and EPP mode with ext2 file types is the fastest at 50 seconds and 40 seconds.

7. Frequently asked questions

These are some of the questions that we get asked a lot. Please read the answers here before asking them again !

7.1 Can I plug a printer into the parallel ZIP drive ?

Yes, but. Currently, you cannot have both the lp and ppa drivers active on the same parallel port. A resource sharing protocol has been designed and is in the early stages of implementation, but will not be available before Linux 2.0 is released.

If you like living on the bleeding edge of things, get the 2.1.x kernel and you can have port sharing. See http://www.cyberelk.demon.co.uk/parport.html

The easy way to do this with 2.0.x kernel is to have 2 parallel ports. I bought my second one for about 15 dollars and it works great. You can use one for printing and one for your ZIP drive. Just build a kernel with both the lp and ppa drivers as modules. I use the newer (1.41) version of ppa, so I just load ppa first (insmod ppa). It then finds the zip drive on the first port. Then I load lp (insmod lp) and it assigns the printer to the second port.

If you use the 0.26 version of the program then upgrade to 1.41 or newer. The only answer you will get on qustions about this driver is to upgrade the program.

7.2 Do you plan to support EPP/ECP ports in PPA ?

Version 0.26 of the program does not support EPP. Version 1.41 **does** support the EPP. This program is also much faster. See the performance section for more details.

7.3 Can I run Linux from a ZIP drive ?

People have done this. It would seem this might make a great rescue disk or a way to test an install.

http://metalab.unc.edu/LDP/HOWTO/mini/ZIP-Install.html

7.4 Can I boot from the ZIP drive ?

This depends on what you are using for a host adapter. If your host adapter has a BIOS ROM with code that can boot from target 5 or 6, then yes, you can boot from the SCSI ZIP drive. You cannot boot from a parallel port drive.

If you have DOS on your main disk and want to boot Linux from a ZIP disk, you can do this using the

TOC

LOADLIN boot loader. Your kernel image must be somewhere on the DOS disk, but the rest of your Linux system could be on the ZIP. Make sure that the kernel you use has the correct drivers built in.

As far as I am aware, there are no installation boot floppies for any of the popular distributions that contain the parallel ZIP driver.

7.5 Why does Iomega use partition number 4 ?

This is one of the most popular questions, but I don't think anyone has a definite answer to this one. Could be they had their head where the sun don't shine. Maybe there is no reason.

7.6 How can I have the disk mounted at boot time ?

All you need to do is to add a line to your `/etc/fstab` file. For instance, if you will always have a DOS disk in the drive when you boot, you could put

```
/dev/sda4   /zip  vfat  defaults  0 0
```

in the `fstab`. Depending on your distribution, the initialization scripts might try to run fsck on partitions listed in your fstab. Be aware that this could cause problems if you forget to put the disk in the drive when you boot, or have the wrong disk there.

7.7 What happens if there is no disk inserted when I boot ?

The kernel will try to read the partition table, but the operation will time out. There is nothing to worry about. Once you have inserted a disk, and try to do something with it the kernel will notice that there is a disk there now and rescan the partition table.

HINT: when you change disks, it is a good idea always to use fdisk to check the partition structure on the new disk.

The BIOS on some SCSI host adapters will attempt to read the partition table on your disk during the system boot. If you cannot disable this check, you may be forced always to boot with a disk in the drive.

7.8 Can I use the parallel drive as a real SCSI disk ?

The PPA-3 parallel-to-SCSI adapter is implemented as a single ASIC chip that Iomega calls a VPI0. It is embedded on the ZIP drive's controller card. There is an actual SCSI bus present, but not in a useful form.

Although I haven't tried to compare the two cards, it seems logical that the VPI0 replaces the conventional electrical buffering circuits that would be required if the SCSI bus were extended outside the package.

TOC

7.9 Can PPA be used with Iomega's parallel port tape drives ?

No. Those drives are floppy-tape drives. There is no SCSI involved.

Several people have indicated an interest in applying my methods to try to determine the protocol and develop a driver for these tapes. The first step in that process is to get the DOS driver working under DOSemu. I have not yet heard that anyone has been successful in doing it.

7.10 Will PPA work with the parallel port SyQuest EZ135 ?

No. The EZ135 is an IDE drive with a ShuttlePort parallel to IDE converter embedded in it.

However there has been some work done on this driver. Check out

http://www.torque.net/linux-pp.html

---

8. Getting more current information

More current versions of the parallel port zip drive (ppa.c) program and current technical information of the driver can be found at

http://www.torque.net/~campbell/

Information of many parallel port external device drivers can be found at

http://www.torque.net/linux-pp.html

Parallel Port sharing or through-port information is found at

http://www.cyberelk.demon.co.uk/parport.html

Installing Linux from a zip drive (for machines without a CD)

http://metalab.unc.edu/mdw/HOWTO/mini/Install-From-ZIP.html

An X frontend to the ziptool program can be found at

http://www.scripps.edu/~jsmith/jazip/

Information on configuration of Lilo can be found at

http://metalab.unc.edu/mdw/HOWTO/mini/LILO.html

Information on using bootprompts can be found at

http://metalab.unc.edu/mdw/HOWTO/BootPrompt-HOWTO.html

If you want to install the Linux OS onto a Zip Drive and run the OS from the Zip drive

http://metalab.unc.edu/LDP/HOWTO/mini/ZIP-Install.html

Iomega's web pages are at

http://www.iomega.com/.

And if you are looking for general information about parallel port programming for the PC, you might want to visit

http://www.lvr.com/parport.htm